

# INTERACTIVE CURVE AND DESIGN

## Forme implicite e forme parametriche

Nel caso di rappresentazioni boundary, ogni faccia possiede degli edges, che possono essere segmenti o segmenti di curva, e la faccia stessa può essere parte di una superficie (*surface patch*).

### Curva Parametrica

Una **Curva Parametrica** (*parametric curve*) nello spazio ha la seguente forma:

$$\mathbf{F}: [0,1] \rightarrow (f(u), g(u), h(u))$$

dove  $f(u)$ ,  $g(u)$  e  $h(u)$  sono tre funzioni a valori reali con  $u$  che varia nell'intervallo  $[0,1]$ . Quindi, ad ogni  $u$  in  $[0,1]$ , corrisponde un punto  $(f(u), g(u), h(u))$  nello spazio. Considereremo  $f(u)$ ,  $g(u)$  e  $h(u)$  sempre polinomiali.

### Osservazione:

Se  $h(u)$  non viene considerata dalla definizione di  $\mathbf{F}(u)$ ,  $\mathbf{F}(u)$  diventa una curva nel piano.

### Esempi

- Una retta può essere definita come  $\mathbf{B} + t\mathbf{d}$ , con  $\mathbf{B}$  un punto e  $\mathbf{d}$  un vettore di direzione. quindi, definita  $\mathbf{F}(u)$  come:

$$f(u) = b_1 + ud_1 \quad g(u) = b_2 + ud_2 \quad h(u) = b_3 + ud_3$$

con  $\mathbf{B} = \langle b_1, b_2, b_3 \rangle$  e  $\mathbf{d} = \langle d_1, d_2, d_3 \rangle$ ,  $\mathbf{F}(u)$  è una curva parametrica che mappa  $[0,1]$  al segmento di retta che va tra  $\mathbf{B}$  e  $\mathbf{B} + \mathbf{d}$ , inclusi.

- Una **circonferenza** ha la seguente forma non polinomiale:

$$x(u) = r \cos(2\pi u) + p \quad y(u) = r \sin(2\pi u) + q$$

con centro in  $(p, q)$  e raggio  $r$ . Poiché  $u$  varia in  $[0,1]$ , il valore di  $2\pi u$  varia in  $[0,2\pi]$ .

Possiamo provare ad eliminare  $u$ .  $x - p = r \cos(2\pi u)$      $y - q = r \sin(2\pi u)$

Elevando al quadrato e sommando  $(x - p)^2 + (y - q)^2 = r^2$

- Una **cubica nello spazio**:

$$f(u) = u \quad g(u) = u^2 \quad h(u) = u^3$$

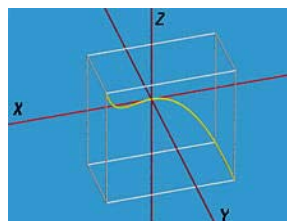


Figura 1 - la curva di equazione precedente nel range  $[-1,1]$

- Una *ellisse circolare*:

$$\mathbf{f}(u) = ( a \cos(u), a \sin(u), bu )$$

Punto iniziale in  $(a, 0, 0)$ , punto finale in  $(a, 0, b)$ . SI osservi che questa curva giace nel cilindro di raggio  $a$  ed altezza l'asse  $z$ .

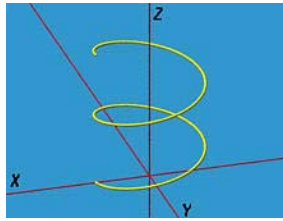
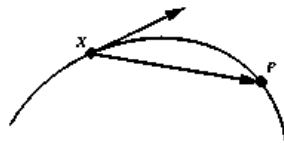


Figura 2 - una ellisse circolare che varia in  $[0, 4u]$

## Vettore Tangente e Retta Tangente

Si consideri un punto fissato  $X$  ed un punto mobile  $P$  su di una curva.

Se  $P$  si avvicina ad  $X$ , il vettore tra  $XP \rightarrow$  si avvicina al vettore tangente in  $X$ . La retta che contiene il vettore tangente è detta **retta tangente**.



La computazione del vettore tangente è molto semplice. Si consideri la derivata di una curva parametrica  $\mathbf{f}(u)$ :

$$\mathbf{f}'(u) = (f'(u), g'(u), h'(u)) \quad \text{con } f'(u) = df/du, g'(u) = dg/du \text{ e } h'(u) = dh/du.$$

Spesso una normalizzazione è necessaria, quindi il vettore tangente di lunghezza unitaria al parametro  $u$ , o al punto  $\mathbf{f}(u)$ , è

$$\mathbf{f}'(u) / |\mathbf{f}'(u)|$$

con  $|\mathbf{x}|$  lunghezza del vettore  $\mathbf{x}$ . La retta tangente ha equazione

$$\mathbf{f}(u) + t \mathbf{f}'(u) \quad \text{oppure} \quad \mathbf{f}(u) + t (\mathbf{f}'(u)/|\mathbf{f}'(u)|)$$

### Esempio

- $\mathbf{f}(u) = ( r \cos(2\pi u) + p, r \sin(2\pi u) + q ), \quad \text{con } u \in [0,1].$

$$\mathbf{f}'(u) = ( -2\pi r \sin(2\pi u), 2\pi r \cos(2\pi u) )$$

**Retta tangente in  $u$ :**

$$\mathbf{f}(u) + t \mathbf{f}'(u) = ( r \cos(2\pi u) + p, r \sin(2\pi u) + q ) + t ( -2\pi r \sin(2\pi u), 2\pi r \cos(2\pi u) )$$

## Vettore Normale e Curvatura

Sia  $\mathbf{f}(u)$  un punto fissato (valore di  $u$  fissato) e  $P, Q$  due punti che si muovono sulla curva parametrica. Questi punti determinano univocamente un piano.

Se  $P$  e  $Q$  si muovono verso  $\mathbf{f}(u)$ , questo piano si muove fino ad arrivare ad una posizione limite, ed in questo caso viene detto **osculating plane** in  $\mathbf{f}(u)$ .

- l'osculating plane in  $\mathbf{f}(u)$  contiene la retta tangente in  $\mathbf{f}(u)$ .

**Teorema:**

l'osculating plane è il piano che passa per  $\mathbf{f}(u)$  e contiene  $\mathbf{f}'(u)$  e  $\mathbf{f}''(u)$ . Più precisamente, ogni punto sul piano soddisfa la seguente equazione:

$$\mathbf{f}(u) + p \mathbf{f}'(u) + q \mathbf{f}''(u) \quad \text{con } p \text{ e } q \text{ parametri}$$

Si definisce **binormal** vector  $\mathbf{b}(u)$  (vettore binormale) il vettore unitario del prodotto vettoriale di  $\mathbf{f}'(u)$  e  $\mathbf{f}''(u)$ :

$$\mathbf{b}(u) = (\mathbf{f}'(u) \times \mathbf{f}''(u)) / |(\mathbf{f}'(u) \times \mathbf{f}''(u))|$$

Quindi  $\mathbf{b}(u)$  è perpendicolare sia a  $\mathbf{f}'(u)$  che a  $\mathbf{f}''(u)$  e, per questo, perpendicolare all'osculating plane. La retta  $\mathbf{f}(u) + t \mathbf{b}(u)$  è detta **binormal line** in  $\mathbf{f}(u)$ .

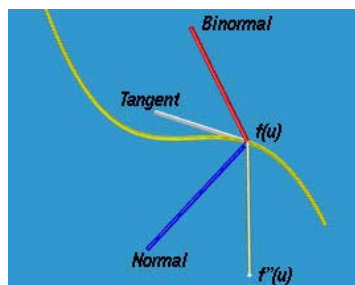
Si definisce il **normal vector** come il vettore perpendicolare ai vettori tangente e binormale con la direzione che segue la regola della mano destra.

Quindi il **normal vector**  $\mathbf{n}(u)$ :  $\mathbf{n}(u) = (\mathbf{b}(u) \times \mathbf{f}'(u)) / |\mathbf{b}(u) \times \mathbf{f}'(u)|$

- $\mathbf{f}(u) + t \mathbf{n}(u)$  è la **normal line** in  $\mathbf{f}(u)$ .

**Osservazione:**

- tangent vector  $\mathbf{f}'(u)$ , normal vector  $\mathbf{n}(u)$  e binormal vector  $\mathbf{b}(u)$  formano un sistema di coordinate con origine in  $\mathbf{f}(u)$ .
- le tangent line, binormal line e normal line sono i tre assi cartesiani con il verso positivo fornito dal tangent vector, binormal vector e normal vector, rispettivamente.
- Tre vettori spesso chiamati **moving triad** o **triad** al punto  $\mathbf{f}(u)$ .
- Moving triad chiamata anche **moving trihedron**.

**Esempio**

Consideriamo una ellisse circolare:

$$\mathbf{f}(u) = (a \cos(u), a \sin(u), b u)$$

Consideriamo la derivata prima e la derivata seconda:

$$\mathbf{f}'(u) = (-a \sin(u), a \cos(u), b) \quad \mathbf{f}''(u) = (-a \cos(u), -a \sin(u), 0)$$

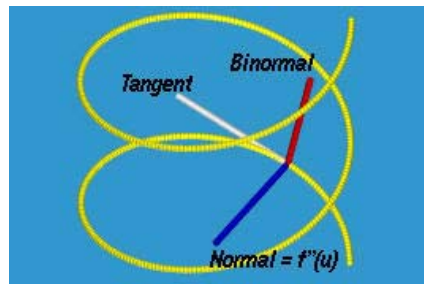
Il vettore binormale non unitario:

$$\mathbf{b}(u) = \mathbf{f}'(u) \times \mathbf{f}''(u) = (a b \sin(u), -a b \cos(u), a^2)$$

Il vettore normale non unitario:

$$\mathbf{n}(u) = \mathbf{b}(u) \times \mathbf{f}'(u) = (-a(a^2 + b^2) \cos(u), -a(a^2 + b^2) \sin(u), 0)$$

- $\mathbf{n}(u)$  ed  $\mathbf{f}''(u)$ , sono paralleli (i loro coefficienti sono proporzionali).
- **Normalizzando** tutti i vettori in questione, il vettore normale e la derivata seconda sono identici.

Figura 3 - i vettori trovati per  $u=1$ .

## Curvatura

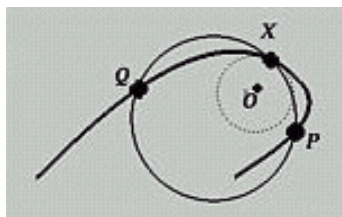
Si consideri  $X$  fisso sulla curva e  $P, Q$  due punti che si muovono su di essa. Fino a quando almeno due non sono allineati i tre punti determinano univocamente una circonferenza.

Quando uno dei due tra  $P$  e  $Q$  si muove verso  $X$ , la circonferenza collassa in una circonferenza limite della **circonferenza di curvatura** in  $X$ .

Il relativo centro  $O$  e raggio  $r$ , sono chiamati **centro e raggio della circonferenza di curvatura**.

### Definizione

$1/r$  è **curvatura** nel punto  $X$  della curva assegnata. Più grande è il raggio di curvatura, minore è la curvatura stessa.

Figura 4 - la curvatura della curva nel punto  $X$ 

### Osservazione

1. Dalla definizione di osculating plane, tale circonferenza di curvatura giace sull'osculating plane.
2. Poiché il cerchio di curvatura è tangente alla curva e quindi alla retta tangente, il centro di curvatura giace sulla **line normale**.

Il valore di curvatura,  $k(u)$ , può essere computata come segue:

$$k(u) = \frac{|\mathbf{f}'(u) \times \mathbf{f}''(u)|}{|\mathbf{f}'(u)|^3}$$

### Esempi

- Retta  $\mathbf{f}(u) = (a + up, b + uq, c + ur)$

Si ha che:

$$\begin{aligned} \mathbf{f}'(u) &= (p, q, r) & |\mathbf{f}'(u)| &= \text{SQRT}(p^2 + q^2 + r^2) \\ \mathbf{f}''(u) &= (0, 0, 0) & \mathbf{f}'(u) \times \mathbf{f}''(u) &= (0, 0, 0) \\ k(u) &= 0 \end{aligned}$$

Quindi la curvatura di una retta è zero ovunque.

## Continuity Issues

Nel caso di rappresentazioni B-rep, archi e facce di un solido possono essere segmenti di curva e patch di superfici piuttosto che semplici segmenti e poligoni. Tre patch curvilinee che toccano fra di loro. Due segmenti di curva si incontrano nel vertice  $X$ . Siano queste due curve descritte dalle equazioni  $\mathbf{f}(u)$  e  $\mathbf{g}(v)$ , con  $u$  e  $v$  valori in intervalli  $[a,b]$  e  $[m,n]$ , rispettivamente. Il problema è spesso assicurarsi che le curve si uniscano in modo "smooth".

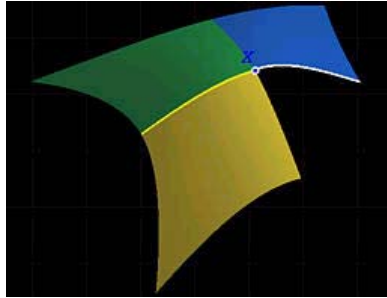


Figura 5 - tre patch uniti insieme per una superficie

### Continuità $C^0$

Si consideri il punto finale a dx di  $\mathbf{f}$ ,  $\mathbf{f}(b)$  e il punto iniziale a sx di  $\mathbf{g}$ ,  $\mathbf{g}(m)$ . Se  $\mathbf{f}(b) = \mathbf{g}(m)$ , allora si può dire che  $\mathbf{f}(u)$  e  $\mathbf{g}(v)$  sono  **$C^0$  continue** in  $\mathbf{f}(b)=\mathbf{g}(m)$ .

### Continuità $C^k$

Per ogni  $i \leq k$ , se il valore dell' $i$ -esima derivata di  $\mathbf{f}(u)$  in  $b$  è uguale al valore dell' $i$ -esima derivata di  $\mathbf{g}(v)$  in  $m$ , allora le curva sono  **$C^k$  continue** in  $\mathbf{f}(b)=\mathbf{g}(m)$ .

### Proprietà

- Due segmenti di curva  $C^k$  continui in  $\mathbf{f}(b)=\mathbf{g}(m)$  sono  $C^i$  continui in  $i \leq k$ .
- Analogamente se la derivata  $k$ -esima non coincide nei punti  $b$  ed  $m$  rispettivamente, le funzioni  $\mathbf{f}$  e  $\mathbf{g}$  non possono essere  $C^i$  continue con  $i \geq k$ .

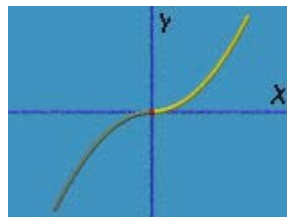
### Esempio Continuità

Consideriamo due curve (**parabole**):

$$\mathbf{f}(u) = (u, -u^2, 0)$$

$$\mathbf{g}(v) = (v, v^2, 0)$$

con domini  $[-1,0]$  e  $[0,1]$ , rispettivamente. Si verifica che  $\mathbf{f}(0) = \mathbf{g}(0)$ . Vediamo di verificare se esse sono di tipo  $C^2$ .



$$\mathbf{f}'(u) = (1, -2u, 0)$$

$$\mathbf{f}''(u) = (0, -2, 0)$$

$$\mathbf{g}'(u) = (1, 2v, 0)$$

$$\mathbf{g}''(v) = (0, 2, 0)$$

- Poiché  $\mathbf{f}(0) = \mathbf{g}(0) = (0, 0, 0)$ , le curve sono  $C^1$  continue in  $(0,0,0)$ .
- Ma  $\mathbf{f}''(0) = (0, -2, 0) \neq \mathbf{g}''(0) = (0, 2, 0)$ , e per questo non sono  $C^2$  continue nell'origine.

Infatti,  $f''(u)$  punta a sud mentre  $g''(u)$  punta a Nord, ed entrambe sono costanti. Quindi, quando un punto che si muove sulla curva attraversa l'origine la derivata seconda cambia direzione.

### Osservazione

Nel caso di forma parametriche una riparametrizzazione o dei cambiamenti di variabile, possono influire molto sulla continuità.

## Geometric Continuity

Molte curve possono essere continue di tipo  $C^1$  e continue di curvatura, ma non continue di tipo  $C^2$  nei punti di congiunzione, alcune volte addirittura può non esistere la derivata seconda. Queste curve però sembrano smooth nei punti di controllo e per questo una certa proprietà deve esistere!

Inoltre dopo un cambiamento di variabile le curve possono diventare di continuità  $C^2$ . È possibile rilassare la condizione di  $C^2$  continuità nel seguente modo:

### Definizione:

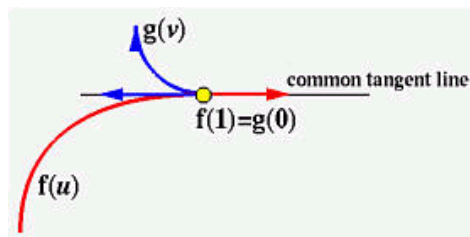
- Due segmenti di curva sono **continui con continuità geometrica  $G^k$**  ai punti di congiunzione se e solo se tutti le derivate  $i$ -esime,  $i \leq k$ , computate con parametri arc length concordano nei punti di giunzione.
- Due segmenti di curva sono continui  $G^k$  ai punti di giunzione se e solo se esistono due parametrizzazioni uno per ogni segmento di curva, in modo che tutte le derivate  $i$ -esime  $i \leq k$ , computate con le nuove parametrizzazioni concordano ai punti di giunzione.

### Consideriamo il caso $k=1$ .

- Due segmenti di curva  $C^0$  sono  $G^1$  continui se e solo se i vettori  $f'(u)$  e  $g'(v)$  sono nella stessa direzione nel punto di giunzione. ( $f'(u)$  e  $g'(v)$  valutati NEL punto di giunzione)
- Poiché i vettori tangenti hanno la stessa direzione, entrambe le curve possiedono la stessa retta tangente nel punto di giunzione.

### IL VICEVERSA NON VALE SEMPRE.

Ovvero se due curve hanno la stessa retta tangente in un punto non è detto che siano  $G^1$  continue.



### Continuità $G^2$

G. Neilson fornisce la seguente definizione per continuità di tipo  $G^2$ :

**Due segmenti di curva  $C^1$  continui sono  $G^2$  continui al punto di giunzione se e solo se il vettore  $f''(u) - g''(v)$  è parallelo al vettore tangente nel punto di giunzione**

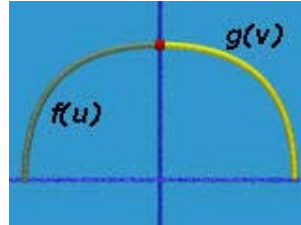
In questo senso la definizione funziona per qualsiasi tipo di parametrizzazione.

## Esempio

Si considerino due archi di parabola che si congiungono nel punto  $(0, 1, 0)$ :

$$\mathbf{f}(u) = (-1 + u^2, 2u - u^2, 0) \quad \mathbf{g}(v) = (2u - u^2, 1 - u^2, 0)$$

con dominio  $[0, 1]$ . Si uniscono in  $\mathbf{f}(1) = \mathbf{g}(0) = (0, 1, 0)$ .



$$\begin{aligned} \mathbf{f}'(u) &= (2u, 2-2u, 0) & \mathbf{f}''(u) &= (2, -2, 0) \\ \mathbf{f}'(u) \times \mathbf{f}''(u) &= (0, 0, -4) & |\mathbf{f}'(u)| &= 2\sqrt{1-2u+2u^2} \\ |\mathbf{f}'(u) \times \mathbf{f}''(u)| &= 4 & k(u) &= 1 / (2(1-2u+2u^2))^{1.5} \\ \mathbf{g}'(v) &= (2-2v, -2v, 0) & \mathbf{g}''(v) &= (-2, -2, 0) \\ \mathbf{g}'(v) \times \mathbf{g}''(v) &= (0, 0, -4) & |\mathbf{g}'(v)| &= 2\sqrt{1-2v+2v^2} \\ |\mathbf{g}'(v) \times \mathbf{g}''(v)| &= 4 & k(v) &= 1 / (2(1-2v+2v^2))^{1.5} \end{aligned}$$

da  $\mathbf{f}'(1) = \mathbf{g}'(0) = (2, 0, 0)$ , si sa che le due curve sono  $C^1$ . Poiché  $\mathbf{f}''(1) = (2, -2, 0)$  è diverso da  $\mathbf{g}''(0) = (-2, -2, 0)$ , **non** sono  $C^2$ . Poiché però le funzioni di curvatura sono le stesse anche senza osservare la figura ci si aspetta che la giunzione sia smooth.

Consideriamo adesso  $G^2$ . Poiché le curve sono  $C^1$ , possiamo considerare  $G^2$  partendo dall'ultima definizione.  $\mathbf{f}'(1) - \mathbf{g}'(0) = (4, 0, 0)$  è parallelo a  $(2, 0, 0)$  vettore tangente al punto di giunzione e per questo, i segmenti di curva sono  $G^2$  continui in  $(0, 1, 0)$ .

## Curve Razionali

Le rappresentazioni parametriche che usando i polinomi **non sono abbastanza potenti**, poiché molte curve (cerchi, ellissi e iperboli) non possono essere ottenute in questo modo. L'unico modo per ovviare al problema è usare le **coordinate omogenee**.

In questo caso, una curva nello spazio (o nel piano) è rappresentata da **quattro (tre)** funzioni piuttosto che tre (due):

$$\begin{aligned} \mathbf{F}(u) &= (x(u), y(u), z(u), W(u)) && \text{Spazio} \\ \mathbf{F}(u) &= (x(u), y(u), W(u)) && \text{Piano} \end{aligned}$$

Con  $u$  parametro che varia in un certo intervallo  $[a, b]$ .

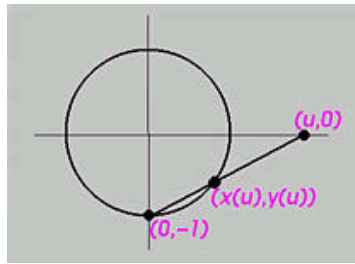
La forma originale si ottiene dividendo tutte le componenti per l'ultima aggiunta:

$$\mathbf{f}(u) = (x(u)/W(u), y(u)/W(u), z(u)/W(u)) \quad \mathbf{f}(u) = (x(u)/W(u), y(u)/W(u))$$

Una curva parametrica nella forma **omogenea** si dice **curva razionale** (prima curva polinomiale).

## Forme razionali di curve comuni

Cerchiamo di calcolare la forma razionale di una circonferenza.



consideriamo il caso di una circonferenza di raggio 1 e centro nell'origine degli assi coordinati cartesiani. Sia  $u$  il parametro di definizione e sia l'asse  $x$  la retta in cui  $u$  si può muovere. Uniamo quindi il punto  $(u, 0)$  (e quindi in corrispondenza del valore  $u$  del parametro) ed il punto  $(0, -1)$ .

Sia  $(x(u), y(u))$  il secondo punto in cui questa retta interseca la circonferenza. Quando  $u$  si muove sull'asse  $x$  il punto si muove sulla circonferenza. La retta che unisce  $(u, 0)$  con  $(0, -1)$  è  $x = u y + u$ .

L'equazione del cerchio è  $x^2 + y^2 = 1$ . Risolvendo in  $y$  si trovano due soluzioni una è il punto con valore di  $y = -1$ , l'altra:  $y = (1 - u^2) / (1 + u^2)$ .

Sostituendo questa  $y$  nell'equazione della retta si ha  $x = (2u) / (1 + u^2)$ . Quindi, per ogni  $u$ , il punto sulla circonferenza è

$$x = 2u / (1 + u^2) \qquad y = (1 - u^2) / (1 + u^2)$$

Quindi anche per la circonferenza esiste una espressione tramite curve razionali.

### ***Teoremi di uniformizzazione***

Le forme razionali possono anche essere convertite in forme polinomiali. Il trucco è eliminare il parametro  $u$  da tutte le equazioni parametriche. Tale processo è chiamato *implicitization*.

**Il viceversa non è sempre valido**, esistono curve che non hanno forme parametriche polinomiali o razionali ( $y^2 = x^3 + x + c$ ). Questo risultato dipende dai teoremi di uniformizzazione.

D'altro canto, le forme razionali e parametriche sono molto più semplici da manipolare.



# CURVE DI BÉZIER

## Introduzione

- La rappresentazione parametrica iniziale non è abbastanza *geometrica*. O meglio, osservando la shape parametrica e abbastanza difficile comprendere in modo immediato la GEOMETRIA che sta al di sotto senza ulteriori analisi.
- I coefficienti delle equazioni non hanno alcun significato geometrico ed è quasi impossibile da predire il cambiamento della shape della curva qualora uno o più coefficienti vengano modificati.
- Progettare una curva in grado di seguire un certo profilo è molto complesso.

Gli utenti che desiderano disegnare curve o superfici complesse, in genere, non si preoccupano degli aspetti matematici: hanno interesse affinché il lavoro sia soddisfacente e che si possa concludere senza troppa fatica!

Un sistema in grado di soddisfare gli utenti che devono creare curve complesse deve essere

1. **Intuitivo:**  
In modo che ogni passo di un algoritmo ed ogni algoritmo possiedano una interpretazione intuitiva e geometrica.
2. **Flessibile:**  
in grado di fornire più controllo in fase di progettazione e visualizzazione della shape di una curva. La metodologia con cui una curva viene creata e visualizzata dovrebbe essere facile e geometrica piuttosto che basata sulla manipolazione di equazioni.
3. **Metodo Unificato:**  
Il modo per rappresentare, creare e visualizzare tipi differenti di curve (rette, sezioni coniche, curve cubiche) deve essere lo stesso.
4. **Invariante:**  
La curva non deve cambiare la geometria relativa se vengono applicate trasformazioni geometriche (traslazione, rotazione e proiezione parallela e prospettica).
5. **Efficienza e Stabilità numerica:**  
Un utente non deve preoccuparsi della "bellezza" della geometria alla base; in più, la computazione non deve "distorcere" la shape della curva (stabilità numerica).

## Vantaggi

Tutte hanno in comune i seguenti vantaggi:

1. Dati i CONTROL POINTS, il sistema riesce a produrre una curva la cui shape segue la shape del poligono di controllo....
2. Cambiando le posizioni dei CONTROL POINTS si modifica la curva.
3. È possibile aggiungere nuovi punti di controllo o altre informazioni **SENZA** cambiare la shape della curva. In questo modo, si ha più libertà per la fase di editing della stessa (aggiungere punti di controllo aumenta il *grado della libertà* della curva).
4. È possibile spezzare la curva in due per un'operazione di micro-editing locale ed unire nuovamente le parti in un secondo tempo con un certo grado di continuità.
5. Esistono algoritmi intuitivi e procedure numericamente stabili per l'individuazione di un punto su di una curva **senza** conoscere l'equazione della stessa.
6. Conoscute le curve, le superfici sono semplici da comprendere e maneggiare. Più precisamente, il passaggio da curva a superficie è semplicissimo, tutto quello imparato sulle curve può essere applicato alle superfici.

## Un po' di storia

- **CURVE DI BÉZIER** : Scoperte da de Casteljau e Pierre E. Bézier alla fine '50 - inizi '60.
- **B-SPLINE**: scoperte da N. Lobachevsky e rivisitate da C. de Boor, M. Cox e L. Mansfield alla fine degli anni '70.
- Curve di Bézier casi speciali di B-spline.
- B-spline e curve di Bézier sono curve parametriche polinomiali. Sappiamo che le forme parametriche polinomiali non possono rappresentare alcuni tipi di curve semplici quale è, ad esempio, la circonferenza.
- L'introduzione di coordinate omogenee, che rendono le B-spline e le curve di Bézier razionali, permettono l'introduzione delle **RATIONAL BÉZIER CURVES** e **NON-UNIFORM RATIONAL B-SPLINES (NURBS)**.
- Le **RATIONAL BÉZIER CURVES** sono più potenti delle curve di Bézier e sono in grado di rappresentare cerchi ed ellissi.
- Le **NURBS** sono più potenti delle B-spline.

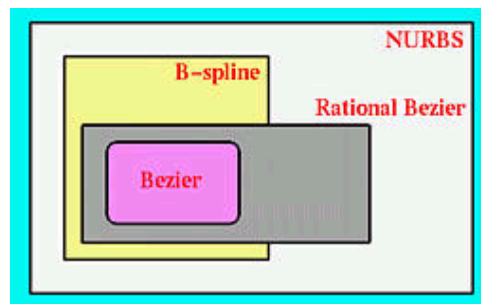


Figura 6 - Rapporto tra i diversi tipi di rappresentazione

## Curve di Bézier

Dati  $n + 1$  punti  $\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2 \dots \mathbf{p}_n$  nello spazio, detti **Control Points**, la **curva di Bézier** definita è:

$$\mathbf{p}(u) = \sum_{i=0}^n B_{n,i}(u) \mathbf{p}_i \quad \text{con } u \in [0, 1]$$

dove i **coefficienti di Bézier** (polinomi di Bernstein) sono definiti come segue:

$$B_{n,i}(u) = \frac{n!}{i!(n-i)!} u^i (1-u)^{n-i}$$

- Il punto che corrisponde ad  $u$  sulla curva di Bézier è la **media pesata di tutti i punti di controllo**, in cui i pesi sono i coefficienti  $B_{n,i}(u)$ .
- I segmenti  $\mathbf{p}_0\mathbf{p}_1, \mathbf{p}_1\mathbf{p}_2, \dots, \mathbf{p}_{n-1}\mathbf{p}_n$ , sono **legs**
- L'unione delle legs è la **polyline di controllo**.

**Bézier Osservazioni:**

- Tutti i coefficienti sono non negativi.
- Sia  $u$  che  $i$  possono essere zero e così pure  $1-u$  e  $n-i$ , per convenzione  $0^0=1$ .
- La curva segue più o meno la polyline.

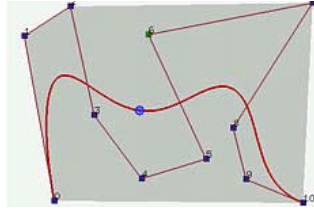


Figura 7 - Curva di Bézier con 11 punti di controllo, ed il punto sulla curva a  $u = 0,4$ .

**Bézier : Proprietà**

1. Dati  $n+1$  punti di controllo, **il grado di una curva di Bézier è  $n$** :  
In ogni coefficiente, l'esponente della  $u$  è  $i+(n-i) = n$ .
2.  $p(u)$  **passa per  $p_0$  e  $p_n$**
3. **Non-negatività:**  
Tutti i coefficienti di Bézier sono non negativi.
4. **Partition of unit:**  
Il numero di coefficienti di Bézier è  $n+1$ .  
La loro somma è 1. Tutti i coefficienti di Bézier sono i coefficienti nell'espansione dell'espressione  $1=(u + (1 - u))^n$ . Quindi, **la loro somma è 1**.
5. Il valore di ogni coefficiente di Bézier è **compreso sempre tra 0 e 1**.

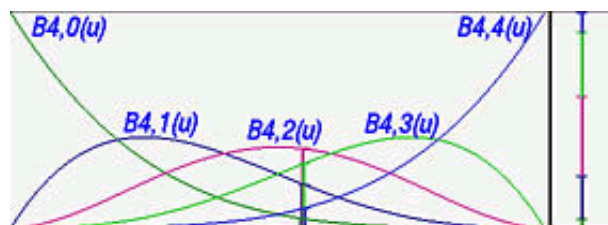
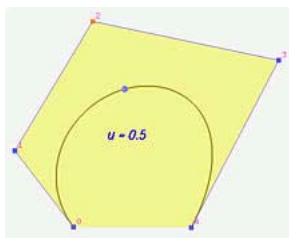


Figura 8 - Curva di Bézier con 5 p.ti di controllo e relativi Coefficienti

**6. Proprietà del Guscio Convesso:**

La curva di Bézier definita dai  $n + 1$  punti di controllo giace completamente nel guscio convesso dei punti di controllo. Il guscio convesso d'un insieme dei punti è il più piccolo insieme convesso che contiene tutti i punti dati. Il segmento riga che unisce due punti arbitrari in un insieme convesso si trova sempre nell'insieme convesso.

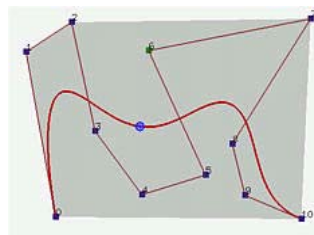


Figura 9 - un esempio di Convex Hull di 11 punti (praticamente lo "sfondo"...)

**Per questa proprietà la curva generata non esce dalla regione e quindi è in un certo senso "PREVEDIBILE"**

### 7. Variation Diminishing Property

Se la curva è contenuta in un piano allora non esistono segmenti che intersecano la curva più volte di quanto intersechino la control polyline.

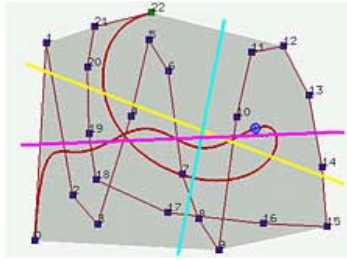


Figura 10 - Variation Diminishing Property

**La curva, girandola e torcendola, non può essere più complessa della relativa polyline.**

### 8. Affine Invariance

Se viene applicata una trasformazione affine ad una curva di Bézier, il risultato può essere costruito dall'immagine affine dei suoi punti di controllo.

#### *u non in [ 0,1 ]*

Certe volte una curva di Bézier può essere definita in un intervallo  $[a,b]$  piuttosto che  $[0,1]$ . Occorre, quindi, un cambiamento di variabile. Trovare una nuova  $\underline{u}$  che si possa muovere nell'intervallo  $[0,1]$  e sia collegata ad  $u$ :

$$\underline{u} = \frac{u - a}{b - a}$$

Inserendo questa  $\underline{u}$  nei coefficienti di Bézier  $B_{n,i}(\underline{u})$  si ottiene quanto segue:

$$B_{n,i}(u) = \frac{n!}{i!(n-i)!} \left( \frac{u - a}{b - a} \right)^i \left( 1 - \frac{u - a}{b - a} \right)^{n-i}$$

Questi nuovi coefficienti di Bézier definiscono una curva di Bézier nell'intervallo  $[a,b]$ .

## Bézier: Riassunto

Per definire una curva di Bézier del grado  $n$ , occorre scegliere  $n+1$  punti di controllo nello spazio in modo che indichino approssimativamente la shape della curva desiderata.

Quando uno o più punti di controllo vengono spostati, la shape della curva di Bézier cambia di conseguenza.

Ma, la curva si trova sempre all'interno del guscio convesso definito dai punti di controllo (proprietà del guscio convesso)

La shape della curva generata è meno complessa della polyline di controllo (Variation Diminishing Property).

## Bézier: Muovere i punti di controllo

Cambiare la posizione d'un punto di controllo cambierà la shape della curva di Bézier relativa.

### Come cambia la shape della curva se un punto di controllo è spostato verso una nuova posizione?

Si supponga che un punto di controllo  $\mathbf{P}_k$  venga spostato in una nuova posizione  $\mathbf{P}_k + \mathbf{v}$ , dove  $\mathbf{v}$  è un vettore che fornisce la direzione e la lunghezza dello spostamento:

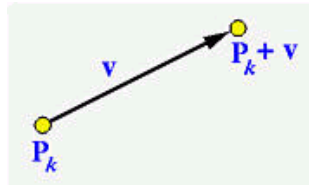


Figura 11 -  $\mathbf{v}$  vettore di spostamento di un punto di controllo

La vecchia curva di Bézier è definita secondo la formula:  $\mathbf{p}(u) = \sum_{i=0}^n B_{n,i}(u) \mathbf{p}_i$

Sia  $\mathbf{C}(u)$  la nuova curva. Inserendo  $\mathbf{P}_k + \mathbf{v}$  l'equazione della curva di Bézier  $\mathbf{C}(u)$  diventa:

$$\begin{aligned} \mathbf{C}(u) &= \sum_{i=0}^{k-1} B_{n,i}(u) \mathbf{P}_i + B_{n,k}(u) (\mathbf{P}_k + \mathbf{v}) + \sum_{i=k+1}^n B_{n,i}(u) \mathbf{P}_i \\ &= \sum_{i=0}^n B_{n,i}(u) \mathbf{P}_i + B_{n,k}(u) \mathbf{v} \\ &= \mathbf{p}(u) + B_{n,k}(u) \mathbf{v} \end{aligned}$$

Poiché solo il  $K$ -esimo termine usa un nuovo punto  $\mathbf{P}_k + \mathbf{v}$ , la nuova curva è la somma della curva originale e d'un termine supplementare  $B_{n,k}(u) \mathbf{v}$ .

**il punto corrispondente di  $u$  sulla nuova curva è ottenuto traducendo il punto corrispondente di  $u$  sulla curva originale nel senso e nella distanza di  $B_{n,k}(u) \mathbf{v}$ .**

Dato  $u$ , si ha  $\mathbf{p}(u)$  sulla curva e  $\mathbf{C}(u)$  sulla nuova tale che  $\mathbf{C}(u) = \mathbf{p}(u) + B_{n,k}(u) \mathbf{v}$ . Poiché  $\mathbf{v}$  dà la direzione di movimento,  $\mathbf{C}(u)$  è il risultato di spostamento di  $\mathbf{p}(u)$  nella direzione di  $\mathbf{v}$ . La lunghezza di questo spostamento è  $B_{n,k}(u) \mathbf{v}$ .

**Esempio:** La distanza fra  $\mathbf{p}(0,5)$  e  $\mathbf{C}(0,5)$  è la lunghezza del vettore  $B(0,5) \mathbf{v} = 8! / (3!(8-3)!) \times 0.5^3 (1-0.5)^{8-3} \mathbf{v} = 0,22 \mathbf{v}$ .

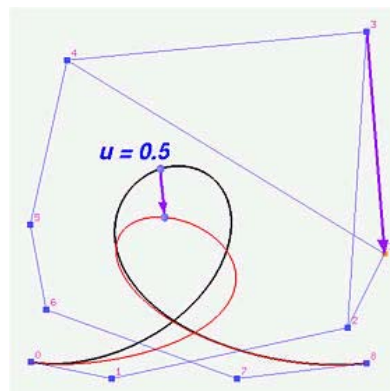


Figura 12 - Spostare un punto di controllo

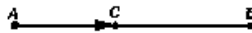
**OSSERVAZIONE:**

Poiché  $B_{n,k}(u)$  è non negativa in  $(0,1)$  (\*\* APERTO \*\*),  $B_{n,k}(u)\mathbf{v}$  non è un vettore nullo dentro  $(0,1)$ . Tranne che per i punti  $\mathbf{p}(0)$  e  $\mathbf{p}(1)$  **tutti i** punti sulla curva vengono spostati.

**Cambiare la posizione di un punto di controllo comporta il cambiamento globale della shape di una curva di Bézier.**

**De Casteljau Algorithm**

- **Scopo:** Determinare un punto su una curva di Bézier assegnata dato un valore di  $u$  fissato.
- Il concetto fondamentale dell'algorithm sta nello scegliere un punto  $C$  in un segmento  $AB$  tale che la distanza fra  $A$  e  $C$  e la distanza fra  $A$  e  $B$  è  $u$ .
- L'algorithm fornisce un modo per determinare  $C$ .



- Il vettore da  $A$  a  $B$  viene indicato con  $\mathbf{B}-\mathbf{A}$ .
- Visto che  $u$  è un rapporto  $\in[0,1]$ , il punto  $C$  è situato a  $u(\mathbf{B}-\mathbf{A})$ .
- Considerando  $A$ , il punto  $C$  è:

$$\mathbf{A} + u (\mathbf{B} - \mathbf{A}) = (1 - u) \mathbf{A} + u \mathbf{B} .$$

$$d(\mathbf{A},\mathbf{C}) / d(\mathbf{A},\mathbf{B}) = u$$

**Schema dell'algorithm**

1. Supponiamo di aver trovato  $\mathbf{p}(u)$ , con  $u$  in  $[0,1]$ .
2. Si consideri dalla polyline,  $00-01-02-03-...-0n$ ,
3. sulla **leg** da  $0i$  a  $0(i+1)$ , si usi la formula suddetta per trovare un punto  $1i$  tale che il rapporto della distanza fra  $0i$  e  $1i$  e la distanza fra  $0i$  e  $0(i+1)$  sia  $u$ .
4. In questo modo, si otterranno ***i* punti  $10,11,12,....,1(n-1)$** . Tali punti definiscono una nuova polyline formata da ***n-1* legs**.

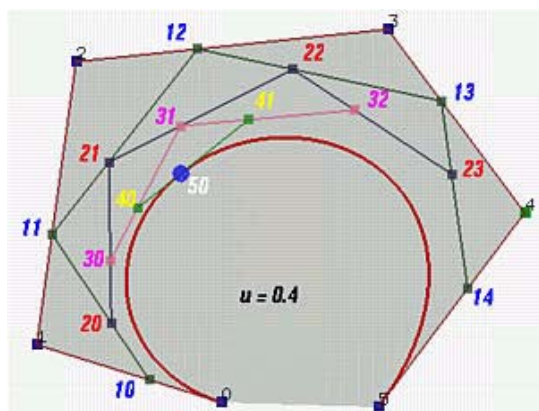


Figura 13 - L'algorithm di De Casteljau

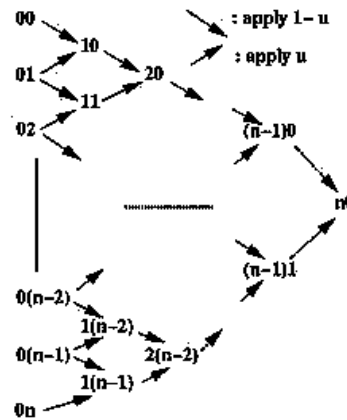
**Schema triangolare:**

Figura 14 - La procedura

**Algoritmo****Input:** allineamento  $\mathbf{p} [ 0: n ]$  di  $n + 1$  punti e numero reale  $u$ **Output:** punto sulla curva,  $\mathbf{p}(u)$ 

```

for i:=0 to n do
    q[i]:=p[i]; // Salvataggio dell'input
for k:=1 to n do
    for i:=0 to n-k do
        q[i] := (1-u)*q[i]+u*q[i+1];
        // applicazione della formula
return q[0];

```

**Derivate d'una curva di Bézier**

Per computare la tangente e le normali in un punto su una curva di Bézier, occorre computare la derivata in quel punto.

$$\mathbf{p}(u) = \sum_{i=0}^n B_{n,i}(u) \mathbf{p}_i \quad \text{curva di Bézier definita da } n+1 \text{ punti di controllo}$$

$$B_{n,i}(u) = \frac{n!}{i!(n-i)!} u^i (1-u)^{n-i} \quad \text{Funzioni Base per la Curve di Bézier}$$

- I punti di controllo sono costanti ed indipendenti dal parametro  $u$ , il calcolo della  $\mathbf{p}(u)$  si riduce a computare la derivata dei coefficienti di Bézier.

$$\frac{d}{du} B_{n,i}(u) = B'_{n,i}(u) = n(B_{n-1,i-1}(u) - B_{n-1,i}(u))$$

Quindi

$$\frac{d}{du} \mathbf{p}(u) = \mathbf{p}'(u) = \sum_{i=0}^{n-1} B_{n-1,i}(u) \{n(\mathbf{p}_{i+1} - \mathbf{p}_i)\}$$

Sia  $\mathbf{q}_0 = n(\mathbf{p}_1 - \mathbf{p}_0)$ ,  $\mathbf{q}_1 = n(\mathbf{p}_2 - \mathbf{p}_1)$ ,  $\mathbf{q}_2 = n(\mathbf{p}_3 - \mathbf{p}_2)$ , ...,  $\mathbf{q}_{n-1} = n(\mathbf{p}_n - \mathbf{p}_{n-1})$ .

$$\mathbf{p}'(u) = \sum_{i=0}^{n-1} B_{n-1,i}(u) \mathbf{q}_i$$

La derivata di  $\mathbf{p}(u)$  è una curva di Bézier di grado  $n-1$  definita dai punti di controllo  $n(\mathbf{p}_1 - \mathbf{p}_0)$ ,  $n(\mathbf{p}_2 - \mathbf{p}_1)$ ,  $n(\mathbf{p}_3 - \mathbf{p}_2)$ , ...,  $n(\mathbf{p}_n - \mathbf{p}_{n-1})$ .

- Chiamata **hodograph** della curva di Bézier originale.

### OSSERVAZIONE

- $\mathbf{p}_{i+1} - \mathbf{p}_i$  è vettore di direzione da  $\mathbf{p}_i$  a  $\mathbf{p}_{i+1}$  e  $n(\mathbf{p}_{i+1} - \mathbf{p}_i)$  è  $n$  volte più lungo della direzione del vettore.
- Una volta decisi i punti di controllo della curva originale, i punti di controllo della relativa derivata possono essere ottenuti immediatamente.

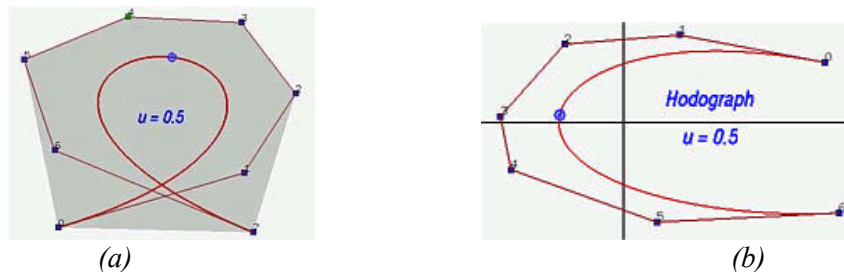


Figura 15 - curva di Bézier di grado 7 (a) e la relativa derivata come curva di Bézier di grado 6 (b)

### Osservazione di Tangenza

Con  $u = 0$  e  $u = 1$  si ha:

$$\mathbf{p}'(0) = n(\mathbf{p}_1 - \mathbf{p}_0) \quad (a) \quad \mathbf{p}'(1) = n(\mathbf{p}_n - \mathbf{p}_{n-1}) \quad (b)$$

- Il vettore tangente a  $u = 0$  è nella direzione di  $\mathbf{p}_1 - \mathbf{p}_0$  moltiplicata per  $n$ , e la prima leg è tangente alla curva di Bézier.
- Il vettore tangente a  $u = 1$  è nella direzione di  $\mathbf{p}_n - \mathbf{p}_{n-1}$  moltiplicato per  $n$  e l'ultima leg è tangente alla curva di Bézier.

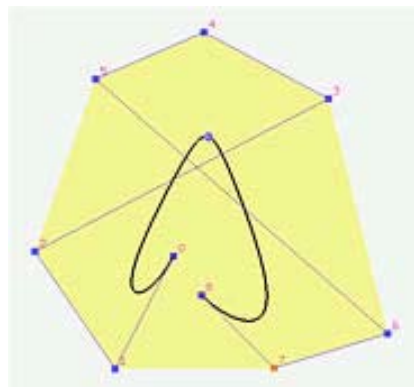


Figura 16 - Tangenza nei punti iniziale e finale

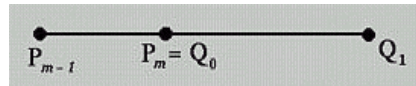


## Unire due curve Bézier con continuità $C^1$

La tangenza nella prima e ultima leg è un aiuto per unire due curve di Bézier. Consideriamo una curva definita da  $m+1$  punti di controllo  $\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_m$  di grado  $m$  ed una altra curva definita da  $n+1$  punti di controllo  $\mathbf{q}_0, \mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_n$  di grado  $n$ .

Se  $\mathbf{p}_m = \mathbf{q}_0$  si ha continuità  $C^0$ .

Nel caso di transizione smussata, i punti  $\mathbf{p}_{m-1}$ ,  $\mathbf{p}_m = \mathbf{q}_0$ ,  $\mathbf{q}_1$  **devono essere sulla stessa riga** in modo che le direzioni dei vettori  $\mathbf{p}_{m-1}\mathbf{p}_m$  e  $\mathbf{q}_0\mathbf{q}_1$  siano le stesse.



La continuità è, in questo caso di tipo  $G^1$ , visto che le due curve possiedono la stessa direzione dei vettore tangenti ed hanno quindi una retta comune di tangenza.

Per realizzare la continuità  $C^1$ , occorre assicurarsi che il vettore di tangenza quando  $u=1$  della prima curva,  $\mathbf{p}'(1)$  ed il vettore di tangenza con  $u=0$  della seconda curva,  $\mathbf{q}'(0)$ , siano identici:  $m(\mathbf{p}_m - \mathbf{p}_{m-1}) = n(\mathbf{q}_1 - \mathbf{q}_0)$

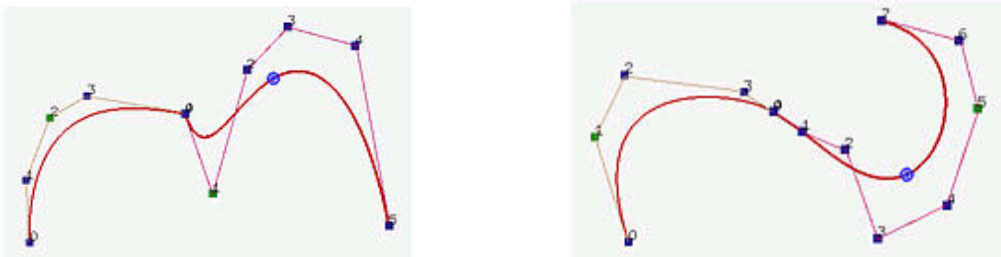


Figura 17 - Continuità  $G^1$  ma non  $C^1$

Se lasciamo il primo ed l'ultimo punto di controllo identici ( $\mathbf{p}_0 = \mathbf{p}_n$ ) e  $\mathbf{p}_1$ ,  $\mathbf{p}_0$  e  $\mathbf{p}_{n-1}$  collineari, la curva di Bézier generata sarà chiusa e unita con continuità  $G^1$  (Figura 18):

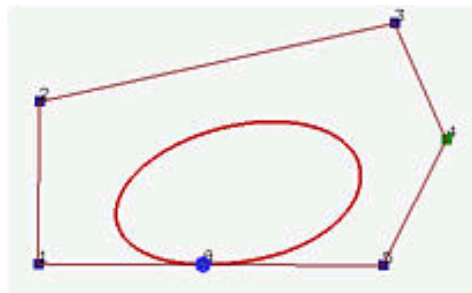


Figura 18 - Curva chiusa con continuità  $G^1$

## Legame tra Derivata e la procedura di de Casteljau

Isoliamo dall'equazione della derivata di una curva di Bézier il termine  $\mathbf{p}_{i+1} - \mathbf{p}_i$ :

$$\begin{aligned} \mathbf{p}'(u) &= \sum_{i=0}^{n-1} B_{n-1,i}(u) \{n(\mathbf{p}_{i+1} - \mathbf{p}_i)\} \\ &= n \left[ \left( \sum_{i=0}^{n-1} B_{n,i}(u) \mathbf{p}_{i+1} \right) - \left( \sum_{i=0}^{n-1} B_{n,i}(u) \mathbf{p}_i \right) \right] \end{aligned}$$

la derivata di una curva di Bézier è la differenza di due curve di Bézier di grado  $n-1$ .

Siano queste due curve  $C_1(u)$  e  $C_2(u)$ :

$$C_1(u) = \sum_{i=0}^{n-1} B_{n,i}(u) \mathbf{p}_{i+1}$$

$$C_2(u) = \sum_{i=0}^{n-1} B_{n,i}(u) \mathbf{p}_i$$

- La prima curva  $C_1(u)$  è definita dai punti di controllo  $\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n$ ,
- La seconda curva  $C_2(u)$  è definita dai punti di controllo  $\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_{n-1}$ .

$$\mathbf{p}'(u) = n(C_1(u) - C_2(u))$$

- Si può computare  $\mathbf{p}'(u)$ , fissato  $u$ , computando  $C_1(u)$  e  $C_2(u)$  con de Casteljau. Quindi, facendo la differenza e moltiplicando per  $n$ .

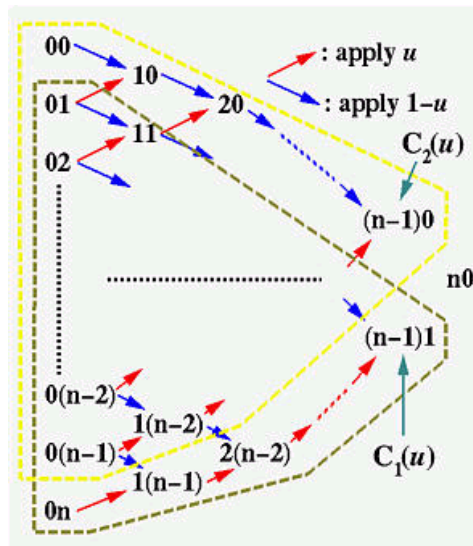


Figura 19 - La procedura di De Casteljau per il computo di  $C_1(u)$  e  $C_2(u)$

- $C_1(u)$  è definita dai punti  $01, 02, \dots, 0n$ ,  $\rightarrow (n-1)1$  è punto sulla curva  $C_1(u)$ .
- $C_2(u)$  è definita dai punti  $00, 01, \dots, 0(n-1)$ ,  $\rightarrow (n-1)0$  è punto sulla curva  $C_2(u)$ .
- Il vettore  $C_1(u) - C_2(u)$  è il vettore dal punto  $(n-1)0$  ad  $(n-1)1$  ed la derivata di  $\mathbf{p}(u)$  è  $n$ -volte  $C_1(u) - C_2(u)$ .
- la polyline da  $(n-1)0$  a  $(n-1)1$  in de Casteljau ha la stessa direzione del vettore tangente a  $\mathbf{p}(u)$ ,

**L'ultima polyline, un segmento, della rete di de Casteljau è tangente alla curva di Bézier  $\mathbf{p}(u)$**

**Un esempio:**

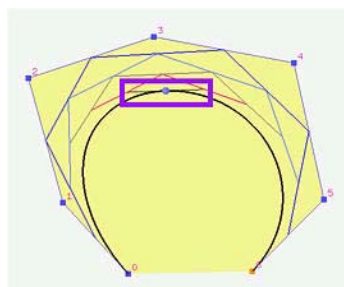


Figura 20

- Il punto segnato in grassetto corrisponde a  $u = 0,5$ .
- L'ultimo segmento è tangente.

## Curve di Bézier : Derivate superiori alla prima

- Lo schema dell'algoritmo di De Casteljau può essere usato anche per la computazione delle derivate superiori alla derivata prima.

Se la derivata  $\mathbf{p}'(u)$  è:

$$\mathbf{p}'(u) = \sum_{i=0}^{n-1} B_{n-1,i}(u) \mathbf{q}_i$$

Applicando la formula alla curva di Bézier derivata:

$$\begin{aligned} \mathbf{p}''(u) &= \sum_{i=0}^{n-2} B_{n-2,i}(u) \{(n-1)(\mathbf{q}_{i+1} - \mathbf{q}_i)\} \\ &= \sum_{i=0}^{n-2} B_{n-2,i}(u) \{n(n-1)(\mathbf{p}_{i+2} - 2\mathbf{p}_{i+1} + \mathbf{p}_i)\} \end{aligned}$$

Definiamo  $\mathbf{D}_i^0$  il punto di controllo  $\mathbf{P}_i$  per  $i=0, \dots, n$ . Allora, definiamo la prima differenza la  $\mathbf{D}_i^1$  come differenza del livello precedente:

$$\begin{aligned} \mathbf{D}_0^1 &= \mathbf{D}_1^0 - \mathbf{D}_0^0 = \mathbf{p}_1 - \mathbf{p}_0 \\ \mathbf{D}_1^1 &= \mathbf{D}_2^0 - \mathbf{D}_1^0 = \mathbf{p}_2 - \mathbf{p}_1 \\ &\vdots \\ \mathbf{D}_{n-2}^1 &= \mathbf{D}_{n-1}^0 - \mathbf{D}_{n-2}^0 = \mathbf{p}_{n-1} - \mathbf{p}_{n-2} \\ \mathbf{D}_{n-1}^1 &= \mathbf{D}_n^0 - \mathbf{D}_{n-1}^0 = \mathbf{p}_n - \mathbf{p}_{n-1} \end{aligned}$$

La  $k$ -esima differenza è definita:

$$\mathbf{D}_i^k = \mathbf{D}_{i+1}^{k-1} - \mathbf{D}_i^{k-1} \quad 0 \leq i \leq n-k$$

Riscriviamo la  $\mathbf{p}'(u)$  come segue, usando la  $\mathbf{D}_i^1$ :

$$\begin{aligned} \mathbf{p}'(u) &= n \sum_{i=0}^{n-1} B_{n-1,i}(u) (\mathbf{p}_{i+1} - \mathbf{p}_i) \\ &= n \sum_{i=0}^{n-1} B_{n-1,i}(u) (\mathbf{D}_{i+1}^0 - \mathbf{D}_i^0) \\ &= n \sum_{i=0}^{n-1} B_{n-1,i}(u) \mathbf{D}_i^1 \end{aligned}$$

Per computare  $\mathbf{p}''(u)$  usiamo  $\mathbf{p}'(u)$ :

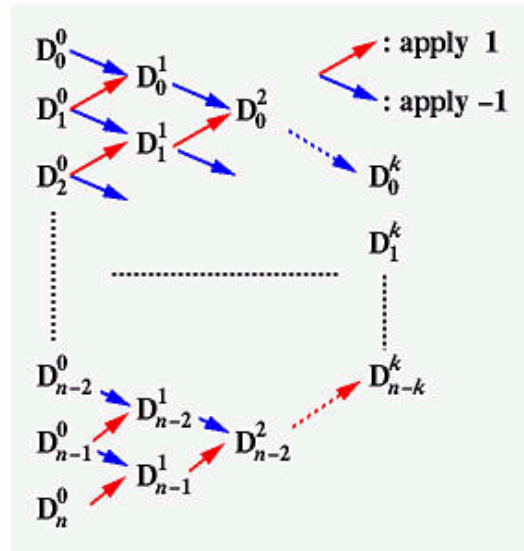
$$\begin{aligned} \mathbf{p}''(u) &= n(n-1) \sum_{i=0}^{n-2} B_{n-2,i}(u) (\mathbf{D}_{i+1}^1 - \mathbf{D}_i^1) \\ &= n(n-1) \sum_{i=0}^{n-2} B_{n-2,i}(u) \mathbf{D}_i^2 \end{aligned}$$

Continuando in questo senso possiamo computare  $\mathbf{p}^{[k]}(u)$ :

$$\begin{aligned} \mathbf{p}^{[k]}(u) &= n(n-1)(n-2) \cdots (n-k+1) \sum_{i=0}^{n-k} B_{n-k,i}(u) (\mathbf{D}_{i+1}^{k-1} - \mathbf{D}_i^{k-1}) \\ &= n(n-1)(n-2) \cdots (n-k+1) \sum_{i=0}^{n-k} B_{n-k,i}(u) \mathbf{D}_i^k \end{aligned}$$

## Schema di computazione

- Quindi per computare la  $k$ -esima derivata di  $p(u)$  ad un particolare valore  $u$ , possiamo computare i  $D_i^k$  ed applicare l'algoritmo di de Casteljau per computare i punti corrispondenti ad  $u$  sulla curve di Bézier definite dai  $D_i^k$ .



## Suddivisione di una curva di Bézier

- Il significato di **suddivisione** di una data curva di Bézier è di tagliare  $p(u)$  per un certo valore di  $u$  in due segmenti di curva, ciascuno dei quali è ancora una curva di Bézier.
- Poiché le curve risultanti devono avere i propri punti di controllo, l'insieme originale dei punti di controllo viene eliminato.
- Poiché la curva originale di Bézier era di grado  $n$ , tagliata in due parti, ciascuna è ancora di grado  $n$ .

### Problema

Dato un insieme di  $n+1$  punti di controllo  $\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n$  e un valore di parametro  $u \in [0, 1]$ , si desidera trovare **due** insiemi di  $n+1$  punti di controllo  $\mathbf{q}_0, \mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_n$  e  $\mathbf{r}_0, \mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_n$  tali che la curva di Bézier definita dai  $\mathbf{q}_i$  ( $\mathbf{r}_i$ ) è la parte della curva originale di Bézier in  $[0, u]$  ( $[u, 1]$ ).

- L'algoritmo di de Casteljau per la valutazione del punto  $\mathbf{p}(u)$  sulla curva fornisce tutte le informazioni necessarie.

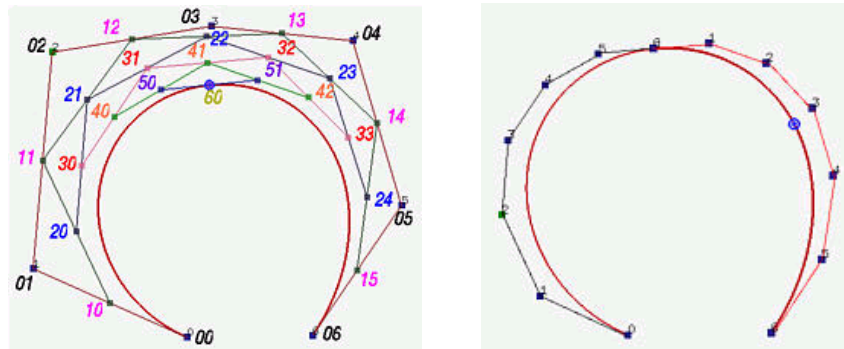


Figura 21 - l'applicazione di De Casteljau e la suddivisione della curva iniziale

- la polyline di sinistra consiste nei punti

$$\mathbf{p}_{00} = \mathbf{p}_0, \mathbf{p}_{10}, \mathbf{p}_{20}, \mathbf{p}_{30}, \mathbf{p}_{40}, \mathbf{p}_{50}, \mathbf{p}_{60} = \mathbf{p}(u),$$

- la polyline di destra consiste nei punti

$$\mathbf{p}_{60} = \mathbf{p}(u), \mathbf{p}_{51}, \mathbf{p}_{42}, \mathbf{p}_{33}, \mathbf{p}_{24}, \mathbf{p}_{15}, \mathbf{p}_{06} = \mathbf{p}_6$$

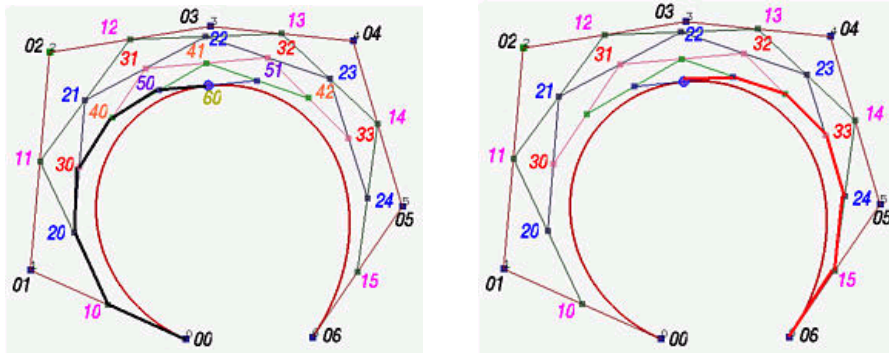


Figura 22 - Selezione dei punti della suddivisione

## De Casteljau per la suddivisione

- Si consideri lo schema triangolare del calcolo dell' algoritmo di de Casteljau. Fissata una  $u$ , si computano  $n$  passi per ottenere  $\mathbf{p}(u)$ .
- Nel corso del calcolo, è possibile raccogliere i primi e gli ultimi punti di ogni colonna e, alle fine del processo, la prima collezione (*l'ultima*) di punti fornisce la suddivisione relativa alla parte della curva originale definita su  $[0, u]$  ( $[u, 1]$ ).

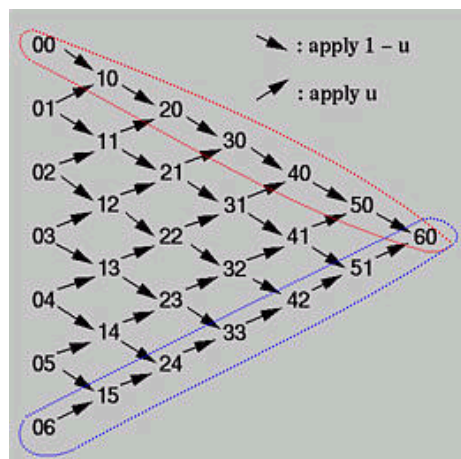


Figura 23 - De Casteljau per trovare i punti per la suddivisione di una curva

## Suddividere: perché?

La suddivisione della curva ha molte applicazioni.

- Computazione dell'intersezione di due curve di Bézier,
- Rappresentazione delle curve di Bézier.

Si consideri il caso di progettazione di una curva, che non è del tutto come la si desiderava. In questo senso, si potrebbe voler suddividerla in due parti in un punto adatto, un lato che ci soddisfa e l'altro no.

Sul lato che non convince, è possibile applicare la suddivisione altre volte, ricordando che per tenere le congiunzioni smooth occorre rendere collineari i segmenti ultimo e primo dei punti di contatto.

## Curva di Bézier: Aumentare il grado

Molte applicazioni che coinvolgono due o più curve di Bézier richiedono che tutte le curve abbiano lo stesso grado.

Inoltre, curve di Bézier di grado elevato richiedono un tempo maggiore di elaborazione ma hanno più flessibilità in fase di progettazione della relativa shape.

- È molto utile poter e saper aumentare il grado di una curva di Bézier **senza** cambiare la relativa shape.

### Problema

Si supponga di avere una curva di Bézier di grado  $n$  definita da  $n+1$  punti di controllo  $\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n$  e che si desideri aumentarne il grado ad  $n+1$  **senza** cambiare la relativa shape.

Una curva di Bézier di grado  $n+1$  è definita da  $n+2$  punti di controllo, per questo occorre trovare un nuovo insieme di punti di controllo.

Ovviamente,  $\mathbf{p}_0$  e  $\mathbf{p}_n$  devono appartenere al nuovo insieme poiché la nuova curva passa per i due punti (come le vecchia).

- Occorrono quindi  $n$  punti di controllo nuovi.

Sia  $\{\mathbf{q}_0, \mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_{n+1}\}$  il nuovi insieme di punti (che comprende anche  $\mathbf{p}_0$  e  $\mathbf{p}_n$ ). Gli  $n$  punti rimanenti vengono computati come segue:

$$\mathbf{q}_i = \frac{i}{n+1} \mathbf{p}_{i-1} + \left(1 - \frac{i}{n+1}\right) \mathbf{p}_i \quad 1 \leq i \leq n$$

Ovvero per ogni  $i$  si ha:

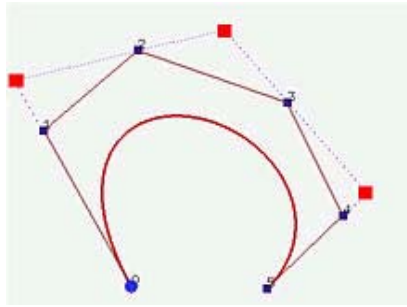
$$\begin{aligned} \mathbf{q}_1 &= \frac{1}{n+1} \mathbf{p}_0 + \left(1 - \frac{1}{n+1}\right) \mathbf{p}_1 \\ \mathbf{q}_2 &= \frac{2}{n+1} \mathbf{p}_1 + \left(1 - \frac{2}{n+1}\right) \mathbf{p}_2 \\ \mathbf{q}_3 &= \frac{3}{n+1} \mathbf{p}_2 + \left(1 - \frac{3}{n+1}\right) \mathbf{p}_3 \\ &\vdots \\ \mathbf{q}_{n-1} &= \frac{n-1}{n+1} \mathbf{p}_{n-2} + \left(1 - \frac{n-1}{n+1}\right) \mathbf{p}_{n-1} \\ \mathbf{q}_n &= \frac{n}{n+1} \mathbf{p}_{n-1} + \left(1 - \frac{n}{n+1}\right) \mathbf{p}_n \end{aligned}$$

**Ogni leg della polyline originale contiene esattamente un nuovo punto di controllo.**

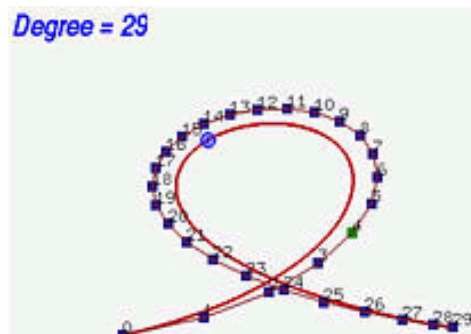
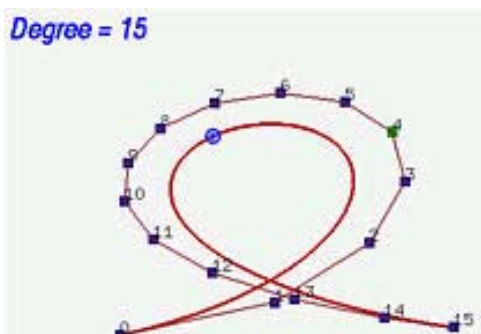
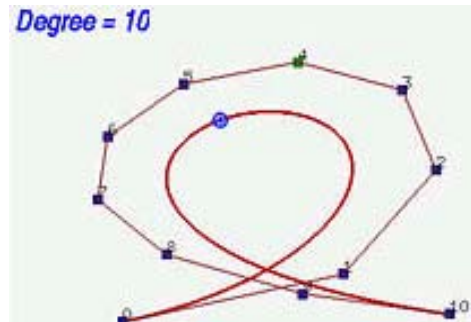
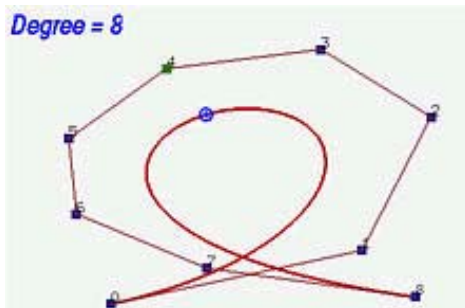
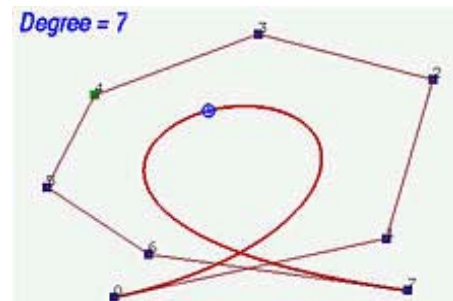
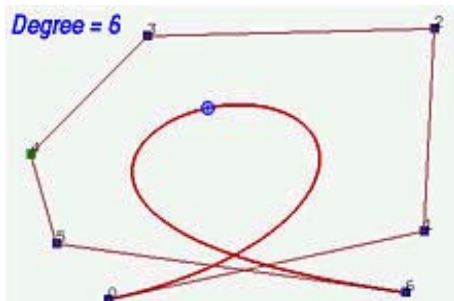
### Osservazioni

- Come nel caso dell'algoritmo di de Casteljau nel quale, un punto  $C$  su un segmento  $AB$  era tale che il rapporto tra  $d(A,C)$  e  $d(A,B)$  è  $u$  (fisso), ( $C = (1-u)A + uB$ ), qui il rapporto della distanza fra  $\mathbf{p}_{i-1}$  e  $\mathbf{q}_i$  e la distanza fra  $\mathbf{p}_{i-1}$  e  $\mathbf{p}_i$  è  $1-i/(n+1)$ .
- L'unica differenza è che, nel caso di De Casteljau il rapporto è fisso, mentre qui varia con il variare di  $i$ .

- Poiché ogni leg della polyline originale di controllo contiene un nuovo punto di controllo, il processo di sostituire l'insieme vecchio di punti con quello nuovo può essere visto come tagliare gli angoli relativi al poligono di controllo originale.



- Con l'aumentare del grado della curva aumenta anche il numero di punti di controllo. La nuova polyline di controllo si muove avvicinandosi alla curva.



# B-SPLINE

## Motivazione

- Consideriamo il caso di dover studiare il profilo di un vaso. Nell'esempio si ha una curva di Bézier di grado 11; in essa è difficile modellare il collo tra  $\mathbf{p}_4$  e  $\mathbf{p}_5$ .
- È possibile aggiungere punti di per avere maggior peso. Questo aumenta il grado, complicando il lavoro.

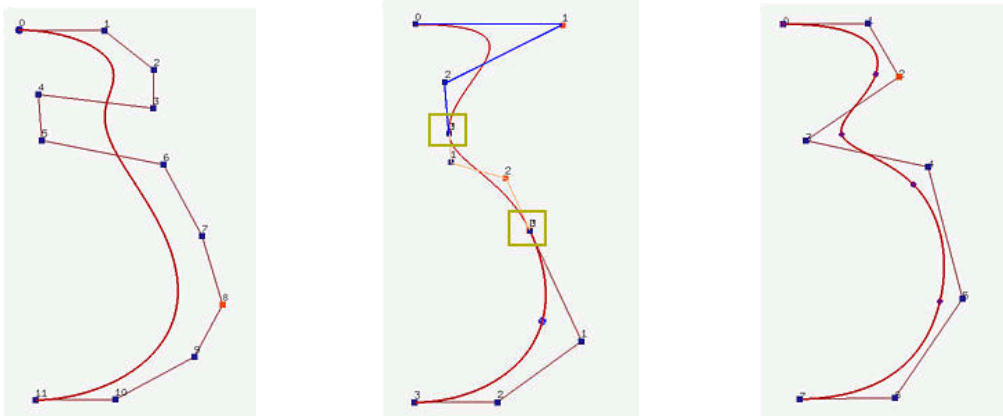


Figura 24 - Necessità delle B-Spline. La terza figura è una B-spline di grado 3 con 8 punti di controllo e 5 segmenti di curva di grado 3 che si uniscono per formarla.

- È possibile dividere la curva in due (o più) parti. Ma mantenere uno stato di collinearità (continuità  $G^1$ ) può essere complesso.
- È possibile mantenere segmenti di curva con grado basso senza doversi preoccupare della collinearità della congiunzioni
- Le **B-spline sono generalizzazioni delle curve di Bézier.**
- È possibile spostare i punti di controllo per modificare la shape della curva come si fa per le curve di Bézier.
- È possibile anche modificare la suddivisione della curva.

## OSSERVAZIONE:

- Le B-spline hanno più alto grado di libertà delle curve di Bézier.
- **La suddivisione diretta della curva è difficile. Invece, è possibile suddividere il dominio della curva.**

Sia il dominio di applicazione  $[0,1]$ , siano fissati  $0 \leq u_0 \leq u_1 \leq \dots \leq u_m \leq 1$  punti detti **knots**. In corrispondenza dei valori  $u_i$ , i punti  $\mathbf{p}(u_i)$  suddividono la curva e, per questo, modificare la suddivisione di  $[0,1]$  comporta una modifica sulla shape della curva.



## RIASSUNTO

Progettare una B-spline, comporta definire:

- un insieme di punti di controllo,
- un insieme di **knots**
- alcuni coefficienti, uno per ogni punto di controllo in modo che tutti i segmenti della curva si uniscano secondo un certo livello di continuità.

Il calcolo dei coefficienti è il punto più complesso delle B-spline perché bisogna accertarsi che sussistano certe condizioni di continuità sui pezzi di curva.

## Funzioni Base per B-spline

Le funzioni base di Bézier sono usate come **pesi** nelle definizioni delle curve. **Le funzioni base B-spline saranno usate nello stesso modo.**

Le B-spline sono costruite in modo che:

- (1) Il dominio è suddiviso in **knots** e
- (2) le funzioni base non sono sempre non negative nell'intervallo assegnato.

Sia  $U$  un insieme di  $m+1$  numeri reali non decrescenti,

$$u_0 \leq u_1 \leq u_2 \leq \dots \leq u_m,$$

- ogni  $u_i$  è detto **knot**,
- l'insieme  $U$  è detto **Knot vector** e
- l'intervallo semi-aperto  $[u_i, u_{i+1})$  è chiamato the **i-th knot span**

Si noti che alcune  $u_i$  possono essere uguali e per questo alcuni intervalli essere nulli.

### Molteplicità di un nodo

Se un nodo  $u_i$  compare  $k$  volte, con  $k > 1$ , è detto **nodo multiplo di molteplicità  $k$** ,  $u_i(k)$ ; altrimenti, è detto **nodo semplice**. Se i nodi sono equidistanti, il vettore dei nodi  $U$  è detto **uniforme**, altrimenti, **non-uniforme**.

- I nodi sono punti di divisione dell'intervallo  $[u_0, u_m]$ .
- Le funzioni base B-spline hanno un dominio all'interno di ogni intervallo.

### Funzioni Base

Per definire le funzioni base B-spline, occorre definire il grado relativo **p**. La **formula di Cox-de Boor** definisce la funzione di grado  $p$ ,  $N_{i,p}(u)$ , come segue:

$$N_{i,0}(u) = \begin{cases} 1 & \text{if } u_i \leq u < u_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u)$$

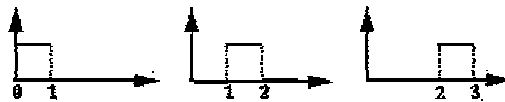
Figura 25 - Cox-de Boor recursion formula.

Se il grado è zero ( $p = 0$ ), sono tutte **funzioni step**. Cioè la funzione base  $N_{i,0}(u)$  è 1 se  $u$  si trova nell'intervallo  $i$ -esimo  $[u_i, u_{i+1})$ .

### Esempio

Dati 4 **knots**  $u_0 = 0, u_1 = 1, u_2 = 2$  ed  $u_3 = 3$ :

- i **knot span** sono  $[0,1), [1,2), [2,3)$
- le funzioni di base di grado 0 sono:
  - $N_{0,0}(u) = 1$  in  $[0,1)$  e 0 altrove,
  - $N_{1,0}(u) = 1$  in  $[1,2)$  e 0 altrove e
  - $N_{2,0}(u) = 1$  in  $[2,3)$  e 0 altrove.



## B-Spline: legame tra De Casteljaeu e le Funzioni Base

Consideriamo il solito schema di **calcolo triangolare**: tutti gli span si trovano sulla prima colonna a sinistra e tutte le funzioni di base di grado zero sulla seconda.

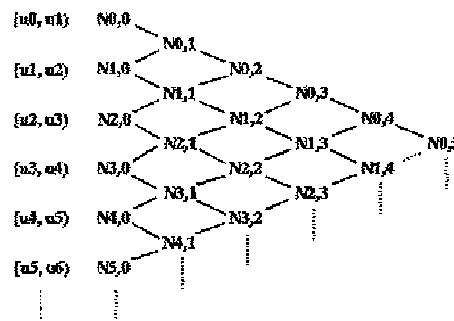


Figura 26 - schema di calcolo triangolare delle funzioni Base

- Per computare  $N_{i,1}(u)$ , devono essere utilizzate  $N_{i,0}(u)$  e  $N_{i+1,0}(u)$ .
- Tutti gli  $N_{i,1}(u)$  vengono scritti nella terza colonna.
- Una volta che tutta la colonna degli  $N_{i,1}(u)$  è stata computata, è possibile computare  $N_{i,2}(u)$ . Questo processo continua fino a che tutti gli  $N_{i,p}(u)$  sono stati computati.

### Esempio

Si provi a computare  $N_{0,1}(u)$  e  $N_{1,1}(u)$ . Per  $N_{0,1}(u)$ ,  $i = 0$  e  $p = 1$ , e dalla definizione si ha

$$N_{0,1}(u) = (u - u_0) / (u_1 - u_0) N_{0,0}(u) + (u_2 - u) / (u_2 - u_1) N_{1,0}(u)$$

Poiché  $u_0 = 0$ , e  $u_1 = 1$  e  $u_2 = 2$ ,

$$N_{0,1}(u) = u N_{0,0}(u) + (2 - u) N_{1,0}(u)$$

- $N_{0,0}(u)$  è non nulla in  $[0,1)$  e
- $N_{1,0}(u)$  è non nulla in  $[1,2)$ ,
- quando  $u$  è in  $[0,1) \cup [1,2)$ , soltanto  $N_{0,0}(u)$  ( $N_{1,0}(u)$ ) contribuisce al calcolo di  $N_{0,1}(u)$ .

- Se  $u$  varia in  $[0,1)$ ,  $N_{0,1}(u)$  è pari a  $uN_{0,0}(u) = u$ , e quando  $u$  varia in  $[1,2)$ ,  $N_{0,1}(u)$  è  $(2 - u)N_{1,0}(u) = (2 - u)$ .

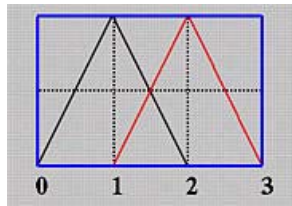


Figura 27 - le righe nere  $N_{0,1}(u)$ , righe rosse  $N_{1,1}(u)$

Disponibili  $N_{0,1}(u)$  e  $N_{1,1}(u)$ , possiamo computare  $N_{0,2}(u)$ . Dalla definizione si ha:

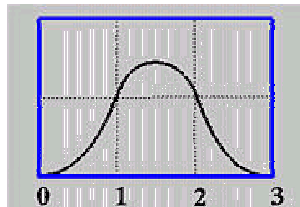
$$N_{0,2}(u) = (u - u_0) / (u_2 - u_0) N_{0,1}(u) + (u_3 - u) / (u_3 - u_1) N_{1,1}(u)$$

$$N_{0,2}(u) = 0.5u N_{0,1}(u) + 0.5(3 - u) N_{1,1}(u)$$

Si noti che  $N_{0,1}(u)$  è non nulla in  $[0,1)$  e  $[1,2)$  e  $N_{1,1}(u)$  è non nulla in  $[1,2)$  e  $[2,3)$ . Quindi si hanno tre casi da considerare:

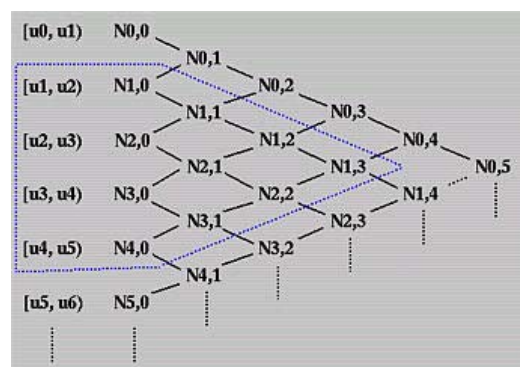
1.  $u$  in  $[0,1)$ :  
In questo caso, soltanto  $N_{0,1}(u)$  contribuisce al calcolo di  $N_{0,2}(u)$ . Poiché  $N_{0,1}(u) = u$ , si ha
 
$$N_{0,2}(u) = 0,5u^2$$
2.  $u$  in  $[1,2)$ :  
In questo caso, sia  $N_{0,1}(u)$  che  $N_{1,1}(u)$  contribuiscono al calcolo di  $N_{0,2}(u)$ . Poiché in  $[1,2)$ ,  $N_{0,1}(u) = 2 - u$  e  $N_{1,1}(u) = u - 1$ , si ha
 
$$N_{0,2}(u) = (0,5u)(2 - u) + 0.5(3 - u)(u - 1) = 0.5(-3 + 6u - 2u^2)$$
3.  $u$  in  $[2,3)$ :  
In questo caso, soltanto  $N_{1,1}(u)$  contribuisce al calcolo di  $N_{0,2}(u)$ . Poiché  $N_{1,1}(u) = 3 - u$  in  $[2,3)$ , si ha

$$N_{0,2}(u) = 0.5(3 - u)(3 - u) = 0.5(3 - u)^2$$



**OSSERVAZIONE:**

- $N_{i,1}(u)$  viene computato partendo da  $N_{i,0}(u)$  e da  $N_{i+1,0}(u)$  ed è in  $[u_i, u_{i+2})$  Similmente  $N_{i,2}(u)$  è non negativa in  $[u_i, u_{i+3})$ .



- La funzione di base  $N_{i,p}(u)$  è non-zero in  $[u_i, u_{i+p+1})$

- $N_{i,p}(u)$  è non-zero nei knot span  $[u_i, u_{i+1}), [u_{i+1}, u_{i+2}), \dots, [u_{i+p}, u_{i+p+1})$ .

Al contrario, dato un intervallo semi-aperto  $[u_i, u_{i+1})$ , per sapere quali funzioni di base sono non nulle in quell'intervallo occorre applicare in senso inverso la regola triangolare.

Nell'intervallo  $[u_i, u_{i+1})$ , al più  $p+1$  funzioni di grado  $p$  sono non nulle:

$$N_{i-p,p}(u), N_{i-p+1,p}(u), N_{i-p+2,p}(u), \dots, N_{i-1,p}(u) \text{ e } N_{i,p}(u).$$

## Funzioni Base B-Spline: Proprietà

$$N_{i,0}(u) = \begin{cases} 1 & \text{if } u_i \leq u < u_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u)$$

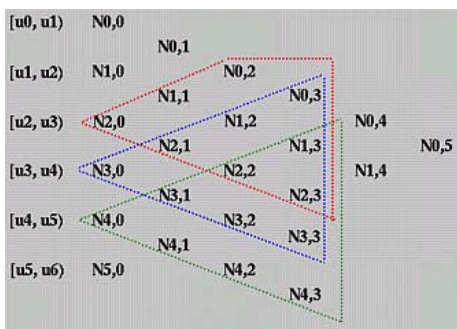
- $N_{i,p}(u)$  è un polinomio di grado  $p$  in  $u$
- **Non-negativity** - per ogni  $i, p$  e  $u$ ,  $N_{i,p}(u)$  è non negativa
- **Local Support** -  $N_{i,p}(u)$  è un polinomio non nullo in  $[u_i, u_{i+p+1})$
- In  $[u_i, u_{i+1})$ , al più  $p+1$  funzioni di grado  $p$  sono non nulle,

$$N_{i-p,p}(u), N_{i-p+1,p}(u), N_{i-p+2,p}(u), \dots, \text{ e } N_{i,p}(u)$$

- **Partition of Unity** - la somma di tutte le funzioni di grado  $p$  non-zero in  $[u_i, u_{i+1})$  è 1: La somma di  $N_{i-p,p}(u), N_{i-p+1,p}(u), N_{i-p+2,p}(u), \dots, \text{ e } N_{i,p}(u)$  è 1.
- Dato il numero di knot  $m+1$ , il grado delle funzioni di base  $p$  il numero di funzioni di grado  $p$  è  $n+1$ , allora  $m = n+p+1$ .

Sia  $N_{n,p}(u)$  l'ultima funzione di grado  $p$ . Tale funzione è non nulla in  $[u_n, u_{n+p+1})$ . Poiché è l'ultima funzione,  $u_{n+p+1}$  deve essere l'ultimo nodo  $u_m$ .

Di conseguenza, si ha  $u_{n+p+1} = u_m$  e  $n + p + 1 = m$ .



Quindi, dato  $m$  e  $p$ , sia  $n = m - p - 1$  e le funzioni di grado  $p$  sono  $N_{0,p}(u), N_{1,p}(u), N_{2,p}(u) \dots \text{ e } N_{n,p}(u)$ .

- La funzione  $N_{i,p}(u)$  è una **curva polinomiale di grado  $p$**  con joining points nei knots definita in  $[u_i, u_{i+p+1})$ .
- Ad un **nodo di molteplicità  $K$** , la funzione di base  $N_{i,p}(u)$  è  $C^{p-k}$  **continua**. Di conseguenza, aumentando la molteplicità di un nodo si diminuisce il grado di continuità  $C$ .

## L'importanza dei nodi di molteplicità $>1$

I nodi multipli hanno un effetto significativo sul calcolo delle funzioni.

- Ogni nodo di molteplicità  $K$  riduce non più di  $K-1$  domini di funzioni base non nulle.
- Per creare un nodo di molteplicità  $k$ , verranno influenzate  $k-1$  funzioni.
- Una di loro perde uno span di influenza (in cui è non-zero), la seconda ne perde due, la terza tre e così via.

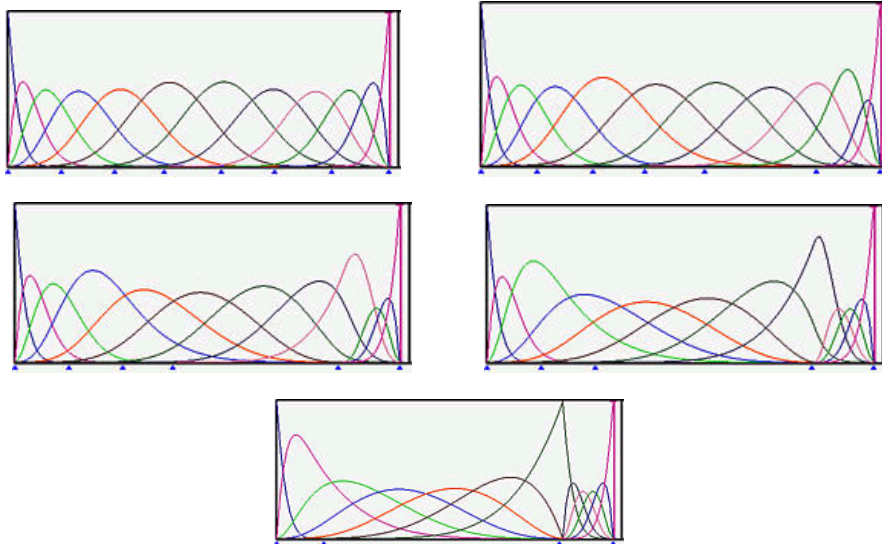


Figura 28 - Funzioni di base di grado 5 dove il primo e l'ultimo knot sono di molteplicità 6 e tutti gli altri sono knot semplici.

## Movimento dei nodi semplici...

- Ad ogni nodo interno di molteplicità  $k$ , corrispondono al più  $p-k+1$  funzioni base non nulle dove  $p$  è il grado delle funzioni.
- Aumentare di uno la molteplicità di un punto diminuirà di uno il numero di funzioni base di grado  $p$  che sono non nulle in quell'intervallo.
- Infine, il numero di funzioni non-zero di base ad un nodo di molteplicità  $K$  è al più

$$(p + 1) - K = p - K + 1.$$

## Esempio 1. Nodi Semplici

- $U = \{0, 0.25, 0.5, 0.75, 1\}$ .
- $m = 4$  e  $u_0 = 0, u_1 = 0.25, u_2 = 0.5, u_3 = 0.75$  ed  $u_4 = 1$ .

Le funzioni base di grado 0 sono semplici da calcolare. Sono 4 e sono le funzioni step definite nei relativi quattro intervalli semi-aperti.

- $N_{0,0}(u), N_{1,0}(u), N_{2,0}(u)$  e  $N_{3,0}(u)$  definite in  $[0,0.25), [0.25,0.5), [0.5, 0.75)$  e  $[0.75, 1)$ , rispettivamente.

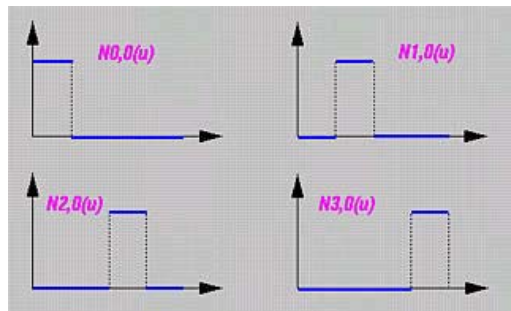


Figura 29 - Funzioni step di grado 0

Consideriamo il calcolo di  $N_{i,1}(u)$ :

| <i>Funzione Base</i> | <i>Intervallo</i> | <i>Equazione</i> |
|----------------------|-------------------|------------------|
| $N_{0,1}(u)$         | $[0, 0,25)$       | $4u$             |
|                      | $[0,25, 0,5)$     | $2(1 - 2u)$      |
| $N_{1,1}(u)$         | $[0,25, 0,5)$     | $4u - 1$         |
|                      | $[0,5, 0,75)$     | $3 - u$          |
| $N_{2,1}(u)$         | $[0,5, 0,75)$     | $2(2u - 1)$      |
|                      | $[0,75, 1)$       | $4(1 - u)$       |

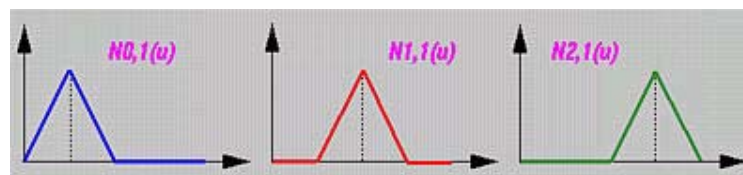


Figura 30 - Funzioni base di grado 1

Una volta ricavate le  $N_{i,1}(u)$ , è possibile computare le funzioni di grado 2.

Poiché  $m = 4$ ,  $p = 2$  e  $m = n + p + 1 \Rightarrow n = 1$  esistono soltanto due funzioni di base di grado 2  $N_{0,2}(u)$  e  $N_{1,2}(u)$ :

| <i>Funzione di Base</i> | <i>Intervallo</i> | <i>Equazione</i>     |
|-------------------------|-------------------|----------------------|
| $N_{0,2}(u)$            | $[0, 0,25)$       | $8u^2$               |
|                         | $[0,25, 0,5)$     | $-1,5 + 12u - 16u^2$ |
|                         | $[0,5, 0,75)$     | $4,5 - 12u + 8u^2$   |
| $N_{1,2}(u)$            | $[0,25, 0,5)$     | $0,5 - 4u + 8u^2$    |
|                         | $[0,5, 0,75)$     | $-1,5 + 8u - 8u^2$   |
|                         | $[0,75, 1)$       | $8(1 - u)^2$         |

Ogni funzione base è una curva composta da tre segmenti di curva di grado 2 ( $N_{0,2}(u)$  è l'unione di tre parabole definite in  $[0,0,25)$ ,  $[0,25,0,5)$  e  $[0,5,0,75)$ ). Nei punti di congiunzione la continuità è di tipo  $C^1$ .

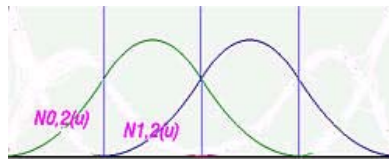


Figura 31 - Funzioni base di grado 2

### Esempio 2 - Nodi Multipli

Nel caso in cui  $U$  (knot vector) contiene nodi multipli, si dovrà ovviare il problema di incontrare una frazione di tipo indeterminato  $0/0$ . In questo caso definiremo  $0/0=0$ .

- Se  $u_i$  è un nodo di molteplicità  $k$  ( $u_i = u_{i+1} = \dots = u_{i+k-1}$ ), allora gli intervalli  $[u_i, u_{i+1}), [u_{i+1}, u_{i+2}), \dots, [u_{i+k-2}, u_{i+k-1})$  non esistono
- $N_{i,0}(u), N_{i+1,0}(u), \dots, N_{i+k-1,0}(u)$  sono tutte funzioni nulle.

Come esempio consideriamo un knot vector  $U = \{ 0, 0, 0, 0.3, 0.5, 0.5, 0.6, 1, 1, 1 \}$ . In questo modo  $0$  e  $1$  sono di molteplicità  $3$  ( $0(3)$  e  $1(3)$ ) e  $0.5$  è di molteplicità  $2$  ( $0.5(2)$ ).

$m = 9$  e si ha che:

|       |       |       |       |       |       |       |       |       |       |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| $u_0$ | $u_1$ | $U_2$ | $u_3$ | $u_4$ | $u_5$ | $u_6$ | $u_7$ | $u_8$ | $u_9$ |
| 0     | 0     | 0     | 0,3   | 0,5   | 0,5   | 0,6   | 1     | 1     | 1     |

Computiamo  $N_{i,0}(u)$ .  $m=9$  e  $p=0$  (funzioni base di grado 0), quindi  $n = m - p - 1 = 8$ . Vi sono soltanto quattro funzioni non-zero:  $N_{2,0}(u), N_{3,0}(u), N_{5,0}(u)$  e  $N_{6,0}(u)$ .

| Funzione Base | Intervallo   | Equazione | Osservazioni                                 |
|---------------|--------------|-----------|--|
| $N_{0,0}(u)$  | tutta la $u$ | 0         | dato che $[u_0, u_1) = [0,0)$ non esiste     |
| $N_{1,0}(u)$  | tutta la $u$ | 0         | dato che $[u_1, u_2) = [0,0)$ non esiste     |
| $N_{2,0}(u)$  | $[0, 0,3)$   | 1         |  |
| $N_{3,0}(u)$  | $[0,3, 0,5)$ | 1         |  |
| $N_{4,0}(u)$  | tutta la $u$ | 0         | dato che $[u_4, u_5) = [0.5,0.5)$ non esiste |
| $N_{5,0}(u)$  | $[0,5, 0,6)$ | 1         |  |
| $N_{6,0}(u)$  | $[0,6, 1)$   | 1         |  |
| $N_{7,0}(u)$  | tutta la $u$ | 0         | dato che $[u_7, u_8) = [1,1)$ non esiste     |
| $N_{8,0}(u)$  | tutta la $u$ | 0         | dato che $[u_8, u_9) = [1,1)$ non esiste     |

Consideriamo anche il calcolo delle funzioni base di grado 1. Poiché  $p$  è 1,  $n = m - p - 1 = 7$ :

| Funzione Base | Intervallo   | Equazione         |
|---------------|--------------|-------------------|
| $N_{0,1}(u)$  | tutta la $u$ | 0                 |
| $N_{1,1}(u)$  | $[0, 0,3)$   | $1 - (10/3) di u$ |
| $N_{2,1}(u)$  | $[0, 0,3)$   | $(10/3) di u$     |
| $N_{3,1}(u)$  | $[0,3, 0,5)$ | $2.5(1 - 2 u)$    |
| $N_{4,1}(u)$  | $[0,3, 0,5)$ | $5 u - 1,5$       |
| $N_{4,1}(u)$  | $[0,5, 0,6)$ | $6 - 10 u$        |
| $N_{5,1}(u)$  | $[0,5, 0,6)$ | $10 u - 5$        |
| $N_{6,1}(u)$  | $[0,6, 1)$   | $2.5(1 - u)$      |
| $N_{6,1}(u)$  | $[0,6, 1)$   | $2,5 u - 1,5$     |
| $N_{7,1}(u)$  | tutta la $u$ | 0                 |

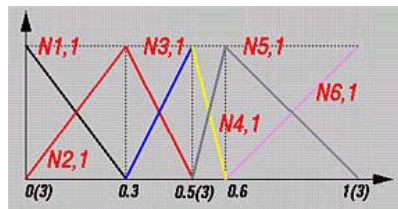


Figura 32 - Funzioni base di grado 1

Consideriamo la computazione delle equazioni base di grado 2:

$$n = m - p - 1 = 6$$

| Funzione Base | Intervallo   | Equazione                    |
|---------------|--------------|------------------------------|
| $N_{0,2}(u)$  | $[0, 0,3]$   | $(1 - (10/3) di u)^2$        |
| $N_{1,2}(u)$  | $[0, 0,3]$   | $(20/3)(u - (8/3) di u)^2$   |
| $N_{2,2}(u)$  | $[0,3, 0,5]$ | $2.5(1 - 2 u)^2$             |
| $N_{3,2}(u)$  | $[0, 0,3]$   | $(20/3) di u^2$              |
| $N_{4,2}(u)$  | $[0,3, 0,5]$ | $-3,75 + 25 u - 35 u^2$      |
| $N_{5,2}(u)$  | $[0,3, 0,5]$ | $(5 u - 1,5)^2$              |
| $N_{6,2}(u)$  | $[0,5, 0,6]$ | $(6 - 10 u)^2$               |
| $N_{7,2}(u)$  | $[0,5, 0,6]$ | $20(-2 + 7 u - 6 u^2)$       |
| $N_{8,2}(u)$  | $[0,6, 1]$   | $5(1 - u)^2$                 |
| $N_{9,2}(u)$  | $[0,5, 0,6]$ | $12.5(2 u - 1)^2$            |
| $N_{10,2}(u)$ | $[0,6, 1]$   | $2.5(-4 + 11,5 u - 7,5 u^2)$ |
| $N_{11,2}(u)$ | $[0,6, 1]$   | $2.5(9 - 30 u + 25 u^2)$     |

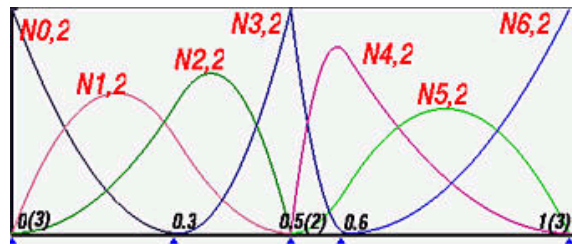


Figura 33 - funzioni base di grado 2

## B-Spline: Definizione

- Dati  $n+1$  punti di controllo  $\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_n$
- Dato un **knot vector**  $U = \{u_0, u_1, \dots, u_m\}$ ,
- la curva **B-Spline** di **grado p** definita da questi punti di controllo e dal vettore  $U$  è:

$$\mathbf{P}(u) = \sum_{i=0}^n N_{i,p}(u) \mathbf{P}_i$$

dove  $N_{i,p}(u)$  è funzioni base B-Spline di grado  $p$ .

- Equazione simile a quella di Bézier.
- Una curva di B-Spline necessita di più informazioni:
- un insieme dei punti di controllo, un vector knot e un grado.



## Osservazioni

- $n, m$  e  $p$  devono soddisfare l'equazione  $m=n+p+1$ .
- Per definire una curva B-Spline di grado  $p$  con  $n+1$  punti di controllo, occorre fornire  $n+p+2$  nodi  $u_0, u_1, \dots, u_{n+p+1}$ .
- $N_{i,p}(u)$  assomiglia a  $B_{n,i}(u)$ , ma il grado delle funzioni base B-Spline è un input, mentre il grado delle funzioni base Bézier dipende solo dal numero di punti di controllo.
- Per cambiare la shape di una curva B-Spline, si possono modificare o le posizione dei punti di controllo, o le posizioni dei nodi o il grado della curva.

## Definizioni

- Una B-spline si dice **OPEN** nel caso in cui il vector knot non abbia alcuna struttura particolare, in questo caso, inoltre la curva non tocca la prima e l'ultima legs
- Nel caso in cui il primo e l'ultimo nodo hanno molteplicità  $p+1$ , la curva passerà per i punti relativi e sarà di tipo **CLAMPED**.
- Nel caso in cui il primo e l'ultimo nodo coincidono la curva si dice di tipo **CLOSED**.

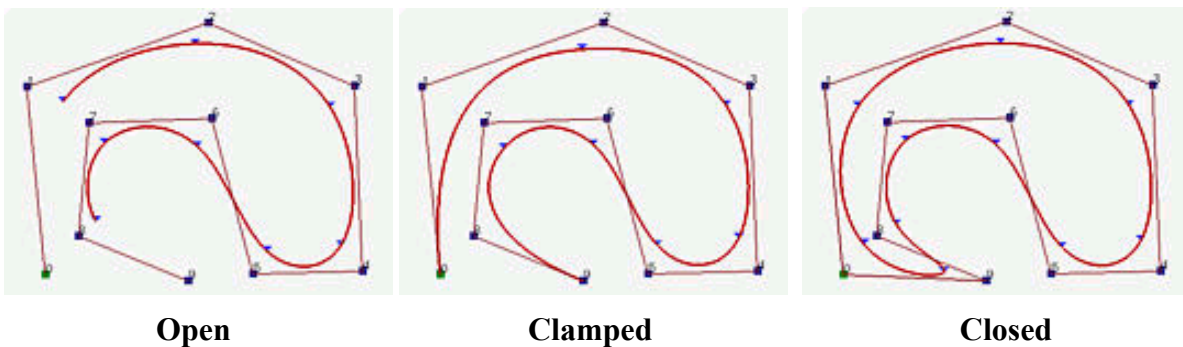


Figura 34 -  $n=9$  e  $p=3$

## B-spline Curves: Proprietà

Tutte le proprietà che sono valide per le curve di Bézier lo sono per le B-Spline, essendo queste una generalizzazione delle Bézier.

Consideriamo il caso di clamped B-spline.

Si consideri il caso di una B-spline di grado  $p$   $\mathbf{p}(u)$  definita da  $n+1$  punti di controllo, un knot vector  $U = \{u_0, u_1, \dots, u_m\}$  con i primi  $p+1$  e gli ultimi  $p+1$  knots "clamped" ( $u_0=u_1=\dots=u_p$  e  $u_{m-p}=u_{m-p+1}=\dots=u_m$ ).

- **B-spline  $\mathbf{p}(u)$  è una curva a pezzi in cui ogni pezzo è una curva di grado  $p$**   
Questa proprietà permette di creare shape abbastanza complesse usando equazioni polinomiali di grado basso.

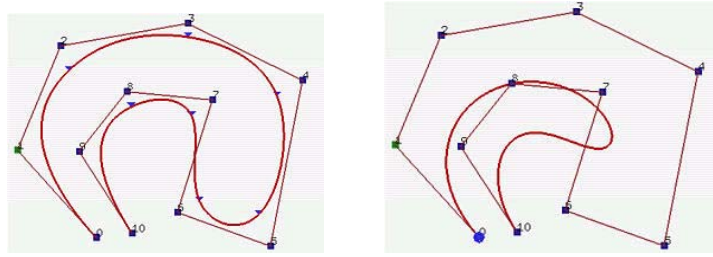


Figura 35 -  $n = 10, m = 14, p = 3$

- minore è il grado più la curva si avvicina al control polygon.
- Quindi quando il grado diminuisce la B-spline si avvicina alla control polyline.

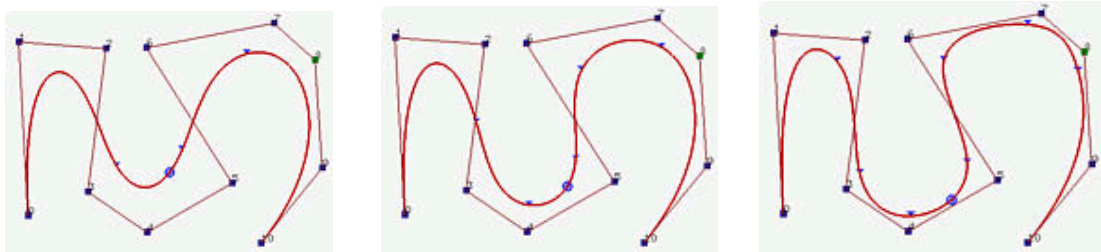
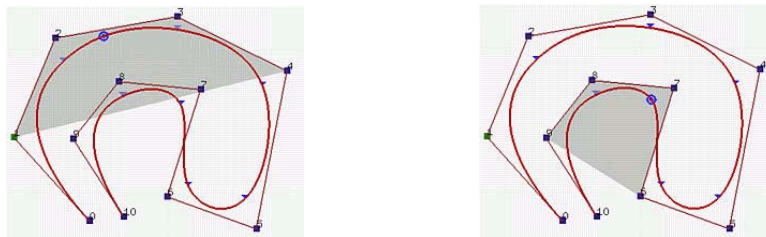
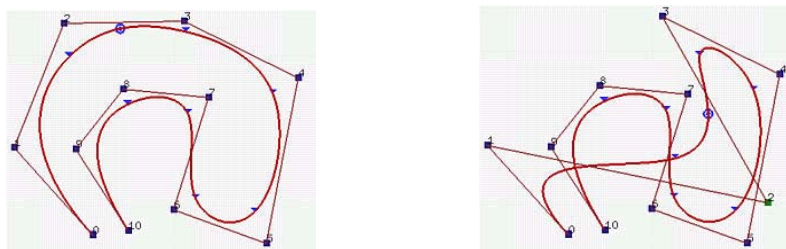


Figura 36 - stessa control polyline stessi knots clamped e spazati uniformemente. grado 7, grado 5 e grado 3.

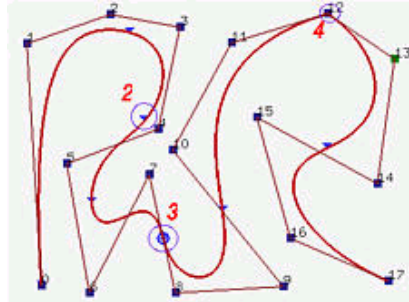
- Deve valere l'equazione  $m=n+p+1$
- **Clamped B-spline  $p(u)$  passa per  $p_0$  e  $p_n$**   
 $N_{0,p}(u)$  è la funzione di coefficiente in  $p_0$  ed è non-zero in  $[u_0, u_{p+1})$ . Poiché  $u_0=u_1=...=u_p=0$  per una clamped B-spline,  $N_{0,0}(u), N_{1,0}(u), \dots, N_{p-1,0}(u)$  sono zero e solamente  $N_{p,0}(u)$  è non-zero. Quindi, quando  $u=0, N_{0,p}(0)$  è 1 e  $p(0)=p_0$ . Analogamente per  $p(1)=p_n$
- **Strong Convex Hull Property:**



- **Local Modification Scheme:** lo spostamento di un punto di controllo cambia la curva solamente nell'intervallo  $[u_i, u_{i+p+1})$

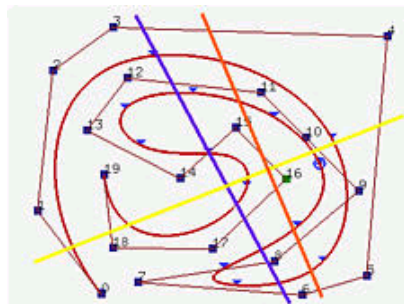


- $\mathbf{p}(u)$  è  $C^{p-k}$  continua in un nodo di molteplicità  $k$   
Se  $u$  non è un knot,  $\mathbf{p}(u)$  si trova in mezzo ad un segmento di curva e per questo è infinitamente differenziabile. Se  $u$  è un knot, nel dominio non-zero di  $N_{i,p}(u)$ , poiché l'ultima è  $C^{p-k}$  continua, lo stesso è  $\mathbf{p}(u)$ .

Figura 37 18 punti di controllo,  $n=17$ ,  $p=4$ .

| $u_0$ to $u_4$ | $u_5$ | $u_6$ and $u_7$ | $u_8$ | $u_9$ to $u_{11}$ | $u_{12}$ | $u_{13}$ to $u_{16}$ | $u_{17}$ | $u_{18}$ to $u_{22}$ |
|----------------|-------|-----------------|-------|-------------------|----------|----------------------|----------|----------------------|
| 0              | 0.125 | 0.25            | 0.375 | 0.5               | 0.625    | 0.75                 | 0.875    | 1                    |

- $\mathbf{p}(u)$  è  $C^4$  continua in ogni punto che non sia un knot
- $C^3$  continua nei nodi semplici,
- $C^2$  continua in  $u_6$ ,
- $C^1$  continua in  $u_9$ ,
- $C^0$  continua in  $u_{13}$ .
- **Variation Diminishing Property:**



- **Le curve di Bézier sono casi speciali di B-spline**  
Se  $n=p$  (grado della b-spline  $n$ , con  $2(p+1) = 2(n+1)$  knots, con  $p+1$  clamped), la B-spline è una curva di Bézier.
- **Affine Invariance**

### Vantaggi

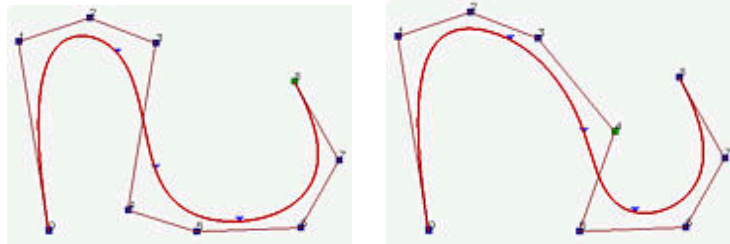
- Una curva di B-Spline può essere una curva di Bézier.
- Le curve B-Spline forniscono maggiore flessibilità di controllo.
- Il grado di una curva B-Spline è separato dal numero di punti di controllo.
- Si possono usare curve di grado più basso ed avere ancora tantissimi punti di controllo.
- È possibile cambiare la posizione di un punto di controllo senza cambiare la shape della curva intera.
- Le curve B-Spline soddisfano la proprietà del guscio convesso, ed hanno un controllo maggiore sulla shape.

## Svantaggi

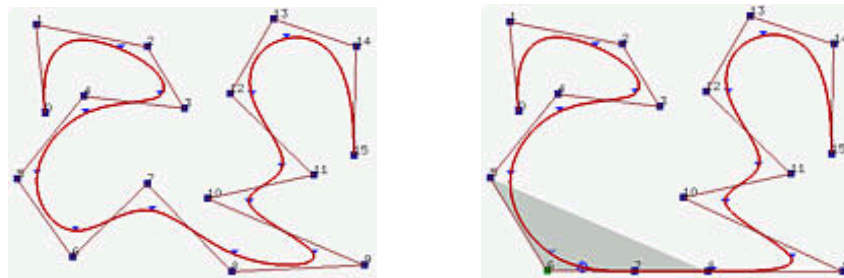
- Le curve B-Spline sono ancora curve polinomiali e le curve polinomiali non possono rappresentare molte curve utili quali i circonferenze e ellissi.
- Una generalizzazione delle B-Spline, **NURBS**, è richiesta.

## Muovere i control points

Lo spostamento dei punti di controllo influenza solo una parte della B-Spline grazie alla proprietà di schema di modifica locale.

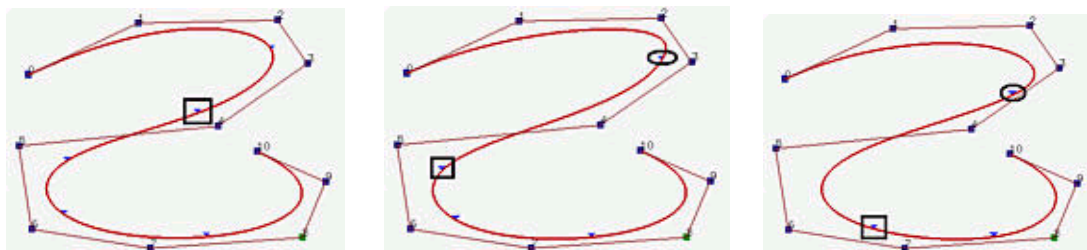


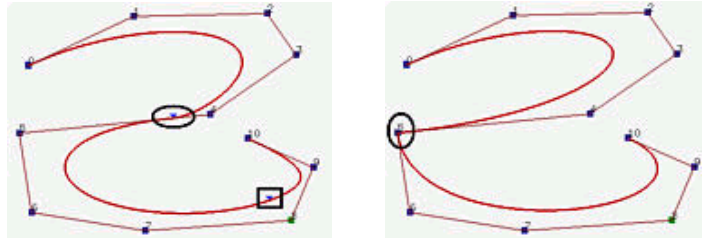
- la proprietà **strong convex hull property** aiuta nel raggiungimento di situazioni particolari per le porzioni di curva da muovere
- un esempio è il caso in cui un segmento di curva deve diventare un segmento.



## Cambiare il Vector Knot

- il cambiamento del vettore dei nodi comporta il cambiamento della shape della curva relativa.
- In particolare aumentare (diminuire) la molteplicità di un singolo knot può cambiare notevolmente la shape.
- Ad esempio se un knot è di molteplicità  $p$  con  $p$  il grado della B-Spline allora esisterà una sola funzione non zero relativa al knot e quindi un unico control point. In questo senso si può forzare la curva a passare per un punto.





## Derivate di una B-Spline

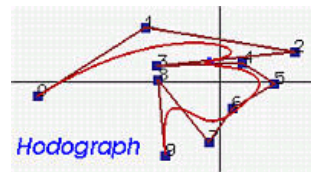
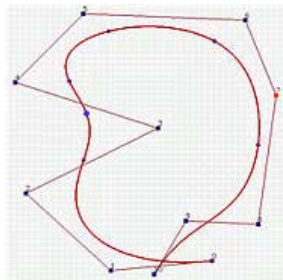
Il calcolo delle derivate di una B-Spline è semplice come nel caso di curva di Bézier

$$\mathbf{p}(u) = \sum_{i=0}^n N_{i,p}(u) \mathbf{p}_i$$

$$\frac{d}{du} N_{i,p}(u) = N'_{i,p}(u) = \frac{p}{u_{i+p} - u_i} N_{i,p-1}(u) - \frac{p}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u)$$

$$\frac{d}{du} \mathbf{p}(u) = \mathbf{p}'(u) = \sum_{i=0}^{n-1} N_{i+1,p-1}(u) \mathbf{q}_i$$

$$\mathbf{q}_i = \frac{p}{u_{i+p+1} - u_{i+1}} (\mathbf{p}_{i+1} - \mathbf{p}_i)$$



### OSSERVAZIONE

Nel caso di Curve B-Spline clumped: **Una B-Spline clumped passa per il primo ed ultimo punto di controllo ed è tangente la primo ed ultimo leg della polyline di controllo.**

### *Derivate superiori alle prima*

Il calcolo delle derivate superiori alla prima è analogo

# NON UNIFORM RATIONAL B-SPLINE

## Motivazione

- le curve di B-spline sono curve polinomiali.
- Sono flessibili e piacevoli da manipolare ma non possono rappresentare la curva più semplice: il cerchio.
- I cerchi possono essere rappresentati con le funzioni razionali (funzioni che sono i quozienti di due polinomiali).
- Per fare fronte ai cerchi, ellissi e molte altre curve che non possono essere rappresentate dalla curve polinomiali, si ha bisogno di una estensione delle B-Spline.

In questo senso nascono le **NURBS**.

## NURBS: Definizione

Dati  $n+1$  punti di controllo  $\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_n$  e  $U = \{u_0, u_1, \dots, u_m\}$  con  $m+1$  knots, la B-spline di grado  $p$  è:

$$\mathbf{p}(u) = \sum_{i=0}^n N_{i,p}(u) \mathbf{p}_i$$

Sia  $\mathbf{p}_i$

$$\mathbf{p}_i = \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix}$$

Nel caso di coordinate omogenee moltiplicare un punto per un valore non zero non cambia la posizione del punto.

$$\mathbf{p}_i = \begin{bmatrix} w_i x_i \\ w_i y_i \\ w_i z_i \\ w_i \end{bmatrix} \quad \mathbf{p}(u) = \sum_{i=0}^n N_{i,p}(u) \mathbf{p}_i = \sum_{i=0}^n N_{i,p}(u) \begin{bmatrix} w_i x_i \\ w_i y_i \\ w_i z_i \\ w_i \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^n N_{i,p}(u) (w_i x_i) \\ \sum_{i=0}^n N_{i,p}(u) (w_i y_i) \\ \sum_{i=0}^n N_{i,p}(u) (w_i z_i) \\ \sum_{i=0}^n N_{i,p}(u) (w_i) \end{bmatrix}$$

$$\mathbf{p}(u) = \begin{bmatrix} (\sum_{i=0}^n N_{i,p}(u) (w_i x_i)) / (\sum_{i=0}^n N_{i,p}(u) (w_i)) \\ (\sum_{i=0}^n N_{i,p}(u) (w_i y_i)) / (\sum_{i=0}^n N_{i,p}(u) (w_i)) \\ (\sum_{i=0}^n N_{i,p}(u) (w_i z_i)) / (\sum_{i=0}^n N_{i,p}(u) (w_i)) \\ 1 \end{bmatrix} = \sum_{i=0}^n \frac{N_{i,p}(u) w_i}{\sum_{i=0}^n N_{i,p}(u) w_i} \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix}$$

$$\mathbf{p}(u) = \frac{1}{\sum_{i=0}^n N_{i,p}(u) w_i} \sum_{i=0}^n N_{i,p}(u) w_i \mathbf{p}_i$$

Questa è la curva NURBS di grado  $p$  definita dai punti di controllo  $\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_n$ , e da  $U = \{u_0, u_1, \dots, u_m\}$  con pesi  $w_0, w_1, \dots, w_n$ .

- I pesi  $w_i$  sono in genere positivi; anche se i pesi negativi hanno una influenza interessante sulla curva.

## Due Risultati Immediati

- **Se tutti i pesi sono uguali a 1, una NURBS riduce ad una B-spline.**
- **Le NURBS sono razionali.** Il valore  $N_{i,p}(u) w_i$  è un polinomio di grado  $p$ . Il denominatore è la somma di tutti i coefficienti e quindi è un polinomio di grado  $p$ . Il coefficiente del punto di controllo  $\mathbf{p}_i$  è il quoziente di due polinomiali di grado  $p$  e la funzione  $\mathbf{p}(u)$  è razionale.
- Le B-spline sono casi particolari di NURBS.

- Le NURBS sono razionali, e quindi le circonferenze, le ellissi e molte altre curve sono descrivibili tramite NURBS

### Un'Interpretazione Geometrica

Una NURBS è la proiezione nel piano  $W=1$  di una B-Spline nello spazio 4-dimensionale.

$$P(u) = \sum_{i=0}^n N_{i,p}(u) P_i = \sum_{i=0}^n N_{i,p}(u) \begin{bmatrix} w_i x_i \\ w_i y_i \\ w_i z_i \\ w_i \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^n N_{i,p}(u) (w_i x_i) \\ \sum_{i=0}^n N_{i,p}(u) (w_i y_i) \\ \sum_{i=0}^n N_{i,p}(u) (w_i z_i) \\ \sum_{i=0}^n N_{i,p}(u) (w_i) \end{bmatrix}$$

### NURBS: Proprietà Importanti

Dati  $n+1$   $p_1, p_2, \dots, p_n$  punti di controllo, ad ognuno dei quali è associato un peso  $W_i$  e dato un vettore  $U = \{u_0, u_1, \dots, u_m\}$  con  $m+1$  knots, la NURBS di grado  $p$  è:

$$P(u) = \sum_{i=0}^n R_{i,p}(u) P_i$$

Con  $R_{i,p}(u)$  è definita come segue:

$$R_{i,p}(u) = \frac{N_{i,p}(u) w_i}{\sum_{j=0}^n N_{j,p}(u) w_j}$$

- $R_{i,p}(u)$  è una funzione razionale di grado  $p$  in  $u$
- Nonnegativity - per ogni  $i, p$ ,  $R_{i,p}(u)$  è non negativa
- Supporto locale -  $R_{i,p}(u)$  è un non-zero in  $[u_i, u_{i+p+1})$   
Poiché la  $N_{i,p}(u)$  è non-zero sopra  $[u_i, u_{i+p+1})$ , anche  $R_{i,p}(u)$ .
- In  $[u_i, u_{i+1})$ , esistono al più  $p+1$  funzioni base di grado  $p$  che siano non-zero,  $R_{i-p,p}(u)$ ,  $R_{i+1-p,p}(u)$ ,  $R_{i+2-p,p}(u)$ ... e  $R_{i,p}(u)$
- La somma di tutte le funzioni non-zero di grado  $p$  in  $[u_i, u_{i+1})$  è 1
- Se i knots sono  $m+1$ , ed il grado delle funzioni di base è  $p$  ed il numero di funzioni  $n+1$ , allora  $m = n+p+1$
- In un nodo di molteplicità  $k$ , la funzione  $R_{i,p}(u)$  è  $C^{p-k}$  continua.
- Una NURBS può essere open, clumped e closed
- La NURBS  $p(u)$  è composizione di curve razionali ognuno di grado  $p$
- $m = n + p + 1$  deve essere soddisfatta
- Una curva clumped NURBS passa per  $p_0$  e  $p_n$
- Strong convex hull property:
- Local modification scheme: cambiare la posizione di un punto di controllo  $p_i$  interessa soltanto la curva  $p(u)$  in  $[u_i, u_{i+p+1})$
- $p(u)$  è  $C^{p-k}$  continua in un nodo di molteplicità  $K$
- Variation Diminishing Property:
- le curve di B-spline e Bézier sono casi speciali delle NURBS  
Se tutti i pesi sono uguali, una curva di NURBS si trasforma in una B-Spline.  
Se  $n=p$  la NURBS si riduce ad una curva di Bézier.

- **Projective Invariance**

Se una trasformazione proiettiva viene applicata ad una NURBS, il risultato può essere costruito dalle immagini proiettate dei relativi punti di controllo.

## NURBS: Modifica dei pesi

**Le NURBS contengono le B-spline come casi speciali e quindi i metodi di modifica di una shape sono analoghi**

Esiste anche un nuovo parametro di modifica nel caso di NURBS, ovvero la modifica dei pesi

$$R_{i,p}(u) = \frac{N_{i,p}(u)w_i}{\sum_{j=0}^n N_{j,p}(u)w_j}$$

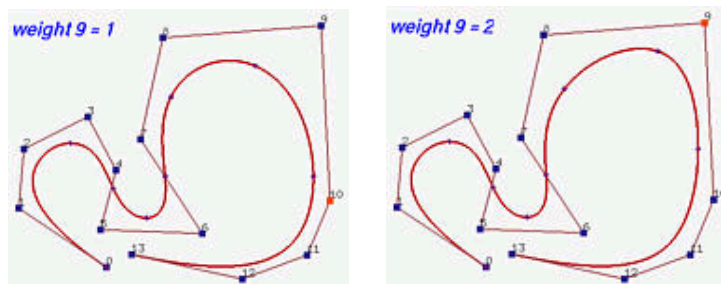


Figura 38 grado della curva =6

**Aumentare (diminuire) il valore dei pesi la curva si avvicina (allontana) dai punti di controllo.**

## Inserimento di un nodo

Quello che è interessante è trovare un algoritmo per l'inserimento di un nodo che non modifichi la shape della NURBS in questione

Il nodo può essere uguale ad uno esistente o essere completamente nuovo.

In ogni caso deve continuare a valere la relazione  $m=n+p+1$ , per questo o il numero dei punti di controllo o il grado della curva deve aumentare.

L'aumento del grado della curva modifica la shape e non viene considerato.

Quindi l'inserimento di un nuovo nodo comporta l'aggiunta di un nuovo punto di controllo



## Inserire un Knot in una B-Spline

Dati  $n+1$  punti di controllo  $\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_n$ , ed un vettore

$U = \{u_0, u_1, \dots, u_m\}$  ed un grado  $p$  si vuole inserire un nuovo knot  $\underline{u}$  nel vettore senza cambiare la shape della B-Spline.

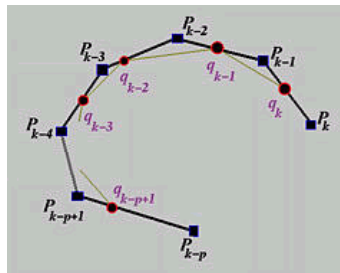
Sia  $\underline{u}$  in  $[u_k, u_{k+1})$ . Questo vuol dire che  $\mathbf{p}(\underline{u})$  si trova nel convex hull definito dai punti di controllo  $\mathbf{p}_k, \mathbf{p}_{k-1}, \dots, \mathbf{p}_{k-p}$  e tutte le altre funzioni di base sono nulle.



Occorre quindi trovare  $\mathbf{q}_k$  su  $\mathbf{p}_{k-1}\mathbf{p}_k$ ,  $\mathbf{q}_{k-1}$  su  $\mathbf{p}_{k-2}\mathbf{p}_{k-1}$ , ..., e  $\mathbf{q}_{k-p+1}$  su  $\mathbf{p}_{k-p}\mathbf{p}_{k-p+1}$  in modo che la vecchia polyline tra  $\mathbf{p}_{k-p}$  e  $\mathbf{p}_k$  è sostituita da  $\mathbf{p}_{k-p}\mathbf{q}_{k-p+1}\dots\mathbf{q}_k\mathbf{p}_k$  tagliando gli angoli in  $\mathbf{p}_{k-p+1}$ , ...,  $\mathbf{p}_{k-1}$ .

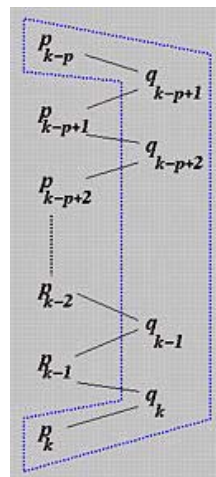
Tutti gli altri punti di controllo rimangono invariati.

$p-1$  control points della vecchia control polyline sono rimossi e sostituiti con  $p$  nuovi control points



Le posizioni dei nuovi punti di controllo sono abbastanza semplici da computare

$$\mathbf{q}_i = (1 - \alpha_i)\mathbf{p}_{i-1} + \alpha_i\mathbf{p}_i \quad \alpha_i = \frac{u - u_i}{u_{i+p} - u_i} \quad \text{for } k - p + 1 \leq i \leq k$$



**OSSERVAZIONE**

Per aggiungere un knot in una NURBS il ragionamento è analogo.

- (1) occorre convertire la NURBS in una B-Spline in 4 dimensioni
  - (2) inserire il knot nella b\_Spline
  - (3) Proiettare il nuovo insieme di punti in 3D
- $n+1$  control points  $\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_n$  con pesi  $w_0, w_1, \dots, w_n$ ,
  - $U$  vettore dei knot e un grado  $p$ .
  - Siano  $\mathbf{p}_i = (x_i, y_i, z_i)$  i punti di controllo.  $\mathbf{P}_i = (w_i x_i, w_i y_i, w_i z_i, w_i)$ , ed il vettore  $U$  definiscono una B-spline di grado  $p$  in 4 dimensioni.
  - Possiamo inserire  $u$  ottenendo  $\mathbf{Q}_i = (X_i, Y_i, Z_i, W_i)$
  - Proiettando questi punti possiamo ottenere la nuova NURBS

## Algoritmo di De-Boor

**Input:** un valore di  $u$

**Output:**  $p(u)$

If  $u \in [u_k, u_{k+1})$  and  $u \neq u_k$ ,

$h = p$ ;

$s = 0$ ;

If  $u = u_k$  and  $u_k$  is a knot of multiplicity  $s$ ,

$h = p - s$ ;

**Copy the affected control points  $p_{k-s}, p_{k-s-1}, p_{k-s-2}, \dots, p_{k-p+1}$  and  $p_{k-p}$  to a new array and rename them as  $p_{k-s,0}, p_{k-s-1,0}, p_{k-s-2,0}, \dots, p_{k-p+1,0}$ ;**

for  $r := 1$  to  $h$  do

    for  $i := k - p + r$  to  $k - s$  do

$a_{i,r} = (u - u_i) / (u_{i+p-r+1} - u_i)$

$p_{i,r} = (1 - a_{i,r}) p_{i-1,r-1} + a_{i,r} p_{i,r-1}$

$p_{k-s,p-s}$  is the point  $p(u)$ .

## Reference

- [1] Rockwood A., Chambers P. Interactive Curves and Surfaces: A Multimedia Tutorial on CAGD. The Morgan Kaufmann Series in Computer Graphics and Geometric Modeling, Brian A. Barsky, Series Editor, 1996
- [2] Piegl L. Tiller W., The NURBS Book, Second Edition, Springer Verlag, 1997.