

# ***Modellazione Grafica***

*Autori:*

*Anna Lamboglia*

*Agostino Vitaglione*

*Mario Vitaglione*

# Introduzione

Negli ultimi anni, con l'avvento di Internet e delle nuove tecnologie, risulta più semplice per l'uomo avere la possibilità di raccogliere dati, diventati una risorsa preziosa per via di numerosi contesti di utilizzo.

## *Come posso disegnare un modello che descriva al meglio l'andamento dei dati?*

Campi come la **Data Science** e il **Machine learning** si basano infatti sulla modellazione e visualizzazione dei dati in 2D e 3D, fornita dallo studio del calcolo numerico.

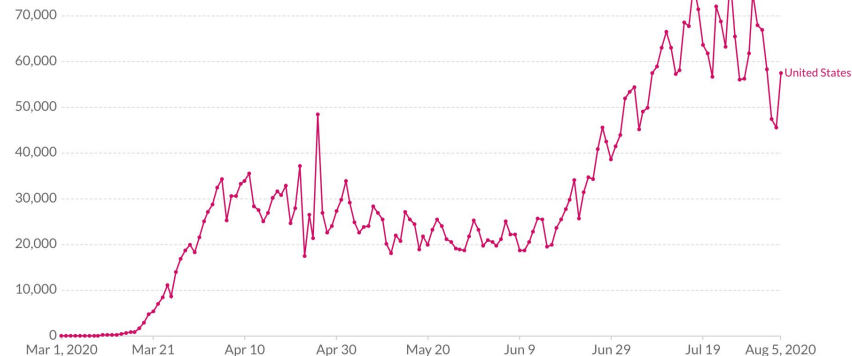
Una rappresentazione opportuna dei dati infatti consente di comprendere

- *le caratteristiche del problema*, in modo da poter effettuare considerazioni e soprattutto previsioni sull'andamento dei dati.
- *la qualità dei dati*: nel caso in cui ci fossero andamenti particolari, risulta possibile valutare se ci sono stati errori di misurazione.
- *la quantità dei dati*.

### Daily new confirmed COVID-19 cases

The number of confirmed cases is lower than the number of actual cases; the main reason for that is limited testing.

LINEAR LOG



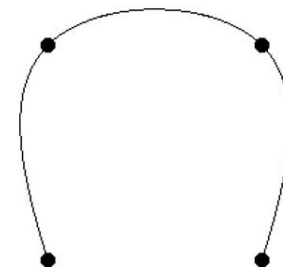
Source: European CDC - Situation Update Worldwide - Last updated 5 August, 10:04 (London time)

# Rappresentazione dei dati

Dobbiamo costruire un modello che descriva i dati in modo attendibile ed abbiamo due tipi di modelli:

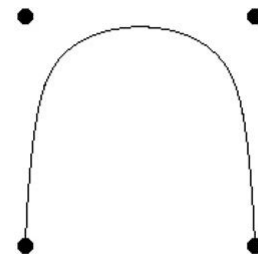
## **Modello Interpolante**

Si costruisce una curva che passa per i punti assegnati assumendo che l'errore dei dati sia trascurabile, poiché di fatto si impone il passaggio della curva per i punti indicati. Di conseguenza, è possibile trascurare l'errore sui dati.



## **Modello Approssimante**

Si costruisce una curva che non passa per i punti assegnati. In questo caso si assume non trascurabile l'errore sui dati.



## **Problema**

Dati  $n + 1$  punti distinti  $(x_i, y_i)_{i=0, \dots, n}$ , si vuole costruire una funzione  $f(x)$  tale che nei nodi  $\{x_i\}_{i=0, \dots, n}$ , siano soddisfatte determinate condizioni chiamate condizioni di interpolazione.

Dunque si vuole tracciare il grafico di una funzione conoscendone solo il valore in un numero finito di punti.

# Interpolazione lineare

Si considerino le funzioni  $\{\varphi_i\}_{i=0,\dots,n} \in \mathbb{R}$

linearmente indipendenti definite in  $[a,b]$ . Siano assegnati i valori  $(x_i, y_i)$  con  $x_i \in [a,b]$  tale che  $x_i \neq x_j \forall i \neq j$ .

Risolvere un problema di interpolazione lineare così posto vuol dire calcolare i coefficienti  $a_0, \dots, a_n \in \mathbb{R}$  tali che la funzione:

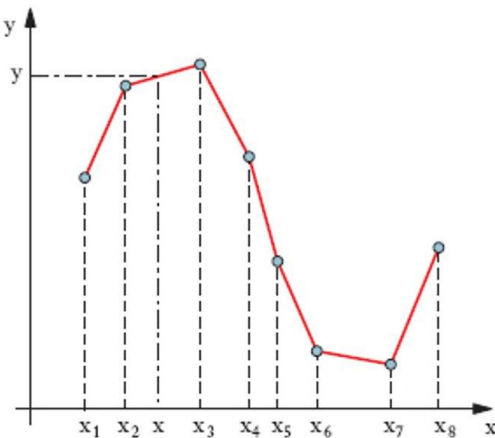
$$f(x) = \sum_{i=0}^n a_i \varphi_i(x)$$

Soddisfi le seguenti condizioni:

$$f(x_i) = y_i, \quad i = 0, \dots, n$$

**Condizioni di interpolazione**

$$\varphi_i(x) = x^i \xrightarrow{\text{Sostituendo}} f(x) = a_0 + a_1 x + \dots + a_n x^n$$



*Il problema della determinazione dei coefficienti del polinomio interpolante si riduce nella risoluzione del seguente sistema lineare:*

$$\begin{bmatrix} 1 & x_0 & x_0^2 & \dots & x_0^n \\ 1 & x_1 & x_1^2 & \dots & x_1^n \\ 1 & x_2 & x_2^2 & \dots & x_2^n \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^n \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \dots \\ a_n \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \dots \\ y_n \end{bmatrix}$$

**Matrice di Vandermonde**



# Interpolazione di Lagrange

Data la funzione  $f(x)$  e  $n+1$  punti:  $\{x_i\}_{i=0,\dots,n}$

Per cui sono noti i valori:  $\{f(x_i)\}_{i=0,\dots,n}$

$$P(x) = \sum_{i=0}^n f(x_i) \prod_{\substack{j=0 \\ j \neq i}}^n \frac{x-x_j}{x_i-x_j}$$

**È definito**

**Polinomio di Lagrange**

**Sostituendo** 
$$l_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^n \frac{x-x_j}{x_i-x_j}$$

$$P(x) = f(x_0)l_0(x) + f(x_1)l_1(x) + \dots + f(x_n)l_n(x)$$

Se si introducono vincoli di comportamento della funzione nell'intorno di quest'ultimi attraverso condizioni sulle derivate, si parla di **interpolazione di Hermite**.

Tali condizioni sono solitamente espresse in forma tabellare.

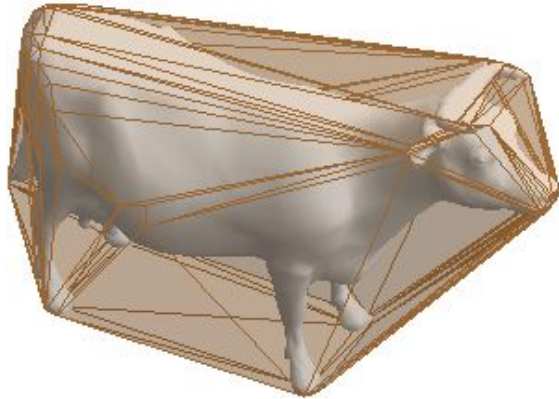
# Punti di controllo

Bézier nel 1960 introdusse i *punti di controllo* per i problemi di interpolazione come un modo naturale per stabilire la dimensione dell'oggetto da disegnare che, fissati opportunamente, *determinano il più piccolo poligono convesso contenente tali punti*.

Tutte le curve sono costruite a partire da questi punti di controllo, disposti a seconda dei vincoli imposti alla curva stessa, sono interamente contenute nell'**inviluppo convesso**.

Per rappresentare matematicamente la curva, bisogna scegliere quali funzioni base  $\varphi_i$  utilizzare.

Una possibile scelta è quella dei **polinomi** poiché molto semplici da manipolare.



## **Forma Standard:**

$$p(x) = a_0 + a_1 x + \dots + a_n x^n \quad \varphi_i(x) = x^i$$

## **Forma di Lagrange:**

$$p(x) = a_0 l_0(x) + a_1 l_1(x) + \dots + a_n l_n(x) \quad \varphi_i = l_i(x)$$

## **Forma di Newton:**

$$p(x) = a_0 + a_1(x - x_1) + \dots + a_n(x - x_1) \dots (x - x_n)$$

$$\varphi_i(x) = \prod_{j=1}^i (x - x_j)$$

# Considerazioni

## Forma Standard

Il determinante della **matrice di Vandermonde**  $V_i$  costruita sui nodi  $x_0, \dots, x_n$  è:

$$\det V_n = \prod_{0 \leq j < i \leq n} (x_i - x_j)$$

Supposti i nodi distinti, la matrice  $V_n$  è invertibile e il sistema lineare ammette una ed una sola soluzione.

Il tempo di calcolo per la risoluzione del sistema lineare con l'algoritmo di Gauss e l'algoritmo di back-substitution è:

$$\mathcal{O}(n^3)$$

Esistono ulteriori algoritmi per la risoluzione di sistemi con matrici di Vandermonde che hanno una complessità computazionale pari:

$$\mathcal{O}(n^2) \quad \mathcal{O}(n \log^2 n)$$

Tuttavia, la matrice di Vandermonde è spesso mal condizionata.

→ **Errori piccoli sui dati vengono amplificati sulla soluzione, rendendole poco attendibili.**

## Forma di Lagrange

Per calcolare  $l_i(x)$  sono necessari  $n-1$  prodotti

→  $\mathcal{O}(2n^2)$

**Computazionalmente oneroso**

## Forma di Newton

→ **È la più utilizzata!**

**Computazionalmente migliore**

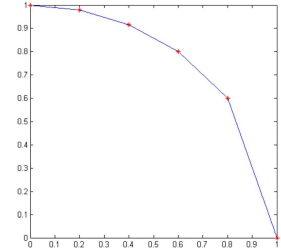
# Rappresentazione

## Rappresentazione Non Parametrica

Le coordinate del punto che si muove lungo la curva sono legate attraverso una *trasformazione geometrica* che porta *univocamente da x ad y*.

Per caratterizzare una curva abbiamo bisogno di calcolare i **coefficienti**, dunque una qualunque trasformazione geometrica sulla curva può essere ottenuta applicando la trasformazione al vettore dei coefficienti.

$$y = \sqrt{1 - x^2}$$



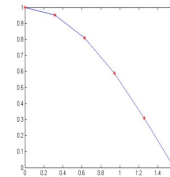
**Con questo tipo di rappresentazione non risulta possibile rappresentare le curve chiuse o con punti multipli**

## Rappresentazione Parametrica

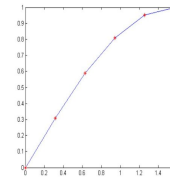
Esiste un parametro, indicato con  $t$ , che definisce il punto di coordinate  $(x(t), y(t))$  che si muove sulla curva al variare della funzione di definizione.

Inoltre è possibile descrivere questa rappresentazione con l'utilizzo di **polinomi** che sono molto flessibili.

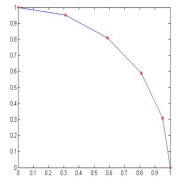
$$\begin{cases} x(t) = \cos(t) \\ y(t) = \sin(t) \end{cases} \quad t \in \left[0, \frac{\pi}{2}\right]$$



$x = \cos(t)$



$y = \sin(t)$



$C(t) = (\cos(t), \sin(t))$

**Con questo tipo di rappresentazione permette di rappresentare le curve chiuse o con punti multipli**



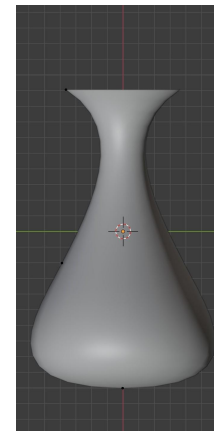
# Curve di Bezier

Le curve di Bézier sono usate per disegnare curve nello spazio a partire da un numero finito di punti.

## Proprietà

Data una successione di punti  $\{P_i\}_{i=0,\dots,n}$  la Curva di Bézier:

- *deve interpolare il primo punto  $P_0$  e l'ultimo punto  $P_n$ ;*
- *i suoi punti devono trovarsi interamente all'interno dell'involucro convesso;*
- *è tangente in  $P_0$  al segmento  $P_0P_1$ ;*
- *è tangente in  $P_n$  al segmento  $P_{n-1}P_n$ ;*
- *la curvatura in  $P_0$  dipende da  $P_0P_1P_2$ ;*
- *la curvatura in  $P_n$  dipende da  $P_{n-2}P_{n-1}P_n$ ;*
- *la derivata  $k$ -esima in  $P_0$  dipende dai primi  $k+1$  punti;*
- *la derivata  $k$ -esima in  $P_n$  dipende dagli ultimi  $k+1$  punti.*



Fissati  $n+1$  punti di controllo, il grado del polinomio che definisce la curva di Bézier è  $n$ .

**Aumentando i punti migliora il modello approssimante realizzato?**

**No, i polinomi tendono ad oscillare sempre di più all'aumentare del grado**

**Fenomeno di Runge**



# Bernstein

**Problema:**

$$r_n(x) = f(x) - p_n(x)$$

**Converge all'infinito all'aumentare del grado**

**Soluzione → Convergenza a zero**

Utilizzo come funzioni base polinomi a tratti di grado  $n$ .

Per ogni punto di controllo si costruisce un polinomio, generando in tal modo  $(n+1)^2$  condizioni per costruire le curve di Bezier.

**Polinomi di Bernstein**

Dati un insieme di punti  $\{P_i\}_{i=0,\dots,n}$

La curva di Bézier è

$$B(t) = \sum_{i=0}^n \binom{n}{i} P_i (1-t)^{n-i} t^i \quad t \in [0, 1]$$

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

**Polinomi di base di Bernstein**

# Algoritmo di de Casteljau

Metodo ricorsivo per valutare il polinomio della forma di Bernstein.  
Numericamente **stabile** ma lento.

Dati 2 punti,  $P_0$  e  $P_1$ , il segmento che li unisce è rappresentato dall'equazione parametrica:

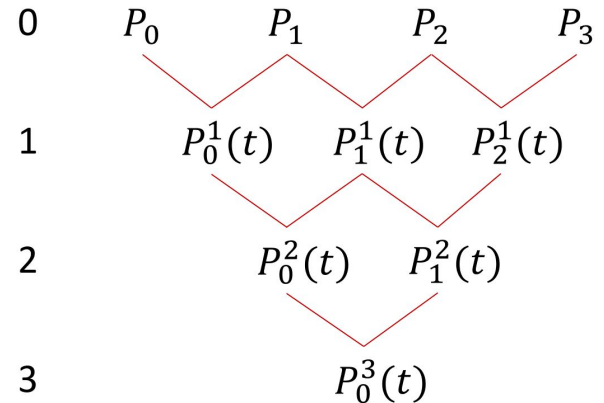
$$B(t) = (1 - t)P_0 + tP_1 \quad t \in [0, 1]$$

Definiti  $n+1$  punti  $P_0, P_1, \dots, P_n$ , poniamo i polinomi intermedi:

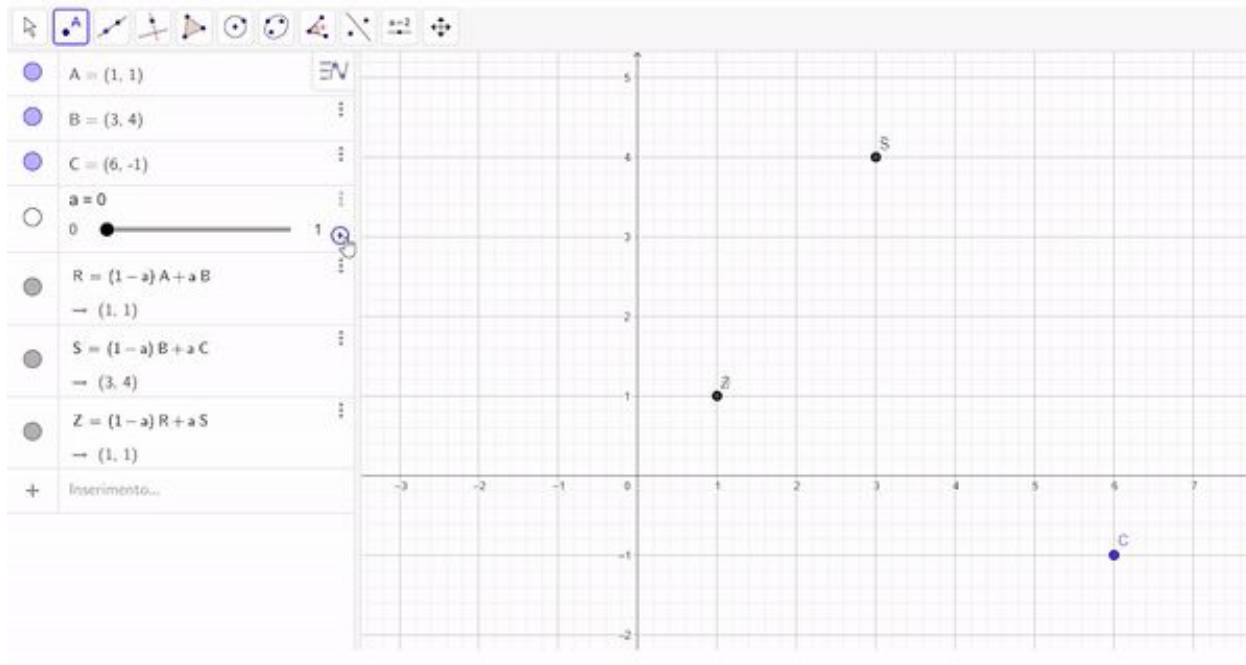
$$P_i^j = (1 - t)P_i^{j-1} + tP_{i+1}^{j-1} \quad i=0, \dots, n-j, \quad j=1, \dots, n$$

La curva finale di Bézier è a questo punto data da:

$$B(t) = P_0^n(t) = (1 - t)P_0^{n-1}(t) + tP_1^{n-1}(t)$$



# Algoritmo di de Casteljau



# Fenomeno di Runge

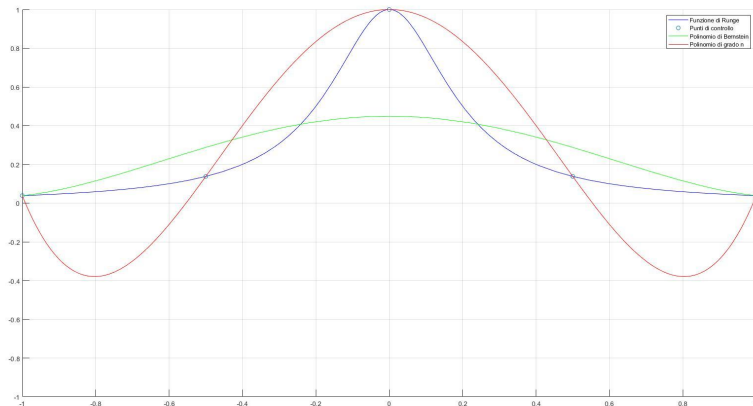
Il fenomeno di Runge è un problema relativo all'interpolazione polinomiale in cui l'errore di interpolazione diverge al crescere del numero dei nodi  $n$ :

$$\lim_{n \rightarrow +\infty} \left( \max_{x \in [-1,1]} |f(x) - P_n(x)| \right) = +\infty.$$

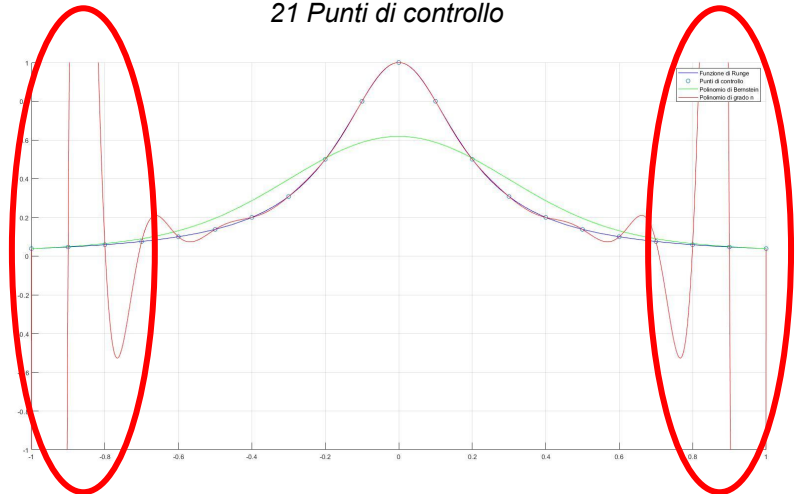
Tanto più aumenta il grado del polinomio interpolante tanto più la funzione oscilla, facendo tendere l'errore d'interpolazione ad infinito.

$$g(x) = 1/(1 + 25x^2) \quad x \in [-1, 1]$$

5 Punti di controllo



21 Punti di controllo

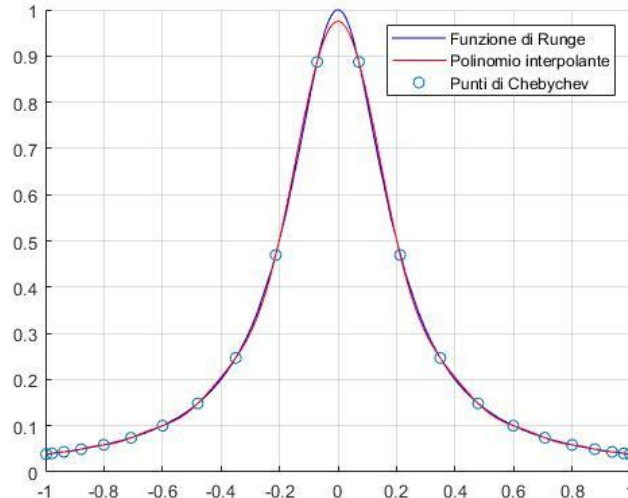


# Nodi di Chebyshev

Per limitare le oscillazioni generate dall'utilizzo dei polinomi interpolanti su nodi equidistanti, Chebyshev propose l'uso di nodi così definiti:

$$x_k = \cos\left(\frac{2k+1}{n+1} \frac{\pi}{2}\right)$$

Questi nodi sono anche definiti come zeri del polinomio di Chebyshev.



***Nodi non  
equidistanziati***

***Non è presente il fenomeno di Runge***

# Funzioni polinomiali interpolanti a tratti

Sia dato un intervallo  $[a, b]$  e sia  $a = t_0 < t_1 < \dots < t_n = b$ .

Una funzione  $f(t)$  si dice *polinomiale a tratti* se la sua restrizione ad ogni intervallo  $[t_k, t_{k+1}]$ , con  $k = 0, \dots, n-1$ , è un polinomio.

## Interpolante polinomiale a tratti di grado $s$

Siano  $x_0 = a < t_1 < \dots < t_n = b$  e si supponga che sia:

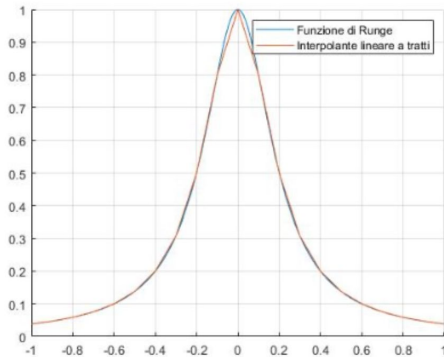
$$t_k = x_{k \cdot s} < x_{k \cdot s + 1} < \dots < x_{k \cdot (s+1) - 1} < x_{k \cdot (s+1)} = t_{k+1}$$

Si consideri la funzione  $p_s^*$  tale che in ogni sottointervallo  $[t_k, t_{k+1}]$ , con  $k = 0, \dots, n$ , sia il polinomio di grado  $s$  che interpola i punti

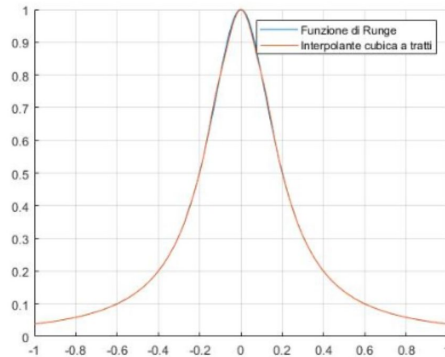
$x_{ks}, \dots, x_{(k+1)s}$  con ordinata  $y_{ks}, \dots, y_{(k+1)s}$ .

Tale  $p_s^*$  si chiama funzione polinomiale a tratti di grado  $s$ , interpolante le coppie  $(x_j, y_j)_{j=0, \dots, n}$ .

*In generale, i punti di raccordo  $x_s, x_{2s}, \dots$  sono punti angolosi della funzione interpolante.*



(a) Interpolazione lineare a tratti



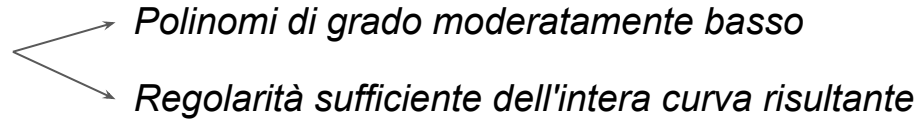
(b) Interpolazione cubica a tratti

**Punti di raccordo  $\rightarrow$  Punti angolosi**

# Spline

Le funzioni che permettono di costruire:

**Modello interpolante ottimale**



Il termine inglese spline significa *listello di legno*; originariamente la spline costituiva uno strumento di disegno formato da lunghe fettucce elastiche che consentiva di tracciare curve regolari per i profili di aerei, navi ...



Sia  $a = x_0 < x_1 < \dots < x_n = b$  una suddivisione dell'intervallo  $[a, b]$ . Una funzione **spline** di grado  $p$  con nodi  $\{x_i\}_{i=0,1,\dots,n}$  è una funzione indicata con  $s_p(x)$  tale che nell'intervallo  $[a, b]$  si ha:

- In ogni sottointervallo di nodi  $[x_i, x_{i+1}]$  con  $i = 0, 1, \dots, n-1$ , la funzione  $s_p(x)$  è un polinomio di grado  $p$ ;
- $s_p(x)$  e le sue prime  $p-1$  derivate sono continue.

**Regolarità nei punti di raccordo:**

$$s_{p,j-1}^{(k)}(x_j) = s_{pj}^{(k)}(x_j) \quad j = 1, \dots, n-1, \quad k = 0, \dots, p-1$$



# Spline cubica interpolante

Dato un insieme di nodi  $K = \{x_0 < x_1 < \dots < x_n\}$  e un insieme di valori  $\{y_i\}_{i=0, \dots, n}$ , determinare la spline cubica  $s_3(x)$  tale che  $s_3(x_i) = y_i$  con  $i=0, \dots, n$  significa determinare in ogni intervallo  $[x_i, x_{i+1}]$  il polinomio di III grado  $s_{3,i}(x)$  che rappresenta la spline, ossia determinare ***i suoi coefficienti in una base di rappresentazione***.

**4 coefficienti per ogni intervallo  $\rightarrow 4n$  incognite da ricavare**

Nella spline cubica si hanno le seguenti condizioni sulla funzione:

- $n + 1$  condizioni di interpolazione  $s_3(x_i) = y_i$ ;
- $3(n-1)$  condizioni di continuità  $\longrightarrow s_{3,i-1}^k(x_i) = s_{3,i}^k(x_i) \quad i = 1, \dots, n-1; \quad k = 0, 1, 2$

Il numero totale di condizioni è dunque:

$$n + 1 + 3(n-1) = 4n-2 \neq 4n \quad \text{Servono due condizioni per determinare univocamente le spline}$$

**Due condizioni al contorno**  $\longrightarrow s^{(2)}(x_1) = s^{(2)}(x_n) = 0$  **Spline cubica naturale**



# B-Spline

Le B-Spline sono funzioni spline a **supporto compatto** *linearmente indipendenti* così definite:

$$B_{i,k}(t) \quad i=0, \dots, n \quad k \text{ è il grado del polinomio ed } i \text{ il nodo considerando}$$

Al variare del grado del polinomio, il supporto compatto è così caratterizzato:

$$B_{i,k} \text{ ha supporto compatto in } [t_i, t_{i+1+k})$$

**Due categorie:**

- **B-Spline uniformi:** gli intervalli sono di lunghezza unitaria.
- **B-Spline non uniformi:** gli intervalli sono di lunghezza differente.

La differenza sostanziale tra le Spline e le B-Spline sta nel fatto che le quest'ultime possono essere funzioni diverse in base all'intervallo, mentre le spline hanno l'obbligo di essere identiche nei vari sottointervalli.

**Ciò permette un migliore adattamento ai dati che si stanno rappresentando.**

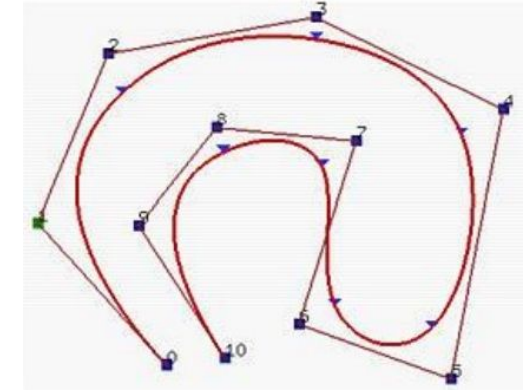
# B-Spline

Dati  $n+1$  punti di controllo  $P = \{P_i\}_{i=0, \dots, n}$  e un vettore di  $m+1$  nodi  $T = [t_0, t_1, \dots, t_m]$ , la curva B-Spline  $C(t)$  definita da  $P$  e  $T$  è:

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

$\downarrow$   
**Funzione B-spline di grado  $k$**   
**definita in  $t \in [t_i, t_{i+h+1}] \rightarrow$  *Nodi***

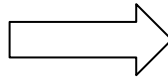
**Curve B-Spline**  
→ **Punti di Controllo**



$k = 3$  grado  $n+1 = 11$  punti di controllo  
 $n+k+2 = 15$  nodi

Sia  $T = (t_0, t_1, \dots)$  il vettore dei nodi si hanno i seguenti parametri:

- $n+1$  punti di controllo  $P_0, \dots, P_n$ ;
- grado  $k$ ;
- vettore dei nodi;
- $T = (t_0, t_1, \dots, t_{n+k+1}), t_i < t_{i+1}$

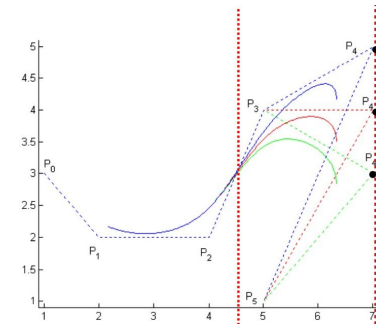
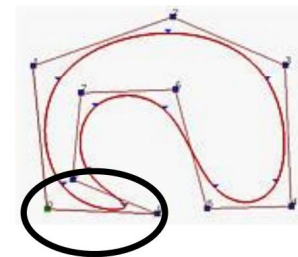


$$\#nodi = \#punti + grado + 1$$

# B-Spline

## Proprietà

1. Deve essere soddisfatta la relazione  $m=n+p+1$ . Ogni punto di controllo ha bisogno di una funzione base e il numero di funzioni base soddisfa  $m=n+p+1$ .
2. La curva non è chiusa ma può diventarlo se il punto di controllo  $P_n$  coincide con  $P_0$ .
3. Come conseguenza del supporto compatto, se  $P_j$  cambia,  $C(u)$  risulta modificata solo in corrispondenza dei punti di definizione della B-spline  $B_{jk}$ . (Controllo locale)
4. In generale la curva non interpola alcun punto di controllo.
5. Possiamo modificare la curva in modo tale che sia tangente al primo e all'ultimo segmento nel primo e nell'ultimo punto di controllo, rispettivamente. Per far ciò, il primo e l'ultimo nodo devono avere molteplicità  $k+1$ .
6. All'aumentare della molteplicità di un punto di controllo, la curva si avvicina al punto multiplo, fino ad interpolarlo.





# Confronto

	Bézier	B-SPLINE
Contenuta nel guscio convesso del poligono di controllo?	Sì	Sì
Intervallo $t$ ?	$[0, 1]$	$[a, b]$
Grado?	Numero vertici poligono - 1	Indipendente dal poligono di controllo
Passa per i punti del poligono di controllo?	Solo per il primo e l'ultimo	In generale no, ma collassando tanti nodi quanto l'ordine della curva, può passare per qualsiasi punto del poligono
La forma della curva rispecchia quella del poligono di controllo?	No, solo approssimativamente	Sì. Con particolari accorgimenti può anche coincidere col poligono di controllo
Trasformazione affine sulla curva?	Basta operare sul poligono di controllo	Basta operare sul poligono di controllo
Controllo locale?	No: se si modifica un punto del poligono di controllo, cambia tutta la curva	Sì: se si modifica un punto del poligono di controllo, cambia solo il tratto di curva relativo
Si possono disegnare circonferenze?	No	Sì, con curve razionali

***Grazie per l'attenzione!***