- $T_{2,m}(x) = \sum_{J=0}^{N} a_J \cos 2J\pi x + b_J \sin 2J\pi x$ $\quad x \in [0,1]$

$b_0 = b_N = \emptyset$

- $a_J = \frac{1}{N} \sum_{K=0}^{2N-1} f_K \cos \frac{2JK\pi}{2N}$

- $b_J = \sum_{K=0}^{N} f_K \sin \frac{2JK\pi}{2N}$ $\quad x_K = \frac{K}{2N}$ $\quad 0 \le K < 2N$

(equispaziati)

⇒ ANALISI di FOURIER (a_J e b_J)

⊛ $T_{2,m}$ → SINTESI di FOURIER

Complessità $\Theta(N^2)$

Scriveremo le somme a_J e b_J e T_{2N} sotto forma di numeri complessi:

$c = a + bi$ — a parte reale, b parte immaginaria

$i^2 = -1$

$\frac{1}{i} = -i$

$c_1 = a_1 + ib_1$

$c_2 = a_2 + ib_2$

$c_1 + c_2 = a_1 + a_2 + i(b_1 + b_2)$

$c_1 \cdot c_2 = a_1 \cdot a_2 + ia_1 b_2 + ia_2 b_1 - b_1 b_2$

quindi

$c_1 \cdot c_2 = a_1 a_2 - b_1 b_2 + i(a_1 b_2 + a_2 b_1)$

formula di Eulero

$e^{ix} = \cos x + i \sin x$

$e^{-ix} = \cos(-x) + i \sin(-x)$

$= \cos(x) - i \sin(x)$

$e^{ix} + e^{-ix} = 2\cos x$ $\quad$ quindi $\quad \cos x = \frac{e^{ix} + e^{-ix}}{2}$

$e^{ix} - e^{-ix} = 2i \sin x$ quindi $\sin x = \frac{e^{ix} - e^{-ix}}{2i}$

Ora riscriviamo analisi e sintesi di Fourier usando esponenziali complessi:

$$f(x) = \sum_{j=0}^{N} a_j \frac{e^{i2j\pi x} + e^{-i2j\pi x}}{2} + b_j \frac{e^{i2j\pi x} - e^{-i2j\pi x}}{2i}$$

$$= \sum_{j=0}^{N} e^{i2j\pi x}\left(\frac{a_j}{2} + \frac{b_j}{2i}\right) + e^{-i2j\pi x}\left(\frac{a_j}{2} - \frac{b_j}{2i}\right) =$$

$$= \sum_{j=0}^{N} e^{i2j\pi x}\left(\frac{a_j - ib_j}{2}\right) + e^{-i2j\pi x}\left(\frac{a_j + ib_j}{2}\right) =$$

$$= \sum_{j=-N}^{N} c_j e^{i2j\pi x}$$

$$c_j = \frac{a_j - ib_j}{2} \quad 0 < j < N$$

$$c_j = \frac{a_j + ib_j}{2} \quad -N < j < 0$$

$$c_{\pm N} = \frac{a_N}{2}$$

$$c_0 = a_0$$

dopo tutte queste trasformazioni resta:

$$T_N(x) = \sum_{j=0}^{N-1} c_j e^{2j\pi x}$$

SINTESI

$$c_j = \frac{1}{N} \sum_{k=0}^{N-1} f_k e^{-\frac{i\,2jk\pi}{N}}$$

ANALISI DI FOURIER

Lavoreremo sull'analisi riscrivendo ulteriormente la formula:

$$c_j = \frac{1}{N} \sum_{k=0}^{N-1} f_k \omega^{jk} \quad \text{con } \omega = e^{-\frac{i\,2\pi}{N}}$$

abbiamo quindi $2N^2$ operazioni per questa formula $[\Theta(N^2)]$

Calcoleremo la formula in maniera un po particolare.

$$c_J = \frac{1}{N} \sum_{k=0}^{\frac{N}{2}-1} f_k \omega^{Jk} + f_{k+\frac{N}{2}} \omega^{J(k+\frac{N}{2})} =$$

$$= \frac{1}{N} \sum_{k=0}^{N/2-1} f_k \omega^{Jk} + f_{k+\frac{N}{2}} \omega^{Jk} \cdot \omega^{J\frac{N}{2}} =$$

abbiamo che $\omega^{\frac{JN}{2}} = e^{\frac{-i2\pi}{N} \cdot \frac{JN}{2}} =$

$$= e^{-iJ\pi} =$$

$$= \cos(-\pi J) + i \sin(-\pi J) =$$

$$= 0$$

perché il cos varia sulla base di J (1 se J è pari -1 se è dispari) $= (-1)^J$

Quindi tornando alla formula ho:

$$= \frac{1}{N} \sum_{k=0}^{N/2-1} f_k \omega^{Jk} + f_{k+\frac{N}{2}} (-1)^J \omega^{Jk}$$

Avendo il comportamento diverso sulla base dei J pari o dispari riscrivo tutto così:

$$c_{2m_1} = \frac{1}{N} \sum_{k=0}^{N/2-1} f_k \omega^{2m_1 k} + f_{k+\frac{N}{2}} \omega^{2m_1 k} \qquad J = 2m_1 \text{ (pari)}$$

$$c_{2m_1+1} = \frac{1}{N} \sum_{k=0}^{N/2-1} f_k \omega^{2m_1 k} \omega^k - f_{k+\frac{N}{2}} \omega^{2m_1 k} \omega^k \qquad J = 2m_1+1 \text{ (dispari)}$$

in pratica ho diviso le 2 sommatorie una somma i pari ed una i dispari (perché il cos mi deve dare valori alternati 1 e -1)

Metto un po' in evidenza:

$$c_{2m_1} = \frac{1}{2}\left[\frac{1}{N/2}\sum_{k=0}^{N/2-1}\left(f_k + f_{k+\frac{N}{2}}\right)\omega_1^{m_1 k}\right] \qquad \left(\omega_1 = \omega^2\right)$$

$$c_{2m_1+1} = \frac{1}{2}\left[\frac{1}{N/2}\sum_{k=0}^{\frac{N}{2}-1}\left(f_k - f_{k+\frac{N}{2}}\right)\omega^k\,\omega_1^{m_1 k}\right]$$

Verifichiamo ora quante operazioni ci vogliono per calcolare le 2 somme.

chiamo $f_k + f_{k+\frac{N}{2}} = f_k^{(1)}$

e $f_k - f_{k+\frac{N}{2}} = f_{k+\frac{N}{2}}^{(2)}$

quindi

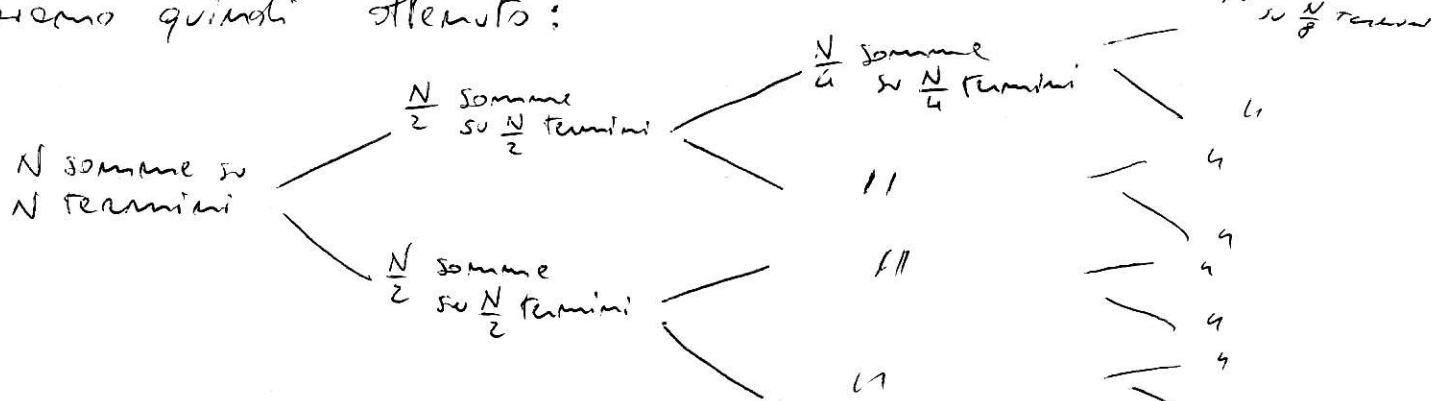
$$c_{2m_1} = \frac{1}{2}\left[\frac{1}{N/2}\sum_{k=0}^{N/2-1} f_k^{(1)}\,\omega_1^{m_1 k}\right]$$

$$c_{2m+1} = \frac{1}{2}\left[\frac{1}{N/2}\sum_{k=0}^{N/2-1} f_{k+\frac{N}{2}}^{(1)}\,\omega_1^{m_1 k}\right]$$

L'ordine in questo caso è $\frac{N^2}{2} + \frac{N}{2} = \frac{1}{2}N^2 \Rightarrow \Theta(N^2)$ operazioni.

Ho risparmiato poco rispetto a $2N^2$ operazioni, e l'ordine è sempre alto.

Abbiamo quindi ottenuto:



Possiamo quindi proseguire nella divisione delle somme

Stiamo passati da N somme su N termini e N somme su $\frac{N}{2}$ termini e N somme su $\frac{N}{4}$ termini ecc. ecc.

PRETENDIAMO CHE PER QUESTO ALGORITMO I DATI SIANO IN NUMERO PARI A POTENZE DI 2.

$$N = 2^L$$

Andando avanti le sommatorie si riducono ad un numero solo: $\frac{N}{2}$; $\frac{N}{4}$; $\frac{N}{8}$ ---- $\frac{N}{N} = 1$

Quindi all'ultimo passo avrò N somme da 1 termine.
Ovviamente per arrivare all'ultimo passo dovrò considerare gli $f_K^{(1)}$ e gli $f_{K+\frac{N}{2}}^{(1)}$

Ad ogni passo faccio N operazioni che sono quelle utili e calcolare la f.
Il numero di passi sarà $\log_2 N$ per il fatto che N = potenza di 2.

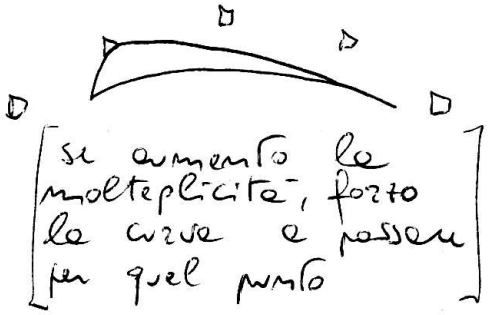
Quindi sono scesa a $N \log_2 N$ ed è un abbattimento notevole della complessità.

Nelle macchinette digitali la trasformata di Fourier è implementata via circuiti integrati.

21/03

Algoritmo di de Boor

Ci consente di calcolare una curva B-S in un punto.
È un algoritmo locale (cioè modifica solo alcuni punti del poligono). È basato sulla tecnica del knot-insertion
(un nodo di molteplicità p, $C(u) = P_i$)



INPUT: u
OUTPUT: $C(u)$
$u \in [u_k, u_{k+1}]$ $\quad u = u_k$

① Sia $h = p$ (cioè inserisco (u) p volte)
$S = 0$ (molteplicità di u)

② Se $u = u_k$ e ha molteplicità S, $h = p - S$ (cioè u ha molteplicità S e devo fare in modo che abbia molteplicità p.

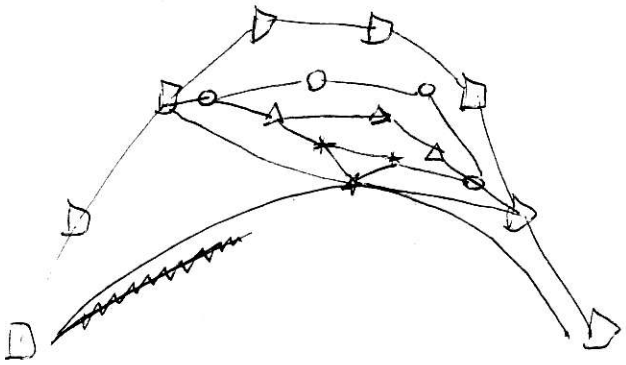
$P_{k-S}, \dots P_{k-p}$ vengono memorizzati $P_{k-S,0} \dots P_{k-p,0}$

```
for r = 1 to h do
    for i = k - p + r to k - S do
    begin a_{i,r} = (u - u_i) / (u_{i+p-r+1} - u_i)
    end
    P_{i,r} = (1 - a_{i,r}) P_{i-1,r-1} + a_{i,r} P_{i,r-1}
end
```

$P_{k-S, p-S}$ è il punto $C(u)$.

esempio:

Si vanno a costruire successivamente nuovi punti del poligono di controllo.



$u \in [0,1]$

Differenze tra de Casteljau e de Boor

Sono entrambi ricorsivi e stabili (piccole perturbazioni non vengono ritenute).

In de Casteljau i coeff. moltiplicativi non dipendevano né da i né da k cosa invece che accade in de Boor

Gli algoritmi di knot-insertion e de Boor sono detti tecniche di corner-cutting

TRASFORMAZIONI GEOMETRICHE

Ampliamo la geometria euclidea introducendo il concetto di infinito.

Consideriamo

(a, w), $w \neq 0 \longrightarrow \frac{a}{w}$ se $w \to 0$ $\frac{a}{w} \to \infty$

esempio:

$(3, 4) \longrightarrow (3w, 4w, w) \longrightarrow \left(\frac{3w}{w}, \frac{4w}{w}\right)$

si può pensare a tali coordinate inserendo questo w che è il divisore

$(3, 4, 5) \longrightarrow \left(\frac{3}{5}, \frac{4}{5}\right)$ è il corrispondente sulle coordinate cartesiane

coordinate omogenee

Si chiamano coordinate omogenee perché:

supponiamo di avere $ax + by + c = 0$ sostituiamo (x, y) con $\left(\frac{x}{w}, \frac{y}{w}\right)$

quindi

$\frac{ax}{w} + \frac{by}{w} + c = 0$ ossia $ax + by + cw = 0$

possiamo a questa coordinate omogenee ma non introdurre il concetto di infinito.

Trasformazioni importanti in grafica

TRASLAZIONE

Dato un punto $\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \longrightarrow \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$ se sposto uno nuovo sono $x' = x + h$, $y' = y + k$

In sostanza sposto il punto

rappresentazione matriciale

rappresentazione matriciale del punto:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & h \\ 0 & 1 & k \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

trasformazione inversa

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -h \\ 0 & 1 & -k \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$$

ROTAZIONE

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Le rappresentazione matriciale è comoda perché si possono comporre facilmente 2 o più trasformazioni

TRASLAZ. + ROTAZ.

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\alpha & -\sin\alpha & h \\ \sin\alpha & \cos\alpha & k \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

In 3 dimensioni la rappresentazione matriciale è:

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 & 0 \\ \sin\alpha & \cos\alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$ rotazione asse z

$$// = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha & 0 \\ 0 & \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$ rotaz. asse x

$$// = \begin{bmatrix} \cos\alpha & 0 & \sin\alpha & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\alpha & 0 & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$ rotaz. asse y

Trasformazioni affini

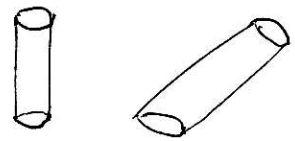
Mentre nelle rotazioni e traslazioni le forme dell'oggetto non cambiano; con le trasf. affini cambiano le forme dell'oggetto.

SCALING (rimpicciolire o ingrandire)

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} p & & & \\ & q & & \\ & & r & \\ & & & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

p, q, r sono dei fattori di rimpicciolimento o ingrandim. e accade che siano maggiori o minori di 1.

SHEARING (l'oggetto viene spinto su un asse)



$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & \alpha & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

direzione x

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ \alpha & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

direzione y

TRASFORMAZIONI PROIETTIVE (il caso + generale di trasf. affini)

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} p_{11}, p_{12}, p_{13} \text{ ---} \\ \text{---} \\ \text{---} \\ \text{---} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

esempio:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & 0 \\ -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix}$$

la coordinata omogenea è zero per cui il punto è andato all'infinito.

CURVE NURBS hanno tutte le proprietà delle B-SPLINES

- ① Sono invarianti per trasformazioni proiettive ~~cosa~~ tale caratteristica non è presente in Bezier e B-Splines.

- non dire che posso lavorare sui punti del poligono di controllo per ~~trasformare~~ trasformare la curva.

$P_0, P_1 \dots P_m$ (punti) $\quad w_0, w_1 \dots w_m$ (pesi)

$U_0, U_1 \dots U_m$ (nodi)

$$C(u) = \frac{\sum_{i=0}^{n} N_{i,p}(u)\, w_i\, P_i}{\sum N_{i,p}(u)\, w_i}$$

in sostanza se voglio dare "importanza" ad un punto per esempio P_1, aumenterò il valore di w_1 e la curva si avvicinerà a P_1. Dando a w_1 il valore 0 è come se non si tenesse conto di P_1 nel disegno della curva.

② Sono razionali, per cui sono utili per modellare cerchi ed ellissi.

Il caso limite è avere tutti i pesi w_i pari a 1. In tal caso la curva si riduce ad una B-SPLINE.

③ Partendo da una B-Spline (4-dim.)

$$C(u) = \sum_i N_{i,p}(u)\, P_i$$

$$P_i = \begin{pmatrix} x_i \\ y_i \\ z_i \\ 1 \end{pmatrix} = P_i^{w_i} = \begin{pmatrix} w_i\, x_i \\ w_i\, y_i \\ w_i\, z_i \\ w_i \end{pmatrix}$$

Nel piano cartesiano in ambo i casi i 2 punti

Sono uguali. Passando delle coordinate omogenee alla conversione ottengo che $P_i = P_i^{w_i} = \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix}$

Quindi

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

Quindi la spline $C(u)$ è uguale a

Per tornare allo spazio a 3 dimensioni:

$$\begin{bmatrix} \frac{\sum N_{i,p}(u) w_i x_i}{\sum N_{i,p}(u) w_i} \\ \frac{\sum N_{i,p}(u) w_i y_i}{\sum N_{i,p}(u) w_i} \\ \frac{\sum N_{i,p}(u) w_i z_i}{\sum N_{i,p}(u) w_i} \end{bmatrix} = \overset{3 \text{ dimensioni}}{\diagdown} = \frac{\sum N_{i,p}(u) w_i P_i}{\sum N_{i,p}(u) w_i}$$

NURBS curve

Quindi una curva NURBS a 3 dimensioni è una B-Splines a 4 dimensioni

Più precisamente:

Le NURBS sono la proiezione 3 dimensionale di una B-SPLINE a 4 dimensioni

28/03

NURBS

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

Si parte da una B-SPLINE di 4 dimensioni

$$P_i = \begin{pmatrix} x_i \\ y_i \\ z_i \\ 1 \end{pmatrix} \quad ; \quad P_i^w = \begin{pmatrix} w_i x_i \\ w_i y_i \\ w_i z_i \\ w_i \end{pmatrix}$$

$$C(u) = \sum_i N_{i,p}(u) \begin{bmatrix} w_i x_i \\ w_i y_i \\ w_i z_i \\ w_i \end{bmatrix} = \begin{bmatrix} \sum N_{i,p}(u) w_i x_i \\ \sum N_{i,p}(u) w_i y_i \\ \sum N_{i,p}(u) w_i z_i \\ \sum N_{i,p}(u) w_i \end{bmatrix}$$

B-SPLINE in 4 dimensioni

$$\begin{bmatrix} \frac{\sum_i N_{i,p}(u) w_i x_i}{\sum N_{i,p}(u) w_i} \\ \frac{\sum_i N_{i,p}(u) w_i y_i}{\sum N_{i,p}(u) w_i} \\ \frac{\sum N_{i,p}(u) w_i z_i}{\sum N_{i,p}(u) w_i} \end{bmatrix} = \frac{\sum N_{i,p}(u) \begin{bmatrix} w_i x_i \\ w_i y_i \\ w_i z_i \end{bmatrix}}{\sum N_{i,p}(u) w_i} =$$

$$= \frac{\sum N_{i,p}(u) w_i P_i}{\sum N_{i,p}(u) w_i}$$

NURBS in 3 dimensioni

Una curva NURBS a 3 dimensioni può essere considerata come una proiezione di una B-SPLINE quadridimensionale

Per calcolare una NURBS in un punto, passo alle relative B-SPLINE in 4 dimensioni, le valuto con l'algoritmo di DE-BOER e poi ripasso alle NURBS.

DERIVATA di UNA B-SPLINE

$$\frac{d}{du} C(u) = C'(u) = \sum_{i=0}^{m-1} N_{i+1,p-1}(u)\, Q_i$$

$$\text{con } Q_i = \frac{p}{u_{i+p+1} - u_{i+1}} (P_{i+1} - P_i)$$

PARAMETRIZZAZIONE (parameter selection) (unità 9)

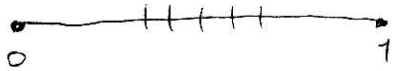
$u_0, u_1 \dots u_m$ $\quad m = n + p + 1$ (n = numero di punti del poligono di controllo)

Vogliamo che la curva passi per gli estremi.

$$u_0 = u_1 = \dots = u_p = 0$$

Come scegliamo i Knots?

1) Uniformemente spaziati $u_{j+p} = \frac{j}{m-p+1}$, $j = 1 \dots m-p$



è una parametrizzazione semplice ma dà cattivi risultati

2) average (media) $u_{j+p} = \frac{1}{p} \sum_{i=j}^{j+p-1} t_i$, $j = 1, \dots m-p$

Vediamo ora come scegliere le quantità t_i

(A) t_i uniformemente spaziati $t_i = \frac{i}{m}$; $i = 0 \dots m$
(non è una buona scelta)

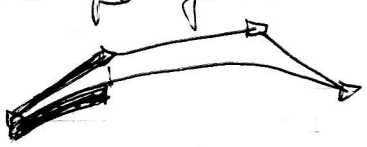
ci si ripropone lo stesso problema di prima.

(B) $t_0 = 0$; $t_n = 1$; $t_k = \frac{1}{L}\left(\sum_{i=1}^{k} |P_i - P_{i-1}|\right)$

con $L = \sum_{i=1}^{n} |P_i - P_{i-1}|$

Metodo chiamato ~~CORD~~ CHORD LENGHT

Si chiama così perché si cerca di mantenere una proporzione tra l'arco della curva e il segmento del poligono di controllo in questione



(C) Parametrizzazione centripeta

Simile alla precedente ma con un esponente in più

$$t_k = \frac{1}{L}\left(\sum_i |P_i - P_{i-1}|^a\right) \qquad a < 1$$

$$L = \sum |P_i - P_{i-1}|^a$$

Si vede che l'accelerazione centripeta della curva, in prossimità degli angoli, si minimizza

Per $a = 1$ $C = B$

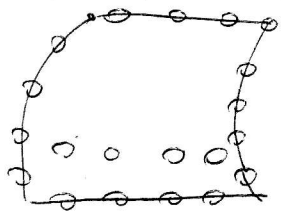
Per il progetto confrontarsi con esempi del capitolo 9.

SUPERFICI

Vediamo le parametriche $f(u,v) = (x(u,v), y(u,v), z(u,v))$
esistono anche le implicite (sono un insieme più vasto costituito da poligoni in 3 variabili)

Assegnamo dei punti detti CONTROL NET (l'equivalente del poligono di controllo per le superfici)

$P_{i,j}$ con $i = 0, \dots, m$
$j = 0, \dots, n$



Superfici di Bezier

Estensione delle curve di Bezier alle superfici.

$$s(u,v) = \sum_{i=0}^{m} \sum_{j=0}^{n} B_{m,i}(u)\, B_{n,j}(v)\, P_{i,j} =$$

$B_{m,i}(u)$, $B_{n,j}(v)$: funzioni di Bezier come nel caso delle curve

- abbiamo quindi 2 curve di Bezier che si "relazionano" tra loro.

$$= \sum_{i=0}^{m} B_{m,i}(u) \left[\sum_{j=0}^{n} B_{n,j}(v)\, P_{i,j} \right] \quad \circledast$$

$\left[\sum_{j=0}^{n} B_{n,j}(v)\, P_{i,j} \right] = q_i$

calcoliamo q_0

$$q_0 = \sum_{j=0}^{n} B_{n,j}(v)\, P_{0,j} \quad \leftarrow \text{1ª riga del control net}$$

Sostanzialmente questa è una curva di Bezier che posso calcolare con de Casteljau.

$$q_1 = \sum_{j=0} B_{m,j}(v)\, P_{0,j}$$

(stessa combinazione di prima)

q_2
$\vdots$
q_m

ed alla fine ottengo le m righe del control net.

Quindi una volta che i q_i sono noti le ⊛ diventa:

$$\sum_{i=0}^{m} B_{m,i}(u)\, q_i$$ ← anche questa è una curva di Bezier

e calcolando il tutto ho ottenuto una superficie di Bezier in 2 variabili.

Proprietà delle SUPERFICI

① $B_{m,i}(u), B_{m,j}(v) \geq 0$

② $\sum \sum B_{m,j}(v)\, B_{m,i}(u) = 1$

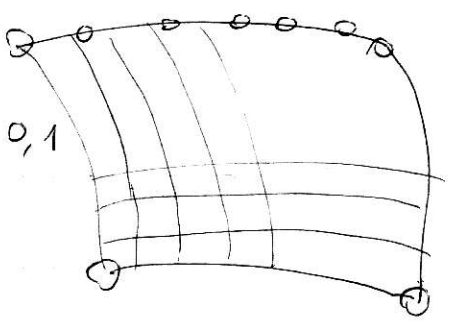
③ CONVEX HULL

④ La superficie passa per gli estremi che sono $P_{0,0}$; $P_{m,m}$; $P_{0,m}$, $P_{m,0}$.

Immaginiamo di avere

$p(u,v)$ $u = 0,1$

$p(0,1,v)$



Variando u ricopro tutte le superfici creando delle curve dette ISOPARAMETRICHE.

— stesse cose le posso fare per il parametro v.

$u = 0,1$; $v = 0,2$ ecc. ——

$p(u,1)$; $p(0,2,v)$ ———

— curve isoparametrica di Bezier è una curve (tutte anche rette)
— si fisso un parametro e faccio scorrere l'altro.

— casi particolari sono le curve isoparametriche di contorno o curve di confine.

$p(0,v)$; $p(1,v)$; $p(u,2)$, $p(u,1)$

$$p(0,v) = \sum B_{m,j}(v)\, P_{0,j}$$
$$p(u,1) = \sum B_{m,i}(u)\, P_{1,m}$$

$P_{0,0}$ $P_{1,1}$ ---
$P_{0,m}$ — rappresentano
$P_{1,m}$ — i bordi del control net.

Possiamo decomporre una superficie di Bezier nel seguente modello

$$p(u,v) = \underbrace{\left[B_{m,0}(u) \cdots B_{m,m}(u)\right]}_{\text{curve Bezier}} \begin{bmatrix} P_{0,0} & P_{0,1} & \cdots & P_{0,m} \\ & & & \\ P_{m,0} & \cdots & \cdots & P_{m,m} \end{bmatrix} \begin{bmatrix} B_{m,0}(v) \\ \vdots \\ B_{m,m}(v) \end{bmatrix}$$

curve Bezier

È importante perché ho realizzato una scomposizione in curve di Bezier unidimensionali.

L'analisi di Fourier mostra i suoi problemi nel momento in cui ci troviamo ad operare con segnali delle frequenze NON STAZIONARIE.

PROBLEMI della TRASFORMATA di FOURIER

- Contiene solo informazioni sulle frequenze, le funzioni sin e cos sono periodiche e non posso limitarle ad un singolo intervallo.
- Le informazioni sul tempo vengono perse.
- Funziona bene con segnali stazionari.
- Crea problemi con segnali non stazionari.

Ci vorrebbe una sorta di trasformata di Fourier nel tempo.

Tale trasformata esiste e si chiama trasformata di Fourier SHORT-TIME.

Suddividiamo il segnale in intervalli considerando ogni funzione con un valore nell'intervallo e nulla all'esterno di esso. Ci potrebbe essere un problema di stima dei coefficienti di Fourier scegliendo gli intervalli (box) netti. Le funzioni box quindi vanno scelte più dolci.

Problemi della STFT

- Scelta di una finestra appropriata
  - finestra piccola => ottima risoluzione in frequenza
  - finestra grande => ottima risoluzione nel tempo

esercizio d'esame

11/04

Interpolazione globale

$m+1$ punti dati $d_0, d_1, d_2 \dots d_m$ } (PUNTI DEL PIANO)

Voglio costruire una curva B-S di grado p fissato che passi attraverso $d_0, d_1 \dots d_m$.

I valori dei parametri $t_0 \dots t_m$ sono calcolati con i metodi noti.

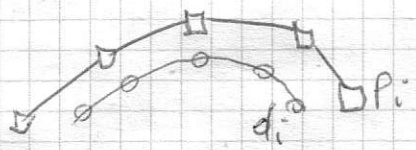
Da questi parametri calcolo il vettore dei knots di $m+1$ punti.

Cerco $m+1$ punti di controllo $P_0, P_1 \dots P_m$ — questi punti rappresentano l'incognita

$$\text{con } p(u) = \sum_{i=0}^{n} N_{i,p}(u)\, P_i$$

$$d_k = p(t_k) = \sum_{i=0}^{m} N_{i,p}(t_k)\, P_i, \quad k = 0 \dots m$$

$$N = \begin{bmatrix} N_{0,p}(t_0) & N_{1,p}(t_0) & \dots & N_{m,p}(t_0) \\ N_{0,p}(t_1) & N_{1,p}(t_1) & \dots & \dots \\ \vdots & & & \\ N_{0,p}(t_m) & \dots & \dots & \dots \end{bmatrix}$$



$$D = \begin{bmatrix} d_{0,1} & d_{0,2} & d_{0,s} \\ d_{1,1} & d_{1,2} & d_{1,s} \\ \dots & \dots & \dots \\ d_{m,1} & d_{m,2} & d_{m,s} \end{bmatrix} \qquad P = \begin{bmatrix} P_{0,1} & \dots & P_{0,s} \\ P_{1,1} & \dots & P_{1,s} \\ \dots & \dots & \dots \\ P_{m,1} & \dots & P_{m,s} \end{bmatrix}$$

d_k trattati come m vettori nello spazio S ($S=2$ o $S=3$) (la dimensione di S è 2 nel piano e 3 nello spazio)

idem?

$\boxed{D = N \times P}$ P rappresenta le incognite.

risolvo il sistema per colonne

d^i i-esima colonna di D $\quad i$

P^i " " " P $\quad d^i = N \cdot P^i$

$i = 0, \dots m$

Esempio del sistema

$k = 0$

$$d_0 = p(t_0) = N_{0,p}(t_0)p_0 + N_{1,p}(t_0)p_1 + \dots + N_{m,p}(t_0)p_m$$

$k = 1$

$$d_1 = p(t_1) = N_{0,p}(t_1)p_0 + N_{1,p}(t_1)p_1 + \dots + N_{m,p}(t_1)p_m$$

ecc. ecc con k che arriva ad n.

Nella scelte di t_0 è il caso di fare il confronto con gli esempi delle dispense.

Possiamo scegliere $S = 2$ e muoverci nel piano

Vediamo ora come risolvere un sistema di equazioni lineari.

$$Ax = b$$

metodo di Gauss

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \\ \quad a_{22}x_2 + a_{23}x_3 + \dots + a_{2n}x_n = b_2 \\ \quad \vdots \qquad\qquad \ddots \\ \quad a_{n-1,n-2}x_{n-1} + a_{n-1,n}x_n = b_{n-1} \\ \quad a_{nn}x_n = b_n \end{cases}$$

MATRICE TRIANGOLARE

x_n è praticamente noto, per cui andando poi a sostituire il tutto a ritroso risolvo il sistema.

in generale

in calcolare $x_k = \dfrac{b_k - \sum_{j=k+1}^{n} a_{kj} x_j}{a_{kk}}$, $k = n-1, \dots$

Quindi dobbiamo cercare di ricondurre una generica matrice ad una triangolare.

In fatti ciò le formule è la seguente:

PREMESSE:

- Quando ad un'equazione del sistema sostituisco una combinazione lineare dell'eq. stessa con un'altra dello stesso sistema il nuovo sistema rimane equivalente al precedente.

In sostanza per ogni equazione si sommano le riga i alla riga $i-1$ moltiplicata per un coefficiente.

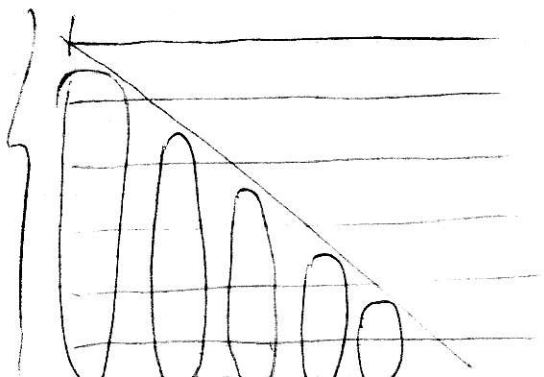
esempio

$$\begin{cases} -\dfrac{a_{21}}{a_{11}}\left(a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n\right) \\ + \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \end{cases}$$

$$0 + x_2\left(\frac{-a_{21}a_{12}}{a_{11}} + a_{22}\right) - - - -$$

Quindi il coefficiente x_1 è scomparso.

Facendo lo stesso per le seguenti equazioni, arriviamo ad una forma triangolare



algoritmo di Gauss

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} + m_{ik} a_{kj}^{(k)}$$

$j = k+1, \dots, n$

$i = k+1, \dots, n$

$$m_{ik} = -\frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}$$

$$b_i^{(k+1)} = b_i^{(k)} + m_{ik} b_i^{(k)}$$

ottengo con tale formule un sistema triangolare e per risolvere il sistema uso l'algoritmo per i sistemi triangolari

gli a_{kk} sono i pivot in caso di pivot pari a zero o molto piccoli si avrà gli problemi.

Uso la strategia del pivoting parziale, in sostanza scrivo i termini nella stessa colonna e considero il pivot di modulo massimo che ad esempio sarà nell'equazione i-esima e a questo punto posso scambiare l'equazione in questione con la i-esima e risolvo il problema