H1 Superfici Parametriche

Questo tipo di modelli sono rappresentano superfici curve, e inoltre non sono lineari a tratti come le mesh, bensì *quadratiche, cubiche, quartiche* a tratti. Quindi non approssimiamo superfici con facce piatte come le mesh, lo facciamo con superfici già curve (***Patch di Bézier***).

Prima guardiamo però il caso in 2D, cioè le ***curve parametriche***, in quanto le **superfici parametriche** sono una loro generalizzazione.

H2 Curve Parametriche

H3 Introduzione

Sono curve descritte da una funzione:

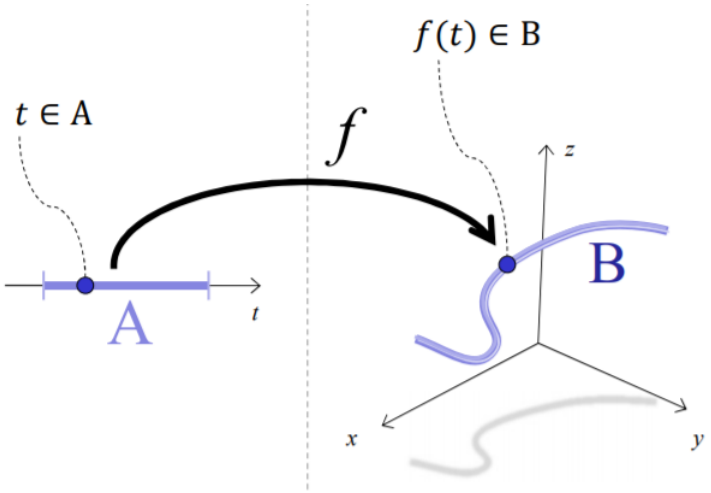
$$f : A \mapsto B$$

dove $A \subset \mathbb{R}$ e:

- $B \subset \mathbb{R}^2 \longrightarrow \quad f(t) = \begin{pmatrix} x \\ y \end{pmatrix}$ sono *curve in 2D*
- $B \subset \mathbb{R}^3 \longrightarrow f(t) = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$ sono *curve in 3D*

In entrambi i casi:

- t è detto *parametro*
- A, il dominio di f, è detto *dominio parametrico*
- B, l'immagine di f, è la curva risultante
- Se $f(t)$ è un polinomio di t, si tratta di una *curva polinomiale* (o *spline*)



H4 Calcolo del vettore tangente a un punto

Si può facilmente dimostrare che il calcolo del vettore tangente a un punto non è altro che la derivata prima di f in quel punto, cioè $f(t)$. Se volessimo che il vettore tangente sia anche unitario ci basterà *normalizzarlo*. In generale quindi avremo:

$$\text{vettore unitario della curva nel punto } f(t) = \frac{f'(t)}{||f'(t)||}$$

Facendo un esempio "giocattolo", prendendo una curva a forma di segmento, avremo la seguente situazione:

Per ottenere tale "curva", la funzione usata sarà una semplice interpolazione lineare:

$$f(t) = p_0 \cdot (1-t) + p_1 \cdot t = \\ p_0 + (p_1 - p_0) \cdot t$$

la cui derivata prima in t è:

$$f'(t) = p_1 - p_0$$

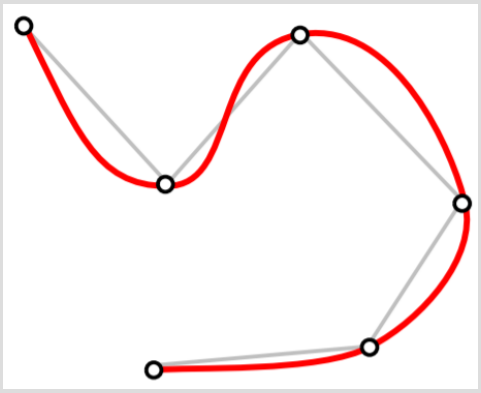
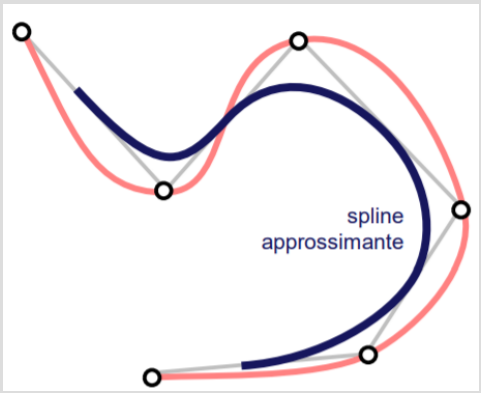
che è proprio il segmento espresso in forma di vettore, ma è ciò che ci aspetteremmo, in quanto la tangente a un segmento è il segmento stesso.

Questo esempio ci aiuta inoltre ad introdurre il concetto di **punti di controllo** di una curva. In questo caso in cui il polinomio è lineare (di grado 1), i punti di controllo sono solo 2 e sono proprio i punti p_0 e p_1. Come vedremo, aumentando il grado del polinomio della funzione f, otterremo più punti di controllo.

H3 Spline

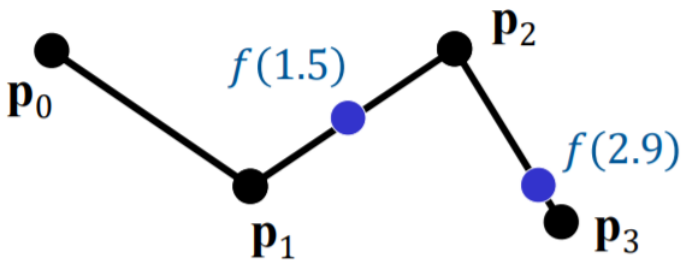
> **Curva polinomiale a tratti** di grado n, divisa in **archi** di grado n controllati da dei **punti di controllo**.

Le spline possono essere **interpolanti**, se passano attraverso i punti di controllo, oppure **approssimanti** se vengono solo "attratte" da essi:

	
spline interpolante	*spline approssimante*

Le spline sono divise in **archi**, i cui punti sono interpolazioni lineari dei relativi punti di controllo.

Facendo un esempio di spline (non specifica) di grado 1, avremo una *linea spezzata*:



La cui funzione sarà:

$$f(t) = \begin{cases} p_0 + (p_1 - p_0)(t-0) & \text{if } t \in [0,1] \\ p_1 + (p_2 - p_1)(t-1) & \text{if } t \in [1,2] \\ p_2 + (p_3 + p_2)(t-2) & \text{if } t \in [2,3] \end{cases}$$

Come è possibile notare, tale linea non è altro che una concatenazione di funzioni che sono *polinomi di grado 1* (i 3 casi di $f(t)$) e che sono l'interpolazione lineare dei rispettivi punti di controllo. Ognuno di quei segmenti è quindi un arco della spline.

Inoltre la linea spezzata in figura è in realtà una particolare spline, ovvero è una **curva di Bézier** di grado 1.

H3 Curve di Bézier

Ci sono molte famiglie di spline, ma fra le più diffuse vi sono le ***curve di Bézier***. Vediamo alcune nozioni a riguardo:

- Una curva di Bézier è composta da *archi di Bézier*
- Un arco di Bézier di grado n è controllato da $n+1$ punti di controllo
- L'arco è una interpolante per il primo e l'ultimo punto di controllo, per gli altri è, in genere, approssimante. Questo significa che è

possibile costruire curve più complesse e comunque continue unendo più archi facendo coincidere l'ultimo punto di controllo di un arco con il primo dell'arco successivo

Le curve di Bézier sono definibili in due modi:

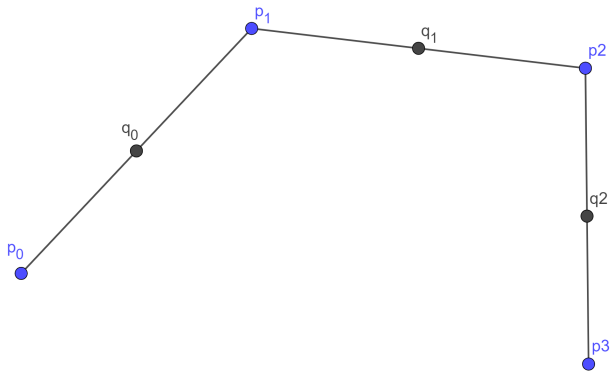
- attraverso l' *Algoritmo di De Casteljau*
- come polinomio (*formulazione di Bézier*)

H4 Algoritmo di De Casteljau

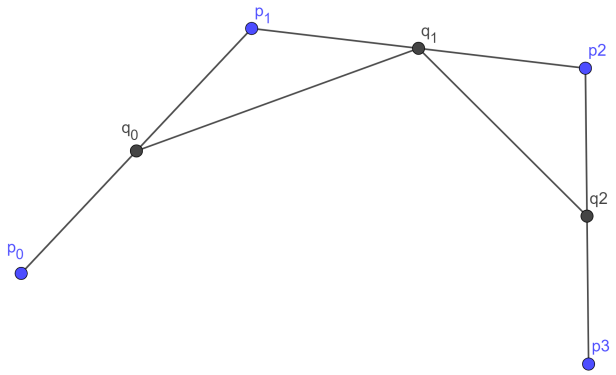
L'algoritmo prende in input i punti di controllo dell'arco e il parametro t , restituendo il punto $f(t)$ di quell'arco.

Dando subito un esempio è facile capire come opera l'algoritmo.
Esempio per un arco di Bézier di grado 3:

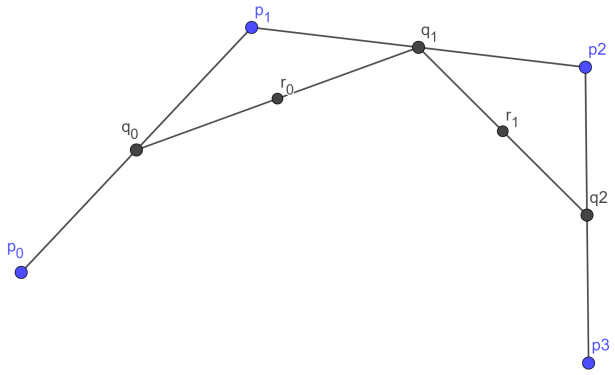
1. Interpolo i punti p_0 e p_1 con il parametro t ; ottengo il punto q_0 ; ripeto per gli altri due segmenti:



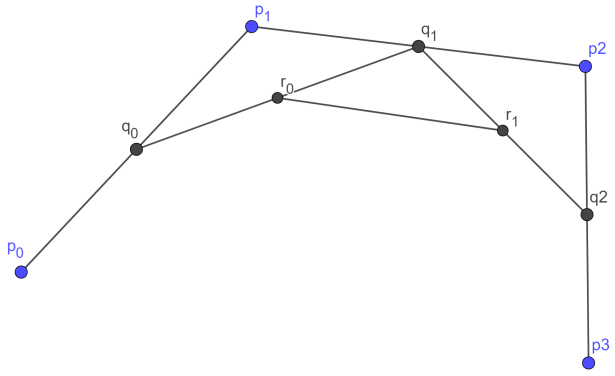
2. Unisco i punti ottenuti con dei segmenti:



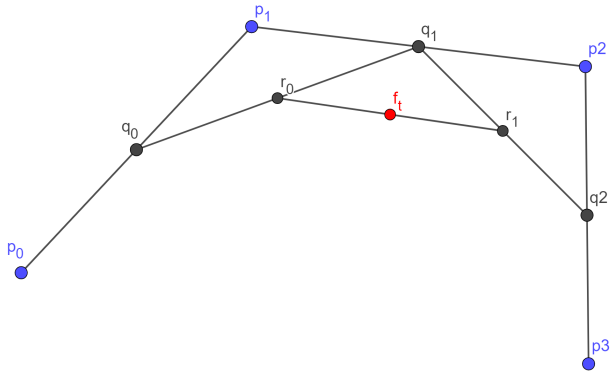
3. Interpolo i punti q_0 e q_1 sempre, con t , e faccio lo stesso con il segmento $q_1 q_2$, ottenendo due nuovi punti r_0 e r_1 :



4. Disegno il segmento $r_0 r_1$:



5. Infine, interpolo $r0$ e $r1$ con t, ottenendo $f(t)$:



In codice, l'algoritmo per questo esempio sarà ad esempio:

```
vec3 bezier(float t, vec3 p0, vec3 p1, vec3 p2, vec3 p3){
    vec3 q0 = mix(p0, p1, t);
    vec3 q1 = mix(p1, p2, t);
    vec3 q2 = mix(p2, p3, t);
    vec3 r0 = mix(q0, q1, t);
    vec3 r1 = mix(q1, q2, t);
    return mix(r0, r1, t);
}
```

H4 Formulazione di Bézier

Partendo da De Casteljau, possiamo ricondurci alle formulazioni di Bézier per ogni grado:

- *grado 1*:

$$f(t) = p_0 + t(p_1 - p_0)$$

- *grado 2*:

$$f(t) = (1-t)q_0 + t \cdot q_1 \tag{1}$$

con

$$q_0 = (1-t)p_0 + t \cdot p_1 \qquad \text{e} \qquad q_1 = (1-t)p_1 + t \cdot p_2$$

quindi sostituendo q_0 e q_1 in (1) avremo:

$$f(t) = p_0 + t(p_1 - 2p_0) + t^2(p_0 - p_1 + p_2)$$

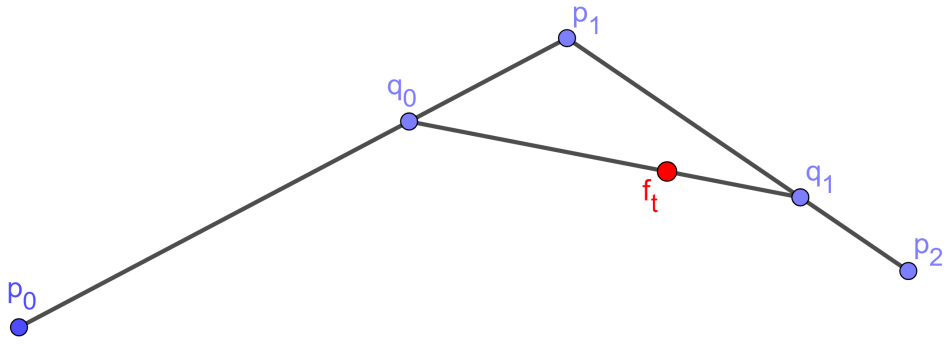
da cui, raccogliendo:

$$f(t) = (1-t)^2 p_0 + (1-t)\,t\,p_1 + t^2\,p_2$$

infine, ponendo $\bar{t} = (1-t)$:

$$f(t) = \bar{t}^2 p_0 + \bar{t}\,t\,p_1 + t^2 p_2 \tag{2}$$

dove p_0, p_1 e p_2 sono i punti di controllo della curva di Bézier di grado 2.



- *grado 3*:

  Per ottenere il grado 3, una sua immagine esempio è quella usata per l'algoritmo di De Casteljau, potremmo procedere come per il grado 2, ma in realtà (2) ci permette di generalizzare il calcolo della formulazione di Bézier per qualsiasi grado. Infatti si tratta di

fare:

$$(\bar{t} + t)^n$$

e poi aggiungere i punti di controllo. Nel nostro caso di grado 3, avremo:

$$(\bar{t} + t)^3 = \\ \bar{t}^3 + 3\bar{t}^2 t + 3\bar{t}t^2 + t^3$$

e aggiungendo $n + 1$ punti di controllo:

$$\bar{t}^3 p_0 + 3\bar{t}^2 t\, p_1 + 3\bar{t}\, t^2 p_2 + t^3 p_3$$

- *grado 4*:

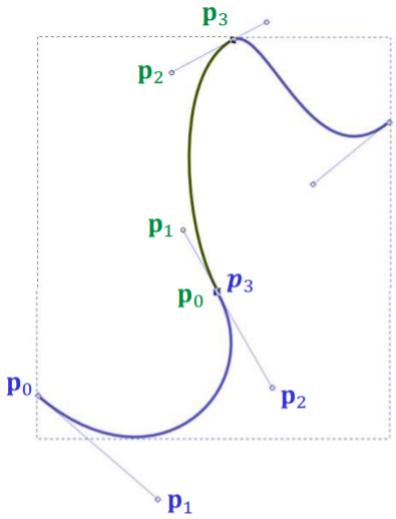
  Allo stesso modo che per il grado 3, avremo:

  $$f(t) = \bar{t}^4 p_0 + 4\bar{t}^3 t p_1 + 6\bar{t}^2 t^2 p_2 + 4\bar{t}\, t^3 p_3 + t^4 p_3$$

- e cosi via...

H4 Curve di Bézier - applicazioni

Le curve di Bézier di grado 3 sono spesso le più usate, in quanto sono le più facili da manipolare. Ciò poiché le tangenti nel primo e nell'ultimo punto coincidono con il primo e l'ultimo segmento che connettono i punti di controllo:



In quest'ultima figura si nota anche come è possibile concatenare più archi di Bézier facendo coincidere il primo e l'ultimo punto di controllo di due archi. In questo modo costruiamo una curva continua. Inoltre se volessimo anche continuità di tangenza (*smoothness*) possiamo far coincidere le tangenti di quei punti (in figura i segmenti p_2p_3 e p_0p_1 sono allineati).

H2 Superfici Parametriche

H3 Introduzione

Le **superfici parametriche** sono un'estensione naturale delle curve parametriche.

Anch'esse sono descritte da una funzione:

$$f : A \mapsto B$$

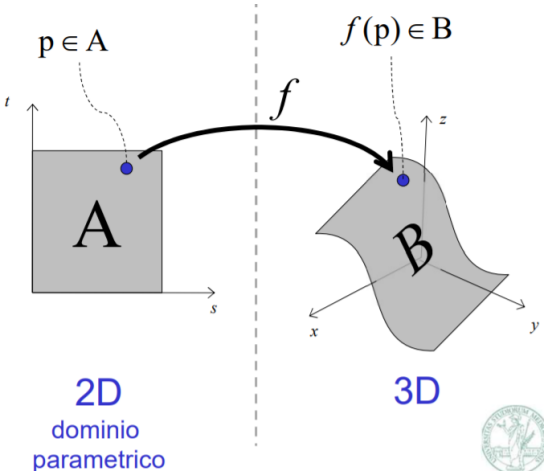
ma questa volta:

$$A \subset \mathbb{R}^2 \quad \text{e} \quad B \subset \mathbb{R}^3$$

quindi, in generale avremo:

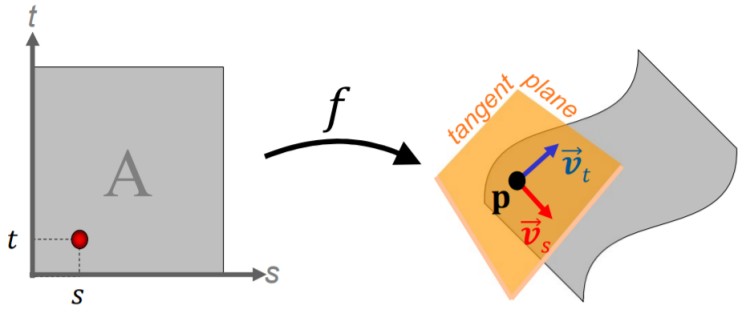
$$f\begin{pmatrix} s \\ t \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

ed avremo invece di una curva nello spazio, una *superficie* nello spazio:



H4 Calcolo della normale in un punto

È possibile ottenere la normale in un punto della superficie usando i vettori che giacciono sul *piano tangente* alla superficie in quel punto. Il prodotto cross normalizzato di tali vettori mi restituirà il vettore normale. In dettaglio:



$$\mathbf{p} = f\begin{pmatrix} s \\ t \end{pmatrix}$$

$$\vec{v_s} = \frac{\partial f}{\partial s}\begin{pmatrix} s \\ t \end{pmatrix} \qquad \vec{v_t} = \frac{\partial f}{\partial t}\begin{pmatrix} s \\ t \end{pmatrix}$$

La normale in $\mathbf{p}$ sarà:

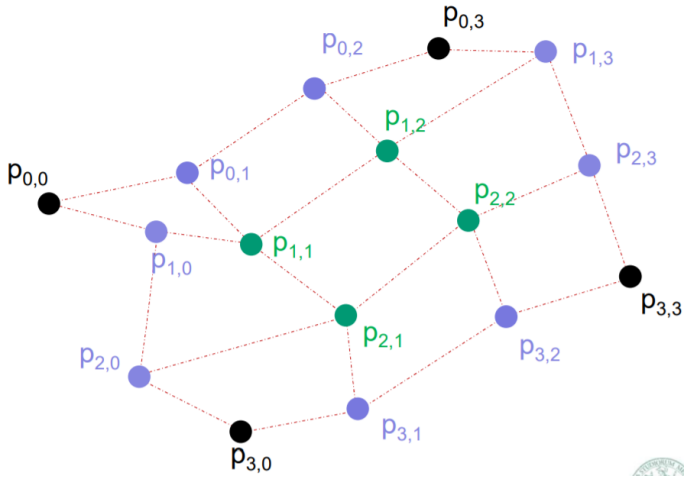
$$\hat{n} = \frac{\overrightarrow{v_s} \times \overrightarrow{v_t}}{||\overrightarrow{v_s} \times \overrightarrow{v_t}||}$$

H3 Patch di Bézier

A livello intuitivo è possibile immaginare le superfici parametriche come un "tessuto di curve parametriche". Esistono quindi le superfici parametriche che seguono lo schema di Bézier, dette ***patch di Bézier***. Ogni punto 3D dato da $f\begin{pmatrix} s \\ t \end{pmatrix}$ è una interpolazione lineare dei punti di controllo 3D. Il numero di punti di controllo di una patch di Bézier di grado n è $(n+1) \times (n+1)$. Ad esempio per una patch di Bézier di grado 3, avremo $(3+1) \times (3+1) = 16$ punti di controllo, elencabili in una griglia 2D:

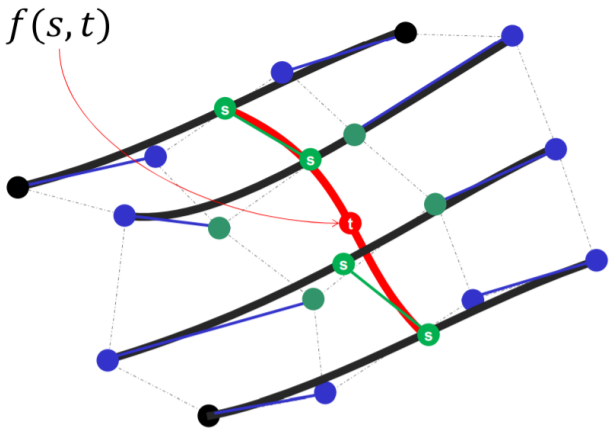
$$\begin{matrix} p_{0,0} & p_{1,0} & p_{2,0} & p_{3,0} \\ p_{0,1} & p_{1,1} & p_{2,1} & p_{3,1} \\ p_{0,2} & p_{1,2} & p_{2,2} & p_{3,2} \\ p_{0,3} & p_{1,3} & p_{2,3} & p_{3,3} \end{matrix}$$

dove ogni riga/colonna è una curva di Bézier.



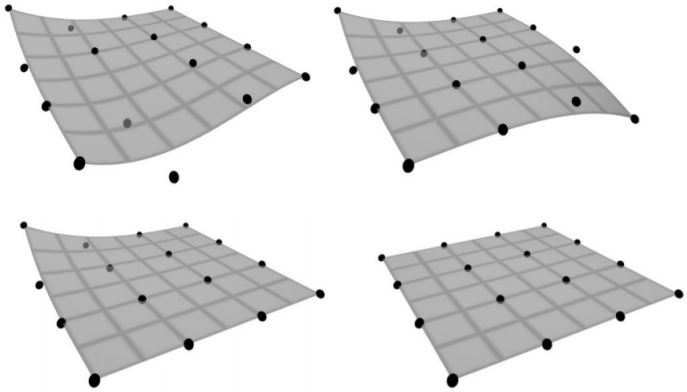
Per ottenere un punto come interpolazione di questi punti di controllo, l'algoritmo è il seguente:

- trovo a come $f(s)$ sulla curva di Bézier $p_{0,0}$ $p_{0,1}$ $p_{0,2}$ $p_{0,3}$
- trovo b come $f(s)$ sulla curva di Bézier $p_{1,0}$ $p_{1,1}$ $p_{1,2}$ $p_{1,3}$
- trovo c come $f(s)$ sulla curva di Bézier $p_{2,0}$ $p_{2,1}$ $p_{2,2}$ $p_{2,3}$
- trovo d come $f(s)$ sulla curva di Bézier $p_{3,0}$ $p_{3,1}$ $p_{3,2}$ $p_{3,3}$
- trovo p come $f(t)$ sulla curva di Bézier a, b, c, d



Patch di Bézier - applicazioni

Le patch di Bézier cubiche sono molto utilizzate, in quanto la patch passa per i 4 angoli, cioè è interpolante, mentre è approssimante per tutti gli altri angoli.



In generale, così come una curva di Bézier è costituita da archi di Bézier, una superficie di Bézier è costituita da patch di Bézier, e per connetterle valgono le considerazioni sulla continuità e sulla *smoothness* viste per le curve.

Vantaggi

✔ Sono facili da manipolare, e sono quindi ottime per il design industriale (CAD)

✔ Sono compatte, in quanto si devono memorizzare solo i punti di controllo

✔ Sono molto espressive, in quanto la superficie è realmente curva

✔ Hanno **risoluzione adattiva**, in quanto il numero di patch determina la risoluzione del modello

✔ Sono già parametrizzate, quindi l'uv-mapping non è necessario

Geometry Processing

Alcune operazioni di geometry processing sono eseguibili, come la **semplificazione automatica** per ridurre il numero di patch, oppure costruire una superficie parametrica a partire da altre strutture dati, anche se quest'ultimo è un task difficile.

È spesso necessario convertire le superfici parametriche in mesh poligonali, ad esempio per il rendering. Fortunatamente è un task semplice:

1. Campiono il dominio parametrico della superficie parametrica in modo regolare, ottenendo delle posizioni (s, t) nel dominio parametrico e formando una griglia regolare.
2. Per ogni posizione campionata (s, t):
    1. creo un vertice della mesh nel punto $f(s, t)$
    2. assegno a ogni vertice la sua normale calcolata con le derivate di f
3. Creo triangolo connettendo i vertici.

La densità del campionamento stabilisce la risoluzione che avrà la mesh risultante: