

## 4.1 Interpolazione, approssimazione, modellazione

Il problema generale è quello di determinare un'espressione analitica o grafica per una funzione  $f(x)$  di cui si conoscono un numero finito di punti del grafico  $(x_i, y_i)$ .

Quindi si cerca una funzione  $f(x)$  tale che

$$f(x_0) = y_0 \quad ; \quad f(x_1) = y_1 \quad ; \quad \dots \quad ; \quad f(x_n) = y_n$$

$$\begin{array}{l|l} x_0 & y_0 \\ x_1 & y_1 \\ x_2 & y_2 \\ \dots & \dots \\ x_n & y_n \end{array}$$

Si vuole che la  $f(x)$  sia *facilmente calcolabile* e che soddisfi le  $n + 1$  eguaglianze o *precisamente* o *approssimativamente* o *modellandosi* su di esse secondo un concetto che vedremo più avanti. Il problema si pone di solito in uno di questi due casi

- I dati sono ottenuti sperimentalmente, per cui  $f(x)$  è da costruire.
- La  $f(x)$  è nota ed è possibile calcolarla anche in altri punti, ma non è facilmente calcolabile (per esempio è la soluzione numerica di un'equazione differenziale) o la sua espressione è comunque assai complessa.

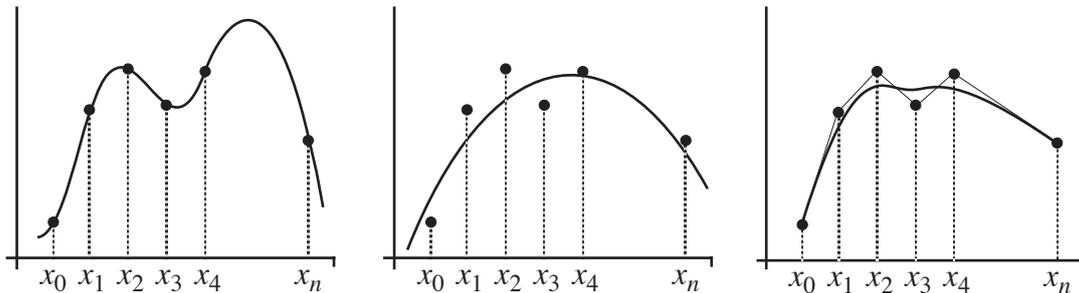
Come detto, le tecniche sono sostanzialmente tre: l'interpolazione, l'approssimazione e la modellazione. Ognuna di esse ha parecchie varianti che possono condurre a risultati diversi. Cerchiamo di dare per ora una definizione intuitiva, che poi approfondiremo, delle tre tecniche.

*Interpolare* significa determinare una funzione che soddisfi precisamente i dati.

*Approssimare* significa determinare una funzione che non soddisfi precisamente i dati, ma se ne discosti il meno possibile.

*Modellare* significa grosso modo determinare una funzione che nel modo più dolce "si inserisca nella poligonale dei dati".

Nel disegno sotto gli stessi dati interpolati, approssimati e "modellati" con qualche tecnica.



### 4.1.1 Interpolazione polinomiale, la matrice di Vandermonde

La prima idea è quella di determinare una funzione polinomiale

$$P(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$$

con  $P(x)$  polinomio di grado minore o uguale a  $n$ . Il polinomio si dirà *polinomio interpolatore dei dati*.

Introducendo i dati si ottiene:

$$\left. \begin{array}{l} P(x_0) = y_0 \Rightarrow a_0 + a_1x_0 + a_2x_0^2 + \dots + a_nx_0^n = y_0 \\ \dots \\ P(x_n) = y_n \Rightarrow a_0 + a_1x_n + a_2x_n^2 + \dots + a_nx_n^n = y_n \end{array} \right\}$$

Questo è un sistema lineare  $(n + 1) \times (n + 1)$  nelle incognite  $a_0, \dots, a_n$ , la cui matrice dei coefficienti è detta *matrice di Vandermonde* della successione  $x_0, \dots, x_n$ . Questa matrice ha determinante diverso da zero se gli  $x_i$  sono distinti; pertanto in tal caso esiste un'unico polinomio di grado minore o uguale a  $n$  che soddisfa i dati (non è detto che abbia grado esattamente  $n$  perché non è detto che si abbia  $a_n \neq 0$ ).

Risolvere il sistema lineare non è però conveniente dal punto di vista calcolativo, sia per la mole dei conti, sia perché la matrice di Vandermonde è particolarmente sensibile agli errori da arrotondamento avendo un numero di condizionamento elevato.

**Esempio 4.1:** Determinare la parabola  $y = a + bx + cx^2$  passante per tre punti  $(x_0, y_0)$ ,  $(x_1, y_1)$ ,  $(x_2, y_2)$  ( $x_i$  distinti). Si ha il sistema lineare nelle incognite  $a, b, c$ :

$$\begin{cases} y_0 = a + bx_0 + cx_0^2 \\ y_1 = a + bx_1 + cx_1^2 \\ y_2 = a + bx_2 + cx_2^2 \end{cases} \quad \text{associato alla matrice} \quad \left( \begin{array}{ccc|c} 1 & x_0 & x_0^2 & y_0 \\ 1 & x_1 & x_1^2 & y_1 \\ 1 & x_2 & x_2^2 & y_2 \end{array} \right)$$

La matrice dei coefficienti è la matrice di Vandermonde della successione  $x_0, x_1, x_2$ .

La soluzione  $(a, b, c)$  del sistema fornisce i coefficienti del polinomio. Ricavare il polinomio in questo modo è elementare, ma presenta dei problemi.

Si può per esempio calcolare che per  $x_0 = 1$ ,  $x_1 = 2$ ,  $x_2 = 3$  la matrice di Vandermonde ha già numero di condizionamento circa 70.92.

### 4.1.2 Interpolazione polinomiale, il polinomio di Lagrange

Esiste una semplicissima formula dovuta a Lagrange per determinare il polinomio in questione:

$$P(x) = y_0 \frac{(x-x_1)(x-x_2)\cdots(x-x_n)}{(x_0-x_1)(x_0-x_2)\cdots(x_0-x_n)} + y_1 \frac{(x-x_0)(x-x_2)\cdots(x-x_n)}{(x_1-x_0)(x_1-x_2)\cdots(x_1-x_n)} + \cdots + y_n \frac{(x-x_0)\cdots(x-x_{n-1})}{(x_n-x_0)\cdots(x_n-x_{n-1})}$$

È evidente che il polinomio ha grado non superiore a  $n$  ed è pure evidente il fatto che esso soddisfa i dati.

La formula di Lagrange, benché elegante ed elementare, non è in generale di uso pratico. Il polinomio non è infatti scritto in una forma che si presti a una semplice algoritmizzazione tipo schema di Ruffini-Hörner per calcolare il polinomio in un punto diverso dagli  $x_i$ .

### 4.1.3 Interpolazione polinomiale, il polinomio di Newton

Esiste un'altra formula, dovuta a Newton, per determinare il polinomio in modo algoritmico ed è la seguente:

$$P(x) = b_0 + b_1(x-x_0) + b_2(x-x_0)(x-x_1) + \cdots + b_n(x-x_0)(x-x_1)\cdots(x-x_{n-1})$$

Prima di spiegare come si calcolano i coefficienti  $b_i$ , osserviamo che, dopo averli determinati, è facile calcolare  $P(x)$  in qualunque punto *senza sviluppare la formula*, in modo simile allo schema di Ruffini-Hörner:

$$P(x) = b_0 + (x-x_0)\left(b_1 + b_2(x-x_1) + \cdots + b_n(x-x_1)\cdots(x-x_{n-1})\right)$$

$$P(x) = b_0 + (x-x_0)\left(b_1 + (x-x_1)\left(b_2 + \cdots + b_n(x-x_2)\cdots(x-x_{n-1})\right)\right) \text{ etc.}$$

Per quanto riguarda i coefficienti  $b_i$ , un conto non difficile, ma laborioso, mostra che essi si possono determinare ricorsivamente nel modo seguente:

$$\begin{aligned} b_0 &= f(x_0) && \stackrel{\text{def}}{=} f[x_0] \\ b_1 &= \frac{f(x_1) - f(x_0)}{x_1 - x_0} && \stackrel{\text{def}}{=} f[x_1, x_0] \\ b_2 &= \frac{f[x_2, x_1] - f[x_1, x_0]}{x_2 - x_0} && \stackrel{\text{def}}{=} f[x_2, x_1, x_0] \\ &\dots && \\ b_n &= \frac{f[x_n, x_{n-1}, \dots, x_1] - f[x_{n-1}, \dots, x_0]}{x_n - x_0} && \stackrel{\text{def}}{=} f[x_n, x_{n-1}, \dots, x_0] \end{aligned}$$

I  $b_i$  si calcolano quindi in modo algoritmico mediante un procedimento detto *calcolo alle differenze finite*. Esplicitiamo l'algoritmo nel caso in cui gli  $x_i$  formino una progressione aritmetica di ragione costante  $d$  cioè si abbia:

$$x_0 \quad x_1 = x_0 + d \quad x_2 = x_1 + d \quad \dots \quad x_n = x_{n-1} + d$$

In questo caso si possono calcolare i  $b_i$  usando lo schema

$$\begin{array}{l} y_0 = y_0^{(0)} \\ y_1 \\ y_2 \\ \dots \\ y_{n-1} \\ y_n \end{array} \left| \begin{array}{l} y_1 - y_0 = y_0^{(1)} \\ y_2 - y_1 = y_1^{(1)} \\ y_3 - y_2 = y_2^{(1)} \\ \dots \\ y_n - y_{n-1} = y_{n-1}^{(1)} \end{array} \right| \begin{array}{l} y_1^{(1)} - y_0^{(1)} = y_0^{(2)} \\ y_2^{(1)} - y_1^{(1)} = y_1^{(2)} \\ \dots \end{array} \left| \dots \right| \begin{array}{l} \dots \\ \dots \\ \dots \end{array} \left| \begin{array}{l} y_1^{(n-1)} - y_0^{(n-1)} = y_0^{(n)} \end{array} \right|$$

e, come si verifica subito, si ha:

$$b_0 = y_0^{(0)} ; b_1 = \frac{y_0^{(1)}}{1! d^1} ; b_2 = \frac{y_0^{(2)}}{2! d^2} ; \dots ; b_n = \frac{y_0^{(n)}}{n! d^n}$$

**Esempio 4.2:** Determiniamo il polinomio  $P(x)$  di grado minore o uguale a 3 tale che:

$$P(1) = 1 \quad P(1.5) = 3 \quad P(2) = 4 \quad P(2.5) = -1$$

In questo caso  $d = 0.5$ . Lo schema delle differenze finite unitarie è:

$$\begin{array}{l} \mathbf{1} \\ 3 \\ 4 \\ -1 \end{array} \left| \begin{array}{l} 3 - 1 = \mathbf{2} \\ 4 - 3 = \mathbf{1} \\ -1 - 4 = -5 \end{array} \right| \begin{array}{l} 1 - 2 = -\mathbf{1} \\ -5 - 1 = -6 \end{array} \left| -6 + 1 = -\mathbf{5} \right.$$

Si ha pertanto

$$b_0 = \mathbf{1} ; b_1 = \frac{\mathbf{2}}{1!(0.5)} = 4 ; b_2 = \frac{-\mathbf{1}}{2!(0.5)^2} = -2 ; b_3 = \frac{-\mathbf{5}}{3!(0.5)^3} = -20/3$$

Il polinomio è quindi:

$$P(x) = 1 + 4(x - 1) - 2(x - 1)(x - 1.5) - \frac{20}{3}(x - 1)(x - 1.5)(x - 2)$$

Se la successione non è a passo costante, lo schema delle differenze finite subisce una semplice modifica che non riportiamo in questa sede.

#### 4.1.4 Il resto nell'interpolazione di Newton

È evidente l'analogia tra la formula di interpolazione di Newton e la nota formula di Taylor. In effetti, come il polinomio di Taylor approssima una funzione con un polinomio avente stesso valore e stesse derivate in un punto  $x_0$ , il polinomio di Newton approssima una funzione con un polinomio che assume nei punti  $x_i$  gli stessi valori della funzione.

In analogia alla formula del resto di Lagrange per il polinomio di Taylor, si ha:

**Proposizione 1** Sia  $f(x)$  una funzione continua nell'intervallo  $[a, b]$ , ivi dotata di derivate continue fino all'ordine  $n + 1$ .

Se  $a = x_0 < x_1 < \dots < x_n = b$  è una suddivisione dell'intervallo e  $P(x)$  è il polinomio di Newton che interpola  $f(x)$  nei punti  $x_i$  (nel senso che  $P(x_i) = f(x_i)$  per ogni  $i$ ), allora, per ogni  $x \in [a, b]$  esiste un punto  $\xi$  nell'intervallo  $[a, b]$  tale che:

$$f(x) = P(x) + \frac{f^{(n+1)}(\xi)}{(n + 1)!} (x - x_0) \dots (x - x_n)$$

### 4.1.5 Interpolazione “spline”

L'interpolazione polinomiale può non essere conveniente per vari motivi. Il primo è che per una lunga serie di dati il polinomio risulta di grado troppo alto, il secondo è che il problema è comunque mal condizionato, nel senso che basta una piccola variazione dei dati per cambiare anche di parecchio i coefficienti del polinomio.

Un modo spesso più efficiente e perciò maggiormente diffuso per interpolare una serie di dati consiste nell'usare una funzione definita a pezzi i cui pezzi siano polinomi di grado basso. Questa interpolazione si dice “spline” (dal nome inglese delle bacchette di legno flessibile usate per l'interpolazione meccanica di una serie di dati).

Si abbia la solita serie di dati da interpolare:

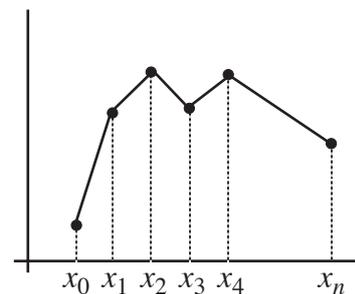
$$f(x_0) = y_0, \dots, f(x_n) = y_n \quad \text{con} \quad x_0 < \dots < x_n$$

Descriveremo tre tipi di interpolazione spline.

#### Interpolazione spline lineare

La più semplice interpolazione spline è quella mediante funzioni lineari. Si può scrivere per ogni  $i$  ( $1 \leq i \leq n$ ) l'equazione della retta passante per i due punti  $(x_{i-1}, y_{i-1})$   $(x_i, y_i)$ .

La funzione così definita a pezzi sugli intervalli  $[x_{i-1}, x_i]$  è evidentemente continua in  $[x_0, x_n]$  e soddisfa le condizioni date.



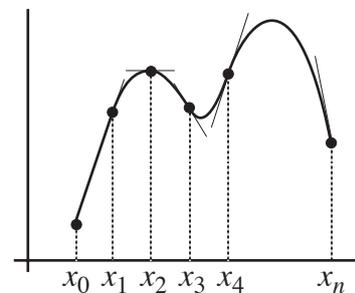
#### Interpolazione spline quadratica

Scriviamo per ogni  $i$  ( $1 \leq i \leq n$ ) le parabole e cioè le funzioni del tipo  $y = a_i + b_i x + c_i x^2$  passanti per i due punti  $(x_{i-1}, y_{i-1})$   $(x_i, y_i)$ . Ne esistono  $\infty^1$  per ogni  $i$ , quindi  $n$  dei  $3n$  coefficienti sono arbitrari. Si può approfittare di questo fatto per imporre che le derivate prime delle parabole coincidano nei punti  $x_1, \dots, x_{n-1}$  e quindi la funzione sia dotata di derivata prima.

Queste sono  $n - 1$  condizioni su  $n$  parametri. Resta pertanto una scelta arbitraria. È uso imporre che la parabola dell'intervallo  $[x_0, x_1]$  degeneri in una retta perché questo modo è facile costruire successivamente le varie parabole.

Si ha pertanto una funzione definita a pezzi sugli intervalli  $[x_{i-1}, x_i]$  che è evidentemente continua e derivabile in  $[x_0, x_n]$  e soddisfa i dati.

La spline quadratica non è molto usata perché spesso fornisce un risultato “saltellante” e quindi poco soddisfacente.



#### Interpolazione spline cubica

La più usata delle interpolazioni spline è quella con funzioni polinomiali di grado tre in quanto consente un calcolo semplice e una approssimazione più che soddisfacente. Inoltre si riesce a fare in modo che la funzione sia di classe  $C^2$ .

Si determina per ogni  $i$  ( $1 \leq i \leq n$ ) la funzione cubica ovvero del tipo  $y = a_i + b_i x + c_i x^2 + d_i x^3$  passante per i due punti  $(x_{i-1}, y_{i-1})$   $(x_i, y_i)$ . Il problema ha  $\infty^2$  soluzioni per ogni  $i$  ( $i = 1, \dots, n$ ), quindi  $2n$  dei  $4n$  coefficienti sono arbitrari.

Imponendo che sia le derivate prime che quelle seconde coincidano nei punti  $x_1, \dots, x_{n-1}$  si hanno altre  $2n - 2$  condizioni lineari; rimangono ancora due scelte arbitrarie ed è uso imporre che la prima e l'ultima parabola cubica abbiano un flesso rispettivamente in  $x_0$  e in  $x_n$  (spline naturale). A volte si danno due condizioni sulle derivate prime in  $x_0$  e in  $x_n$  (spline vincolata).

Si ha pertanto una funzione definita a pezzi sugli intervalli  $[x_{i-1}, x_i]$  che è evidentemente continua e derivabile due volte in  $[x_0, x_n]$  e soddisfa i dati.

Accenniamo brevemente al procedimento per calcolare in maniera relativamente veloce la spline cubica naturale.

L'idea base è quella di fare in modo che le incognite siano solo le derivate seconde delle parabole cubiche nei punti  $x_1, \dots, x_{n-1}$ . Poniamo per semplicità di notazione:

$$q_1 = f''(x_1), \dots, q_{n-1} = f''(x_{n-1})$$

Scriviamo la *derivata seconda* della funzione cubica  $f_i(x)$  che congiunge il punto  $(x_{i-1}, y_{i-1})$  col punto  $(x_i, y_i)$ . È una funzione lineare che possiamo scrivere così (usando la formula di Lagrange) in modo da evidenziare i valori che la derivata seconda stessa assume nei punti  $x_i$ :

$$f_i''(x) = q_{i-1} \frac{x - x_i}{x_{i-1} - x_i} + q_i \frac{x - x_{i-1}}{x_i - x_{i-1}}$$

Integriamo due volte rispetto a  $x$  e scriviamo opportunamente le due costanti di integrazione  $h_i$  e  $k_i$  ottenendo così le  $f_i(x)$ :

$$f_i(x) = \frac{q_{i-1}(x - x_i)^3}{6(x_{i-1} - x_i)} + \frac{q_i(x - x_{i-1})^3}{6(x_i - x_{i-1})} + h_i(x_i - x) + k_i(x - x_{i-1})$$

Le due costanti così scritte  $h_i$  e  $k_i$  si determinano imponendo che  $f_i(x_i) = y_i$  e che  $f_i(x_{i-1}) = y_{i-1}$ . Svolti i conti si ottiene:

$$h_i = \frac{y_{i-1}}{x_i - x_{i-1}} - \frac{q_{i-1}(x_i - x_{i-1})}{6} \quad k_i = \frac{y_i}{x_i - x_{i-1}} - \frac{q_i(x_i - x_{i-1})}{6}$$

Rimangono da determinare tutti i  $q_i$ . Imponendo che le derivate prime coincidano in tutti i punti  $x_i$  ( $i \neq 0, n$ ) si ottiene:

$$(x_i - x_{i-1})q_{i-1} + 2(x_{i+1} - x_{i-1})q_i + (x_{i+1} - x_i)q_{i+1} = 6 \left( \frac{y_{i+1} - y_i}{x_{i+1} - x_i} + \frac{y_{i-1} - y_i}{x_i - x_{i-1}} \right)$$

Queste sono  $n - 1$  relazioni lineari tra i  $q_i$ . Se si impone che la spline sia naturale, si ha  $q_0 = q_n = 0$ ; le incognite sono quindi  $n - 1$  e la matrice delle  $n - 1$  relazioni lineari è tridiagonale, per cui la risoluzione del sistema è particolarmente agevole. Inoltre la forma delle singole  $f_i(x)$  è particolarmente adatta al calcolo con uno schema tipo Ruffini-Hörner.

Nel caso particolarmente frequente in cui gli  $x_i$  siano in progressione aritmetica di ragione  $d$ , il sistema nelle incognite  $q_1, \dots, q_{n-1}$  è associato alla matrice tridiagonale simmetrica

$$\left( \begin{array}{cccccc|cccc} 4d & d & 0 & 0 & \dots & 0 & 0 & 6(y_0 - 2y_1 + y_2)/d & & \\ d & 4d & d & 0 & \dots & 0 & 0 & 6(y_1 - 2y_2 + y_3)/d & & \\ 0 & d & 4d & d & \dots & 0 & 0 & 6(y_2 - 2y_3 + y_4)/d & & \\ \dots & & & & & & & \dots & & \\ 0 & 0 & 0 & 0 & & d & 4d & 6(y_{n-2} - 2y_{n-1} + y_n)/d & & \end{array} \right)$$

La spline cubica naturale è in un certo senso la miglior funzione di classe  $C^2$  che interpola i dati. Più precisamente si dimostra che:

**Proposizione 2** *Siano  $x_0 < x_1 < \dots < x_n$ . Consideriamo tutte le funzioni  $g(x)$  di classe  $C^2$  nell'intervallo  $[x_0, x_n]$ , tali che  $g(x_i) = y_i$ , per  $i = 1, \dots, n$ ; allora la quantità*

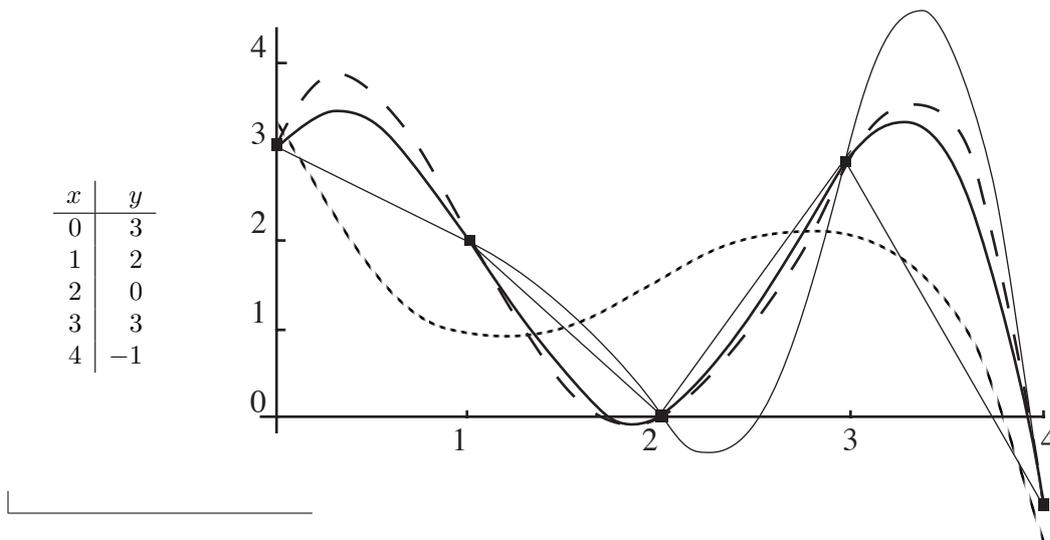
$$\int_{x_0}^{x_n} (g''(x))^2 dx$$

*è minima qualora  $g(x)$  sia la spline cubica naturale.*

**Esempio 4.3:** Un esempio di costruzione di interpolazioni polinomiali con splines o altro comporterebbe solo una serie di lunghi calcoli e non sarebbe di aiuto alla comprensione dei metodi esposti. Ci accontentiamo perciò di fornire un grafico elaborato al calcolatore che illustra varie interpolazioni polinomiali della serie di dati sotto.

Nel grafico (che per motivi di chiarezza non è monometrico), sono disegnati con vari tipo di tratto:

- La spline lineare che interpola i dati.
- La spline quadratica che interpola i dati (il primo tratto coincide con quello lineare)
- La spline cubica che interpola i dati.
- - - Il polinomio di Newton (quindi di grado 4) che interpola i dati.
- ..... Il polinomio di grado 3 che *approssima* i dati ai minimi quadrati (vedi paragrafo successivo).



#### 4.1.6 Approssimazione ai minimi quadrati

Invece di trovare un polinomio *di grado*  $n$  che interpoli  $n + 1$  dati, se ne può trovare uno *di grado inferiore* che non passi esattamente per i punti dati, ma se ne discosti per poco “nel senso dei minimi quadrati”.

Più precisamente non si pretende che il polinomio  $P(x)$  di grado  $d \leq n$  soddisfi precisamente le eguaglianze  $P(x_0) = y_0$  ;  $P(x_1) = y_1$  ;  $\dots$  ;  $P(x_n) = y_n$ , ma ci si accontenta che la quantità

$$\left(P(x_0) - y_0\right)^2 + \left(P(x_1) - y_1\right)^2 + \dots + \left(P(x_n) - y_n\right)^2$$

sia la minima possibile.

Si dimostra che esiste unico un polinomio di grado  $d < n$  con questa proprietà ed è detto polinomio che approssima i dati *ai minimi quadrati*.

Questo modo di approssimare i dati è usato soprattutto quando i dati sono frutto di osservazioni sperimentali e quindi soggetti a probabile errore. Particolarmente noto è il caso in cui il polinomio ha grado 1, quindi si ha l'approssimazione lineare ai minimi quadrati.

Ci sarebbe molto da dire, ma ci accontentiamo di riportare la tecnica più semplice per trovarlo (anche se non è la più numericamente stabile). Come nel caso di Vandermonde si cerca un polinomio

$$P(x) = a_0 + a_1x + a_2x^2 + \dots + a_dx^d$$

di grado minore o uguale a  $d$ . Introducendo i dati si ottiene:

$$\left. \begin{array}{l} P(x_0) = y_0 \Rightarrow a_0 + a_1x_0 + a_2x_0^2 + \dots + a_dx_0^d = y_0 \\ \dots \\ P(x_n) = y_n \Rightarrow a_0 + a_1x_n + a_2x_n^2 + \dots + a_dx_n^d = y_n \end{array} \right\}$$

Osserviamo che questa volta la matrice dei coefficienti  $A$ , dato che  $d \leq n$ , è una matrice rettangolare con più righe che colonne e che quindi il sistema  $Au = y$  è un sistema con più equazioni che incognite

e quindi quasi sicuramente senza soluzioni. La *soluzione ai minimi quadrati* è però l'unica soluzione del sistema quadrato con matrice invertibile

$$A^T A u = A^T y$$

dove la matrice  $A^T A$  è una matrice  $d \times d$  simmetrica. Se il polinomio cercato è di grado 1, la matrice è  $2 \times 2$  e in questo caso la retta è la famosa *regressione lineare* spesso usata in statistica.

**Esempio 4.4:** Determinare la retta e la parabola  $y = a + bx + cx^2$  che approssimino ai minimi quadrati i dati a lato.

|   |   |
|---|---|
| 1 | 0 |
| 3 | 3 |
| 4 | 2 |
| 6 | 3 |

Per l'approssimazione lineare si scrivono la matrice di Vandermonde dei numeri 1, 3, 4, 6 arrestata alla potenza 1 e la matrice dei dati  $y$  in colonna:

$$A = \begin{pmatrix} 1 & 1 \\ 1 & 3 \\ 1 & 4 \\ 1 & 6 \end{pmatrix} \quad y = \begin{pmatrix} 0 \\ 3 \\ 2 \\ 3 \end{pmatrix}$$

Il sistema lineare  $Au = y$  non ha soluzioni, ma il sistema lineare  $2 \times 2$   $(A^T A)u = (A^T y)$  ha l'unica soluzione  $u = (0.1154, 0.5385)$ . Quindi la retta  $y = 0.1154 + 0.5385x$  approssima i dati ai minimi quadrati.

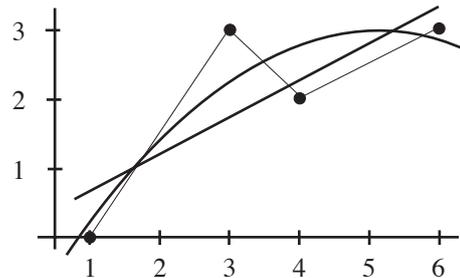
Analogamente, per l'approssimazione quadratica si scrivono la matrice di Vandermonde dei numeri 1, 3, 4, 6 arrestata ai quadrati e la matrice dei dati  $y$  in colonna:

$$A = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 3 & 9 \\ 1 & 4 & 16 \\ 1 & 6 & 36 \end{pmatrix} \quad y = \begin{pmatrix} 0 \\ 3 \\ 2 \\ 3 \end{pmatrix}$$

Come sopra, ma il sistema lineare  $3 \times 3$   $(A^T A)u = (A^T y)$  ha l'unica soluzione  $u = (-1.3846, 1.7051, -0.1667)$ .

Quindi la parabola  $y = -1.3846 + 1.7051x - 0.1667x^2$  approssima i dati ai minimi quadrati.

Nel disegno sono riportati i punti, la retta e la parabola.



## 4.2 Curve di Bézier e B-spline

Se invece di interpolare o approssimare i dati, vogliamo “modellare” una curva sui dati, lo strumento base è quello delle curve di Bézier.

Grosso modo *modellare* significa determinare una curva “dolce” che si inserisca nella poligonale, detta *poligono di controllo* che interpola i punti dati.

### 4.2.1 Polinomi di Bézier

Iniziamo con le *funzioni* polinomiali di Bézier. Esse rappresentano un caso assai particolare e limitato, ma costituiscono in un certo senso il passaggio dall'interpolazione alla modellazione. Queste funzioni “modellano”, *mediante una funzione*, dati del tipo  $f(x_0) = y_0; \dots; f(x_n) = y_n$  nel caso in cui  $x_0, x_1, x_2, \dots$  sia una successione a passo costante.

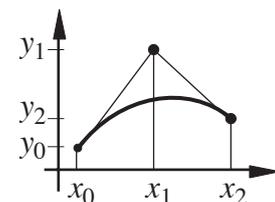
#### Polinomio quadratico di Bézier

Siano  $x_0, x_1, x_2$  tre numeri *equidistanti* e  $y_0, y_1, y_2$  tre numeri qualunque. Chiamiamo  $P_0$  il punto  $(x_0, y_0)$  e così via.

Esiste una e una sola funzione quadratica  $f(x) = ax^2 + bx + c$ , il cui grafico è una parabola passante per  $P_0$ , passante per  $P_2$ , tangente alla retta  $\overline{P_0 P_1}$  in  $P_0$  e tangente alla retta  $\overline{P_1 P_2}$  in  $P_2$ .

In realtà sembra strano che esista una parabola che soddisfi 4 condizioni, perché i coefficienti sono solo tre, ma ciò è dovuto al fatto che  $x_1$  è il punto medio tra  $x_0$  e  $x_2$ .

Se  $P_0, P_1, P_2$  sono allineati, la parabola degenera in una retta.



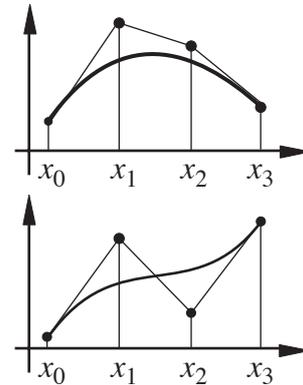
**Polinomio cubico di Bézier**

Siano  $x_0, x_1, x_2, x_3$  quattro numeri *equidistanti* e  $y_0, y_1, y_2, y_3$  quattro numeri.

Esiste una e una sola *funzione cubica*  $f(x) = ax^3 + bx^2 + cx + d$ , passante per i punti  $P_0(x_0, y_0), \dots, P_3(x_3, y_3)$  e tangente alla retta  $\overline{P_0 P_1}$  in  $P_0$  e alla retta  $\overline{P_2 P_3}$  in  $P_3$ .

La cubica è unica perché i coefficienti sono quattro di fronte a quattro condizioni indipendenti e potrebbe anche avere un flesso o degenerare in una parabola quadratica o in una retta.

È possibile costruire allo stesso modo i polinomi di Bézier di grado  $n$  che passano per  $n + 1$  punti *equidistanti* anche se non è facile dare un'interpretazione geometrica delle proprietà di queste curve.



La costruzione dei polinomi di Bézier viene di solito effettuata mediante una delle due tecniche seguenti: quella analitica (i polinomi di Bernstein) o quella grafica (l'algoritmo di de Casteljau).

Ci limitiamo per ora a introdurre i polinomi di Bernstein, riservando la descrizione dell'algoritmo di de Casteljau al caso ben più interessante delle *curve di Bézier*.

**4.2.2 I polinomi di Bernstein**

Sia  $[a, b]$  un intervallo della retta reale e sia  $n \geq 2$ . Definiamo i polinomi di Bernstein di grado  $n$  nell'intervallo  $[a, b]$ .

Gli  $n + 1$  polinomi di Bernstein di grado  $n$  si denotano normalmente con  $B_i^n(x)$ , ma li scriveremo semplicemente  $B_i(x)$  per non appesantire la notazione.

**Definizione:** I polinomi di Bernstein di grado  $n$  nell'intervallo  $[a, b]$  sono

$$B_i(x) = \binom{n}{i} \frac{(b-x)^{n-i}(x-a)^i}{(b-a)^n} \quad i = 0, 1, \dots, n$$

I polinomi di Bernstein di grado  $n$  dipendono dall'intervallo  $[a, b]$  e si ha:

$$B_0(a) = 1 \quad \text{e} \quad B_i(a) = 0 \text{ per } i > 0 \quad \quad B_i(b) = 0 \text{ per } i < n \quad \text{e} \quad B_n(b) = 1$$

I polinomi di Bernstein costituiscono una base per lo spazio vettoriale costituito dai polinomi di grado  $\leq n$ , nel senso che ogni altro polinomio di grado minore o uguale a  $n$  si può scrivere *in modo unico* come loro combinazione lineare.

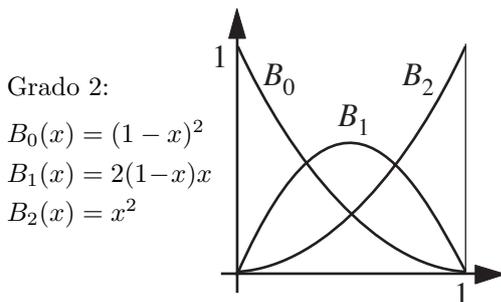
Si ha poi:  $B_0(x) + B_1(x) + \dots + B_n(x) \equiv 1$  (per ogni  $x$ ).

Il polinomio di Bézier di grado  $n$  generato dai punti  $(x_0, y_0), \dots, (x_n, y_n)$  ( $x_i$  equidistanti) è, come si può facilmente verificare, combinazione lineare a coefficienti  $y_0, \dots, y_n$  dei polinomi di Bernstein di grado  $n$  nell'intervallo  $[a, b] = [x_0, x_n]$

$$\text{Bez}(x) = y_0 B_0(x) + \dots + y_n B_n(x)$$

**Osservazione:** Nel seguito, per generare le curve di Bézier, useremo esclusivamente i polinomi di Bernstein nell'intervallo  $[0, 1]$ .

I polinomi di Bernstein di grado 2 e di grado 3 in  $[0, 1]$  sono



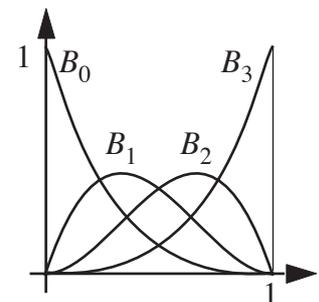
Grado 3:

$$B_0(x) = (1-x)^3$$

$$B_1(x) = 3(1-x)^2x$$

$$B_2(x) = 3(1-x)x^2$$

$$B_3(x) = x^3$$



Osserviamo che:  $B_0(x) + B_1(x) + B_2(x) \equiv 1$  (vale 1 per ogni  $x$ ) nel caso quadratico.

Analogamente:  $B_0(x) + B_1(x) + B_2(x) + B_3(x) \equiv 1$  nel caso cubico.

### 4.2.3 Le curve di Bézier

Svincoliamoci ora dall'ipotesi che  $x_0, x_1, \dots, x_n$  siano equidistanti. Esiste una curva, detta *curva di Bézier*, che modella i punti, anche se si non può pretendere che sia una semplice funzione polinomiale  $y = a_0 + a_1x + \dots + a_nx^n$ .

Occorre lavorare bidimensionalmente ed esprimere la curva in forma parametrica.

Si può supporre che le funzioni  $x(t)$  e  $y(t)$  siano definite nell'intervallo  $[0, 1]$  e che in 0 e in 1 assumano i valori  $(x_0, y_0)$  e  $(x_n, y_n)$  rispettivamente.

La successione dei punti  $P_0(x_0, y_0), \dots, P_n(x_n, y_n)$  verrà detta *poligono di controllo* della curva di Bézier.

A questo punto non è più neanche necessario che gli  $x_i$  siano né ordinati né distinti, basta che siano distinti i punti  $P_i(x_i, y_i)$ .

Siano quindi  $P_0(x_0, y_0), \dots, P_n(x_n, y_n)$ ,  $n + 1$  punti distinti nel piano.

**Definizione:** È detta *curva di Bézier* generata dal poligono di controllo  $(x_0, y_0), \dots, (x_n, y_n)$  la curva avente come rappresentazione parametrica

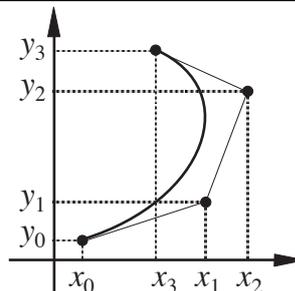
$$\begin{cases} x(t) = x_0B_0(t) + \dots + x_nB_n(t) \\ y(t) = y_0B_0(t) + \dots + y_nB_n(t) \end{cases}$$

dove i  $B_i(t)$  sono i polinomi di Bernstein di grado  $n$  nell'intervallo  $[0, 1]$ .

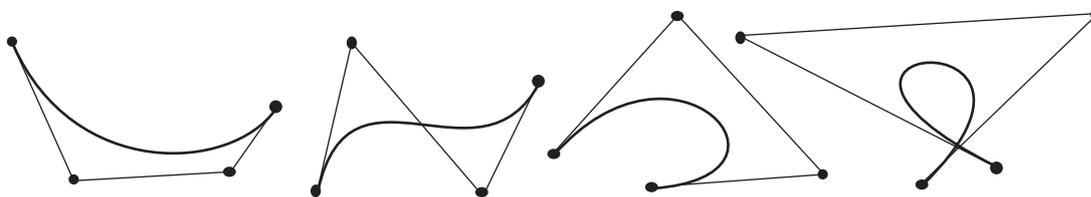
Per esempio nel caso cubico, che è anche uno dei più usati in pratica, il poligono di controllo sarà costituito da quattro punti  $(x_0, y_0), (x_1, y_1), (x_2, y_2), (x_3, y_3)$  e la curva avrà rappresentazione parametrica

$$\begin{cases} x(t) = x_0B_0(t) + x_1B_1(t) + x_2B_2(t) + x_3B_3(t) \\ y(t) = y_0B_0(t) + y_1B_1(t) + y_2B_2(t) + y_3B_3(t) \end{cases} \quad t \in [0, 1]$$

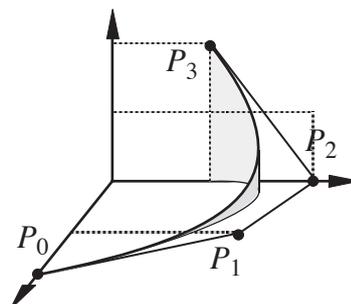
dove i  $B_i(t)$  sono i polinomi cubici di Bernstein definiti nell'intervallo  $[0, 1]$ .



Qui di seguito alcuni esempi di curve di Bézier cubiche con il loro poligono di controllo; la quarta è addirittura nodata, cosa che può capitare se il poligono è intrecciato, del resto anche la terza è nodata, anche se il nodo cade esternamente alla porzione utile.



Questa generalizzazione permette anche di costruire curve di Bézier nello spazio. Si dovrà aggiungere una terza funzione  $z(t)$ , ma tutto funziona esattamente come nel caso planare. Si tenga presente che, mentre una curva di Bézier quadratica è sempre un arco di parabola e perciò una curva piana giacente nel piano dei tre punti del poligono di controllo, una curva di Bézier cubica può essere una curva sghemba e quindi dotata di vera torsione tridimensionale se i quattro punti del poligono di controllo non sono complanari.



### 4.2.4 Algoritmo di de Casteljau per costruire curve di Bézier

L'algoritmo di de Casteljau permette di costruire quanti punti si vuole di una curva di Bézier con un semplice procedimento che ha un'immediata interpretazione grafica.

Il concetto base è semplicemente la parametrizzazione segmentaria della retta. Data una retta  $r$  passante due punti  $A$  e  $B$ , si considera la parametrizzazione

$$P(t) = A + t(B - A)$$

che pone il segmento  $\overline{AB}$  in corrispondenza biunivoca con l'intervallo  $[0, 1]$  e fornisce  $A$  per  $t = 0$  e  $B$  per  $t = 1$ .

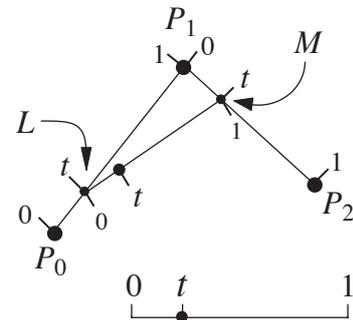
#### Cominciamo col caso quadratico

Il poligono di controllo sarà costituito da tre punti  $P_0, P_1, P_2$ .

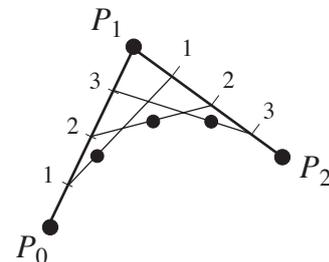
Si pone il segmento  $\overline{P_0 P_1}$  in corrispondenza biunivoca coll'intervallo  $[0, 1]$  in modo che  $P_0$  corrisponda a 0 e  $P_1$  a 1, ovvero con la parametrizzazione  $P(t) = P_0 + t(P_1 - P_0)$ . Allo stesso modo anche  $\overline{P_1 P_2}$  è posto in corrispondenza con  $[0, 1]$ .

Si fissa un numero  $t$  compreso tra 0 e 1 e si considerano sui segmenti  $P_0 P_1$  e  $P_1 P_2$  i due punti corrispondenti a  $t$  che chiamiamo per ora  $L$  e  $M$ .

Si costruisce il segmento che ha come estremi questi due punti  $L$  e  $M$  e lo si pone in corrispondenza biunivoca coll'intervallo  $[0, 1]$ . Il punto del segmento  $LM$  corrispondente a  $t$  fa parte della curva quadratica di Bézier. Facendo variare  $t$  nell'intervallo  $[0, 1]$  si ottengono tutti i punti della parabola.



**Esempio 4.5:** I due segmenti sono stati divisi in 4 parti uguali, quindi si usano i tre valori  $t = 1/4, 2/4, 3/4$  compresi tra 0 e 1. I punti su  $P_0 P_1$  e su  $P_1 P_2$  sono stati chiamati 1, 2, 3. I segmenti  $\overline{11}, \overline{22}, \overline{33}$  sono posti in corrispondenza con  $[0, 1]$  e sul segmento  $\overline{11}$  viene considerato il punto corrispondente a  $t = 1/4$ , sul segmento  $\overline{22}$  il punto corrispondente a  $t = 2/4$ , sul segmento  $\overline{33}$  il punto corrispondente a  $t = 3/4$ . I 5 punti così trovati (si aggiungono i due estremi) appartengono alla parabola di Bézier.



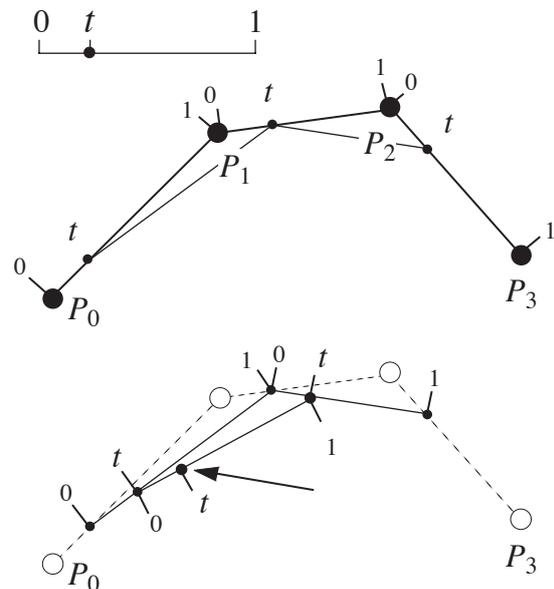
#### Proseguiamo col caso cubico

Il poligono di controllo sarà costituito da quattro punti  $P_0, P_1, P_2, P_3$ .

Come nel caso quadratico, si pongono i segmenti  $P_0 P_1, P_1 P_2, P_2 P_3$  in corrispondenza biunivoca coll'intervallo  $[0, 1]$  in modo che rispettivamente  $P_0$  corrisponda a 0,  $P_1$  a 1 etc.

Si fissa un numero  $t$  compreso tra 0 e 1 e si cercano sui segmenti  $P_0 P_1, P_1 P_2, P_2 P_3$  i tre punti corrispondenti a  $t$ .

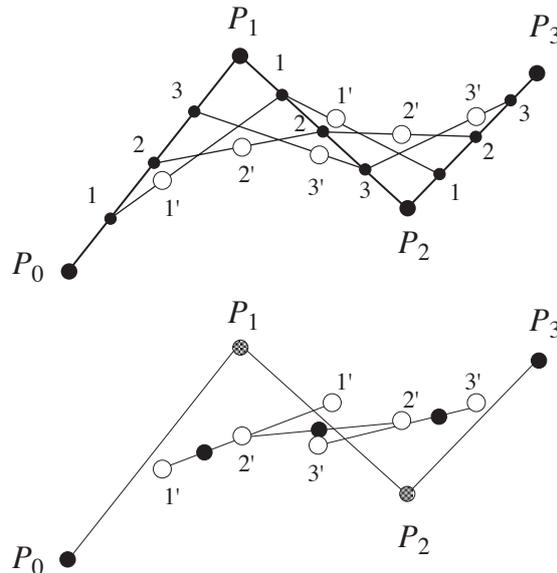
Si costruiscono quindi i due segmenti che hanno come estremi questi tre punti nell'ordine. Come nella seconda figura si pongono i due segmenti in corrispondenza biunivoca coll'intervallo  $[0, 1]$ . A questo punto si prosegue come per l'algoritmo di de Casteljau nel caso quadratico, cercando sui due segmenti i punti corrispondenti a  $t$  e congiungendoli con un segmento che va posto in corrispondenza biunivoca con  $[0, 1]$ . In corrispondenza di questo  $t$  si ha il punto della cubica di Bézier.



**Esempio 4.6:** È dato un poligono di controllo  $P_0, P_1, P_2, P_3$ . Si dividono i lati del poligono di controllo in 4 parti uguali, ovvero si usano tre valori  $t = 1/4, 2/4, 3/4$  compresi tra 0 e 1. I punti vengono chiamati egualmente 1, 2, 3 su ciascuno dei tre segmenti.

I 6 segmenti  $\overline{11}, \overline{22}, \overline{33}, \overline{11}, \overline{22}, \overline{33}$  vengono divisi in 4 parti uguali, ma su ciascuno dei due segmenti  $\overline{11}$  viene considerato il punto corrispondente a  $t = 1/4$  e abbiamo i due punti di nome  $1'$ . Su ciascuno dei due segmenti  $\overline{22}$  viene considerato il punto corrispondente a  $t = 2/4$  e abbiamo i due punti di nome  $2'$ . Su ciascuno dei due segmenti  $\overline{33}$  viene considerato il punto corrispondente a  $t = 3/4$  e abbiamo i due punti di nome  $3'$ . A questo punto consideriamo il secondo disegno identico al primo, ma dove, per chiarezza, sono stati eliminati i segmenti  $\overline{11}$  etc.

I segmenti  $\overline{1'1'}, \overline{2'2'}, \overline{3'3'}$  vengono divisi in 4 parti uguali, ma sul segmento  $\overline{1'1'}$  viene considerato il punto corrispondente a  $t = 1/4$ , sul segmento  $\overline{2'2'}$  il punto per  $t = 2/4$  e sul segmento  $\overline{3'3'}$  quello per  $t = 3/4$ . I 5 punti così trovati (si aggiungono i due estremi  $P_0$  e  $P_3$ ) appartengono alla cubica di Bézier.



#### 4.2.5 Le curve B-spline

Se il poligono da modellare è costituito da molti punti, non conviene costruire una curva di Bézier di grado elevato, ma è meglio costruire diverse curve di Bézier di ordine basso (3 è il più usato) e raccordarle insieme nel modo migliore possibile.

La curva ottenuta in questo modo e che quindi è una curva costituita da varie curve di Bézier è detta *curva B-spline*.

La teoria è assai vasta; ci limitiamo ai due casi più semplici, le B-spline quadratiche uniformi e non uniformi e le B-spline cubiche uniformi e non, avvertendo che anche sulle curve non uniformi si possono fare variazioni di rilievo rispetto alla semplice trattazione che segue.

#### 4.2.6 Le B-spline quadratiche

La cosa più complicata è capire quale uso fare dei dati iniziali, perché la B-spline modella una serie di punti, senza necessariamente passare per essi, ma “raddolcendo” il loro andamento.

Nel caso più elementare sono assegnati  $n + 1$  punti distinti (o meglio tali che tre consecutivi siano distinti) che costituiscono il cosiddetto *poligono di de Boor*

$$P_0, P_2, P_4, \dots, P_{2n}$$

I punti del poligono hanno indici pari. Definiremo ora i punti  $P_i$  con  $i$  dispari e costruiremo una B-spline in cui ogni pezzo è una curva quadratica di Bézier con poligono di controllo  $P_{2i-1}, P_{2i}, P_{2i+1}$  (il centrale ha indice pari).

Per costruire i punti di indice dispari esistono vari criteri.

**Caso uniforme**

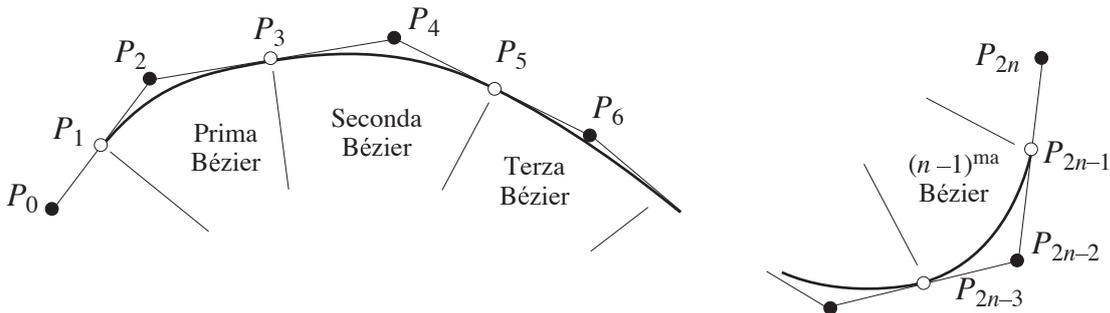
Nel caso più elementare i punti di indice dispari saranno semplicemente i punti medi

$$\text{Quindi porremo: } P_1 = \frac{P_0 + P_2}{2} \quad P_3 = \frac{P_2 + P_4}{2} \quad \dots \quad P_{2n-1} = \frac{P_{2n-2} + P_{2n}}{2}$$

e costruiremo le  $n - 2$  curve di Bézier con poligono di controllo  $P_{2i-1}, P_{2i}, P_{2i+1}$  mediante i polinomi di Bernstein o l'algoritmo di de Casteljau. Vedremo nel caso non uniforme che è possibile parametrizzare tutta la B-spline partendo un apposito intervallo esattamente come l'algoritmo di de Casteljau parametrizza una curva di Bézier usando l'intervallo  $[0, 1]$ .

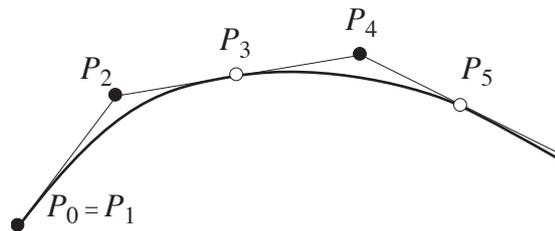
La B-spline così costruita è detta *B-spline quadratica uniforme*. L'aggettivo uniforme si riferisce al fatto che i punti di indice dispari sono presi come *punti medi* dei segmenti.

La B-spline è continua e di classe  $C^1$  per costruzione.



**Osservazioni: :**

1. La B-spline quadratica ha un controllo *semi-locale* dei punti, nel senso che cambiando uno dei  $P_i$  subiscono variazioni solo due pezzi di Bézier della curva e non l'intera curva.
2. La curva non passa per i punti iniziali della spezzata. Se si vuole ottenere questo, si può agire in due modi: o, come fanno alcuni, facendo semplicemente coincidere  $P_1$  con  $P_0$  e  $P_{2n-1}$  con  $P_{2n}$  (e definendo tutti gli altri punti come sopra) oppure mediante l'uso di opportune successioni nodali come vediamo nel seguito.



**Caso non uniforme e successioni nodali**

Invece di prendere i punti medi dei segmenti  $\overline{P_0 P_2}$  etc., si possono prendere altri punti più o meno distanti dagli estremi ed avere una curva più o meno aderente al poligono di de Boor e quindi più adatta a certe esigenze. Si tenga presente che cambiando un punto si cambiano solo due curve di Bézier della B-spline, e si mantiene quindi un controllo *semi-locale* su tutta la curva.

Potremmo semplicemente dire quale punto prendiamo su ogni segmento, ma conviene introdurre le *successioni nodali*, perché più utili per il seguito e indispensabili, come vedremo, nel caso delle B-spline cubiche.

Supponiamo di avere il poligono di de Boor  $P_0, P_2, \dots, P_{2n}$  costituito da  $n + 1$  punti. Si chiama successione nodale una successione non decrescente di  $n + 2$  numeri  $u_0 \leq u_1 \leq \dots \leq u_{n+1}$ .

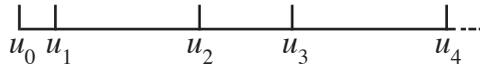
Per definire i punti  $P_1, P_3, \dots$  si adopera la successione nodale nel seguente modo:

$$P_1 = \frac{(u_2 - u_1)P_0 + (u_1 - u_0)P_2}{u_2 - u_0} \quad P_3 = \frac{(u_3 - u_2)P_2 + (u_2 - u_1)P_4}{u_3 - u_1} \quad \text{etc.}$$

Notiamo che, se la successione nodale è a passo costante, per esempio  $0 < 1 < 2 < 3 < \dots$ , si riottengono gli stessi  $P_i$  del caso uniforme.

Per illustrare graficamente il funzionamento della successione nodale, conviene visualizzare la

successione mediante un righello.



I segmenti del poligono di de Boor vengono messi in corrispondenza biunivoca con i sottosegmenti del righello nel seguente modo:

Il segmento  $P_0 P_2$  con la porzione  $[u_0, u_2]$  del righello, il segmento  $P_2 P_4$  con la porzione  $[u_1, u_3]$  del righello e così via.

Analiticamente ciò equivale a parametrizzare la retta  $P_0 P_2$  nel seguente modo

$$P(t) = P_0 + \frac{t - u_0}{u_2 - u_0}(P_2 - P_0)$$

Questa parametrizzazione di  $P_0 P_2$  fa ottenere  $P_0$  per  $t = u_0$  e  $P_2$  per  $t = u_2$ . Su  $P_2 P_4$  si ottiene in modo analogo  $P_2$  per  $t = u_1$  e  $P_4$  per  $t = u_3$  e così via per gli altri segmenti  $P_i P_{i+2}$ .

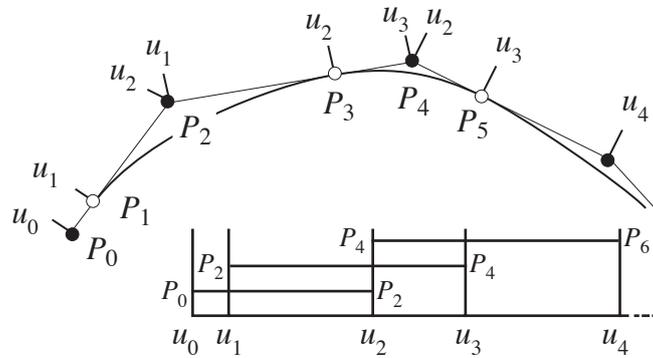
Il segmento  $P_0 P_2$  viene ad avere un punto intermedio corrispondente a  $u_1$ , il segmento  $P_2 P_4$  un punto intermedio corrispondente a  $u_2$  etc.

Questi punti, come si vede in figura, sono i punti  $P_1, P_3, \dots$  e ciò chiarisce il significato geometrico delle formule sopra che definiscono i  $P_i$  dispari.

La B-spline varia cambiando la successione nodale. La variazione di un elemento della successione nodale ha effetto solo su tre curve di Bézier della curva (una o due se siamo agli estremi).

**Osservazione:** Se nella successione nodale si ha  $u_0 = u_1$ , allora  $P_0 = P_1$  e la curva passa per il primo punto del poligono.

In generale però è bene che nella successione nodale non ci siano coincidenze fuori dagli estremi perché queste causano punti angolosi nella B-spline e comunque  $u_i$  troppo ravvicinati causano bruschi cambiamenti di curvatura.



**L' algoritmo di De Casteljaun nel caso quadratico non uniforme**

La successione nodale funge anche da parametro per la parametrizzazione della curva risultante nel senso che ogni punto dell'intervallo  $[u_1, u_n]$  (escludendo cioè gli estremi) fornisce un punto della B-spline in modo analogo all'algoritmo di De Casteljaun per le Bézier quadratiche.

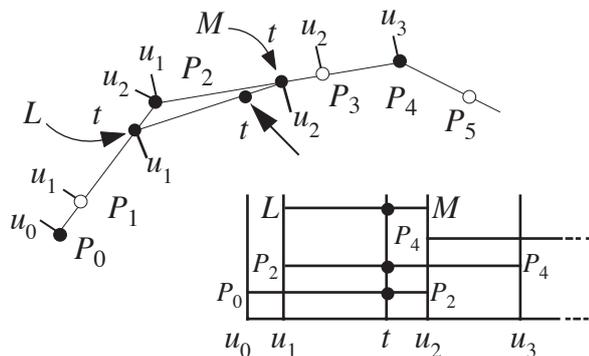
Per esempio scegliamo un  $t$  nell'intervallo  $[u_1, u_2]$ . Questo determina un punto della prima curva di Bézier della B-spline nel seguente modo:

Sul segmento  $P_0 P_2$  che è parametrizzato dall'intervallo  $[u_0, u_2]$ , si considera il punto  $t \in [u_1, u_2]$  che chiamiamo  $L$ .

Analogamente sul segmento  $P_2 P_4$  che è parametrizzato dall'intervallo  $[u_1, u_3]$ , si considera il punto  $t \in [u_1, u_2]$  che chiamiamo  $M$ .

Il segmento  $LM$  viene ora parametrizzato dall'intervallo  $[u_1, u_2]$ . In corrispondenza di  $t$  su  $LM$  si determina il punto della B-spline corrispondente a  $t$ .

Facendo variare  $t$  nell'intervallo  $[u_1, u_2]$  si ottengono tutti i punti della B-spline compresa tra  $P_1$  e  $P_3$ , facendolo variare in  $[u_2, u_3]$  si ottengono i punti compresi tra  $P_3$  e  $P_5$  e così via.



### 4.2.7 Le B-spline cubiche

Le B-spline quadratiche danno risultati abbastanza soddisfacenti e, grazie alla flessibilità data dalle successioni nodali, si adattano facilmente a molte esigenze.

Le B-spline più usate comunque sono quelle cubiche. A tal proposito riportiamo due osservazioni:

**Osservazione 1:** Anche se le B-spline quadratiche sono curve di classe  $C^1$  e quindi non presentano spigoli nei punti di giunzione delle varie curve di Bézier che la compongono, esse non sono quasi mai di classe  $C^2$ .

Le curve non di classe  $C^2$ , pur non avendo spigoli, risultano spesso sgradevoli per il fatto che il raggio di curvatura, che dipende dalla derivata seconda, può variare bruscamente nei punti di giunzione.

Le B-spline cubiche saranno invece costruite in modo da essere di classe  $C^2$ .

**Osservazione 2:** Le parabole sono curve piane. Se si crea una curva nello spazio raccordando tra loro rami di parabola, il passaggio da una parabola all'altra può risultare assai brusco perché cambia di colpo il piano di giacenza.

Le cubiche nello spazio invece sono dotate di *torsione* e quindi la giacitura sul piano osculatore varia con continuità nello spazio e nelle B-spline cubiche la variazione è continua anche nei punti di giunzione.

Nel seguito, i punti del poligono di de Boor saranno non necessariamente nel piano e quindi potranno avere una terza coordinata  $z$ , e ciò senza alcuna variazione degli algoritmi di costruzione delle B-spline.

#### Il poligono di de Boor

Come dati iniziali sono assegnati  $n+1$  punti distinti (o meglio tali che tre consecutivi siano distinti) che costituiscono il *poligono di de Boor*.

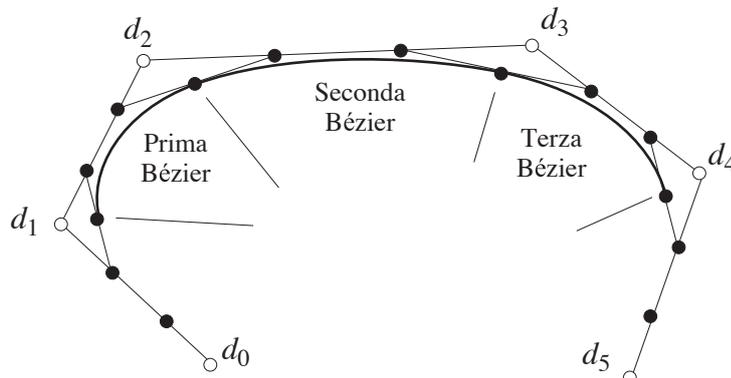
$$d_0, d_1, d_2, \dots, d_{n-1}, d_n$$

Costruiremo una curva B-spline costituita da  $n-2$  curve cubiche di Bézier ognuna delle quali ha un poligono di controllo  $A, B, C, D$  col secondo e terzo punto situati sui segmenti  $d_i d_{i+1}$  ( $i \neq 0, n$ ).

I punti dei poligoni di controllo andranno scelti osservando certe regole se si vuole fare in modo che la B-spline risultante sia di classe  $C^2$ . Questo rende la costruzione leggermente più complessa che nel caso quadratico.

#### Caso uniforme

Nel caso più semplice divideremo i segmenti  $d_i d_{i+1}$  in tre parti uguali. Poi considereremo i segmenti che hanno estremi due punti consecutivi delle divisioni e li divideremo in due parti uguali. La B-spline sarà costituita dalle curve di Bézier cubiche che hanno come punti di controllo questi. La figura dovrebbe chiarire quali sono i poligoni di controllo.



La curva così costruita è detta *B-spline cubica uniforme*. L'aggettivo uniforme si riferisce al fatto che i punti dei poligoni di controllo sono presi con suddivisioni uniformi dei segmenti del poligono di de Boor.

La B-spline è continua e di classe  $C^1$  per costruzione. Si può dimostrare che, se i punti  $P_i$  sono costruiti come sopra, e cioè dividendo in tre e in due parti uguali, essa è anche di classe  $C^2$ .

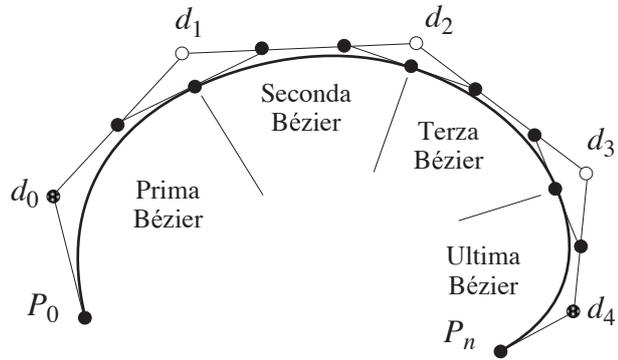
Osserviamo che la curva non passa per nessuno dei punti del poligono di de Boor ed è anche lontana dagli estremi.

Esiste una costruzione alternativa che consente di far passare la curva per gli estremi del poligono, mantenendo la classe  $C^2$ .

Questa costruzione privilegia gli estremi del poligono, quindi è bene chiamare il poligono

$$P_0, d_0, d_1, \dots, d_{n-1}, d_n, P_n$$

Il primo lato  $d_0 d_1$  viene diviso in due parti e non in tre e così pure l'ultimo. I punti estremi  $P_0$  e  $P_n$  fanno parte del poligono di controllo delle Bézier estreme.



Per il resto tutto è come nel caso sopra. La figura dovrebbe chiarire la costruzione.

Comunque vedremo che l'uso delle successioni nodali nel caso di B-spline non uniformi fornirà un modo più efficace per costruire una B-spline con questa proprietà.

**Caso non uniforme**

Invece di dividere i segmenti  $\overline{d_i d_{i+1}}$  in tre parti uguali e i segmenti intermedi in due parti uguali, si possono fare altre scelte e ottenere una modellazione diversa con un controllo *semi-locale*.

Si tenga però presente che scelte casuali della suddivisione dei vari segmenti possono far sì che la B-spline risultante non sia più di classe  $C^2$  e, come abbiamo detto, le curve non di classe  $C^2$  sono da evitare.

Per ottenere ciò si fa uso di una *successione nodale* che consente di effettuare modifiche, mantenendo la classe  $C^2$ .

Se il poligono di de Boor è  $d_0, \dots, d_n$ , la successione nodale è una successione non decrescente di  $n + 3$  numeri positivi  $u_0 \leq u_1 \leq \dots \leq u_{n+2}$ . Definiremo quindi i punti di controllo delle Bézier cubiche usando gli  $u_i$ . La costruzione è analoga al caso quadratico, anche se più complessa.

Per motivi di chiarezza illustreremo la procedura mediante un semplice poligono di de Boor di soli quattro vertici. Si otterrà una B-spline cubica elementare costituita da una sola curva di Bézier.

I quattro vertici per praticità saranno denotati  $A, B, C, D$  anziché  $d_0, d_1, d_2, d_3$

La successione nodale sarà costituita da 6 numeri  $u_0 \leq u_1 \leq \dots \leq u_5$ .

Visualizziamo, come prima, la successione

$$u_0, u_1, \dots, u_5.$$

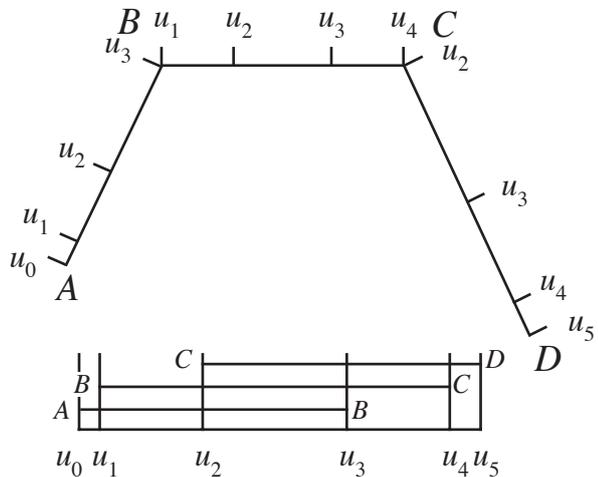
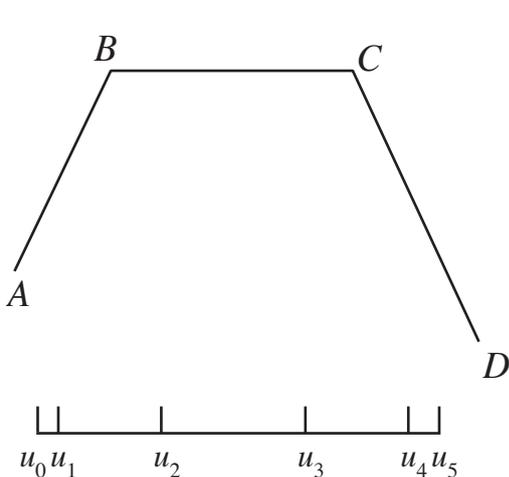
su un righello che useremo per parametrizzare i tre lati.

Parametizziamo:

il segmento  $AB$  mediante  $[u_0, u_3]$

il segmento  $BC$  mediante  $[u_1, u_4]$

il segmento  $CD$  mediante  $[u_2, u_5]$

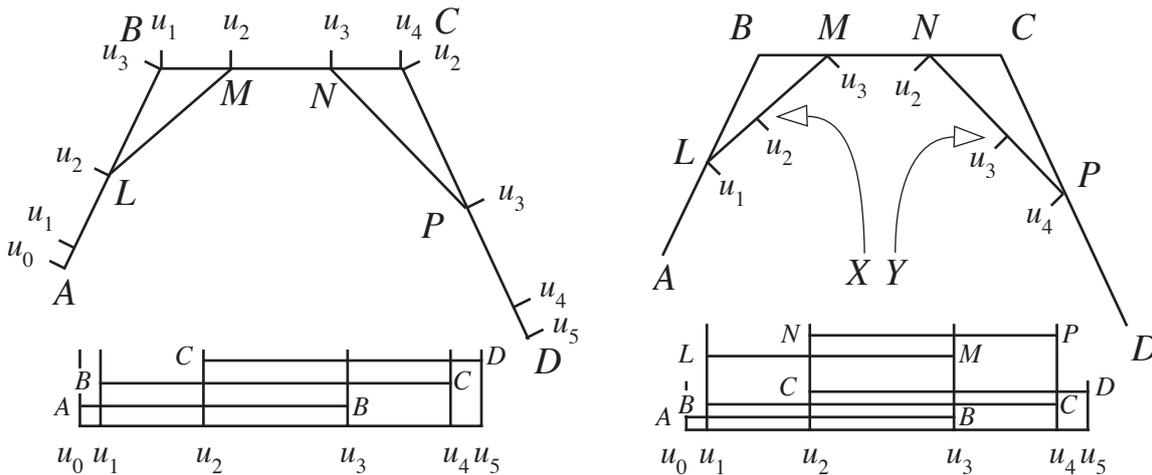


Su ognuno dei lati  $AB, BC, CD$  ci sono due punti ottenuti rispettivamente in corrispondenza di  $u_2$  e di  $u_3$ .

Escludendo  $B$  ottenuto su  $AB$  per  $t = u_3$  e  $C$  ottenuto su  $CD$  in corrispondenza di  $u_2$ , abbiamo 4 punti che chiamiamo  $L, M, N, P$  come nella prima figura sotto.

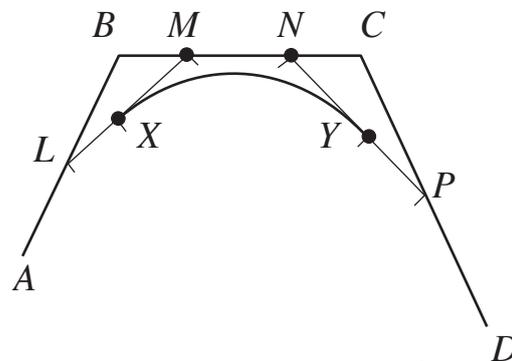
$$\text{Analiticamente: } L = A + \frac{u_2 - u_0}{u_3 - u_0} (B - A) \quad M = B + \frac{u_2 - u_1}{u_4 - u_1} (C - B) \quad \text{etc.}$$

I due segmenti  $LM, NP$  vanno parametrizzati rispettivamente mediante l'intervallo  $[u_1, u_3]$  e mediante l'intervallo  $[u_2, u_4]$  come nella seconda figura sotto.

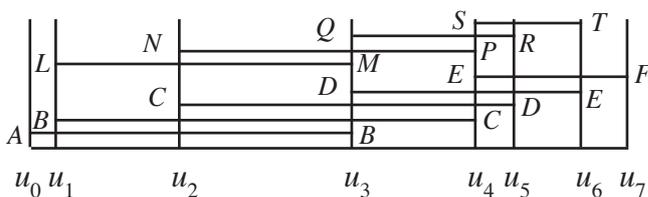
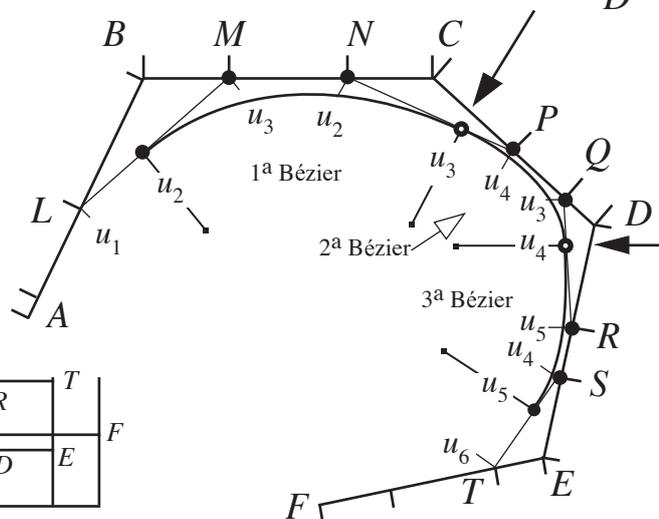


Fissiamo l'attenzione sul punto ottenuto su  $LM$  per  $t = u_2$  che chiamiamo  $X$  e su quello ottenuto su  $NP$  per  $t = u_3$  che chiamiamo  $Y$  (seconda figura sopra).

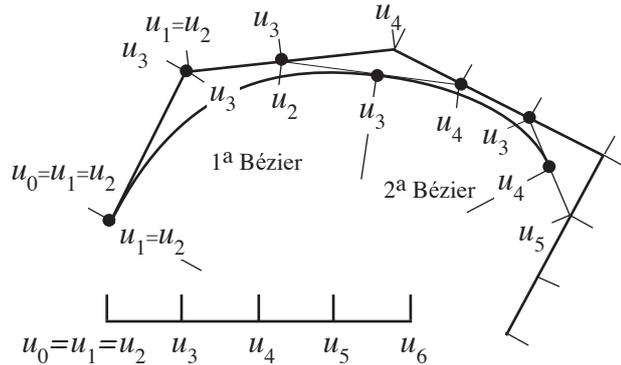
Come nel caso quadratico, costruiamo la curva di Bézier che ha come poligono di controllo quello costituito dai quattro punti  $X, M, N, Y$  segnati nella figura a lato. Osserviamo solo che, se la successione nodale è a passo costante, si riottiene il caso uniforme, con suddivisione dei tre segmenti  $AB, BC, CD$  in tre parti uguali e dei due segmenti  $LM, NP$  in due parti uguali.



Nella figura accanto vediamo questo procedimento portato avanti per un poligono di 6 vertici  $ABCDEF$  e quindi usando una successione nodale di 8 numeri  $u_0, \dots, u_7$ . La B-spline conseguente è costituita da tre curve di Bézier. Sono indicati i due punti di giunzione delle tre curve. Qui sotto il righello coi segmenti parametrizzati mediante la successione nodale.



Per concludere vediamo un esempio con poligono di de Boor di 5 vertici e una successione nodale  $u_0, \dots, u_6$  in cui  $u_0 = u_1 = u_2$ . Questo fa sì che la curva passi per il primo punto del poligono di de Boor. Per il resto abbiamo scelto che la successione nodale sia di passo costante. Fondamentalmente si ritrova la costruzione alternativa del caso uniforme che permette di far passare la B-spline per il primo vertice del poligono.

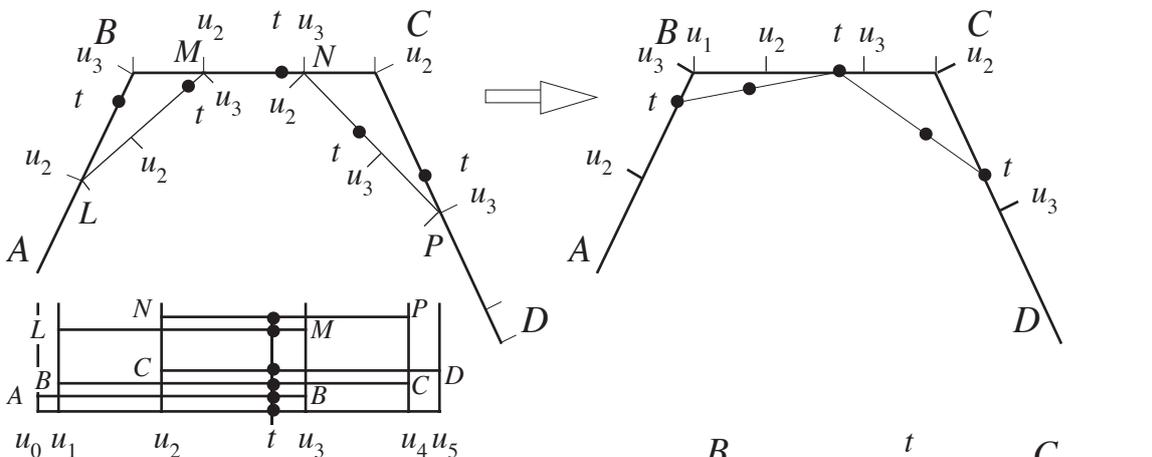


**L'algorithmo di De Casteljau per le B-spline cubiche non uniformi**

Invece di costruire le curve di Bézier dati i loro poligoni di controllo, può essere più conveniente parametrizzare la curva, come nel caso quadratico.

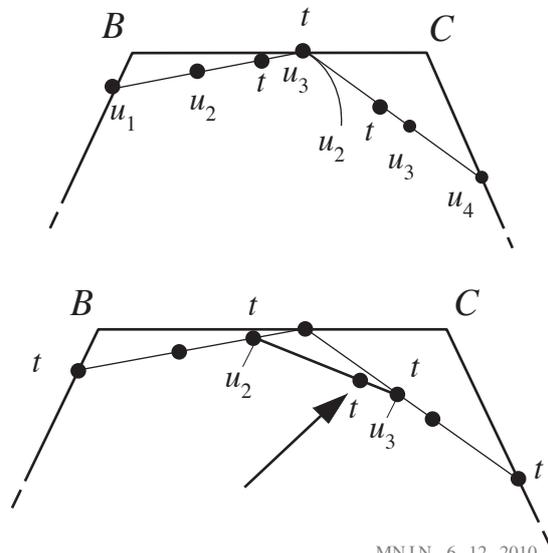
Riprendiamo quindi il caso del poligono  $ABCD$  dal momento in cui sono stati trovati i punti  $L, M, N, P$ . Parametrizzeremo la curva con un parametro  $t$  che varia nell'intervallo  $[u_2, u_3]$  della successione nodale.

Scegliamo  $t \in [u_2, u_3]$  e riportiamolo su tutti i segmenti che sono stati posti in corrispondenza con questo intervallo. Ci sono 5 intervalli  $[u_2, u_3]$ , quindi fissiamo l'attenzione sui 5 punti corrispondenti a  $t$ . I tre punti su  $AB$ , su  $LM$  e su  $BC$  risultano allineati e così pure i tre su  $BC$ , su  $NP$  e su  $CD$ . Questo è conseguenza di un famoso teorema geometrico, noto come *Teorema di Menelao*.



Consideriamo dunque le due rette e poniamole in corrispondenza rispettivamente con gli intervalli  $[u_1, u_3]$  e  $[u_2, u_4]$  della successione nodale. Sempre mediante il teorema di Menelao, si può dimostrare che i punti intermedi per questa parametrizzazione degli intervalli sono proprio quelli che corrispondono rispettivamente a  $u_2$  e a  $u_3$ . Su questi segmenti individuiamo ancora una volta il punto  $t$  compreso tra  $u_2$  e  $u_3$ .

Per concludere consideriamo l'ultimo segmento che ha come estremi i due punti  $t$  e poniamolo in corrispondenza con l'intervallo  $[u_2, u_3]$ . Su questo segmento individuiamo il punto  $t$  compreso tra  $u_2$  e  $u_3$ . Questo è finalmente un punto della B-spline.



## 4.2.8 Cenno sulle curve di Bézier razionali

Le B-spline cubiche così definite degenerano correttamente in rette o in parabole se i punti del poligono sono disposti in maniera particolare, per esempio se sono allineati, ma non possono mai rappresentare correttamente un arco di circonferenza o di ellisse per il semplice motivo che né circonferenze, né ellissi ammettono una parametrizzazione mediante funzioni polinomiali, mentre le parametrizzazioni delle curve di Bézier sono polinomiali essendo combinazione lineare di polinomi di Bernstein. Questa è la ragione per cui spesso vengono utilizzate le curve B-spline razionali che sono a pezzi curve di Bézier razionali. Accenniamo brevemente a queste ultime.

Per definire una curva di Bézier razionale di ordine  $n$  occorrono un poligono  $P_0, P_1, \dots, P_n$  e una successione di numeri positivi  $w_1, \dots, w_n$  detti *pesi*. La curva ha rappresentazione parametrica

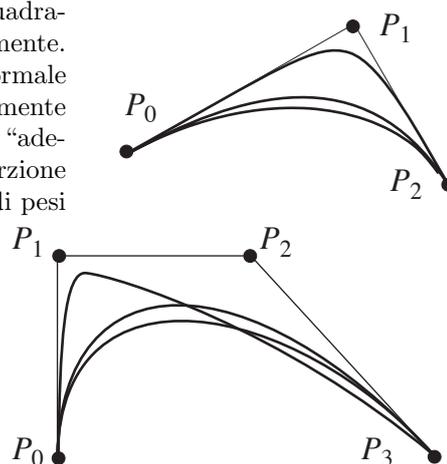
$$P(t) = \frac{w_0 P_0 B_0(t) + \dots + w_n P_n B_n(x)}{w_0 B_0(t) + \dots + w_n B_n(x)}$$

dove i  $B_i$  sono i polinomi di Bernstein di ordine  $n$ . Osserviamo che, per le proprietà dei polinomi di Bernstein, il denominatore vale 1 se tutti i pesi sono 1, per cui in questo caso si ottiene la solita curva di Bézier. Assegnando opportunamente i pesi si riesce a fare “aderire” più o meno la curva ai vertici del poligono ottenendo spesso risultati più flessibili di quelli delle curve di Bézier semplici e riuscendo per esempio a descrivere archi di coniche diversi dalle parabole nel caso di curve di Bézier quadratiche.

Senza entrare nei dettagli mostriamo due esempi.

Sul poligono  $P_0 P_1 P_2$  sono state costruite le curve di Bézier quadratiche razionali con pesi  $[1 \ 1 \ 1]$   $[1 \ 1 \ 2]$   $[1 \ 4 \ 1]$  rispettivamente. La prima è la più distante da  $P_1$  ed è la curva di Bézier normale (quindi una parabola), la seconda è la mediana ed è precisamente un quarto di ellisse inserito nel poligono di controllo, la terza “aderisce” al punto centrale del poligono di controllo ed è una porzione di iperbole. La curva con pesi  $[2 \ 1 \ 1]$  coincide con quella di pesi  $[1 \ 1 \ 2]$  anche se con diversa parametrizzazione.

Sul poligono  $P_0 P_1 P_2 P_3$  sono state costruite le curve di Bézier cubiche razionali con pesi  $[1 \ 1 \ 1 \ 1]$   $[3 \ 2 \ 2 \ 3]$   $[1 \ 5 \ 1 \ 1]$  rispettivamente. La prima è la mediana ed è la cubica di Bézier normale. La seconda è la più bassa ed è un arco di ellisse, la terza “aderisce” al secondo punto del poligono di controllo.



## 5.1 Integrazione ed equazioni differenziali

È un capitolo assai vasto dell'analisi numerica, anche perché le equazioni differenziali sono uno strumento essenziale in moltissime questioni.

Ci limitiamo ai casi più semplici illustrando tecniche che comunque sono abbastanza antiche, anche se il loro studio ha ricevuto enorme impulso con l'avvento del calcolo automatico.

Il problema è che di raro è possibile risolvere le equazioni differenziali in modo esatto e quindi i metodi numerici sono quasi sempre indispensabili.

### 5.1.1 Richiami sugli integrali

Il più semplice problema differenziale è il seguente:

Data una funzione  $f(t)$  definita in un punto  $t_0$  e in un suo intorno (destra, sinistra o comprendente  $t_0$ ), e un numero  $y_0$ , determinare una funzione  $y(t)$  definita in un intervallo comprendente il numero  $t_0$  tale che

$$y' = f(t) \quad y(t_0) = y_0$$

La funzione  $y(t)$  è detta *primitiva* di  $f(t)$ .

Come è ben noto, il teorema fondamentale del calcolo integrale fornisce la soluzione del problema:

**Proposizione 3** Se  $f(t)$  è integrabile in un intorno di  $t_0$ , allora

$$y(t) = y_0 + \int_{t_0}^t f(u)du$$

Il problema è quindi quello di calcolare l'integrale. Se la primitiva  $y(t)$  è ricavabile mediante le note tecniche di integrazione indefinita, allora la proposizione fornisce semplicemente la nota formula

$$\int_{t_0}^t f(u)du = y(t) - y(t_0)$$

È noto che in molti casi  $y(t)$  non è calcolabile elementarmente. In altri casi lo è, ma la sua espressione è comunque complessa, per cui ci proponiamo di ricavare degli algoritmi numerici per il calcolo dell'integrale definito.

### 5.1.2 Integrazione numerica: formule di Newton-Cotes

Vogliamo calcolare numericamente l'integrale definito

$$\int_a^b f(t)dt$$

dove  $f(t)$  è una funzione continua nell'intervallo  $[a, b]$  (ma basterebbe continua a tratti e limitata).

L'idea base è sempre quella di sostituire a  $f(t)$  un polinomio di grado  $n$  passante per  $n + 1$  punti dell'intervallo  $[a, b]$  e quindi di usare la primitiva del polinomio che è calcolabile elementarmente, avvertendo che, in genere, *non è necessario esplicitare il polinomio* per calcolare l'area sottesa.

A seconda del grado usato e del criterio di scelta dei punti si possono avere numerosissimi metodi di integrazione numerica.

Se tra gli  $n + 1$  punti ci sono gli estremi si parla di *metodo chiuso*.

Se gli  $n + 1$  punti sono scelti dividendo l'intervallo in parti uguali, le formule ricavate sono dette *formule di quadratura di Newton-Cotes*.

Ci limiteremo a quest'ultimo caso ed esamineremo in dettaglio i casi  $n = 0, 1, 2$ .

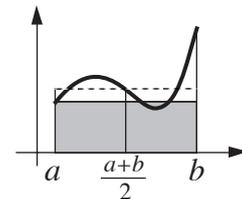
#### Metodo del rettangolo (o di Cauchy) ( $n = 0$ )

Il polinomio ha grado 0 è cioè una costante, quindi va scelto un solo punto dell'intervallo, per esempio il punto  $a$ .

Quindi  $f(t)$  è sostituita dalla funzione costante  $y = f(a)$  e notoriamente  $\int_a^b f(a)dt = f(a)(b - a)$  (area del rettangolo).

Benché il metodo sia grossolano e banale, vedremo poi la sua controparte nel caso di un'equazione differenziale qualunque.

Osserviamo solo che, se invece di scegliere il punto  $a$  si sceglie il punto medio dell'intervallo  $(a + b)/2$ , si ottiene una formula assai simile alla successiva e in molti casi più precisa.



#### Metodo del trapezio (o di Bézout) ( $n = 1$ )

Il polinomio ha grado 1 e ha come grafico una retta, quindi vanno scelti due punti dell'intervallo che, nel caso chiuso di Newton-Cotes sono i due punti  $a, b$ .

Quindi  $f(t)$  è sostituita dalla funzione che rappresenta la retta passante per i punti  $(a, f(a))$  e  $(b, f(b))$ . L'integrale di questa funzione è l'area del trapezio in figura che vale  $\frac{f(a) + f(b)}{2} (b - a)$

