

Dispense di Grafica Computazionale

Relative all'anno accademico 2009 - 2010

**Corsi di laurea in:
Informatica
Tecnologie Informatiche**

Docente: Prof.ssa *B. Della Vecchia*

Rielaborazione e appunti a cura degli studenti:

D'Epifanio Stefano	(sdepifanio@hotmail.it)
Esposito Giuseppe	(bear852@gmail.com)
Gobbi Matteo	(matteogobbi@matteogobbi.it)
Hima Erin	(alfasite777@hotmail.com)
Perfetto Stefano	(stefano.perfo@gmail.com)
Veschini Andrea	(a.veschini@gmail.com)

Capitolo 1: **Curve di Bézier**

- 1.1 Storia
- 1.2 Definizione
- 1.3 Proprietà di una curva di Bézier:
- 1.4 Metodi di tracciamento della curva
 - 1.4.1 Algoritmo di De Casteljau
 - 1.4.1.1 Interpretazione geometrica dell'algoritmo di De Casteljau
 - 1.4.1.2 Algoritmo di De Casteljau: definizione
- 1.5 Algoritmo del Degree Elevation
- 1.6 Algoritmo di suddivisione
- 1.7 Derivata di una curva di Bézier
- 1.8 Continuità
- 1.9 Curve di Bézier Razionali

Capitolo 2: **Curve B-Splines**

- 2.1 Introduzione
- 2.2 Le funzioni Base
 - 2.2.0 Premesse
 - 2.2.1 Calcolo delle Funzioni Base
 - 2.2.2 Esempio B-spline
- 2.3 Definizione delle Curve B-Spline
- 2.4 Proprietà delle curve B-Spline
 - 2.4.1 Osservazioni sulle Curve B-Spline
- 2.5 Curve di Bézier vs. Curve B-Spline
- 2.5 Algoritmi delle B-Spline
 - 2.5.1 Knot insertion: Introduzione
 - 2.5.1.1 Knot Insertion: definizione
 - 2.5.2 Algoritmo di De Boor
 - 2.5.2.1 Algoritmo di De Boor in dettaglio
 - 2.5.3 Algoritmo di suddivisione per le B-Spline

- 3.0 Coordinate omogenee e trasformazioni geometriche
 - 3.0.1 Coordinate omogenee
 - 3.0.2 Trasformazioni geometriche
 - 3.0.2.1 Trasformazioni euclidee
 - 3.0.2.2 Trasformazioni affini
 - 3.0.2.3 Trasformazioni proiettive
 - 3.0.3 Comodità delle moltiplicazioni e trasformazioni nella forma matriciale

Capitolo 3:

Curve Nurbs

- 3.1 Un po' di Storia
- 3.2 Perché usare le curve NURBS?
- 3.3 Modellare le curve NURBS
- 3.4 Definizione di curva NURBS
- 3.5 Proprietà importanti delle curve NURBS
- 3.6 Relazione tra una curva B-Spline e una curva NURBS
- 3.7 Calcolo di una curva NURBS
- 3.8 Curve di Bézier vs. Curve B-Spline vs. Curve NURBS

Capitolo 4:

Superfici

- 4.1 Superfici
 - 4.1.1 Superfici parametriche
 - 4.1.2 Superfici implicite
- 4.2 Le superfici di Bézier
 - 4.2.1 Definizione di superficie di Bézier
 - 4.2.2 Proprietà delle superfici di Bézier
 - 4.2.3 Limitazioni delle superfici di Bézier
 - 4.2.4 La soluzione - Le patches di Bézier
- 4.3 Algoritmo di de Casteljau
- 4.4 Curve isoparametriche
 - 4.4.1 Curve isoparametriche di bordo
- 4.5 Prodotto Tensoriale
- 4.6 Superfici B-Spline
 - 4.6.1 Definizione superfici B-Spline
 - 4.6.2 Proprietà delle superfici B-Spline
 - 4.6.3 Algoritmo per ricavare la superficie

Capitolo 5:

Sistemi di proiezione

- 5.1 Sistemi di riferimento
 - 5.1.1 Definizione punti
- 5.2 Coordinate camera
- 5.3 Trasformazione punti del mondo nel sistema camera
 - 5.3.1 Soluzioni del sistema
- 5.4 Scelta della prospettiva
- 5.5 Progetto

Capitolo 6:
Simulazione

- 6.1 Simulazione
- 6.2 Cosa vuol dire simulazione?
- 6.3 Cosa vuol dire modello?
- 6.4 Complessità e accuratezza: la relazione tra simulazione e modello
- 6.5 Limiti dei modelli per la simulazione
- 6.6 Esempi di modello
 - 6.6.1 Il modello preda predatore
 - 6.6.2 Il modello Lotka - Volterra

Capitolo 7:
Automati Cellulari

- 7.1 Introduzione
- 7.2 Definizione
 - 7.2.1 Definizione formale
 - 7.2.2 Parole chiave
- 7.3 Le origini del modello
- 7.4 Dettagli e caratteristiche
 - 7.4.1 Possibilità di un automa cellulare
- 7.5 Utilizzi
 - 7.5.1 Automi e algoritmi decentralizzati
 - 7.5.2 Automi e crittografia
 - 7.5.3 Automi e sistemi biotici naturali
 - 7.5.4 Automi e chimica
 - 7.5.5 Automi e processori nei computer

Capitolo 8:
Modelli basati su agenti

- 8.1 Introduzione
- 8.2 Cos'è un agente?
- 8.3 Background degli ABM
- 8.4 ABM vs modelli differenziali
- 8.5 Perché utilizzare i modelli basati su agente?
 - 8.5.1 Benefici dei modelli basati su agente
- 8.6 Topologia alla base delle interazioni sociali
- 8.7 ABM, alcune applicazioni
 - 8.7.1 Evacuazioni
 - 8.7.2 Interazioni sociali
 - 8.7.3 Controllo dei flussi
 - 8.7.4 Organizzazioni
 - 8.7.5 Catena di servizi
 - 8.7.6 Stormi
 - 8.7.7 Mercato dell'energia Elettrica

Capitolo 9:
Game of life

- 9.1 Le origini del gioco
- 9.2 Esempi di configurazione

Appendice:
Domande frequenti per l'esame

Conclusioni

Capitolo 1

Curve di Bézier

Nel campo matematico dell'analisi numerica una curva di Bézier è un'importante curva parametrica polinomiale usata nella computer graphics.

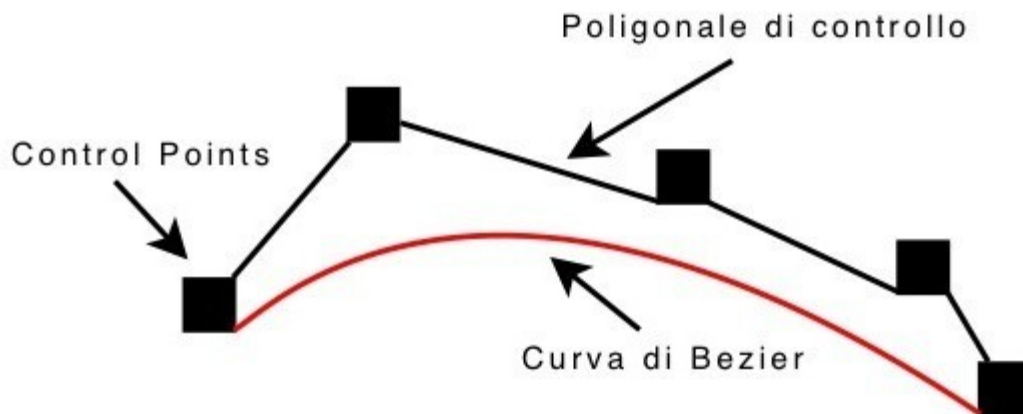
1.1 Storia

Le curve di Bézier sono largamente pubblicizzate nel 1962 dal francese Pierre Bézier, un ingegnere meccanico francese che lavora per la fabbrica Renault, che le usò per disegnare le carrozzerie delle automobili. I problemi con cui Bézier aveva a che fare riguardavano la gestione delle macchine a controllo numerico che tranciavano i pezzi di lamiera con cui fare le carrozzerie delle auto.

Le curve furono realizzate nel 1959 da Paul Casteljaou usando l'algoritmo di De Casteljaou. Bézier stesso, molto onestamente, ammise che alle stesse curve era giunto anche il suo collega della Citroen De Casteljaou, ma i vincoli di segretezza imposti da questa fabbrica hanno fatto sì che tali curve restassero legate al suo nome. Bézier stabilì un modo di realizzare le curve che partiva da due punti e una linea vettoriale appunto, un sistema innovativo che permette ancora oggi agli operatori grafici di realizzare disegni curvilinei bellissimi e precisi.

1.2 Definizione

Dati $n+1$ punti P_i denominati Punti di Controllo [Control Points], denominiamo poligono di controllo la linea spezzata che unisce gli $n+1$ punti P_i .



Dati i punti P_0, P_1, \dots, P_n , la curva di Bézier è descritta attraverso la formula in rappresentazione parametrica:

$$B_n(t) = \sum_{k=0}^n b_{n,k}(t) P_k$$

dove:

- P_k rappresentano i punti del poligono di controllo

$P_k \in \mathbb{R}^d$ con $d \geq 2$

- $B_{n,k}(t)$ rappresentano le funzioni di blending basate su polinomi di Bernstein di grado n

$$b_{n,k}(t) = \binom{n}{k} t^k (1-t)^{n-k} \quad \text{con } t \in [0,1]$$

Osservando la formula che descrive una curva di Bézier, notiamo che il punto che corrisponde ad t sulla curva è la media "pesata" di tutti i punti di controllo, dove i pesi sono i coefficienti $B_{n,k}(t)$.

1.3 Proprietà di una curva di Bézier:

1. La curva giace nel dominio convesso [Convex hull].

Questa proprietà assicura che la curva di Bézier definita da $n+1$ punti di controllo assegnati giace completamente nel dominio convesso formato dai punti di controllo. Il dominio convesso di un insieme di punti è il più piccolo insieme convesso che contiene tutti i punti.

La proprietà è sempre verificata quando:

$$b_{n,k}(t) \geq 0$$

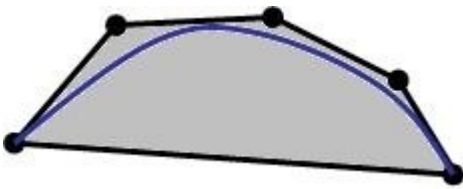
Ovvero:

$$b_{n,k}(t) = \binom{n}{k} t^k (1-t)^{n-k} \geq 0 \quad t \in [0,1]$$

La proprietà è importante perché siamo garantiti che la curva generata giacerà in una regione definita e calcolabile abbastanza vicina al poligono di controllo e non fuoriuscirà da essa.

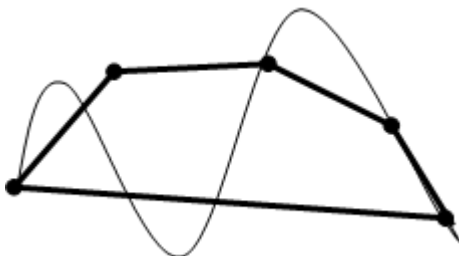
Esempio 1:

Nella figura seguente il dominio convesso dei 5 punti di controllo è mostrato in grigio. La curva corrispondente giacerà completamente in questa regione di piano.



Esempio 2:

Nella figura seguente la curva non rispetta la proprietà in quanto fuoriesce dal dominio convesso.



2. La curva è ben definita.

Questa proprietà assicura che la curva è ben definita cioè è indipendente dalla scelta dell'origine delle coordinate e dipendente solo dai punti P_i . Quindi se traslo tutti i punti P_i di un vettore V , anche la curva corrispondente sarà spostata rispetto all'origine di una quantità V .

Occorre verificare che le funzioni di blending in somma diano 1:

$$\sum_{k=0}^n b_{n,k}(t) = 1$$

Ovvero:

$$\sum_{k=0}^n \binom{n}{k} t^k (1-t)^{n-k} = 1$$

Essendo:

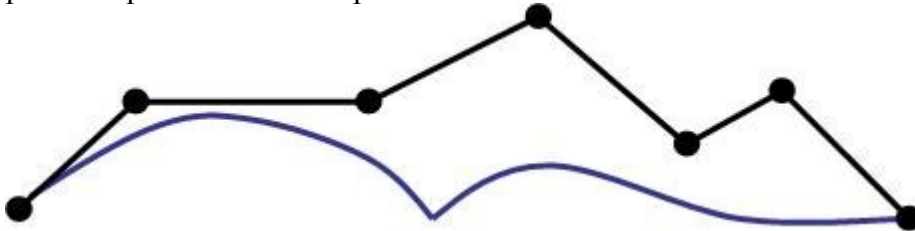
$$(a+b)^n = \sum_{k=0}^n \binom{n}{k} a^k b^{n-k}$$

se sostituiamo $a = t$ e $b = 1 - t$ otteniamo:

$$(t+1-t) = \sum_{k=0}^n \binom{n}{k} t^k (1-t)^{n-k}$$

3. Regolarità della curva(smoothness).

Questa proprietà afferma che la curva di Bézier deve essere dolce e regolare e non deve presentare picchi o discontinuità; questo risulta possibile in quanto la funzione $B_n(t)$ è derivabile più volte perché somma di polinomi.

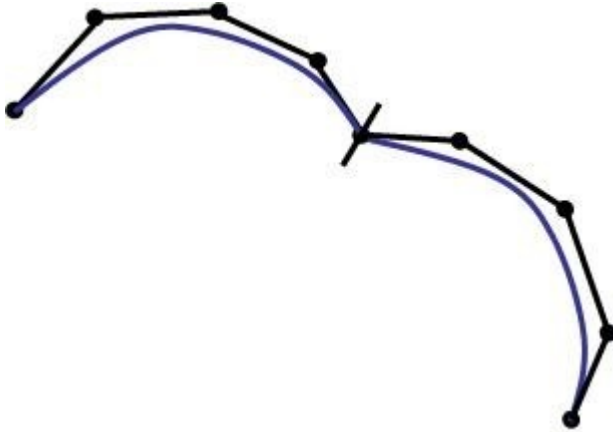


4. Interpolazione agli estremi.

Questa proprietà afferma che le curve di Bézier passano sempre attraverso il primo e l'ultimo punto del poligono di controllo.

Questo è utile per raccordare bracci di curve diverse.

Per esempio nel caso in cui si hanno due curve distinte:



per unire le due curve basta far coincidere i due punti estremi.

$$b_{n,k}(0) = \begin{cases} 0, & k \neq 0 \\ 1, & k = 0 \end{cases} \quad b_{n,k}(1) = \begin{cases} 0, & k \neq n \\ 1, & k = n \end{cases}$$

E' necessario sostituire 0 e 1 nella formula $B_n(t)$.

5. Simmetria di una curva.

Questa proprietà assicura che invertendo l'ordine dei punti del poligono di controllo non cambia la forma della curva. Ciò è assicurato dal fatto che

$$b_{n,k}(t) = b_{n,n-k}(1-t)$$

ovvero:

$$B_n(P_n, \dots, P_0, t) = B_n(P_0, \dots, P_n, (1-t))$$

6. Conservazione di punti e rette.

Questa proprietà afferma che le curve di Bézier riproducono punti e rette, cioè se tutti i punti P_i del poligono di controllo collassano nel primo punto P_0 anche la curva coinciderà con quell'unico punto. Ciò è garantito da:

$$\sum b_{n,t}(t) = 1 \quad \begin{matrix} P_0 \\ \bullet \end{matrix}$$

Discorso analogo per le rette, nel caso in cui i punti sono allineati a formare una retta, anche la curva seguirà l'andamento. Ciò è garantito da:

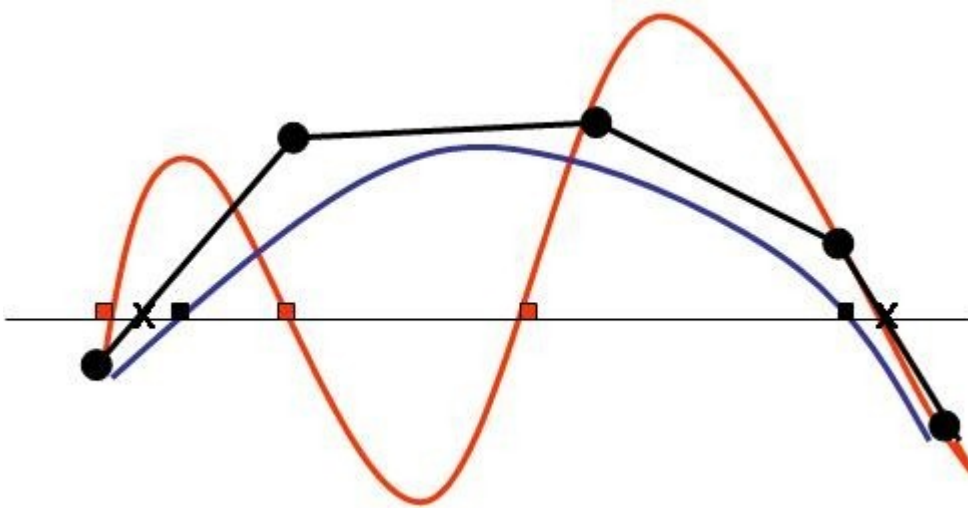
$$\sum b_{n,k}(t)k = nt$$

7. Riduzione delle oscillazioni [Variation diminishing].

La proprietà afferma che il numero di intersezioni di una qualsiasi retta con la curva è sempre minore o uguale al numero di intersezioni della stessa retta con il poligono di controllo. Questo ci garantisce che la curva non oscilla troppo rispetto al poligono di controllo.

Esempio:

Nella figura seguente è possibile notare che la curva in rosso non rispetta la proprietà in quanto la retta interseca la curva un numero di volte maggiore rispetto al numero di intersezioni della stessa retta con il poligono di controllo. Diversamente dalla curva in rosso, la curva in blu rispetta la proprietà in quanto il numero di volte che la retta interseca la curva è pari al numero di volte che la stessa retta interseca il poligono di controllo.

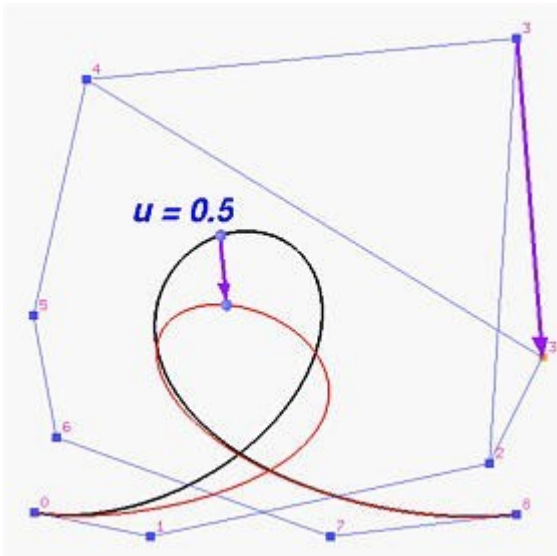


8. Controllo Locale[Proprietà negativa].

Questa proprietà è ritenuta negativa poiché variando la posizione di un punto di controllo la forma della curva di Bézier cambia in ogni punto.

Esempio:

La figura seguente illustra quest'effetto. Sia la curva nera che quella rossa sono curve di Bézier di grado 8 definite mediante 9 punti di controllo. Quella nera è l'originale. Se il suo punto di controllo 3 viene spostato in una nuova direzione come indicato dal vettore viola, la curva nera si modifica in quella rossa. Su ognuna delle due curve vi è il punto corrispondente a $u = 0.5$.



9. Dipendenza grado-punti di controllo[Proprietà negativa].

Questa proprietà afferma che la curva ha un grado che dipende dal numero dei punti di controllo. Il grado di una curva di Bézier definita da $n + 1$ punti di controllo è n quindi per esempio se il poligono di controllo è costituito da 100 punti, la corrispondente curva ha grado 99. Questo motivo comporta un grande svantaggio per le curve di Bézier poiché polinomi di grado elevato sono più difficili da gestire per motivi di instabilità numerica. Lo scopo è invece lavorare con curve con grado basso.

10. Invarianza per trasformazioni affini.

Questa proprietà stabilisce che quando occorre una trasformazione geometrica o anche affine (ovvero una rotazione, traslazione..etc) su una curva di Bézier, si opera **solamente** su i punti

di controllo, e una volta ottenuti i punti di controllo trasformati, la curva di Bézier trasformata è quella definita da questi nuovi punti. Pertanto non occorre lavorare sui punti della curva per trasformarla.

1.4 Metodi di tracciamento della curva

1.4.1 Algoritmo di De Casteljau

L'algoritmo di De Casteljau permette di costruire la curva di Bézier associata al vettore di punti di controllo assegnato, lavorando su combinazioni lineari.

1.4.1.1 Interpretazione geometrica dell'algoritmo di De Casteljau

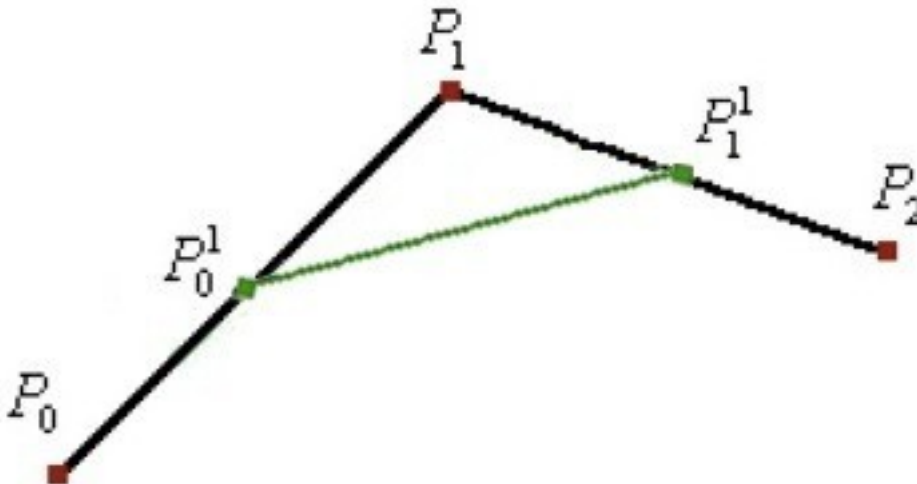
Vediamo ora alcuni esempi:

CURVA DI GRADO 2: arco di parabola

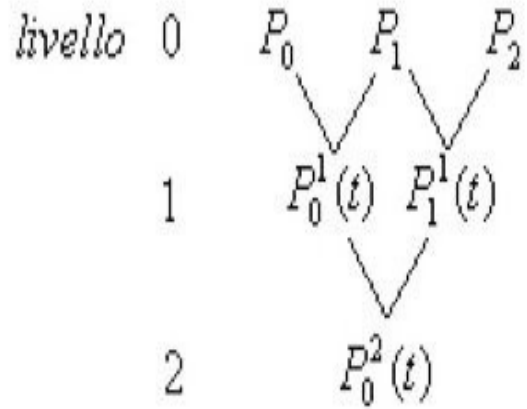
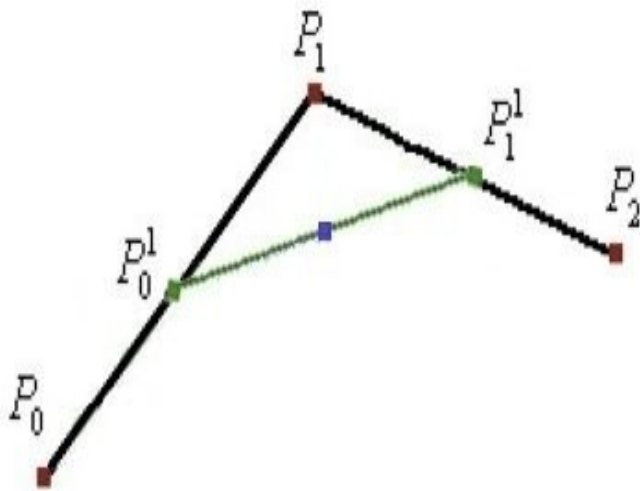
Il poligono di controllo è dato da 3 punti P_0, P_1, P_2 .

Esempio con $t = \frac{1}{2}$.

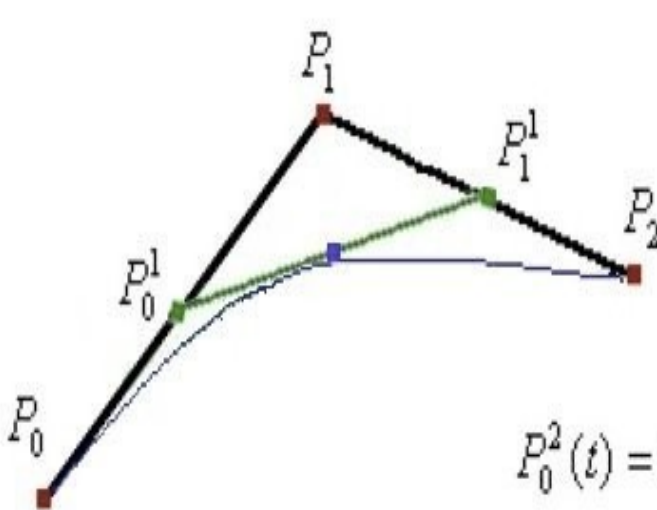
Considerato il valore $t = \frac{1}{2}$, si definiscono due punti sui segmenti P_0P_1 e P_1P_2 ; █



Ora, sul segmento congiungente questi due punti, si definisce un terzo punto, sempre in funzione di t . Il terzo punto è quello corrispondente alla curva di Bézier di grado 2 per il valore di t .



Ripetendo questo algoritmo per ogni $t = \frac{1}{2}$, ottengo l'intera curva. La formula generale è ottenuta per sostituzione:



$$P_0^1(t) = (1-t)P_0^0 + tP_1^0$$

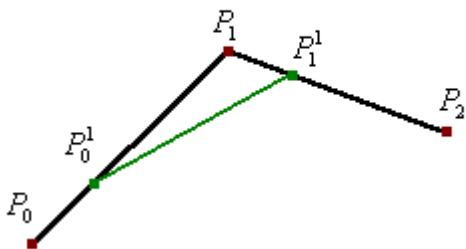
$$P_1^1(t) = (1-t)P_1^0 + tP_2^0$$

$$P_0^2(t) = (1-t)P_0^1(t) + tP_1^1(t)$$

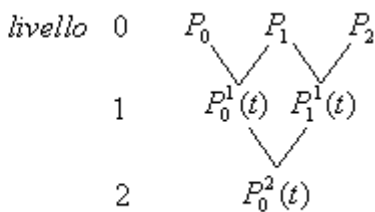
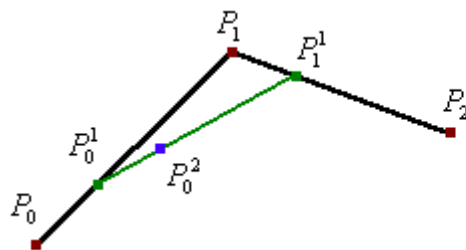
$$P_0^2(t) = (1-t)^2 P_0^0 + 2(1-t)tP_1^0 + t^2 P_2^0$$

Esempio con $t = \frac{1}{3}$.

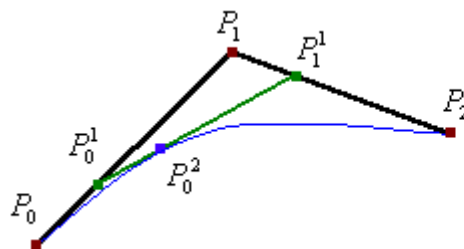
Considerato il valore $t = \frac{1}{3}$, si definiscono due punti sui segmenti P_0P_1 e P_1P_2 ; il punto all'interno del segmento si chiamerà in questo modo:
 $(0, t, 1) = (P_i, P_i^1, P_{i+1})$ *poligonale primo livello*



Ora, sul segmento congiungente questi due punti, si definisce un terzo punto, sempre in funzione di t . Il terzo punto è quello corrispondente alla curva di Bézier di grado 2 per il valore di t .



Ripetendo questo algoritmo per ogni $t = \frac{1}{3}$, ottengo l'intera curva. La formula generale è ottenuta



$$P_0^1(t) = (1-t)P_0 + tP_1$$

$$P_1^1(t) = (1-t)P_1 + tP_2$$

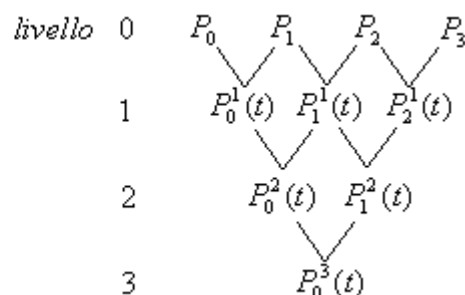
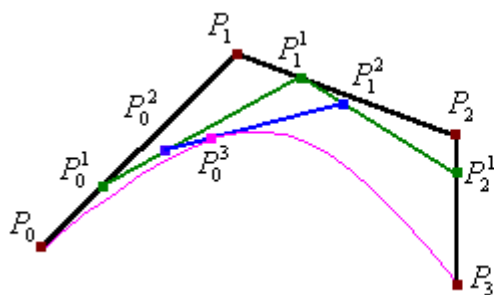
$$P_0^2(t) = (1-t)P_0^1(t) + tP_1^1(t)$$

$$P_0^2(t) = (1-t)^2 P_0 + 2(1-t)tP_1 + t^2 P_2$$

per sostituzione:

CURVA DI GRADO 3:

Il poligono di controllo è dato da 4 punti. Considerando un valore $t \in [0,1]$, si definiscono altri punti in modo da lavorare su curve di grado minore, come nell'esempio precedente:



La formula della curva è: $P_0^3(t) = (1-t)^3 P_0 + 3(1-t)^2 tP_1 + 3(1-t)t^2 P_2 + t^3 P_3$

1.4.1.2 Algoritmo di De Casteljau: definizione

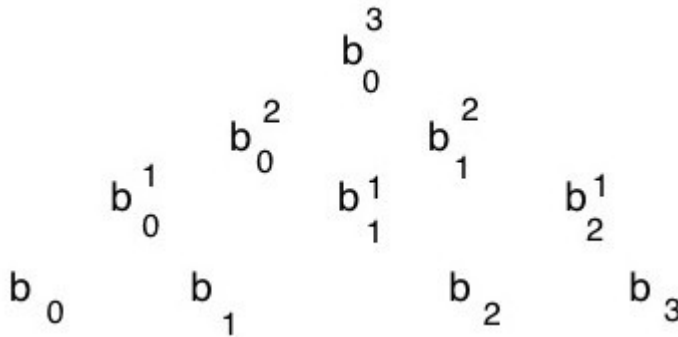
Dati b_i^0 punti di controllo di partenza, si costruiscono i punti successivi in tal maniera:

$$b_i^r = (1-t)b_i^{r-1}(t) + tb_{i+1}^{r-1}(t) \quad \begin{cases} r = 1, \dots, n \\ i = 0, \dots, n-r \end{cases}, \quad t \in \mathbb{R}$$

La curva di Bézier potrà essere calcolata in b_0^r ovvero:

$$b_0^r = b^t(t)$$

Infine i coefficienti intermedi $b_i^r(t)$ possono essere ordinati secondo lo *schema a triangolo di De Casteljau*:



Considerazioni:

L'algoritmo si presenta:

- STABILE in quanto i coefficienti $1-t$ e t sono compresi tra 0 e 1 e quindi non c'è amplificazione dell'errore.
- RICORSIVO in quanto viene applicata la stessa formula per ogni passaggio.

1.5 Algoritmo del Degree Elevation.

Supponiamo di aver disegnato una curva di Bézier di grado n e vogliamo modificare solo una parte della curva. Per aver maggior controllo della parte interessata aggiungiamo un ulteriore punto, lasciando allo stesso tempo inalterata la forma della curva.

L'algoritmo del degree elevation (elevazione di grado) nasce con l'idea di aumentare la flessibilità della curva costruendo un nuovo poligono di controllo, lasciando il primo e l'ultimo punto di controllo immutati e aggiungendo nuovi punti intermedi.

In particolare si parte da un poligono con n punti e si arriva ad un poligono che ne ha $n+1$.

Quindi:

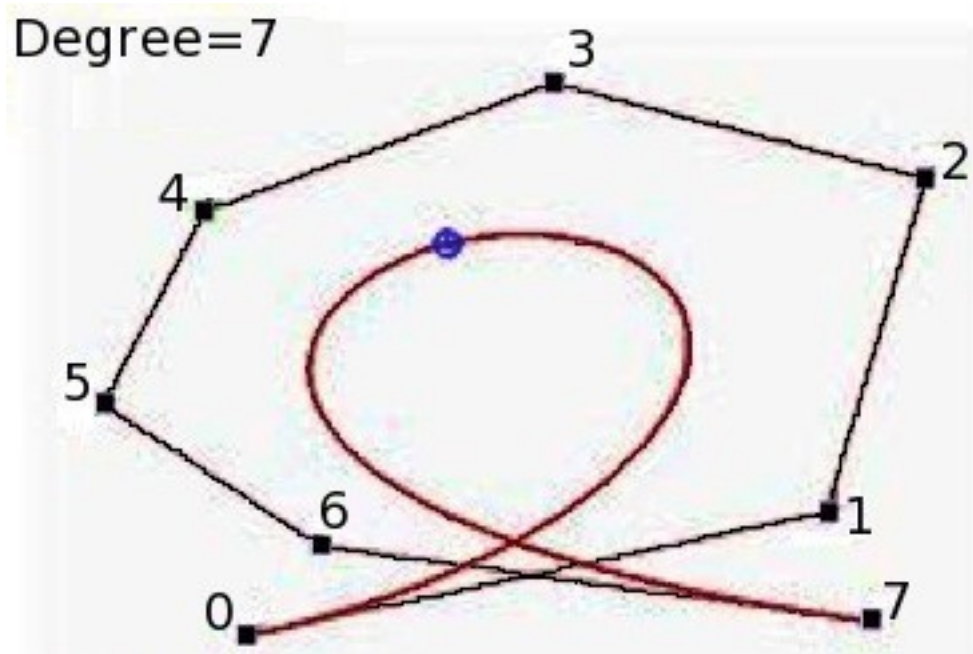
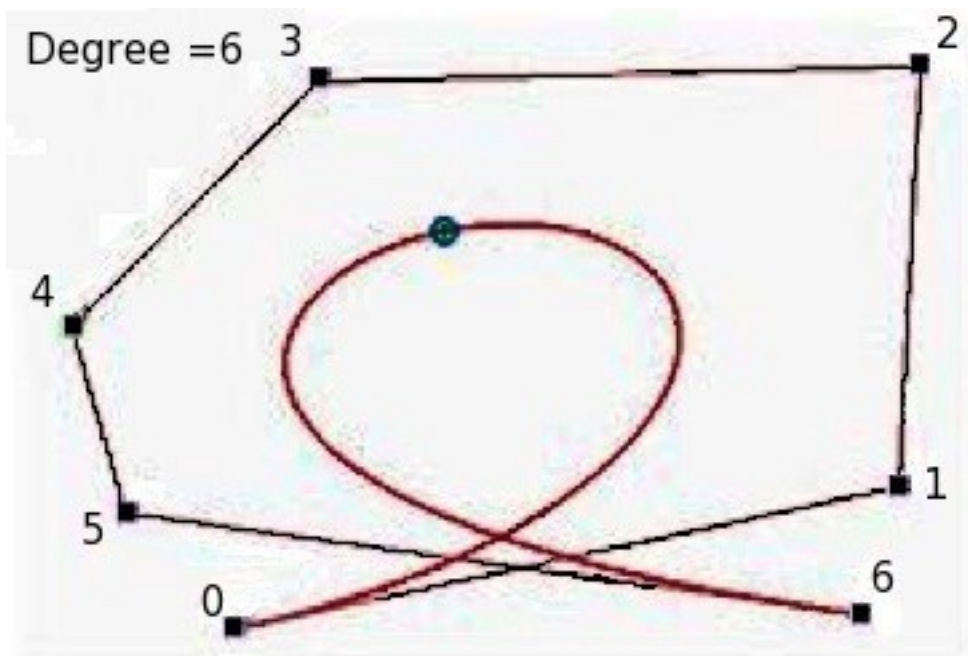
$$b_j^1 = \frac{j}{n+1} b_{j-1} + \left(\frac{j}{n+1}\right) b_j$$

con $j = 1, \dots, n$

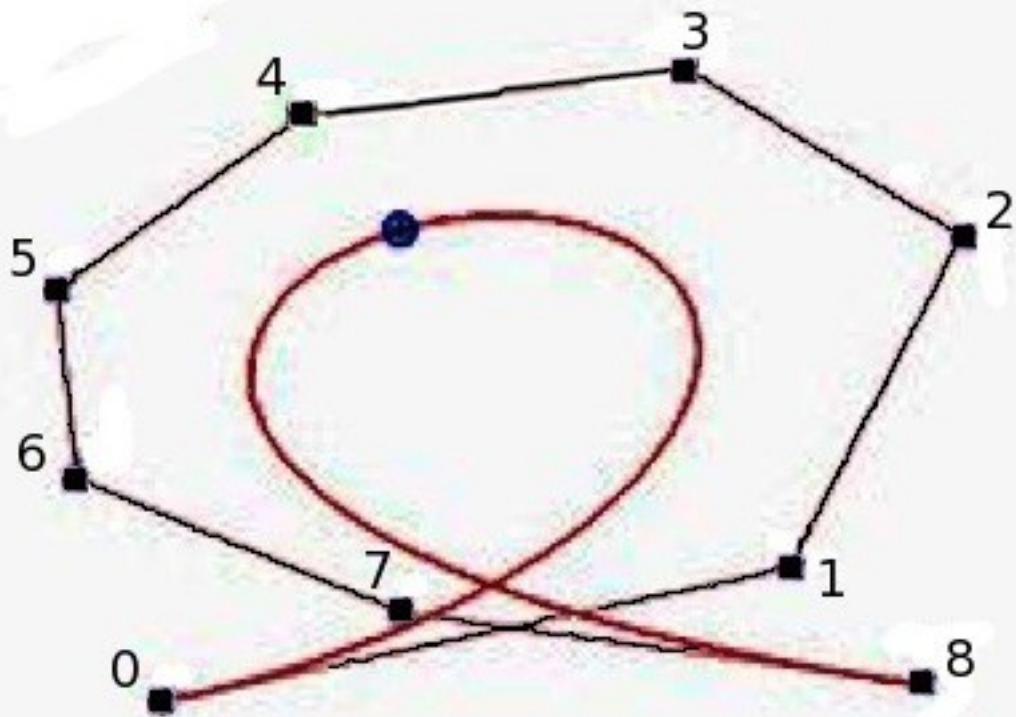
La formula permette di legare un punto ai precedenti. Il nuovo punto deve essere compreso tra b_0 e b_{n+1} , compresa quindi tra gli estremi che non devono essere toccati.

Iterando la procedura all'infinito il poligono di controllo si avvicinerà sempre più alla curva di Bézier arrivando a coincidere con essa.

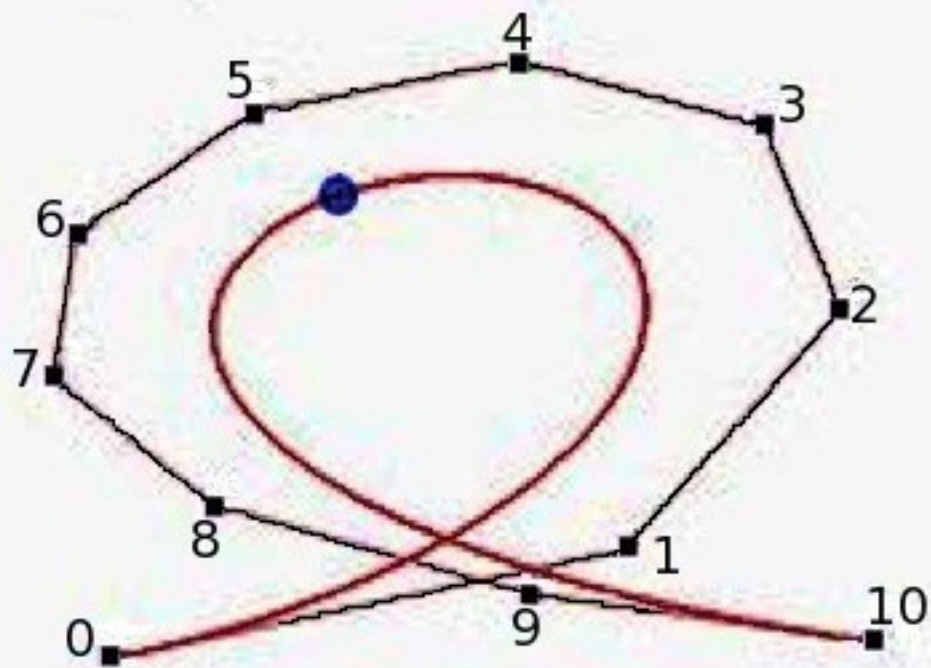
Vediamo alcuni esempi pratici:



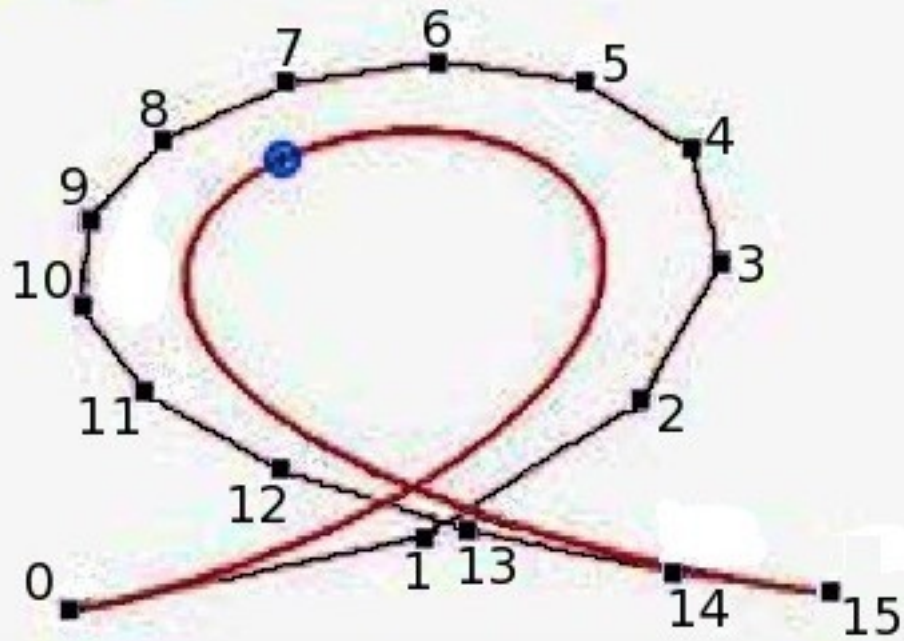
Degree=8



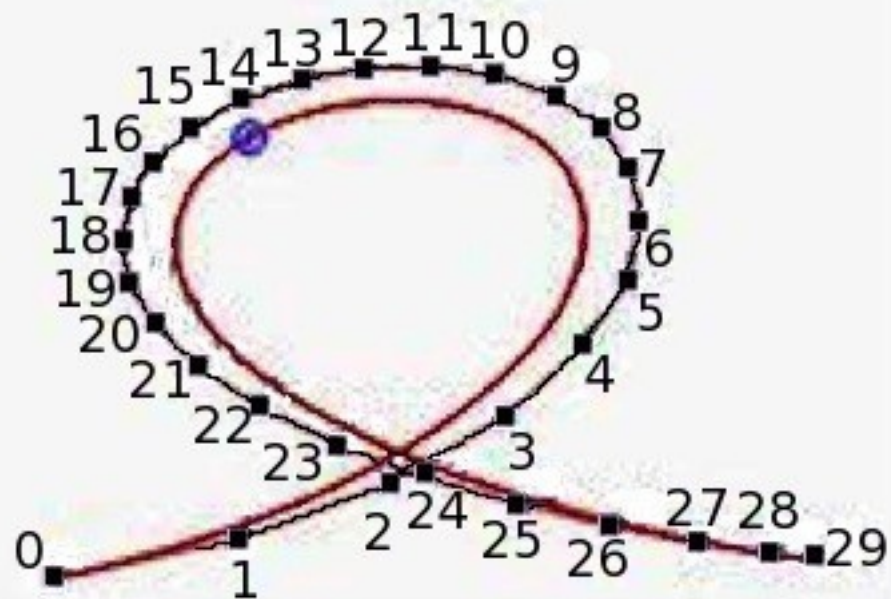
Degree=10



Degree=15



Degree=29



Considerazioni:

L'algorithmo si presenta:

- **STABILE**

- **RICORSIVO**

Svantaggi:

L'algorithmo si presenta **lento** in quanto se voglio avere un'ottima approssimazione della curva di Bézier devo iterare più volte la procedura. Per questo motivo questo algorithmo è meno usato in grafica.

1.6 Algoritmo di suddivisione

L'algorithmo di suddivisione permette, come l'algorithmo del Degree Elevation, di modellare una determinata parte della curva di Bézier. Supponiamo che il designer non sia soddisfatto di un solo tratto della curva. Possiamo allora voler suddividere la curva in due parti in un punto appropriato, una soddisfacente e l'altra no. Possiamo quindi dimenticarci quella soddisfacente e concentrarci su quella non soddisfacente. In particolare, occorre determinare nuovi punti di controllo

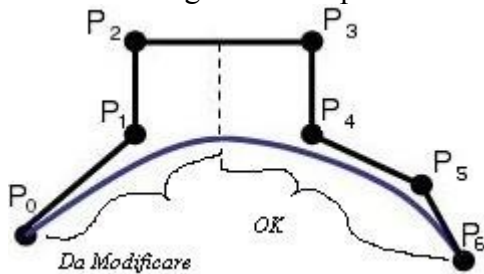
$r = 0, \dots, n$, definiti in un nuovo intervallo, tali che la nuova curva di Bézier $C_n(s)$ sia coincidente con la curva originaria nell'intervallo di riferimento.

Esempio:

Dato un poligono di controllo P_0, \dots, P_6 costruiamo la corrispondente curva di Bézier $B_n(t)$ e

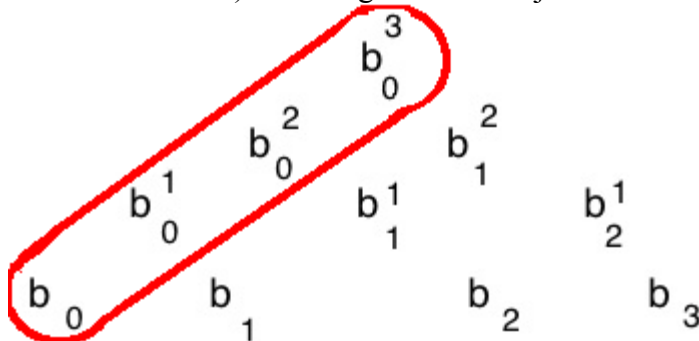
fissiamo un numero $c = \frac{1}{2}$ compreso tra 0 e 1. L'intervallo $[0, \frac{1}{2}]$ rappresenta il tratto della curva che stiamo prendendo in considerazione. Vogliamo ora, costruire una nuova curva di Bézier

$C_n(s)$ che deve essere uguale a $B_n(t)$ quando $t \in [0, \frac{1}{2}]$, e costruire dei vertici del poligono di controllo che generano la parte della curva da modificare.



Calcoliamo i coefficienti C_j con $j = 0, \dots, n$ che generano la curva $C_n(s)$.

I coefficienti $C_j = b_0^n(c)$ possono essere ottenuti considerando il lato sinistro (nella figura cerchiato in rosso) del triangolo di Casteljau.



1.7 Derivata di una curva di Bézier

$$B_n(t) = \sum_{i=0}^n b_{n,i}(t) P_i$$

Sia

calcoliamo la derivata per un punto fissato.

$$\frac{d}{dt} B_n(t) = n \sum_{i=0}^{n-1} b_{n-1,i}(t) \Delta P_i$$

Quindi abbiamo

dove $\Delta P_i = P_{i+1} - P_i$ indica la differenza tra due punti continui.

Osservazione: La derivata rappresenta un **odografo** ovvero un vettore tangente alla curva di Bézier calcolato per ogni punto.

Analizzando la parte destra dell'equazione, si può notare che la curva risultante è ancora una curva di Bézier relativa ora ad un altro insieme di punti (cioè $n(P_{i+1} - P_i)$), di grado di una unità in meno:

$$\sum_{i=0}^{n-1} b_{n-1,i}(t) Q_i$$

Vediamo ora il caso particolare in cui voglio calcolare la derivata di una curva di Bézier nel primo e nell'ultimo punto:

$$\frac{d}{dt} B_n(0) = n(P_1 - P_0) \quad \text{con } t \in [0,1]$$

$$\frac{d}{dt} B_n(1) = n(P_n - P_{n-1})$$

1.8 Continuità

Spesso in grafica per riprodurre forme particolari occorre raccordare più curve. Un metodo matematico utilizzato per fare ciò sono le derivate in quanto permettono di far congiungere due tratti di curva conservando una certa continuità.

Esistono due tipi di continuità: parametrica e geometrica.

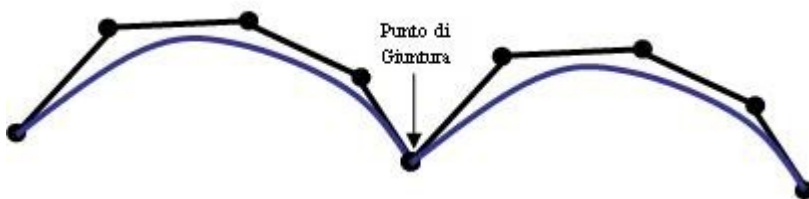
a) CONTINUITÀ PARAMETRICA:

In generale due curve continue parametricamente, saranno continue anche geometricamente, ma il viceversa non è verificato. Si parla di continuità parametrica quando le due curve adiacenti hanno uguali le derivate parametriche di grado i nei punti di giuntura; per avere che due derivate sono uguali con continuità parametrica dobbiamo imporre che i due vettori abbiano la stessa direzione e lo stesso modulo.

La continuità parametrica viene indicata con C_i dove i ne indica il grado. In particolare:

C_0 indica che le due curve hanno un punto in comune.

C_1 indica che le due curve hanno sia un punto in comune, sia la derivata prima uguale in direzione e modulo.



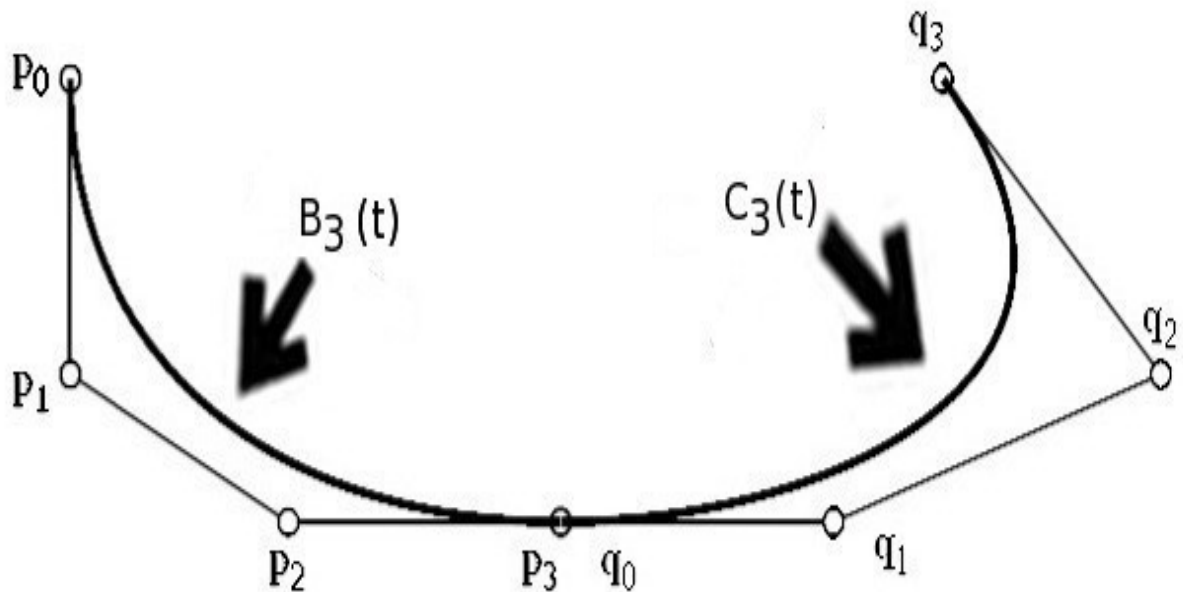
1° Poligono

2° Poligono

Osservazione: E' richiesta la continuità C_0 per avere un punto di giuntura tra le due curve e in genere serviranno altre continuità C_1 e C_2 per avere i due tratti di curva raccordati con una certa dolcezza nel profilo.

b) CONTINUITÀ GEOMETRICA:

Questo tipo di continuità è espresso con G ed è una continuità debole. Richiede che le derivate nei punti di giuntura abbiano la stessa direzione ma non necessariamente lo stesso modulo.



Dal disegno è facile vedere che la C_0 è garantita dall'uguaglianza $P_3 = Q_0$.

Deriviamo la prima curva $B_3(t)$ nel punto finale:

$$\frac{d}{dt}B_3(1) = 3(P_3 - P_2)$$

Deriviamo la seconda curva $C_3(t)$ nel punto di partenza:

$$\frac{d}{dt}C_3(0) = 3(q_1 - q_0)$$

sostituendo $q_0 = P_3$ otteniamo:

$$\frac{d}{dt}C_3(0) = 3(q_1 - P_3)$$

Considerazioni:

- per avere la continuità parametrica le due quantità devono essere uguali in modulo e direzione.
- per avere la continuità geometrica i due bracci $(P_2 - P_3)$ e $(q_0 - q_1)$ possono avere anche lunghezza diversa ma la direzione deve essere uguale.

1.9 Curve di Bézier Razionali

Alcune curve che sembrano semplici, come il cerchio, non possono essere descritte da una curva di Bézier, quindi abbiamo bisogno di maggiori gradi di libertà.

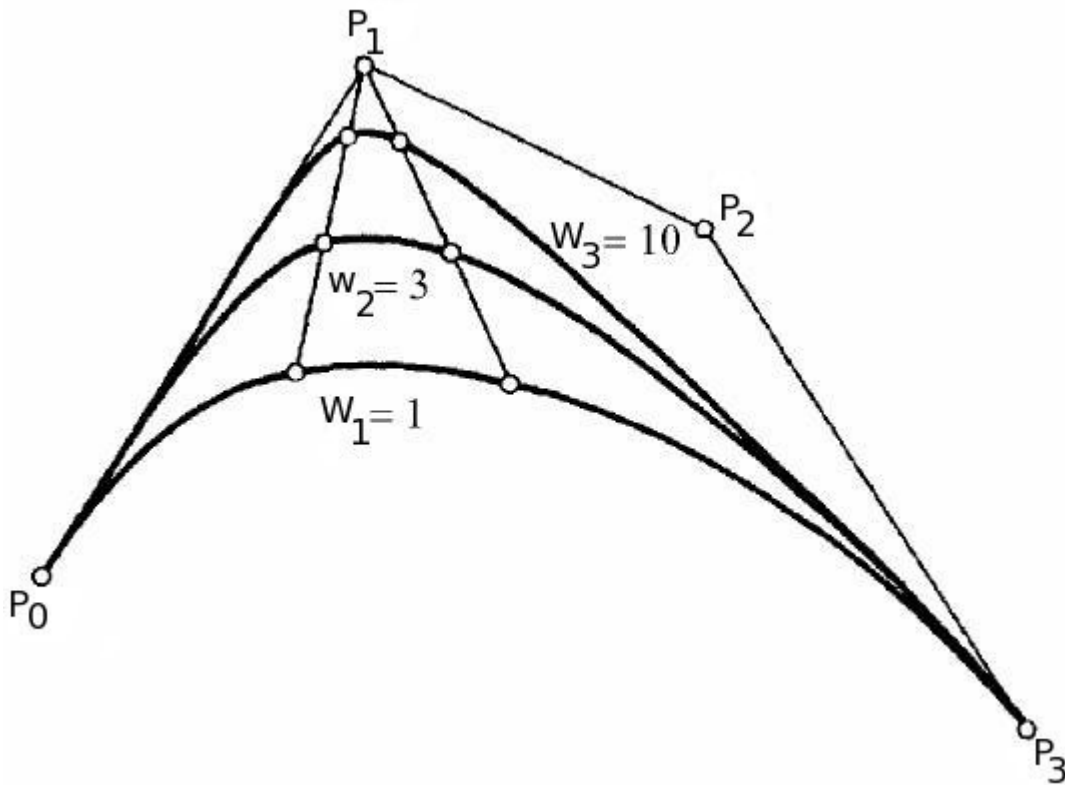
La curva di Bézier Razionale aggiunge dei pesi $w_i \geq 0$ che vengono utilizzati per poter modellare l'oggetto scegliendo un peso per un determinato punto.

Dati $n + 1$ punti di controllo P_i , la curva di Bézier razionale è data da:

$$B(t) = \frac{\sum_{i=0}^n b_{i,n}(t) P_i w_i}{\sum_{i=0}^n b_{i,n}(t) w_i}$$

Il numeratore è una curva di Bézier in forma di Bernstein pesata e il denominatore è una somma pesata di polinomi di Bernstein.

Esempio:



Osservazione: All'aumentare del valore di w_i , $B(t)$ si spinge verso P_i mentre al diminuire di w_i , $B(t)$ si allontana da P_i .

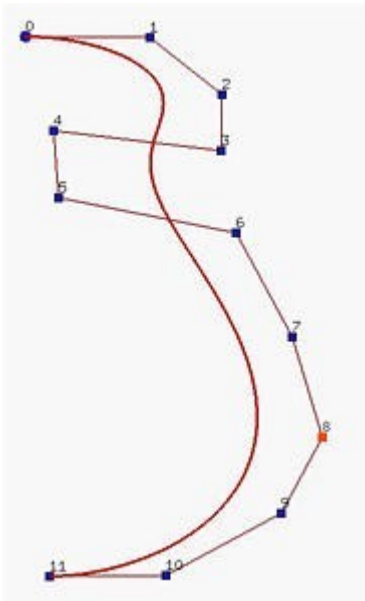
Capitolo 2

Curve B-Splines

2.1 Introduzione

Nel campo della CAGD, una curva può avere una forma così complessa da non poter essere rappresentata da una curva di Bèzier poichè avremo principalmente due limitazioni correlate tra loro ovvero il controllo globale e il grado della curva.

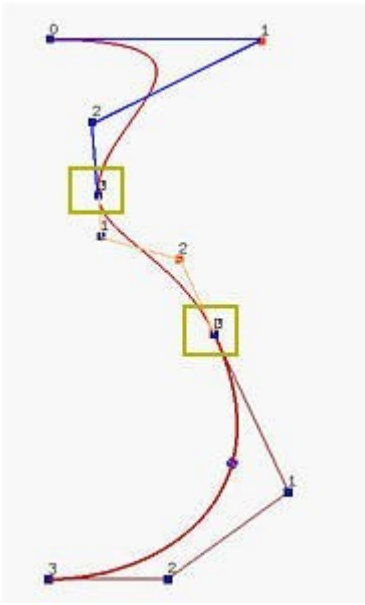
Per esempio supponiamo di voler disegnare un profilo di un vaso. La figura qui di seguito è una curva di Bèzier di grado 11; ma abbiamo una difficoltà a modellare il “collo” in corrispondenza del punto P_4P_5 . Naturalmente, utilizzando l'algorithmo del degree elevation possiamo aggiungere ulteriori punti di controllo vicino a questo segmento per incrementare il peso di quella regione. Questo comunque causerà un aumento del grado della curva e del corrispondente polinomio.



In molti casi, non vale la pena usare un polinomio di grado così alto.

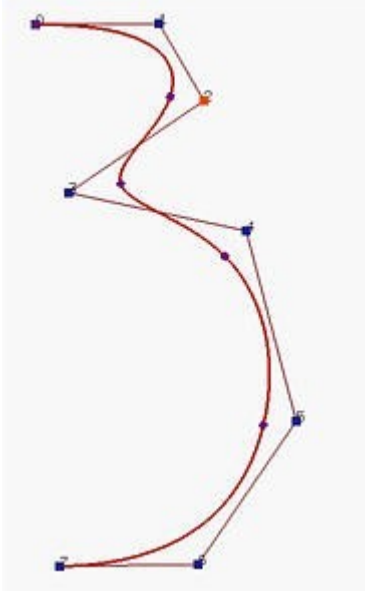
Possiamo unire due curve di Bèzier. Finché l'ultima parte della prima curva e la prima parte della seconda hanno la stessa direzione, possiamo ottenere almeno la G^1 continuità poiché i vettori tangenti hanno la stessa direzione ma possono avere lunghezza diversa (cioè, se anche la lunghezza è la stessa, otteniamo la C^1 continuità). La figura sottostante usa questa idea. Essa è composta da tre segmenti di una curva di Bèzier di grado 3 uniti nei punti marcati con rettangoli gialli.

Questo ci mostra che con più segmenti di una curva di Bèzier di grado basso che soddisfano la condizione della G^1 continuità, possiamo ancora disegnare forme complesse. Ma, mantenere la condizione di G^1 continuità può essere noioso e sgradito.



In questi casi si ricorre alle curve di B-Spline che sono generalizzazioni delle curve di Bezièr. Le curve B-Spline prendono il proprio nome dalle “spline”, flessibili asticcioline di legno, utilizzate per la

progettazione degli scafi delle navi. Fissati alcuni punti sulla sezione trasversale dello scafo, il resto della curva veniva determinato forzando, tramite alcuni chiodi, queste asticcioline a passare in corrispondenza di tali punti, lasciando l’asta libera di disporsi per il resto del profilo secondo la sua naturale tendenza; si determinava così una curva regolare e “smooth” (dolce) con comportamento preassegnato in certe posizioni.



In figura osserviamo una curva B-spline di grado 3 definita da 8 punti di controllo. Questi punti

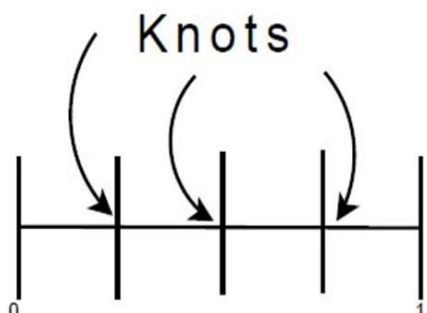
suddividono la curva B-spline in 5 segmenti di una curva di Bèzier. Le curve B-Spline permettono di disegnare una curva con un alto grado di libertà poichè offrono la possibilità di poter modificarne la suddivisione in segmenti e la sua relativa forma lavorando sui suoi punti di controllo.

2.2 Le funzioni Base

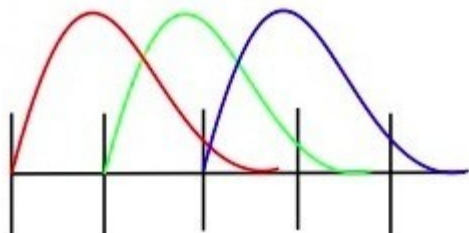
2.2.0 Premesse

Prima di analizzare le nostre curve è utile fare alcune premesse:

1) Il dominio della curva viene suddiviso da punti chiamati Knots.



2) Le funzioni base non sono non-nulle sull'intero intervallo. Questo implica che le funzioni assumono valori diversi da zero solo per una parte dell'intervallo di riferimento e sono, di conseguenza, completamente "locali".



2.2.1 Calcolo delle Funzioni Base

Le funzioni base sono definite da:

1)

Un **vettore dei knots** denominato U composto da $m+1$ numeri non-decrescenti chiamati **knots**, $u_0 \leq u_1 \leq u_2 \leq u_3 \leq \dots \leq u_m$. L'intervallo semi-aperto $[u_i, u_{i+1})$ corrispondente tra due knots si chiama ***i*-esimo knot span**.

Possono presentarsi diversi casi:

- se un knot u_i appare k volte (cioè, $u_i = u_{i+1} = \dots = u_{i+k-1}$), dove $k > 1$, u_i è un **knot multiplo** di **molteplicità** k , scritto come $u_i(k)$.

- se un knot u_i appare solo una volta, è un **knot semplice**.

- se tra i knots c'è sempre la stessa distanza (cioè, $u_{i+1} - u_i$ è una costante per $0 \leq i \leq m-1$), il vettore dei knots (o la sequenza dei knots) è detto **uniforme**; altrimenti, esso è **non-uniforme**.

- se il knot $u_0 = 0$ e il knot $u_m = 1$ si opera con un *vettore standard*.

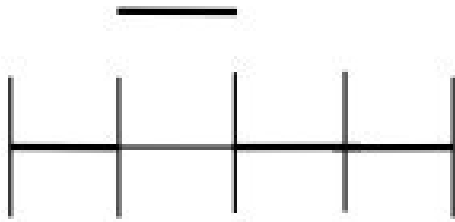
2)

Un parametro p che indica il grado della funzione base B-spline; il grado viene fissato indipendentemente dai punti di controllo. Generalmente il suo valore è 2 oppure 3. Questo rende possibile il disegno di curve regolari.

L' i -esima funzione base di una B-spline di grado p , scritta come $N_{i,p}(u)$, è definita ricorsivamente come segue:

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

Considerando $p = 0$ (il che indica che il polinomio associato avrà grado 0 e quindi la funzione sarà costante e parallela all'asse delle x):



N.B : Il risultato ,come si nota dalla figura, rappresenta funzioni a gradino.

Generalizzando, si introduce la *formula di ricorsione di Cox-de Boor* :

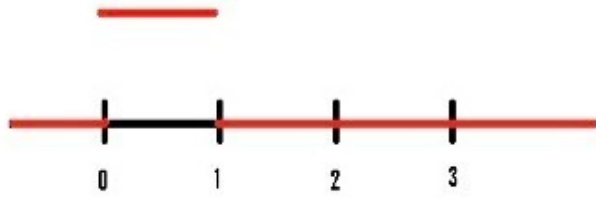
$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1} + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}$$

2.2.2 Esempio B-spline

Prendiamo un vettore di knot $U = (u_0, u_1, u_2, u_3)$ dove $u_0 = 0, u_1 = 1, u_2 = 2, u_3 = 3$ e consideriamo l'intervallo $[0,3]$.

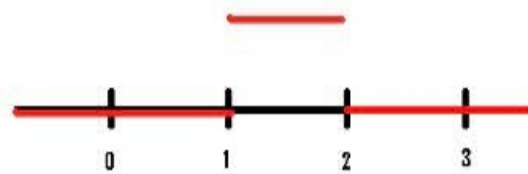
Calcoliamo la $N_{i,p}(u)$ nonché la curva per $i = 0, p = 0$ in $[0,1)$

$$N_{0,0}(u) = \begin{cases} 1 & \text{tra } 0 \text{ e } 1 \\ 0 & \text{altrimenti} \end{cases}$$



Consideriamo $i = 1, p = 0$ in $[1, 2)$

$$N_{1,0}(u) = \begin{cases} 1 & \text{tra } 1 \text{ e } 2 \\ 0 & \text{altrimenti} \end{cases}$$



Consideriamo $i = 2, p = 0$ in $[2, 3)$

$$N_{2,0}(u) = \begin{cases} 1 & \text{tra } 2 \text{ e } 3 \\ 0 & \text{altrimenti} \end{cases}$$



Vediamo che in questa maniera non riusciamo a disegnare delle curve, allora proviamo ad aumentare il grado della curva da 0 a 1.

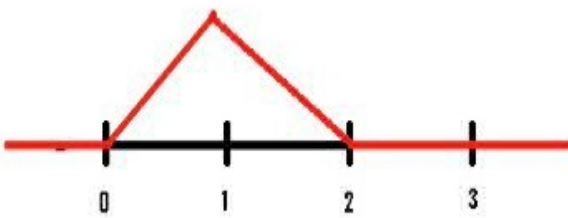
Consideriamo $i = 0, p = 1$ ed utilizziamo la formula di Cox-De Boor che diventa:

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

Vediamo come si comporta questa funzione in $[0, 1)$, in $[1, 2)$ e in $[2, 3)$. Se ci trovassimo nel primo intervallo descritto la nostra funzione diventerebbe:

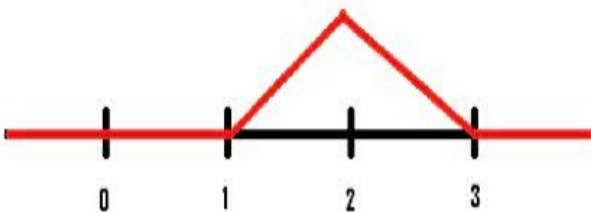
$$N_{0,1}(u) = u \cdot (1) + (2 - u) \cdot (0)$$

Questo perché in $[0,1)$, come si può verificare nei calcoli precedenti, $N_{0,0}$ vale 1 mentre $N_{1,0}$ vale 0. Procedendo in questa maniera per tutti gli altri intervalli in studio possiamo stabilire che:

$$N_{0,1}(u) = \begin{cases} u & \text{in } [0, 1) \\ 2 - u & \text{in } [1, 2) \end{cases}$$


Allo stesso modo calcoliamo $N_{1,1}$:

$$N_{1,1}(u) = \frac{u - u_1}{u_2 - u_1} N_{1,0} + \frac{u_3 - u}{u_3 - u_2} N_{2,0} = (u - 1) N_{1,0} + (3 - u) N_{2,0}$$

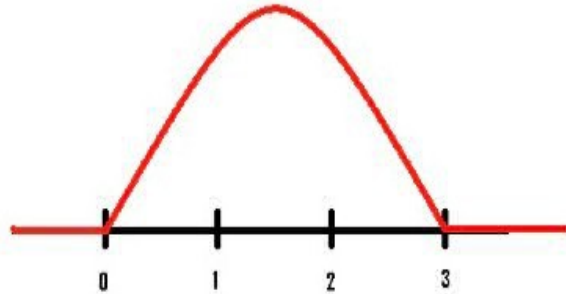
$$N_{1,1}(u) = \begin{cases} u - 1 & \text{in } [1, 2) \\ 3 - u & \text{in } [2, 3) \end{cases}$$


Nuovamente questa rappresentazione evidenzia la difficoltà di modellare una qualsiasi curva: si effettua quindi un nuovo incremento di p .

Per $p = 2$:

$$N_{0,2}(u) = \frac{u - u_0}{u_2 - u_0} N_{0,1} + \frac{u_3 - u}{u_3 - u_1} N_{1,1} = 0.5u \cdot N_{0,1} + 0.5(3 - u) \cdot N_{2,0}$$

$$N_{0,2}(u) = \begin{cases} \frac{1}{2}u^2 & \text{in } [0, 1) \\ \frac{1}{2}u^2(-2u^2 + 6u - 3) & \text{in } [1, 2) \\ \frac{1}{2}(3 - u)^2 & \text{in } [2, 3) \end{cases}$$



Vogliamo dimostrare come la curva che stiamo calcolando sia continua nella sua derivata prima e nella sua derivata seconda e come la sua rappresentazione sia la curva definita nel disegno sovrastante.

Calcoliamo allora $N'_{0,2}(u)$ e $N''_{0,2}(u)$:

$$N'_{0,2}(u) = \begin{cases} u & \text{in } [0, 1) \\ \frac{1}{2}(-4u + 6) & \text{in } [1, 2) \\ u - 3 & \text{in } [2, 3) \end{cases}$$

$$N''_{0,2}(u) = \begin{cases} 1 & \text{in } [0, 1) \\ -2 & \text{in } [1, 2) \\ 1 & \text{in } [2, 3) \end{cases}$$

Abbiamo allora dimostrato che la nostra curva è **continua e derivabile**. Se la derivata prima è uguale in un punto, la curva proseguirà dolcemente e non avrà più la forma a spigoli come è possibile notare nei casi dei disegni precedenti.

Infatti vediamo come la derivata prima nell'intervallo $[1, \frac{3}{2})$ sia maggiore di zero, in $\frac{3}{2}$ sia uguale a 0 e successivamente in $[\frac{3}{2}, 3)$ sia minore di zero. Prendendo in esame queste considerazioni è possibile disegnare la curva sopraindicata.

2.3 Definizione delle Curve B-Spline

Dati n punti di controllo P_0, \dots, P_n , un vettore di Knots $U = \{u_0, \dots, u_m\}$ e un grado p definiamo una curva B-Spline come:

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

N.B: La forma di una curva B-Spline è molto simile a quella di una curva di Bèzier (somma, funzioni base, punti).

2.4 Proprietà delle curve B-Spline

1. $N_{i,p}(u)$ è una funzione polinomiale a tratti di grado p in u . La proprietà implica che per ogni intervallo viene definito un polinomio. In totale si otterrà un polinomio a tratti raccordabile.
2. $N_{i,p}(u) \geq 0$. La proprietà afferma che le funzioni base essendo positive possono essere considerate come pesi, e ogni punto della curva si otterrà come la media pesata dei punti del poligono di controllo.
3. **Supporto Locale.** $N_{i,p}(u)$ è un polinomio non nullo nell'intervallo $[u_i, u_{i+1})$. Questa proprietà determina il supporto locale implicando la possibilità di modellare bene una curva poiché non tutte le $N_{i,p}(u)$ giocano un ruolo fondamentale.
4. **Partizione dell'unità.** La proprietà afferma che la somma di tutte le funzioni base non-nulle di grado p sul generico span $[u_i, u_{i+1})$ è 1. Quindi la curva non dipende dalla scelta dell'origine.

$$\sum N_{i,p}(u) = 1$$

5. $m = n + p + 1$. La proprietà afferma che su ogni knot-span $[u_i, u_{i+1})$ al più ci sono $p + 1$ funzioni base non nulle.

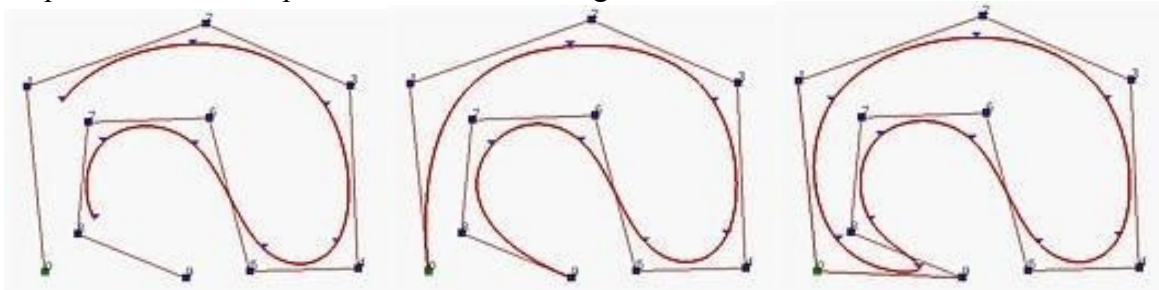
6. $p - k + 1$. La proprietà afferma in ogni knot di molteplicità k il numero di funzioni base diverse da 0 è al più $p - k + 1$.

7. $N_{i,p}(u) \in C^{p-k}$. La proprietà afferma che le funzioni godono della continuità parametrica (stessa direzione e stesso modulo). La “smoothness” dipende dal valore di k ovvero dalla molteplicità. Aumentando si perde regolarità. Di conseguenza all’aumentare del grado p cresce la continuità ovvero la regolarità.

2.4.1 Osservazioni sulle Curve B-Spline

Sebbene restano valide le proprietà già elencate e discusse per le curve di Bèzier, vengono indicate alcune proprietà particolari che dimostrano la superiorità funzionale delle curve B-Spline.

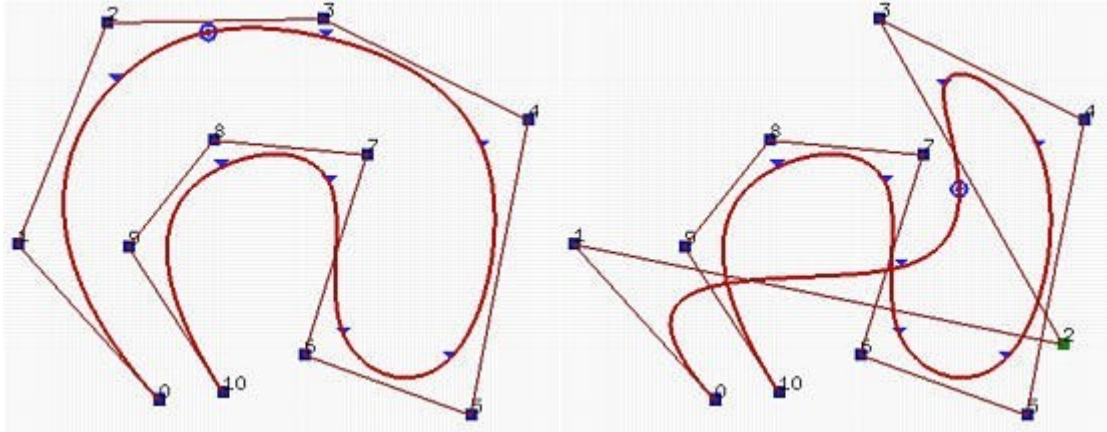
- > Una curva B-Spline non passa necessariamente per i punti estremi del poligono di controllo. La strategia da usare per imporre questo passaggio è assegnare molteplicità p ai punti che coincidono con gli estremi del poligono. Ponendo u_0 e u_m con molteplicità p la curva passerà per i punti estremi del poligono di controllo.
- > Se le curve non toccheranno i punti estremi del poligono di controllo, le B-spline sono chiamate curve *aperte* (come mostrato nella prima figura sottostante). Nel caso in le curve passino per il primo e l’ultimo poligono di controllo (vd. prima osservazione elencata) le curve sono definite *strette* (figura centrale). Ripetendo alcuni knots e punti di controllo la curva generata può essere *chiusa*. In questo caso, l’inizio e la fine della curva generata si uniscono per formare un loop come mostrato nella figura sottostante a destra.



- > **Controllo Locale.** La proprietà garantisce una modifica locale di una parte della curva infatti variando la posizione del punto di controllo P_i si influenza solo la curva $C(u)$ nell'intervallo $[u_i, u_{i+p+1})$.

Esempio:

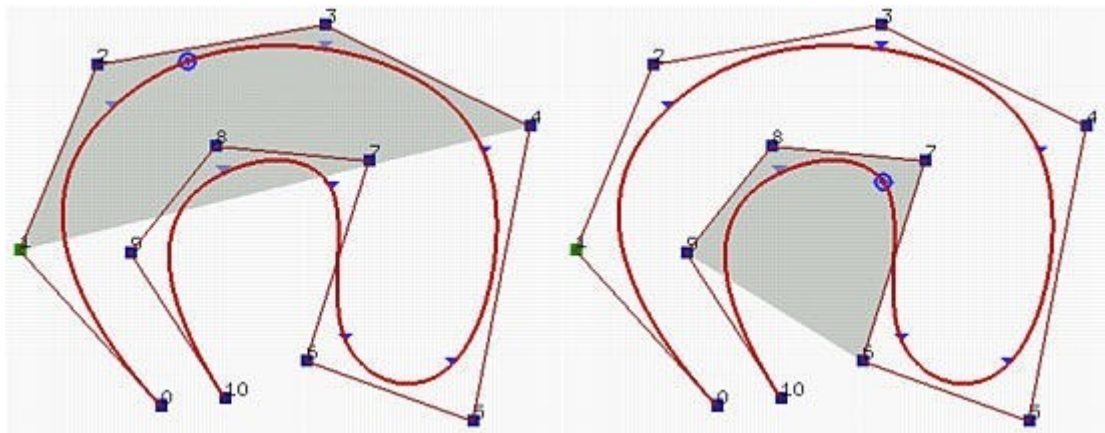
La figura a destra mostra il risultato dello spostamento di P_2 all'angolo destro più basso. Come è possibile vedere, solo il primo, il secondo e il terzo segmento di curva cambiano la loro forma mentre tutti i rimanenti segmenti di curva rimangono nella loro posizione originaria senza alcun cambiamento.



> **Indipendenza grado-punti di controllo.** La proprietà afferma che la curva ha un grado che non dipende dal numero dei punti di controllo. Il grado infatti viene fissato dal designer e corrisponde al parametro p .

> **Strong Convex Hull.** La proprietà garantisce che la curva è contenuta nel dominio convesso del poligono di controllo.

Data la curva B-Spline $C(u)$ con $u \in [u_i, u_{i+1})$, la curva giace nel dominio convesso definito dai punti P_{i-p}, \dots, P_i .



> **Proprietà della derivata**

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

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

Come è possibile vedere l'indice della nostra funzione base è $n+1$, quindi non ci troviamo davanti una vera e propria curva B-Spline anche se di grado inferiore. Se volessimo confrontarla con la derivata di Bèzier:

$$\frac{d}{dt} B_n(t) = n \sum_{k=0}^{n-1} b_{n-1,k}(t) \Delta P_k$$

possiamo vedere che differiscono sia per l'indice della funzione base (in questo caso $n-1$) sia per il modo in cui vengono definiti i punti Q_i e ΔP_k .

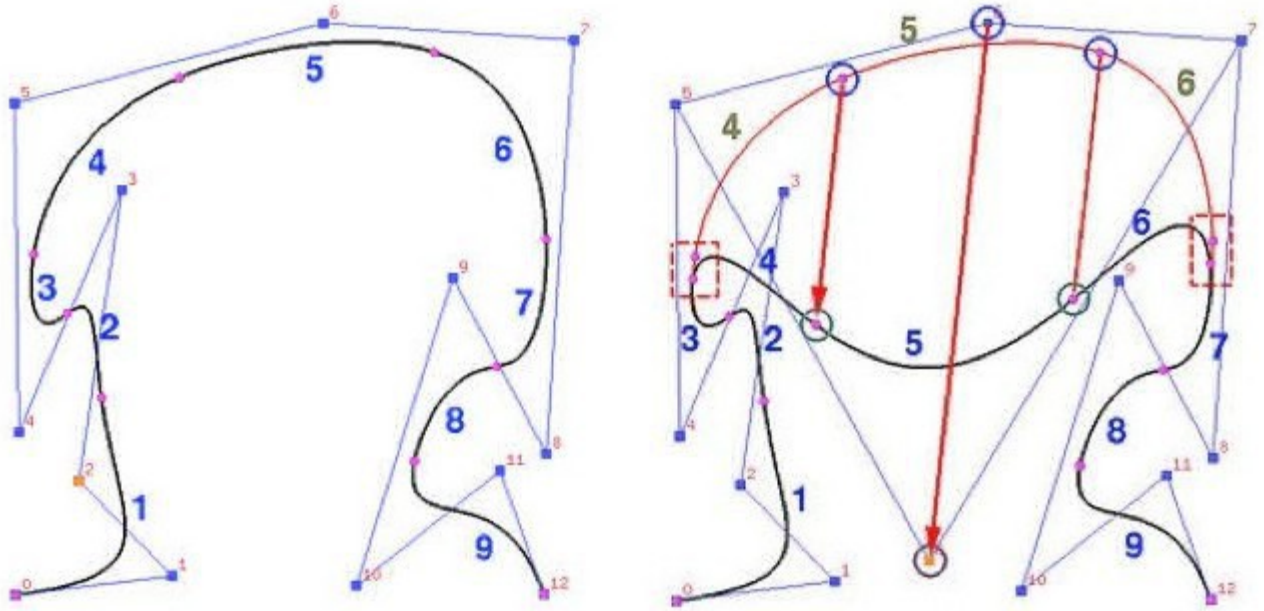
2.5 Curve di Bèzier vs. Curve B-Spline

Analizziamo qui di seguito quali sono i punti che accomunano le curve di Bèzier e le B-Spline e quali invece le differenze:

1. La curva è contenuta all'interno dell'involucro convesso del poligono di controllo?
 - Bèzier: SI
 - B-Spline: SI
2. Come si lega il grado della curva rispetto al poligono di controllo?
 - Bèzier: Grado = numero di vertici - 1
 - B-Spline: Indipendente dal poligono di controllo
3. La curva tocca il poligono di controllo?
 - Bèzier: Sì, nel primo e nell'ultimo punto
 - B-Spline: No, ma può essere fatto, collassando tanti nodi quanto il grado della curva. Operando questa routine, difatti si può farla passare per qualsiasi punto del poligono.
4. La forma della curva e del poligono sono in qualche modo correlate?
 - Bèzier: No, solo in modo approssimato
 - B-Spline: Sì, e lì si può addirittura far coincidere, con particolari accorgimenti
5. Si può avere una trasformazione affine alla curva?
 - Bèzier: Sì, operando sui vertici del poligono di controllo
 - B-Spline: Sì, operando sui vertici del poligono di controllo

6. Si può avere il controllo locale sulla curva?

- Bèzier: NO
- B-Spline: SI



Esempio di controllo locale della curva B-Spline

7. Si possono disegnare cerchi ed ellissi?

- Bèzier: NO
- B-Spline: NO

2.5 Algoritmi delle B-Spline

2.5.1 Knot insertion: Introduzione

L'algoritmo Knot insertion permette di inserire un nuovo nodo nel vettore dei knots senza cambiare la forma della curva; questo nuovo nodo potrà essere uguale ad uno già esistente e, in questo caso, la molteplicità di tale nodo sarà aumentata di 1.

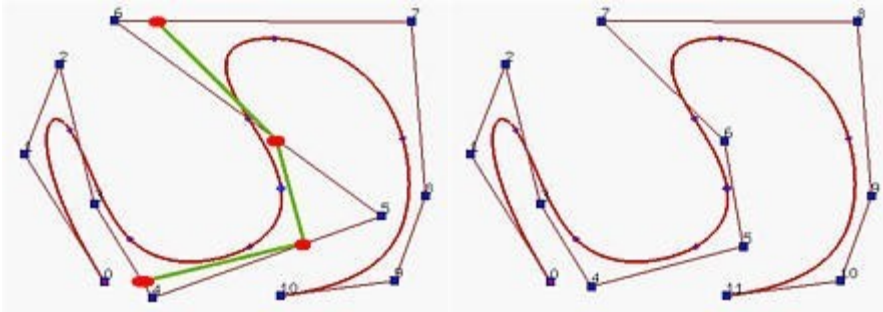
Dall'equazione $m = n + p + 1$ si vede chiaramente che all'aumentare di m segue un aumento di n in quanto il valore di p è fissato. Questo mostra come aumentando di x il numero di knots, aumenta di x anche il numero di punti nel poligono di controllo.

Il poligono di controllo si avvicinerà quindi di più alla curva (ovviamente sempre mantenendo e, anzi, rafforzando la proprietà dello strong convex hull) grazie al "corner cutting".

Esempio:

L'effetto dell'inserimento di un nuovo punto di controllo e il conseguente corner cutting viene mostrato dalle figure sottostanti; dove viene inserito un nuovo nodo $u = 0,5$.

La figura di sinistra mostra anche il poligono di controllo prima e dopo l'inserimento. Come si vede bene, la forma della curva non cambia. Quello che cambia è il poligono di controllo. Infatti 4 nuovi punti (in rosso) rimpiazzano i punti originali P_4 , P_5 e P_6 , e tre segmenti (in giallo) "tagliano" gli angoli (corner cutting) in corrispondenza dei vecchi punti P_4 , P_5 e P_6 .



Si avrà una curva più facilmente modellabile grazie all'aumento dei punti di controllo.

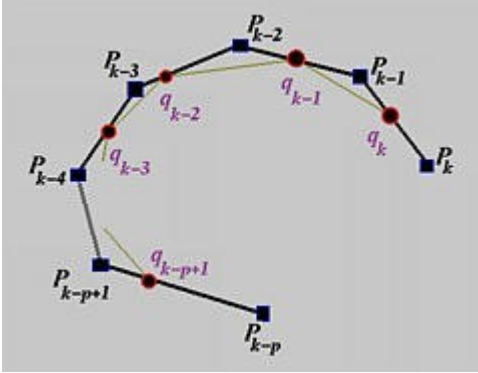
2.5.1.1 Knot Insertion: definizione

Dati $n + 1$ punti di controllo P_0, P_1, \dots, P_n , un vettore di $m + 1$ nodi $U = \{u_0, u_1, \dots, u_m\}$ e un grado p della curva, vogliamo inserire un nuovo nodo t nel vettore dei nodi senza cambiare la forma della curva $C(u)$.

Supponiamo che il nuovo nodo t si trovi nel knot span $[u_k, u_{k+1})$. Dalla proprietà dello strong convex hull $C(t)$ giace nel dominio convesso definito dai punti di controllo $P_k, P_{k-1}, \dots, P_{k-p}$ e le funzioni base degli altri punti di controllo sono uguali a 0 (e quindi nulle). Quindi l'inserimento viene fatto considerando soltanto i punti $P_k, P_{k-1}, \dots, P_{k-p}$.

Per inserire t dobbiamo trovare p nuovi punti di controllo Q_k nel segmento tra P_{k-1} e P_k , Q_{k-1} nel segmento tra P_{k-2} e P_{k-1} , \dots , e Q_{k-p+1} nel segmento tra P_{k-p} e P_{k-p+1} in modo tale da sostituire il vecchio poligono di controllo tra P_{k-p} e P_k (in nero nella figura sottostante) con i nuovi punti $P_{k-p}, Q_{k-p+1}, \dots, Q_k, P_k$ (in arancione) tagliando gli angoli dei punti $P_{k-p+1}, \dots, P_{k-1}$.

Tutti gli altri punti di controllo rimangono invariati.



E' da notare che $p - 1$ punti dell'originale poligono di controllo sono rimossi e rimpiazzati da p nuovi punti di controllo.

Fortunatamente, le posizioni dei nuovi punti di controllo Q_i sono facili da calcolare. La formula per calcolare la posizione del punto Q_i nel segmento tra P_{i-1} e P_i è la seguente:

$$Q_i = (1 - a_i)P_{i-1} + a_iP_i$$

dove il coefficiente a_i è calcolato come segue:

$$a_i = \frac{t - u_i}{u_{i+p} - u_i} \quad \text{for } k - p + 1 \leq i \leq k$$

In definitiva, per inserire un nuovo nodo t , per prima cosa troviamo il knot span $[u_k, u_{k+1})$ che contiene t . Poi vengono calcolati i nuovi punti di controllo Q_{k-p+1}, \dots, Q_k con la formula sopra. Ed infine, l'originale poligono di controllo tra P_{k-p} e P_k viene sostituito con il nuovo poligono di controllo definito da $P_{k-p}, Q_{k-p+1}, Q_{k-p+2}, \dots, Q_{k-1}, Q_k, P_k$.

Notiamo che dopo aver inserito un nuovo nodo, il vettore dei knots diventa $U = \{u_0, u_1, \dots, u_k, t, u_{k+1}, \dots, u_m\}$. Nel caso in cui t sia uguale a u_k , la molteplicità di u_k viene aumentata di 1.

Esempio:

Supponiamo di avere una B-Spline di grado 3 il cui vettore dei nodi è il seguente:

u_0 to u_3	u_4	u_5	u_6	u_7	u_8 to u_{11}
0	0.2	0.4	0.6	0.8	1

Vogliamo inserire un nuovo nodo $t = 0.5$. Dato che t si trova all'interno del knot span $[u_5, u_6)$, i punti di controllo presi in considerazione per il calcolo del nuovo punto sono P_5, P_4, P_3 e P_2 . Dovremo calcolare tre nuovi punti Q_5, Q_4 e Q_3 (2 per sostituire i vecchi P_3 e P_4 e uno per il nuovo punto), dobbiamo calcolare a_5, a_4 e a_3 come segue:

$$a_5 = \frac{t - u_5}{u_8 - u_5} = \frac{0.5 - 0.4}{1 - 0.4} = \frac{1}{6}$$
$$a_4 = \frac{t - u_4}{u_7 - u_4} = \frac{0.5 - 0.2}{0.8 - 0.2} = \frac{1}{2}$$
$$a_3 = \frac{t - u_3}{u_6 - u_3} = \frac{0.5 - 0}{0.6 - 0} = \frac{5}{6}$$

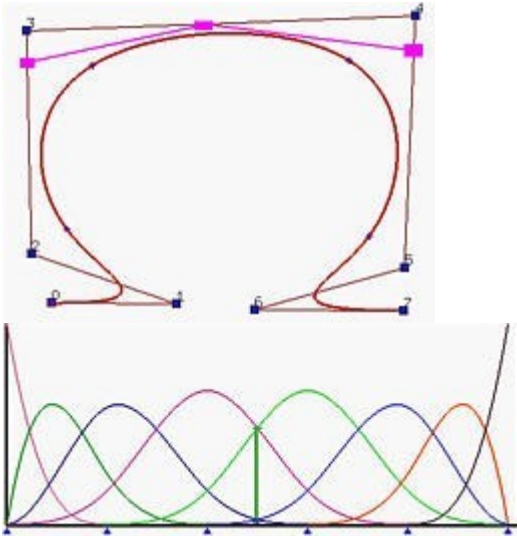
I tre nuovi punti di controllo sono:

$$Q_5 = \left(1 - \frac{1}{6}P_4\right) + \frac{1}{6}P_5$$
$$Q_4 = \left(1 - \frac{1}{2}P_3\right) + \frac{1}{2}P_4$$
$$Q_3 = \left(1 - \frac{5}{6}P_2\right) + \frac{5}{6}P_3$$

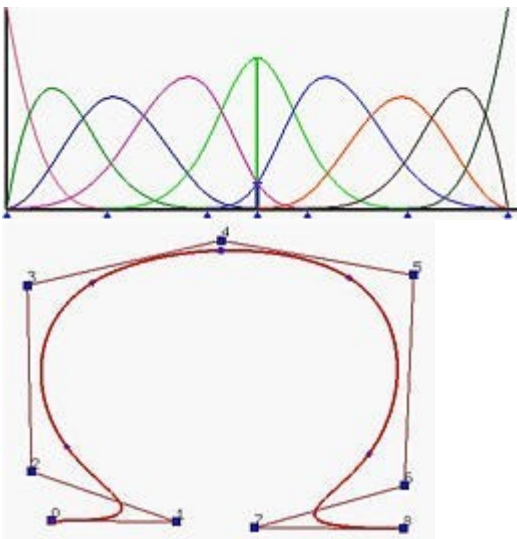
Il nuovo poligono di controllo è composto dai punti $P_0, P_1, P_2, Q_3, Q_4, Q_5, P_5, \dots$, e il nuovo vettore di nodi è:

u_0 to u_3	u_4	u_5	u_6	u_7	u_8	u_9 to u_{12}
0	0.2	0.4	0.5	0.6	0.8	1

La B-Spline originale aveva $p = 3$ e $m = 11$, quindi abbiamo $n = m - p - 1 = 11 - 3 - 1 = 7$. E quindi 8 punti di controllo. Di seguito vediamo una B-Spline che soddisfa tali condizioni e le sue funzioni base:



I tre rettangoli violetti sono Q_3 , Q_4 e Q_5 . Dopo l'inserimento di $t = 0,5$ gli angoli corrispondenti a P_3 e P_4 vengono "tagliati". Nell'immagine seguente vediamo il nuovo poligono di controllo e le nuove funzioni base. E' da notare che la forma della curva, come ci si aspetta, non cambia.



2.5.2 Algoritmo di De Boor

L'algoritmo di De Boor è una generalizzazione dell'algoritmo di De Casteljeau; un modo veloce e numericamente stabile per trovare un punto in una B-Spline dato un u appartenente al dominio ($u \in [0,1)$).

Ricordiamo che aumentando la molteplicità di un knot interno decresce il numero di funzioni base non nulle che attraversano questo knot. Infatti se la molteplicità di questo knot è k , ci sono al più $p - k + 1$ funzioni base non nulle che attraversano questo knot. Questo implica che in un nodo di molteplicità p , ci sarà solo un funzione base (essendo $p - p + 1 = 1$) non nulla il cui valore in corrispondenza di tale knot sarà uguale a 1 per il principio della partizione dell'unità. Chiamiamo questo knot u_i .

Se $u = u_i$, poichè $N_{i,p}(u) \neq 0$ in $[u_i, u_{i+1})$, il punto sulla curva $C(u)$ viene influenzato da esattamente un punto di controllo P_i .

Quindi $C(u) = N_{i,p}(u) P_i = P_i$.

Se un nodo u viene inserito ripetutamente in modo che la sua molteplicità sia p , l'ultimo nuovo punto di controllo generato è il punto della curva che corrisponde ad u .

Dopo aver inserito u più volte (fino a far arrivare la sua molteplicità a p), la curva deve passare per questo nuovo punto, esso è il punto che sulla curva corrisponde ad u .

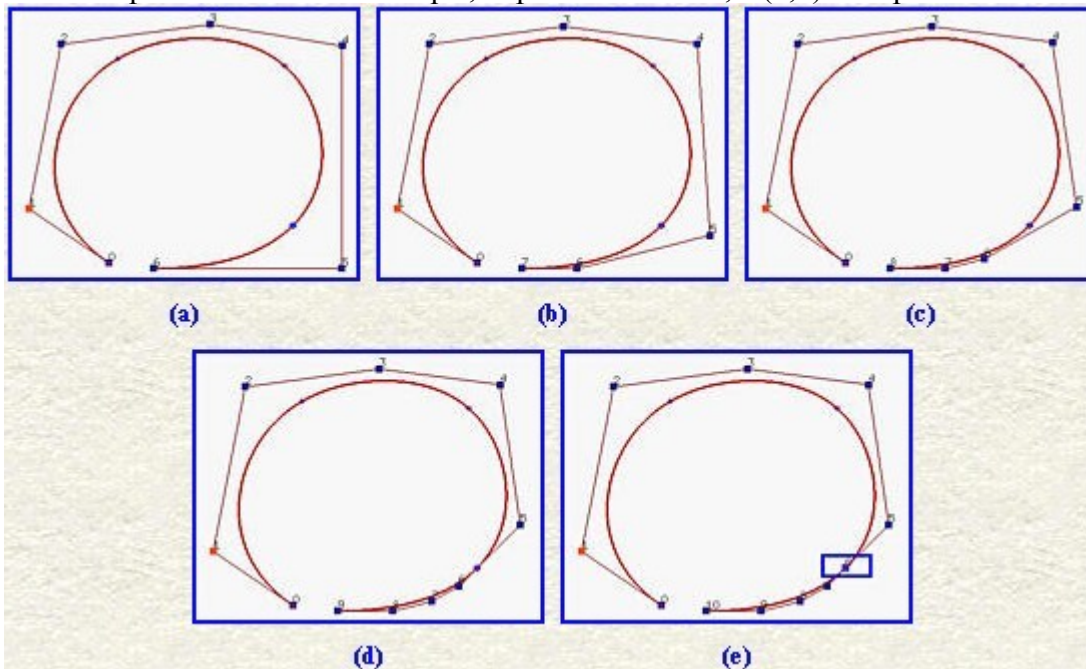
Questa osservazione ci fornisce una **tecnica per trovare $C(u)$** : inseriamo u finchè la sua molteplicità non diventa pari a p e l'ultimo punto è esattamente $C(u)$!

L'algorithmo di inserimento dei knots può essere facilmente modificato per soddisfare il nostro scopo.

Dobbiamo inserire il nodo u solo il numero di volte necessario per farlo diventare un nodo di molteplicità p . Se u è un nodo già esistente di molteplicità s allora sarà sufficiente inserirlo $p - s$ volte.

Vediamo un esempio:

La figura (a) è una B-Spline di grado 4 definita da 7 punti di controllo. Per calcolare $C(0,9)$, dove 0,9 non è un nodo, $u = 0,9$ viene inserito 4 (il grado) volte. Le figure (b) e (c) mostrano il risultato dopo il primo e il secondo inserimento. Quindi 2 nuovi punti di controllo sono inseriti nella zona inferiore destra della figura. Notiamo come ora il poligono di controllo sia molto più vicino alla curva rispetto all'originale. Il risultato del terzo inserimento è mostrato in figura (d). Il quarto inserimento (figura (e)) "trova" il punto nella curva. Dunque, dopo 4 inserimenti, $C(0,9)$ è un punto di controllo.



2.5.2.1 Algoritmo di De Boor in dettaglio

Input: un valore u (di solito $0 \leq u \leq 1$)

Output: un valore $C(u)$ (il valore della curva in u)

if u non è un nodo già esistente then

$h = p; s = 0; //$ (inseriamo u esattamente p volte)

else

$h = p - s; //$ (inseriamo u $p - s$ volte, dove s è la molteplicità del nodo già esistente)

copiamo i punti di controllo che saranno influenzati dall' algoritmo $P_{k-s}, P_{k-s-1}, P_{k-s-2}, \dots, P_{k-p+1}$ e P_{k-p} in un nuovo array e li rinominiamo come: $P_{k-s,0}, P_{k-s-1,0}, P_{k-s-2,0}, \dots, P_{k-p+1,0}, P_{k-p,0}$ dove lo 0 indica lo step iniziale (che ovviamente crescerà ad ogni passo dell' algoritmo);

for r from 1 to h do

for i from $k - p + r$ to $k - s$ do

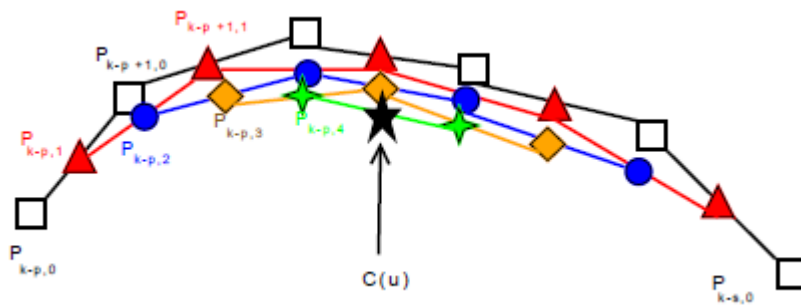
$$a_{i,r} = \frac{u - u_i}{u_{i+p-r+1} - u_i}$$

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

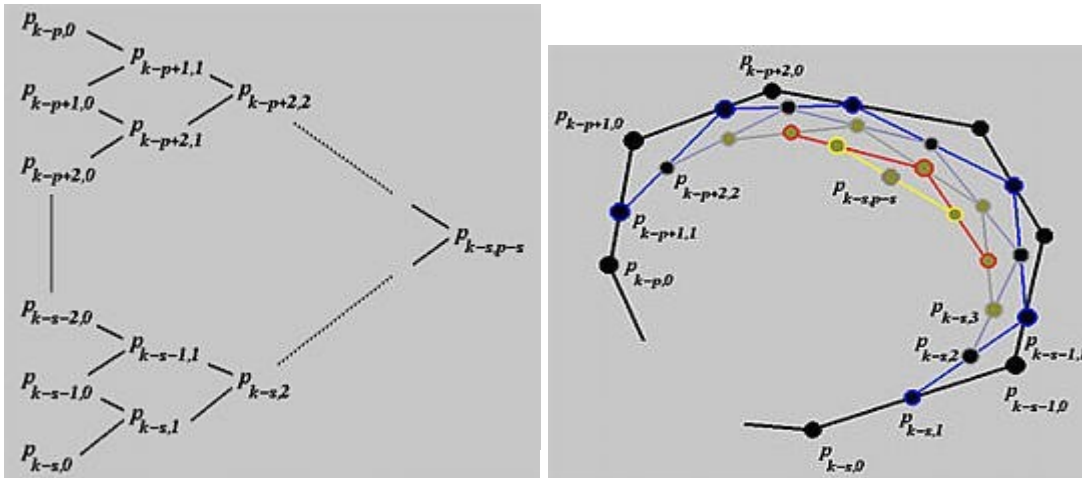
end for

end for

$P_{k-s,p-s}$ è il punto $C(u)$.



Nel diagramma sinistro delle figure sottostanti, tutti i punti $P_{i,0}$ sono sulla prima colonna. Da questa colonna (colonna degli 0) e dai coefficienti $a_{i,1}$, si può calcolare la colonna dei $P_{i,1}$ (la seconda colonna). Dalla seconda colonna e dai coefficienti $a_{i,2}$ si calcola la terza colonna e così via. Dato che nella prima colonna (colonna degli 0) ci sono $(k - s) - (k - p) = p - s + 1$ punti, e, dato che ogni colonna ha un punto in meno della colonna precedente, servono $p - s$ colonne per ridurre il numero di punti in una colonna ad 1. Questo spiega perché l'ultimo punto è $P_{k-s,p-s}$. Il diagramma di sinistra mostra graficamente cosa succede al poligono di controllo durante l' algoritmo mettendo in evidenza il processo di knot insertion e corner cutting.



Considerazioni:

A differenza dell'algoritmo di Casteljau nel quale la dipendenza era solo sull'indice i , in De Boor il punto di controllo $P_{i,r}$ dipende sia dall'indice i che dal contatore r il quale indica il numero di iterazioni necessarie a portare u ad avere molteplicità p .

L'algoritmo di De Boor si presenta:

- *stabile*: la stabilità permette di non amplificare l'errore di elaborazione, dato che si lavora con una combinazione lineare positiva di somma 1.

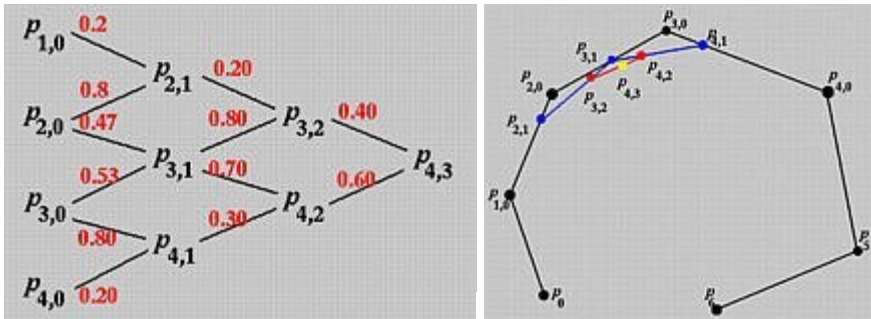
- *ricorsivo*: la ricorsività implica la riapplicazione dell'algoritmo a se stesso. Ovviamente l'algoritmo non modifica in alcun modo la forma della curva.

ESEMPIO:

Supponiamo di avere una B-Spline di grado 3 ($p = 3$) definita da 7 punti di controllo P_0, \dots, P_6 ($n = 6$) e il seguente vettore di 11 nodi ($m = 11$):

$u_0 = u_1 = u_2 = u_3$	u_4	u_5	u_6	$u_7 = u_8 = u_9 = u_{10}$
0	0.25	0.5	0.75	1

Basandoci su queste informazioni dovremmo calcolare $C(0,4)$. Per l'algoritmo di De Boor questo equivale ad inserire un nodo $u = 0,4$ esattamente 3 volte. Dato che $u = 0,4$ non è un nodo già esistente ($s = 0$) e u si trova in $[u_4, u_5)$, i punti di controllo influenzati dall'algoritmo saranno P_4, P_3, P_2 e P_1 . Il seguente è uno schema rappresentativo dello svolgimento dell'algoritmo:



Calcoliamo la seconda colonna. I coefficienti a coinvolti saranno:

$$a_{4,1} = \frac{u - u_4}{u_{4+3} - u_4} = 0.2$$

$$a_{3,1} = \frac{u - u_3}{u_{3+3} - u_3} = 8/15 = 0.53$$

$$a_{2,1} = \frac{u - u_2}{u_{2+3} - u_2} = 0.8$$

La prima colonna viene calcolata come segue:

$$\mathbf{P}_{4,1} = (1 - a_{4,1})\mathbf{P}_{3,0} + a_{4,1}\mathbf{P}_{4,0} = 0.8\mathbf{P}_{3,0} + 0.2\mathbf{P}_{4,0}$$

$$\mathbf{P}_{3,1} = (1 - a_{3,1})\mathbf{P}_{2,0} + a_{3,1}\mathbf{P}_{3,0} = 0.47\mathbf{P}_{2,0} + 0.53\mathbf{P}_{3,0}$$

$$\mathbf{P}_{2,1} = (1 - a_{2,1})\mathbf{P}_{1,0} + a_{2,1}\mathbf{P}_{2,0} = 0.2\mathbf{P}_{1,0} + 0.8\mathbf{P}_{2,0}$$

Per calcolare la seconda colonna abbiamo bisogno dei seguenti coefficienti:

$$a_{4,2} = \frac{u - u_4}{u_{4+3-1} - u_4} = 0.3$$

$$a_{3,2} = \frac{u - u_3}{u_{3+3-1} - u_3} = 0.8$$

I punti sono:

$$\mathbf{P}_{4,2} = (1 - a_{4,2})\mathbf{P}_{3,1} + a_{4,2}\mathbf{P}_{4,1} = 0.7\mathbf{P}_{3,1} + 0.3\mathbf{P}_{4,1}$$

$$\mathbf{P}_{3,2} = (1 - a_{3,2})\mathbf{P}_{2,1} + a_{3,2}\mathbf{P}_{3,1} = 0.2\mathbf{P}_{2,1} + 0.8\mathbf{P}_{3,1}$$

Il punto finale è:

$$\mathbf{P}_{4,3} = (1 - a_{4,3})\mathbf{P}_{3,2} + a_{4,3}\mathbf{P}_{4,2} = 0.4\mathbf{P}_{3,2} + 0.6\mathbf{P}_{4,2}$$

Questo è il punto della curva corrispondente a $u = 0,4$.

Osservazioni:

Notiamo la somiglianza tra i poligoni di controllo intermedi generati dall' algoritmo di De Boor e quelli generati dall' algoritmo di De Casteljeau. È da notare anche che u non è un nodo in questo esempio; se lo fosse stato, il calcolo sarebbe stato molto più semplice, dato che avremmo avuto un numero inferiore di punti di controllo da prendere in considerazione per il calcolo.

2.5.3 Algoritmo di suddivisione per le B-Spline

Suddividere una curva B-Spline richiede esattamente lo stesso processo che si utilizza per le curve di bezier, con l'evidente differenza che si utilizza l' algoritmo di De Boor invece di quello di De Casteljeau.

• Selezione dei punti di controllo:

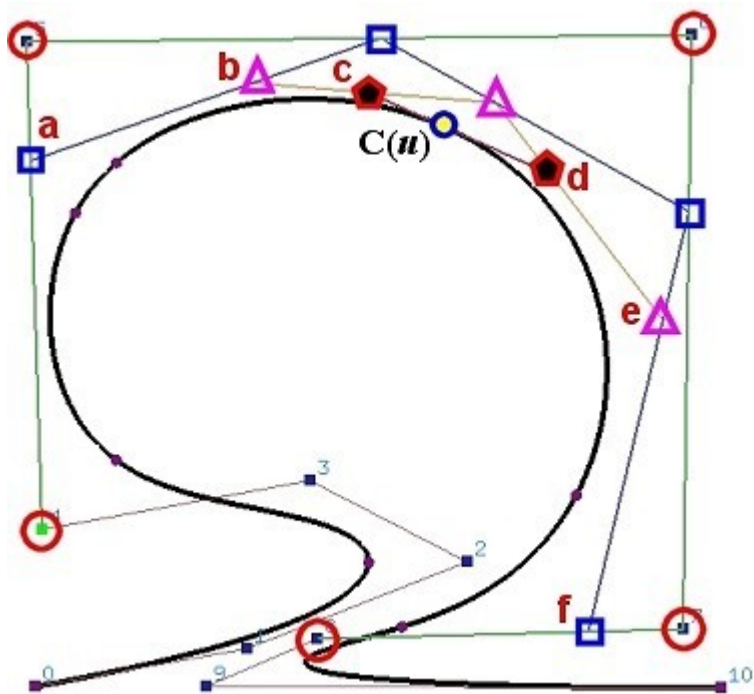
Supponiamo di voler suddividere una B-Spline in u in due curve, una in $[0, u]$ e l'altra in $[u, 1]$. La prima cosa da fare è applicare l' algoritmo di De Boor in u . Notiamo che se u è in $[u_k, u_{k+1})$, allora al massimo $p+1$ punti di controllo saranno usati nell' algoritmo, precisamente: $\mathbf{P}_k, \mathbf{P}_{k-1}, \mathbf{P}_{k-2}, \dots, \mathbf{P}_{k-p}$. Per trovare i punti di controllo della B-Spline in $[0, u]$ iniziamo con \mathbf{P}_0 e seguiamo il poligono di controllo.

Quando incontriamo il primo dei punti di controllo generati dal primo step dell' algoritmo di De Boor lo selezioniamo come il primo dei nuovi punti di controllo.

Continuiamo cercando il primo punto di controllo generato dalla seconda iterazione dell' algoritmo di de Boor e lo selezioniamo come secondo dei nuovi punti.

Continuiamo in questo modo fino a raggiungere $\mathbf{C}(u)$ che selezioniamo come ultimo punto della curva in $[0, u]$. Un esempio di questo procedimento sono i punti **a, b, c e $\mathbf{C}(u)$** nella figura sottostante che rappresenta una B-Spline di grado 3 con 11 ($n = 10$) punti di controllo .

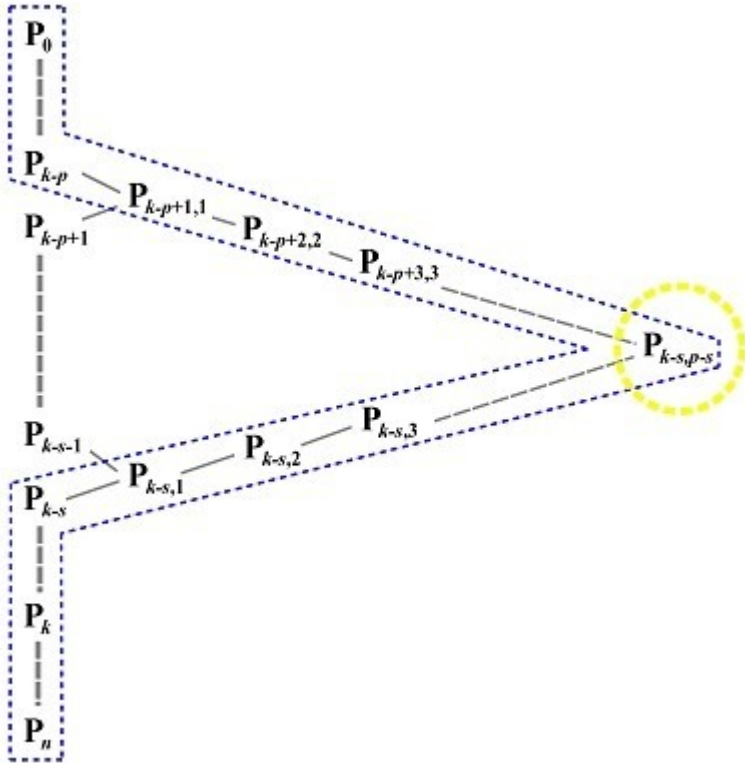
Per la curva in $[u, 1]$ applichiamo lo stesso processo ma selezionando come primo punto della curva $\mathbf{C}(u)$, poi il punto di controllo generato dalla penultima iterazione dell' algoritmo di De Boor e così via fino a quando ritroviamo un punto dell' originale poligono di controllo. Ancora una volta vediamo nella figura seguente come i punti selezionati siano in ordine $\mathbf{C}(u), d, e, f$.



Per capire meglio osserviamo come lo svolgimento dell'algorithm di de Boor formi una "rete" di De Boor (molto evidente nell'immagine precedente).

I Punti di controllo della prima delle due nuove curve (quella in $[0, u]$) sono i primi punti non influenzati dall'algorithm di De Boor, più il **primo** punto di controllo di ogni poligono di controllo della rete di De Boor. Similmente i punti di controllo della seconda delle due curve ($[u, 0]$) sono gli **ultimi** punti di controllo di ogni poligono di controllo della rete di De Boor, più gli ultimi punti del poligono originale che non sono stati influenzati dall'algorithm.

Per una interpretazione leggermente più matematica utilizziamo il seguente schema:



N.B. Se avessimo costruito uno sche ma del genere per la curva nell'immagine precedente, i punti evidenziati con dei cerchi nell'immagine sarebbero quelli nella prima colonna (colonna degli zeri), quelli nella seconda colonna sarebbero evidenziati con i quadratini, nella terza colonna i triangoli, nella quarta i pentagoni e nell'ultima $C(\mathbf{u})$. Questo per far vedere la corrispondenza tra lo schema triangolare e la rete di De Boor.

Dato che i punti di un poligono di controllo nella rete di De Boor sono il primo e l'ultimo della stessa colonna, essi possono essere estratti una volta calcolata un'intera colonna.

Nel diagramma precedente possiamo immediatamente estrarre i punti da \mathbf{P}_0 a \mathbf{P}_{k-p} per la prima curva ($[0, \mathbf{u}]$) e i punti da \mathbf{P}_{k-s} a \mathbf{P}_n per la seconda curva ($[\mathbf{u}, 0]$).

una volta calcolata la prima colonna, possiamo estrarre $\mathbf{P}_{k-p+1,1}$ (prima curva) e $\mathbf{P}_{k-s,1}$ (seconda).

Dopo aver calcolato la seconda colonna passiamo alla terza estraendo $\mathbf{P}_{k-p+2,2}$ (prima curva) e $\mathbf{P}_{k-s,2}$ (seconda). Continuiamo con questo processo fino a raggiungere $\mathbf{P}_{k-s,p-s} = C(\mathbf{u})$ che sarà selezionato per entrambe le curve. Graficamente vediamo che i punti di controllo delle nuove curve sono quelli circondati dalla linea tratteggiata blu, da \mathbf{P}_0 a $\mathbf{P}_{k-s,p-s}$ per la prima curva e da $\mathbf{P}_{k-s,p-s}$ a \mathbf{P}_n per la seconda.

- **Selezione dei knots:**

Il vettore dei nodi della curva in $[0, \mathbf{u}]$ includerà tutti i knots (della curva originale) in $[0, \mathbf{u}]$ seguiti da $p+1$ copie di \mathbf{u} , mentre il vettore dei nodi della curva in $[\mathbf{u}, 0]$ sarà formato da $p+1$ copie di \mathbf{u} seguite da tutti i nodi (della curva originale) in $(\mathbf{u}, 1]$.

Per esempio: l'originale vettore dei nodi di una B-Spline di grado 4 è $\{0, 0, 0, 0, 0, 0.3, 0.4, 0.6, 0.7, 0.85, 0.9, 1, 1, 1, 1, 1\}$.

Se la curva è suddivisa in due curve in $u = 0.65$, la curva in $[0, 0.65]$ avrà il seguente vettore dei nodi: $\{0, 0, 0, 0, 0, 0.3, 0.4, 0.6, 0.65, 0.65, 0.65, 0.65, 0.65\}$; mentre la curva in $[0.65, 1]$ avrà come vettore dei nodi $\{0.65, 0.65, 0.65, 0.65, 0.65, 0.7, 0.85, 0.9, 1, 1, 1, 1\}$.

Può essere dimostrato che le curve componenti costruite in questo modo sono anch'esse delle curve B-Spline con grado uguale a quello della curva originale e sono i due bracci della curva originale rispettivamente in $[0, u]$ e $[u, 1]$. Dunque lasciano la forma della curva assolutamente invariata.

3.0 Coordinate omogenee e trasformazioni geometriche

3.0.1 Coordinate omogenee

Uno dei vantaggi di usare le coordinate omogenee è di poter estendere le coordinate euclidee aggiungendo un concetto fondamentale: quello dell'infinito.

Utilizzando il concetto di infinito molti concetti e calcoli geometrici possono essere molto semplificati.

Con le coordinate euclidee non si può rappresentare un punto all'infinito; con quelle omogenee si può.

Come già detto, l'idea è quella di estendere il concetto classico di coordinate:

Nello spazio ad UNA sola dimensione consideriamo la coppia di valori (a, w) come il punto della retta \mathbf{R} indicato da a / w (con $a \in \mathbf{R}$).

Se $w \neq 0$, il punto indicato avrà valore esattamente a / w .

Se $w = 0$ allora il punto che stiamo considerando sarà all'infinito.

Questo estende concettualmente la trattazione dello spazio di riferimento \mathbf{R} .

Nel caso in 2 dimensioni se in coordinate omogenee abbiamo una tripla $(3, 4, 5)$ dove $x = 3$, $y = 4$, $w = 5$: essa è un punto espresso in coordinate omogenee in due dimensioni. Ebbene per trasformare in coordinate euclidee basterà dividere le prime 2 coordinate del punto per la terza ovvero divideremo x ed y per il valore di w . Dunque il punto espresso in coordinate euclidee sarà quello di coordinate $((3 / 5), (4 / 5))$.

Al contrario quale sarà il corrispondente punto di coordinate euclidee (x, y) espresso in coordinate omogenee?

Sarà il punto di coordinate omogenee $(x, y, 1)$! Cioè la componente w è uguale a 1. Però in realtà la risposta NON è unica! Infatti la coordinata omogenea di un punto euclideo (x, y) è (xw, yw, w) per ogni $w \neq 0$.

Basta infatti vedere che riconvertendo, le coordinate euclidee risultano essere (x, y) come ci si aspettava.

Quindi: **la conversione da coordinate omogenee ad euclidee è unica, mentre la conversione da euclidee ad omogenee non lo è!**

Prendiamo in considerazione una funzione $f(x, y) = 0$ se rimpiazziamo x ed y con i valori x / w e y / w la funzione diventerà $f(x / w, y / w) = 0$. Se la funzione è polinomiale, moltiplicandola per w^n (dove n è il grado del polinomio) tutti i denominatori diventeranno uguali a 1 (in pratica scompariranno).

Ad esempio: supponiamo di avere una retta $Ax + By + C = 0$, in coordinate omogenee la coppia (x, y) diventa $(x / w), (y / w)$ e quindi:

$$A(x / w) + B(y / w) + C = 0,$$

Moltiplicando per w avremo:

$$Ax + By + Cw = 0.$$

Sia l'equazione data un polinomio di grado 2 $Ax^2 + 2Bxy + Cy^2 + 2Dx + 2Ey + F = 0$.

Dopo aver sostituito x ed y con x / w e y / w e dopo aver moltiplicato l'espressione per w^2

avremo:

$$Ax^2 + 2Bxy + Cy^2 + 2Dxw + 2Eyw + Fw^2 = 0.$$

Se guardiamo attentamente questi 2 polinomi vedremo che il grado di ognuno dei termini è uguale.

Nel caso di una retta, i termini x , y e w sono di grado 1, mentre nel caso del polinomio di grado 2, tutti i termini (x^2 , xy , xw , y^2 , yw e w^2) sono di grado 2.

Dato un polinomio di grado n , dopo aver introdotto w , tutti i termini diventano di grado n .

Di conseguenza, questi polinomi vengono chiamati **polinomi omogenei** e le coordinate (x, y, w) vengono chiamate **coordinate omogenee**.

Dato un polinomio di grado n nel sistema di coordinate omogenee, dividendolo per w^n e rimpiazzando x/w e y/w con x ed y rispettivamente avremo il polinomio convertito da omogeneo a convenzionale.

Per esempio, se il polinomio omogeneo di grado 3 è il seguente:

$$x^3 + 3xy^2 - 5y^2w + 10w^3 = 0$$

Il risultato è:

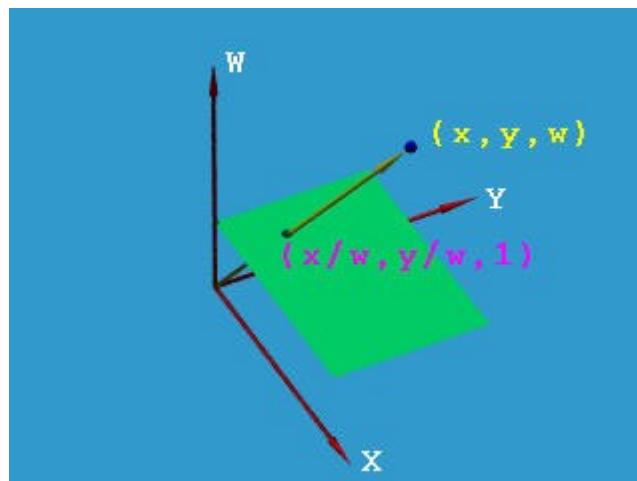
$$x^3 + 3xy^2 - 5y^2 + 10 = 0$$

Questo procedimento funziona anche per lo spazio a 3 dimensioni. Si può rimpiazzare un punto (x, y, z) con $(x/w, y/w, z/w)$ e moltiplicare il risultato per w elevato ad una certa potenza. Il polinomio risultante è un **polinomio omogeneo**. Convertire un polinomio omogeneo di grado n in x, y, z e w di nuovo nella forma convenzionale è esattamente identico al caso in 2 dimensioni.

Una semplice interpretazione geometrica:

Data una coordinata omogenea (x, y, w) di un punto nel piano euclideo, consideriamo (x, y, w) come un punto nello spazio a 3 dimensioni le cui coordinate sono appunto x, y e w rispettivamente per gli assi X, Y e W .

La retta che passa per questo punto e l'origine degli assi interseca il piano di equazione $w = 1$ nel punto di coordinate $(x/w, y/w, 1)$ come illustrato nella seguente figura.



Questa trasformazione tratta un punto in 2 dimensioni in coordinate omogenee come un punto

in 3 dimensioni e proietta (dall'origine degli assi) tale punto in 3 dimensioni nel piano $w = 1$. La figura illustra chiaramente anche il concetto espresso in precedenza che, mentre la conversione da coordinate euclidee a omogenee è unica, la conversione inversa non lo è perché tutti i punti della retta passante per l'origine e il punto (x,y,w) saranno proiettati nel punto $(x/w, y/w, 1)$.

3.0.2 Trasformazioni geometriche

Dividiamo le trasformazioni geometriche da noi studiate in:

- trasformazioni euclidee:
 - . traslazioni
 - . rotazioni
- trasformazioni affini:
 - . scaling
 - . shear
- trasformazioni proiettive.

3.0.2.1 Trasformazioni euclidee

Sono quelle usate più comunemente.

Lasciano invariate la lunghezza e gli angoli; quindi la forma degli oggetti non cambia (gli oggetti vengono soltanto “spostati”).

- Traslazioni: si mettono in pratica traslando un punto (x,y) del piano in un nuovo punto aggiungendo un vettore $\langle h,k \rangle$.

E' facile capire che la relazione tra il vecchio punto (x,y) e quello nuovo (x',y') sarà:

$$x' = x + h \text{ e } y' = y + k.$$

Utilizziamo una forma di rappresentazione simile alle coordinate omogenee: un punto diventa un vettore con una terza componente uguale ad 1.

Dunque il punto (x,y) diventa il seguente:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

possiamo dunque mettere la relazione tra il punto (x,y) ed il punto (x',y') nella forma matriciale nel seguente modo:

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

Possiamo dunque riportare il punto (x',y') in coordinate euclidee ed avremo eseguito la traslazione di un punto.

Si capisce facilmente che per traslare un intero oggetto basta applicare ad ogni suo punto la trasformazione che abbiamo appena eseguito per un solo punto.

- Rotazioni: per ruotare un punto (x,y) di un angolo a rispetto all'origine e farlo diventare un nuovo punto (x',y') si applica la seguente trasformazione:

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

Ed ovviamente si riporta il punto in coordinate euclidee.

Chiaramente le espressioni in forma matriciale sono più compatte e possono includere più istruzioni. Ad esempio si possono combinare insieme una rotazione e una traslazione in un'unica forma matriciale nel seguente modo:

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

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

L'espressione di cui sopra esprime una rotazione seguita da una traslazione. Se si volesse eseguire prima la traslazione e poi la rotazione si dovrebbe applicare la seguente trasformazione:

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

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

Dunque traslazione e rotazione NON sono commutative!

Traslazioni e rotazioni in 3 dimensioni:

La traslazione nello spazio a 3 dimensioni è simile alla versione nel piano:

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & p \\ 0 & 1 & 0 & q \\ 0 & 0 & 1 & r \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & -p \\ 0 & 1 & 0 & -q \\ 0 & 0 & 1 & -r \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix}$$

La formula sopra trasla i punti aggiungendo un vettore $\langle p, q, r \rangle$.

La rotazione nello spazio è più complessa di quella nel piano perchè si può ruotare rispetto all'asse **X**, all'asse **Y** o all'asse **Z**. Quando si ruota rispetto all'asse **Z** soltanto le coordinate della **x** e della **y** cambieranno, mentre la coordinata **z** rimarrà inalterata. In effetti il risultato sarà esattamente una rotazione rispetto all'origine nel piano **XY**. Quindi, l'equazione della rotazione nello spazio rispetto all'asse **Z** sarà la seguente:

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos a & -\sin a & 0 & 0 \\ \sin a & \cos a & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} \cos a & \sin a & 0 & 0 \\ -\sin a & \cos a & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix}$$

Con queste equazioni: sia **a** un angolo di 90°; il punto (1,0,0) ruoterà in (0,1,0); mentre il punto (0,1,0) ruoterà in (-1,0,0). Dunque l'asse **X** ruoterà nell'asse **Y** e l'asse **Y** ruoterà nella direzione negativa dell'originale asse **X**.

Questo è l'effetto di una rotazione di 90° rispetto all'asse **Z**.

Basandoci sulla stessa idea, ruotare un punto (x,y,z) di un angolo **a** rispetto all'asse **X** sarà:

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos a & -\sin a & 0 \\ 0 & \sin a & \cos a & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos a & \sin a & 0 \\ 0 & -\sin a & \cos a & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix}$$

Verifichiamo nuovamente con $a = 90^\circ$. $(0,1,0)$ ruoterà in $(0,0,1)$ e $(0,0,1)$ ruoterà in $(0,-1,0)$.
Nuovamente vediamo che l'asse Y ruoterà nell'asse Z e l'asse Z ruoterà nella direzione negativa dell'originale asse Y .

Ruotare rispetto all'asse Y è un po' diverso a causa del fatto che tradizionalmente gli angoli rispetto all'asse Y si misurano partendo dall'asse X andando verso l'asse Z (che è il contrario di quello che si fa con gli altri 2 assi). Quindi l'equazione della rotazione di un punto rispetto all'asse Y di un angolo a è la seguente:

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos a & 0 & \sin a & 0 \\ 0 & 1 & 0 & 0 \\ -\sin a & 0 & \cos a & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} \cos a & 0 & -\sin a & 0 \\ 0 & 1 & 0 & 0 \\ \sin a & 0 & \cos a & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix}$$

Di nuovo con un angolo di 90° $(1,0,0)$ ruota in $(0,0,-1)$ e $(0,0,1)$ in $(1,0,0)$. Quindi l'asse X ruota nella direzione dell'asse Z originale, mentre l'asse Z ruota nell'asse X originale.

3.0.2.2 Trasformazioni affini

Le trasformazioni affini sono generalizzazioni delle trasformazioni euclidee.

Con le trasformazioni affini le linee; ma i cerchi diventano ellissi. La lunghezza e gli angoli non vengono preservati.

Quindi le forme degli oggetti cambiano!

- Scaling: Lo scaling “espande” o “contrae” un dato oggetto e cambia la lunghezza e gli angoli.

L'obiettivo dello scaling è quello di creare una nuova scala di una coordinata p volte più grande.

In altre parole, la coordinata x viene “ingrandito” (o rimpicciolito) di p volte.

Questo requisito soddisfa l'equazione $x' = px$ e quindi anche $x = x' / p$.

Lo scaling può essere applicato ad ogni asse ognuno con un fattore di scaling diverso.

Per esempio se l'asse X , Y e Z vengono modificati ognuno di un fattore rispettivamente p , q ed r , la matrice di scaling sarà la seguente:

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} p & 0 & 0 & 0 \\ 0 & q & 0 & 0 \\ 0 & 0 & r & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} 1/p & 0 & 0 & 0 \\ 0 & 1/q & 0 & 0 \\ 0 & 0 & 1/r & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix}$$

- Shear: L'effetto di una trasformazione shear dà l'idea di aver “spinto” un oggetto geometrico nella direzione parallela ad uno dei piani (3D) o ad un asse (2D).

Nella figura sottostante il cilindro rosso è il risultato di uno shear applicato al cilindro giallo:



Quanto “lontano” l'oggetto venga “spinto” viene deciso da un fattore di shear.

Nel piano si può spingere nella direzione x (positiva o negativa) e lasciare la direzione y invariata (o viceversa).

La seguente matrice è quella che si usa per una trasformazione shear nella direzione x con un fattore di shear pari ad a :

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & a & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & -a & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$$

mentre la seguente è uno shear in direzione y con fattore pari a b :

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ b & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ -b & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$$

In 3 dimensioni si può spingere in 2 direzioni lasciando la terza invariata.

La matrice per uno shear nelle direzioni x ed y con fattori rispettivamente a e b che lasci la direzione z invariata è la seguente:

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & a & 0 \\ 0 & 1 & b & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -a & 0 \\ 0 & 1 & -b & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix}$$

Dunque le equazioni risultanti dalla matrice sono le seguenti:

$$x' = x + az$$

$$y' = y + bz$$

$$z' = z$$

Trasformazione shear nelle direzioni x e z :

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & a & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & c & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \qquad \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & -a & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -c & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix}$$

Trasformazione shear nelle direzioni y e z :

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ b & 1 & 0 & 0 \\ c & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \qquad \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -b & 1 & 0 & 0 \\ -c & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix}$$

3.0.2.3 Trasformazioni proiettive

Sono le più generali e richiedono l'uso delle coordinate omogenee.

Dato un punto nello spazio di coordinate omogenee (x,y,z,w) e la sua immagine dopo una trasformazione proiettiva (x',y',z',w') , una trasformazione proiettiva ha la seguente forma:

$$\begin{bmatrix} x' \\ y' \\ z' \\ w' \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \\ p_{41} & p_{42} & p_{43} & p_{44} \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \qquad \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \\ p_{41} & p_{42} & p_{43} & p_{44} \end{bmatrix}^{-1} \cdot \begin{bmatrix} x' \\ y' \\ z' \\ w' \end{bmatrix}$$

Questa matrice 4×4 deve essere invertibile. Dunque: le trasformazioni proiettive sono più generali di quelle affini dato che l'ultima riga non deve avere per forza valori $0 \ 0 \ 0 \ 1$.

Con le trasformazioni affini si possono portare punti finiti nell'infinito e viceversa.

Esempio:

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

Questa matrice manda il punto $(x,y,w) = (1,0,1)$ nel punto $(x',y',w') = (1,-1,0)$; che in coordinate euclidee vuol dire mandare il punto $(1,0)$ nel piano all'infinito nella direzione $\langle 1,-1 \rangle$.

Le trasformazioni proiettive hanno la stessa capacità operativa delle altre trasformazioni con l'aggiunta del cambiamento di forma (es. un cerchio diventa un'ellissi e viceversa).

3.0.3 Comodità delle moltiplicazioni e trasformazioni nella forma matriciale

Abbiamo introdotto diverse trasformazioni geometriche, mostrando sempre 2 forme; una da x a x' e l'inversa da x' a x . In molti casi, si può aver bisogno di molte trasformazioni per portare l'oggetto dallo stato attuale a quello desiderato. Per esempio, poniamo di aver bisogno di una matrice di trasformazione A che porti il punto p in q ($q = Ap$) seguita da una seconda matrice B che porti q in r ($r = Bq$); seguita da un'altra ancora che porti r in s ($s = Cr$).

L'effetto $p \rightarrow q \rightarrow r \rightarrow s$ può essere sintetizzato in una singola trasformazione rappresentato dal prodotto di tutte le matrici coinvolte nel processo.

Notiamo che la prima (rispettivamente l'ultima) matrice di trasformazione è quella più a destra (rispettivamente sinistra) nella sequenza di moltiplicazione.

$$s = Cr = C(Bq) = CBq = CB(Ap) = CBAp$$

Quindi, per calcolare l'effetto risultante, calcoliamo solamente CBA e usiamo la matrice risultante come una singola matrice di trasformazione che porta p direttamente in s .

Vediamo un esempio:

Vogliamo applicare la seguente trasformazione ad un oggetto:

1. scalare nella direzione x con un fattore di scale uguale a 5 (ingrandire di 5 volte)
2. ruotare rispetto all'asse z di 30°
3. fare uno shear nelle direzioni x e y rispettivamente di un fattore 2 e 3
4. traslare di un vettore $\langle 2,1,2 \rangle$.

Siano le matrici di scaling, rotazione, shearing e trslazione rispettivamente A , B , C e D .

Riconducendoci al discorso fatto precedentemente avremo la seguente matrice $H = DCBA$:

$$A = \begin{bmatrix} 5 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad B = \begin{bmatrix} \sqrt{3}/2 & -1/2 & 0 & 0 \\ 1/2 & \sqrt{3}/2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad C = \begin{bmatrix} 1 & 0 & 2 & 0 \\ 0 & 1 & 3 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad D = \begin{bmatrix} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H = DCBA = \begin{bmatrix} 5\sqrt{3}/2 & -1/2 & 2 & 2 \\ 5/2 & \sqrt{3}/2 & 3 & 1 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

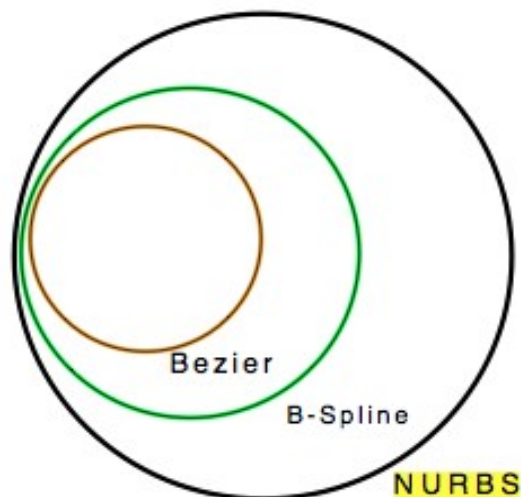
Quindi, l'effetto della trasformazione di un punto x dell'oggetto iniziale nel corrispondente punto x' dopo le 4 trasformazioni sopra illustrate è calcolato come $x' = Hx = DCBAx$.

Capitolo 3

Curve Nurbs

L'acronimo NURBS, sta per “**Not Uniform Rational B-Spline**” traducibile in "B-Splines razionali non uniformi".

La parola *Rational*, indica che le funzioni sono razionali, cioè del tipo “polinomio fratto polinomio” (anche di grado diverso). Questo tipo di curve, sono una generalizzazione delle curve B-Spline e delle curve di Bèzier. Le curve Nurbs, modellano in maniera dettagliata, curve ed ellissi, restituendo un risultato migliore rispetto alle altre curve affrontate (più avanti vedremo un esempio con cui potremo renderci conto di questa differenza). In un certo senso, possiamo rappresentare in questo modo, la relazione che c'è tra i vari tipi di curve:



3.1 Un po' di Storia

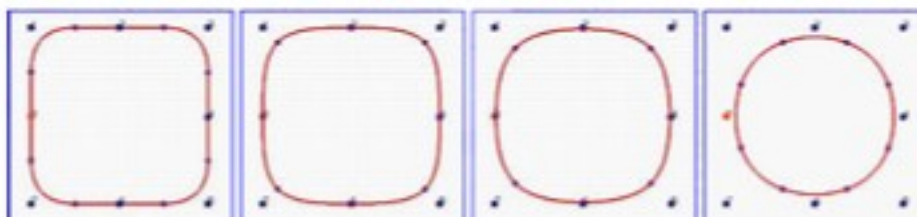
La paternità delle NURBS è dibattuta. Vi sono almeno due candidati che competono per questo titolo:

- Ken Versprille si autopromuove come inventore asserendo di aver ideato le NURBS quando anni fa lavorava per ComputerVison, al tempo una delle maggiori aziende CAD del mondo. Il 5 aprile 2005 Ken ha ricevuto dalla CAD Society il CAD Society Lifetime Award per questo suo contributo alla comunità CAD.
- -La comunità scientifica riconosce unanimemente in Les A. Piegl e Wayne Tiller i padri delle NURBS. A testimonianza del loro contributo fondamentale a questa tecnologia, vi è il poderoso volume The NURBS Book di ben 645 pagine, che costituisce l'indisputato testo di riferimento per chiunque si occupi di NURBS. Les A. Piegl è professore di computer science e ingegneria presso l'University of South Florida mentre Wayne Tiller è attualmente occupato in Geomware, un'azienda creata per sviluppare e commercializzare la libreria NURBS chiamata Nlib, descritta nel volume.

3.2 Perché usare le curve NURBS?

Come già visto in precedenza, le curve B-Spline, non hanno espressività tale da rendere possibile la modellazione di ellissi e cerchi, sebbene siano molto flessibili e ricche di proprietà interessanti. I cerchi e le ellissi, possono essere rappresentate dunque, solamente con funzioni **RAZIONALI**. Occorre quindi estendere ulteriormente le curve B-Spline, e arrivare alle Nurbs.

Ma andiamo a vedere un esempio dettagliato, in cui si nota perfettamente che i risultati ottenuti cercando di modellare un cerchio utilizzando ad esempio le curve B-Spline, non sono dei migliori:



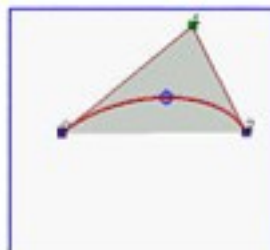
- La prima immagine, rappresenta il risultato di una curva B-Spline di grado 2. Appare chiaro che il grado, è fin troppo basso per rappresentare un cerchio; si ottiene piuttosto, un quadrato con gli angoli smussati.
- La seconda immagine, rappresenta il risultato di una curva B-Spline di grado 3, si nota solo un leggerissimo miglioramento, ma siamo ancora lontani dal nostro obiettivo.

- Con il grado 5 (terza immagine) il risultato cambia di pochissimo.
- Solo con il grado 10 (quarta immagine) si inizia ad intravedere una sorta di “cerchio”. E' evidente però, che non ci siamo ancora, e che in ogni caso, un grado così alto è inaccettabile.

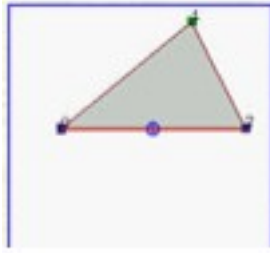
3.3 Modellare le curve NURBS

Le curve NURBS, vengono modellate usando:

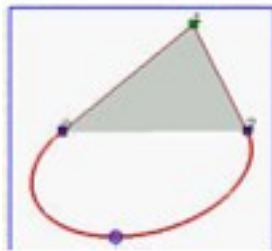
- $\mathbf{u}_0 \dots \mathbf{u}_n$: vettore dei nodi;
- $N_{i,p}(\mathbf{u})$: funzione base;
- p : grado fissato;
- $w_i \geq 0$: pesi che rappresentano **l'importanza del nodo associato**. Nel paragrafo successivo, vedremo infatti che per definire un curva NURBS, sarà necessario associare a ogni punto del poligono di controllo, un peso di valore arbitrario, scelto dal Designer, che identificherà l'apporto dato alla modellazione della curva. Vediamo l'esempio che segue, dove il peso che andremo a considerare, sarà quello associato a l'unico **punto centrale** del poligono di controllo:
 - Con w alto, la curva tenderà ad avvicinarsi al punto di controllo ad esso associato;



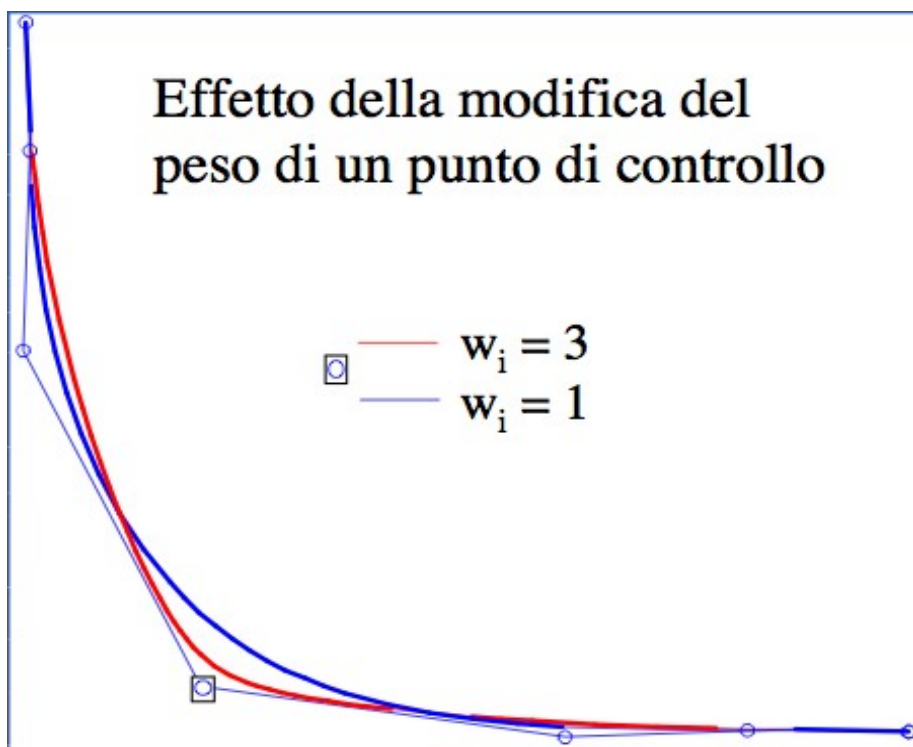
- Con $w = 0$, la curva non sarà influenzata da quel punto di controllo indicato;



- $w < 0$, è una forma poco utilizzata, possiamo notare dalla figura, cosa succederebbe.



Osserviamo anche nell'immagine che segue, la variazione della curva in base al peso:



3.4 Definizione di curva NURBS

Una curva NURBS, viene definita come “**Somma di funzioni base per i punti del poligono di controllo** (moltiplicati per il peso associato)” tutto fratto ovviamente la somma delle funzioni base per i pesi dei punti del poligono di controllo:

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

Senza il polinomio al denominatore, la funzione non sarebbe razionale, e quindi non sarebbe una curva NURBS.

Ricordiamo che in corrispondenza di ogni Punto di Controllo vi è associato un peso.

Un caso particolare si pone quando tutti i pesi abbiano un valore pari a 1. In questo caso infatti la curva NURBS, non sarebbe più tale, ma diventerebbe una curva B-Spline:

$$C(u) = \frac{\sum_{i=0}^n N_{i,p}(u) \cdot P_i}{1}$$

Questo perché così facendo, otterremmo al denominatore una banale somma di funzioni base, che ci restituirebbe il valore 1. La funzione dunque, non sarebbe più razionale (cioè del tipo “polinomio fratto polinomio”). Questo è banale, poiché i control points in coordinate omogenee, sarebbero identici al loro equivalente Cartesiano.

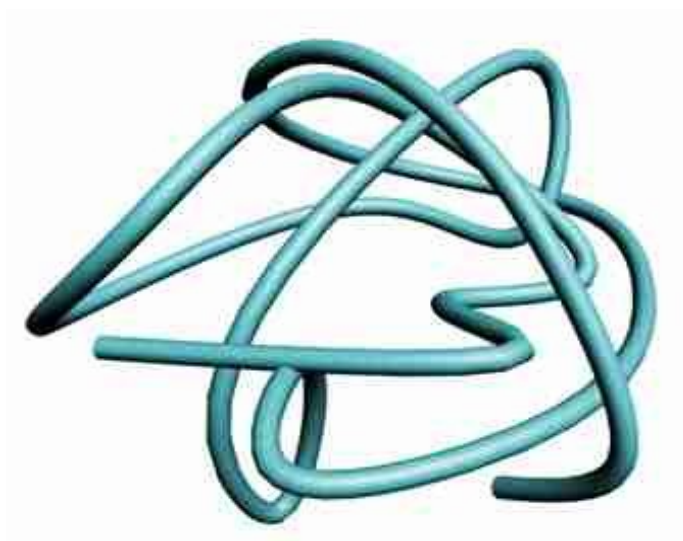
Da tutto ciò possiamo comprendere quanto possano essere importanti e caratterizzanti i pesi nelle curve NURBS.

Esempi di modellazione con curve NURBS:

Nell'immagine seguente, notiamo la “dolcezza” e la flessibilità di queste curve: si nota chiaramente, la rappresentazione del numero “6”:



In questa immagine invece, possiamo apprezzare l'utilizzo delle curve NURBS in uno spazio 3D:



3.5 Proprietà importanti delle curve NURBS

Dato un insieme con $n + 1$ control points P_0, P_1, \dots, P_n , ad ognuno dei quali è associato un peso non negativo w_i (per esempio P_i ha peso $w_i \geq 0$), e un vettore di knot $U = u_0, u_1, \dots, u_m$ costituito da $m + 1$ knots, la curva NURBS di grado p è definita come segue:

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

e $R_{i,p}$ è definita come segue:

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

Si è scelto di utilizzare una notazione differente poiché si vuole riscrivere la definizione di una curva NURBS in una forma molto vicina ad una curva B-Spline. La notazione $R_{i,p}(u)$, rappresenta le funzioni base della curva NURBS. A questo punto è chiaro che le curve NURBS sono una generalizzazione delle curve B-Spline, per questo motivo hanno tutte le loro proprietà. Le proprietà che seguono, enfatizzano questo concetto:

1. $R_{i,p}(u)$ è una funzione **razionale** di grado p in u ;
2. **Non negatività**: per ogni i e p , $R_{i,p}(u)$ è non negativa;
3. **Supporto locale**: $R_{i,p}$ assume valori definiti solo nell'intervallo $[u_i, u_{i+p+1})$, mentre all'esterno dell'intervallo, il valore è 0. Questo implica un maggiore controllo della curva, poiché non tutte le funzioni base hanno un ruolo fondamentale;
4. **Partizione dell'unità**: La somma di tutte le funzioni base nell'intervallo è $[u_i, u_{i+1})$ pari a 1;
5. Se il numero dei knots è $m + 1$, il grado delle funzioni base è p , e il loro numero è $n + 1$, allora $m = n + p + 1$;

6. La funzione base $R_{i,p}(u)$ è una curva composta da funzioni razionali di grado p con punti di funzione ai knots nell'intervallo $[u_i, u_{i+p+1})$.
7. Al knot di molteplicità k , la funzione base $R_{i,p}$ è continua C^{p-k} . Questo implica che aumentando la molteplicità, decresce il livello di continuità e aumentando il grado aumenta la continuità.
8. Se $w_i = c$ per tutte le i , dove c è una costante diversa da 0, $R_{i,p}(u) = N_{i,p}(u)$.

A questo punto, diviene necessario elencare quali sono le caratteristiche peculiari delle curve NURBS, ovvero:

- **Le funzioni base sono razionali;**
- **Supporto locale;**
- **Partizione unità;**
- **Variation diminishing:** Se la curva è contenuta in un piano, significa che nessuna linea interseca una curva NURBS più volte di quanto essa intersechi il poligono di controllo.
- **Strong Convex Hull:** La curva è contenuta nel dominio convesso generato dai suoi punti di controllo. In aggiunta, se u è contenuto all'interno dell'intervallo $[u_i, u_{i+1})$, allora $C(u)$ è nel dominio convesso dei control points $P_{i-p}, P_{i-p+1}, \dots, P_i$. Con questo appare chiaro che tutti i pesi devono essere non negativi. Se qualcuno di essi lo fosse, la *Strong Convex Hull*, o comunque la *Convex Hull*, non sarebbe presente.
- **Invarianza per trasformazioni proiettive:** se volessi applicare una trasformazione proiettiva, dovrei applicare il calcolo matriciale a tutti i punti della curva. Questa proprietà, garantisce invece un risultato corretto, effettuando il calcolo solamente sui punti del poligono di controllo.

3.6 Relazione tra una curva B-Spline e una curva NURBS

In questo paragrafo mostriamo la relazione esistente tra una curva **b-spline** ed una curva **nurbs**:

1. Si parte da una curva B-Spline:

$$C(u) = \sum_{i=0}^n N_{i,p}(u)P_i \quad (3.1)$$

2. Esprimiamo P_i mediante le sue **coordinate euclidee**:

$$P_i = \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix}$$

3. A questo punto (grazie alle coordinate omogenee viste in precedenza) è possibile riscrivere P_i come un vettore colonna costituito da quattro componenti con la quarta componente pari ad 1:

$$P_i = \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix}$$

4. Ora, consideriamo un peso w_i ; poiché sappiamo che moltiplicando le coordinate omogenee per uno scalare diverso da 0, non cambia la posizione del punto nello spazio, possiamo scrivere:

$$P_i = \begin{bmatrix} w_i \cdot x_i \\ w_i \cdot y_i \\ w_i \cdot z_i \\ w_i \end{bmatrix}$$

5. Quindi, sostituendo il tutto alla formula iniziale (3.1) otteniamo:

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

6. Abbiamo dunque ottenuto in questo modo una curva B-Spline 4-dimensionale. Ripassando ora alle coordinate cartesiane, occorre dividere le prime tre coordinate per l'ultima. Quindi:

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

Otteniamo quindi la **definizione di una curva NURBS**, che può essere considerata come la **proiezione di una curva B-Spline in uno spazio 4-dimensionale**.

3.7 Calcolo di una curva NURBS

Vogliamo ora trovare un algoritmo efficiente per il calcolo di una curva nurbs. Poiché nel paragrafo precedente abbiamo visto la relazione che lega una curva Nurbs ad una curva b-spline (quindi che **una curva NURBS**, può essere considerata come la **proiezione di una curva B-Spline in uno spazio 4-dimensionale**), cerchiamo di sfruttare questa proprietà per trovare un algoritmo per il calcolo di una curva Nurbs:

1. Si parte dalla definizione di NURBS:

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

Formula 3.7.1

2. Dunque sappiamo che il punto di controllo **P_i** può essere immaginato come un punto in coordinate omogenee con la quarta pari ad uno:

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

Formula 3.7.2

3. Poiché sappiamo che moltiplicando le coordinate omogenee per uno scalare diverso da 0, non cambia la posizione del punto nello spazio, moltiplichiamo ogni coordinata, per la quantità:

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

che è il denominatore visto nelle 4 coordinate della formula 3.7.2, e otteniamo una curva B-Spline 4-dimensionale:

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

Formula 3.7.4

4. Posso quindi calcolare le 4 coordinate in (3.7.4), mediante l'algoritmo di **De Boor**. A questo punto ripassiamo alle coordinate cartesiane, dividendo come noto le prime tre coordinate in (3.7.4) per l'ultima, cioè:

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

ottengo quindi:

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

che è proprio la definizione della Nurbs di partenza (3.7.1).

3.8 Curve di Bèzier vs. Curve B-Spline vs. Curve Nurbs

Analizziamo qui di seguito quali sono i punti che accomunano le curve di Bèzier, le B-Spline e le curve Nurbs, e quali sono invece le differenze:

1. La curva è contenuta all'interno dell'involucro convesso del poligono di controllo)
 - Bezièr: SI
 - B-Spline: SI
 - Nurbs: SI (a patto che il peso associato ai punti del poligono di controllo sia maggiore di 0)
2. Come si lega il grado della curva rispetto al poligono di controllo?
 - Bezièr: Grado = Numero di vertici - 1
 - B-Spline: Indipendente dal poligono di controllo
 - Nurbs: indipendente dal poligono di controllo
3. La curva tocca il poligono di controllo?
 - Bezièr: SI, nel primo e nell'ultimo punto
 - B-Spline: NO, ma può essere fatto, collassando tanti nodi quanto il grado della curva. Operando questa routine, difatti si può farla passare per qualsiasi punto del poligono
 - Nurbs: NO, ma può essere fatto, collassando tanti nodi quanto il grado della curva. Operando questa routine, difatti si può farla passare per qualsiasi punto del poligono
4. La forma della curva e del poligono sono in qualche modo correlate?
 - Bezièr: NO, solo in modo molto approssimato
 - B-Spline: SI e li si può addirittura far coincidere con particolari accorgimenti
 - Nurbs: SI e li si può addirittura far coincidere con particolari accorgimenti

5. Si può avere una trasformazione affine sulla curva?

- Bezièr: SI, operando sui vertici del poligono di controllo
- B-Spline: SI, operando sui vertici del poligono di controllo
- Nurbs: SI, operando sui vertici del poligono di controllo.
Inoltre è invariante anche per trasformazioni proiettive.

6. Si può avere il controllo locale sulla curva?

- Bezièr: NO
- B-Spline: SI
- Nurbs: SI

7. Si possono disegnare cerchi ed ellissi?

- Bezièr: NO
- B-Spline: NO
- Nurbs: SI

Capitolo 4

4.1 Superfici

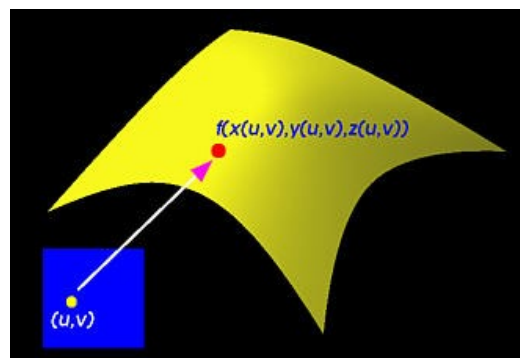
Esistono due tipi di superfici usate nella modellazione di sistemi :

- Superfici parametriche
- Superfici implicite

4.1.1 Superfici parametriche

Le superfici parametriche sono definiti da un insieme di tre funzioni, uno per ogni coordinata.

$$f(u, v) = [x(u, v); y(u, v); z(u, v)]$$



Per i nostri scopi assumeremo che $u, v \in [0, 1]$. Cioè (u, v) sarà un punto nel quadrato definito dai punti $(0, 0)$; $(1, 0)$; $(0, 1)$; $(1, 1)$.

Se $x(u, v)$, $y(u, v)$, $z(u, v)$ sono polinomi, le superfici parametriche non sono utili per rappresentare la maggior parte delle superfici es. sfera (al posto delle superfici parametriche in questi casi si usano le superfici implicite).

4.1.2 Superfici implicite

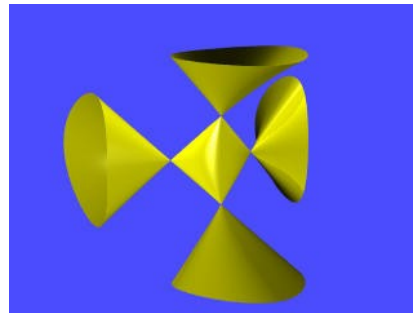
Le superfici implicite sono definite da un polinomio in tre variabili:

$$p(x, y, z)$$

Molte superfici implicite non hanno una forma parametrica. Di conseguenza le superfici implicite hanno un potere espressivo molto maggiore delle superfici parametriche.

A destra vediamo un esempio di superficie implicita di grado 3, la quale equazione implicita è :

$$8x^2 - xy^2 + xz^2 + y^2 + z^2 - 8 = 0$$



4.2 Le superfici di Bézier

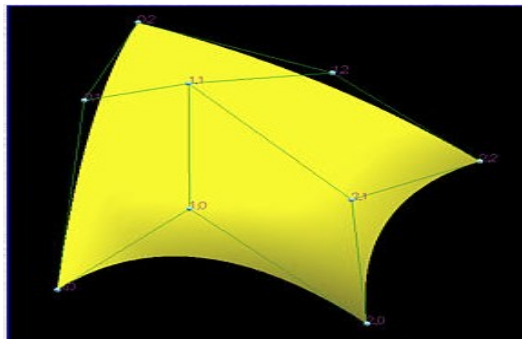
Le superfici di Bézier sono uno strumento molto potente per la progettazione di superfici, quindi ampiamente utilizzate nella computer graphics assieme alle curve di Bézier. L'idea che sta alla base della teoria di Bézier è di definire una superficie a partire da un numero finito di punti P_0, P_1, \dots, P_n nello spazio, in modo che il suo supporto segua l'andamento del poliedro, detto **poliedro di controllo** individuato da questi. Anche nel caso delle superfici di Bézier ci sono diversi modi per definirle e implementarle. In questa sede ci si limita alla costruzione delle superfici di Bézier mediante i polinomi di Bernstein.

4.2.1 Definizione di superficie di Bézier

Le superfici di Bézier vengono definite considerando un insieme di punti p_{ij} per $i=0, \dots, m$ e $j=0, \dots, n$. Questi punti sono distribuiti su di un reticolo che prende il nome di poliedro di controllo (control net). La superficie di Bézier viene quindi così definita:

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

Dal momento che il grado di $B_{m,i}(u)$ è m e il grado di $B_{n,j}(v)$ è n , possiamo affermare che il grado di una superficie di Bézier è (m, n) . La figura che segue mostra una superficie di Bézier definita da 3 righe e 3 colonne ovvero 9 control points. La superficie ha grado $(2, 2)$.



4.2.2 Proprietà delle superfici di Bézier

Le superfici di Bézier godono di alcune proprietà che derivano da quelle dei polinomi di Bernstein usati per costruire le superfici stesse e dalle proprietà delle curve di Bézier, in quanto le superfici non sono altro che un'estensione delle curve di Bézier; in particolare:

1. Convex Hull:

Ovviamente il prodotto $B_{m,i}B_{n,j}(u) \geq 0$ nell'intervallo $[0, 1]$; ovvero le funzioni base sono tutte positive: questa proprietà garantisce che la superficie sarà contenuta nel dominio convesso del control net.

2. Partizione dell'unità:

$$\sum \sum B_{m,i} B_{n,j} = 1$$

Quindi la superficie dipende unicamente dalla scelta dei control points e non dalla scelta delle origini delle coordinate.

3. Interpolazione degli estremi:

Come la curva di Bézier, la superficie di Bézier interpola, cioè passa attraverso, i primi e gli ultimi punto di controllo, mentre approssima gli altri punti. Infatti, si ha che:

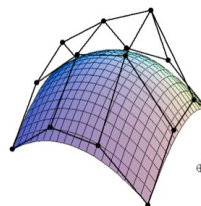
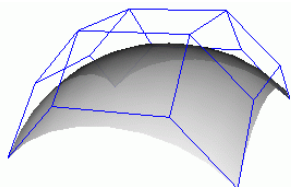
$$\begin{aligned} P(0, 0) &= P_{0,0} & P(1, 0) &= P_{h,0} \\ P(0; 1) &= P_{0;k} & P(1; 1) &= P_{h;k} \end{aligned}$$

Questo ci suggerisce che la superficie è facilmente raccordabile.

4. Invarianza affine:

Le superfici di Bézier sono invarianti per trasformazioni affini.

5. Il bordo delle superfici di Bézier è dato dalle curve di Bézier descritte dalle poligoni che formano il bordo del poliedro di controllo. Invece le curve di Bézier generate dalle poligoni interne al poliedro di controllo di solito non vivono sulla superficie.



6. No variation diminishing!

Le superfici di Bézier, diversamente dalle curve di Bézier, non godono della proprietà di variation diminishing.

4.2.3 Limitazioni delle superfici di Bézier

Le superfici di Bézier hanno alcune limitazioni:

7. Non si possono rappresentare sfere o calotte sferiche, né superfici di rotazione.

8. Le superfici di Bézier non ammettono controllo locale, pertanto la variazione di un solo punto del poliedro di controllo porta ad un rilevante cambiamento di tutta la superficie.

9. Per rappresentare forme complesse, usando una sola superficie, sono necessari molti punti di controllo. Questo significa che il grado dei polinomi usati nella rappresentazione di Bézier può diventare molto alto e quindi difficile da trattare al calcolatore.

4.2.4 La soluzione - *Le patches di Bézier*

La soluzione a questo problema consiste nell'incollare più superfici di grado minore secondo condizioni di continuità per dare l'effetto di un solo foglio con grado apparentemente alto di complessità, la cui elaborazione risulta però meno complessa:

.

Le patches di Bézier: Un esempio di patches di Bézier sono quelle triangolari che usano i polinomi di Bernstein generalizzati, presenti nel capitolo riguardante i polinomi di Bernstein, adatti proprio a computare superfici di questo tipo. Le condizioni imposte all'atto dell'incollaggio hanno la funzione di rendere la superficie più o meno levigata, in particolare:

- La continuità C^0 impone che le due superfici condividano lo stesso bordo, quindi in prossimità di tale incollamento è possibile avere punti angolosi. Se consideriamo i bordi $v = 1$ del patch \mathbb{P}_1 e $v = 0$ del patch \mathbb{P}_2 , la condizione diventa per ogni $u \in [0; 1]$:

$$\bar{P}_1(u, 1) = \bar{P}_2(u, 0)$$

- La continuità C^1 impone, oltre alla C^0 , che la derivata lungo il bordo sia la stessa; viene eliminata la presenza di punti angolosi:

Queste limitazioni non risultano così svantaggiose, dato che, come abbiamo visto, ci sono dei metodi per aggirarle. Infatti le superfici di Bézier, assieme alle curve di Bézier, sono uno strumento fondamentale per la rappresentazione grafica e per la modellistica perché la loro semplicità di implementazione e la facilità con cui possono essere manipolate superano di gran lunga i loro difetti.



4.3 Algoritmo di de Casteljau

L'algoritmo di de Casteljau può essere esteso per calcolare anche le superfici di Bézier. Più precisamente l'algoritmo può essere applicato diverse volte finché non si ottiene il punto corrispondente sulla superficie di Bézier $p(u, v)$ dato (u, v) .

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

Invece di calcolare la superficie effettuando le operazioni in cascata, porremo:

$$q_i(v) = \sum_j B_{n,j}(v) p_{i,j}$$

Come si può notare $q_i(v)$ è un punto sulla curva di Bézier definita dai punti di controllo $p_{i,0}, p_{i,1}, \dots, p_{i,n}$

A questo punto utilizzando l'algoritmo di De Casteljau, calcoliamo q_i per ogni i :

per $i = 0$ $q_0(v) = \sum_j B_{n,j}(v) p_{0,j}$ $\begin{matrix} * & * & * & * & * \\ \circ & \circ & \circ & \circ & \circ \\ \circ & \circ & \circ & \circ & \circ \\ \circ & \circ & \circ & \circ & \circ \\ \circ & \circ & \circ & \circ & \circ \end{matrix}$

che rappresenta la prima riga (contraddistinta dalle $(*)$) presa in esame nella control net.

per $i = 1$ $q_1(v) = \sum_j B_{n,j}(v) p_{1,j}$ $\begin{matrix} \circ & \circ & \circ & \circ & \circ \\ * & * & * & * & * \\ \circ & \circ & \circ & \circ & \circ \\ \circ & \circ & \circ & \circ & \circ \\ \circ & \circ & \circ & \circ & \circ \end{matrix}$

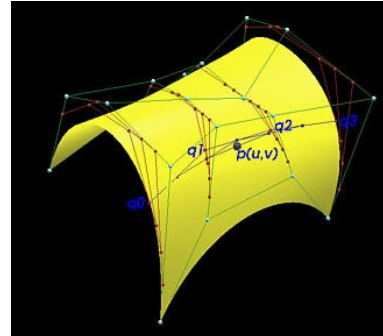
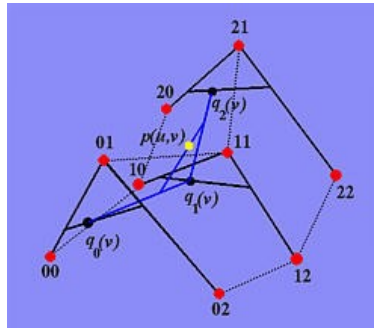
etc...

Si ottiene, quindi:

$$P(u, v) = \sum_i B_{m,i}(u) q_i(v)$$

Possiamo nuovamente utilizzare l'algoritmo di De Casteljaeu, poiché si ottiene una curva definita sui nuovi punti di controllo.

Per calcolare un punto $p(u, v)$ sulla superficie di Bézier, possiamo determinare $m+1$ punti $q_0(v), q_1(v), \dots, q_m(v)$ e da questi calcolare poi $p(u, v)$.



4.4 Curve isoparametriche

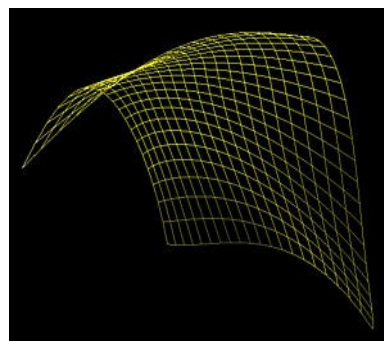
Abbiamo visto che una superficie parametrica è definita come:

$$f(u, v) = [x(u, v); y(u, v); z(u, v)]$$

Se fisso il parametro u ad un punto costante a e faccio variare l'unico parametro libero v , genero un'unica curva sulla superficie definita da $f(a, v)$. Questa è una curva isoparametrica in cui $u = \text{cost.}$ in direzione v .

Analogamente, fissato $v = \text{cost.}$ e variando il parametro u , ottengo una curva isoparametrica in cui $v = \text{cost.}$ e $u = 0 \dots 1$

Facendo variare u e v in $[0 \dots 1]$ si ottengono una serie di curve che finiscono per ricoprire l'intera superficie:



4.4.1 Curve isoparametriche di bordo

Ci sono quattro curve isoparametriche speciali: $p(0, v)$; $p(1, v)$; $p(u, 0)$; $p(u, 1)$. Queste vengono chiamate curve isoparametriche di bordo visto che mappano i bordi della nostra superficie unitaria.

Impostando $u = 0$ e $u = 1$, $v = 0$ e $v = 1$ si ottengono le seguenti equazioni delle curve di bordo:

$$\begin{aligned}
 p(0, v) &= \sum_{j=0}^n B_{n,j}(v) p_{0,j} & p(u, 0) &= \sum_{i=0}^m B_{m,i}(u) p_{i,0} \\
 p(1, v) &= \sum_{j=0}^n B_{n,j}(v) p_{m,j} & p(u, 1) &= \sum_{i=0}^m B_{m,i}(u) p_{i,n}
 \end{aligned}$$

4.5 Prodotto Tensoriale

A questo punto siamo pronti per introdurre il concetto di **prodotto tensoriale**

Ricordiamo qui sotto la definizione della superficie di Bézier:

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

Se riscriviamo l'insieme dei punti di controllo in una matrice di $m+1$ righe e $n+1$ colonne otteniamo:

$$\begin{bmatrix}
 p_{0,0} & p_{0,1} & \cdot & \cdot & p_{0,n} \\
 p_{1,0} & p_{1,1} & \cdot & \cdot & p_{1,n} \\
 \cdot & \cdot & \cdot & \cdot & \cdot \\
 \cdot & \cdot & \cdot & \cdot & \cdot \\
 p_{m,0} & p_{m,1} & \cdot & \cdot & p_{m,n}
 \end{bmatrix}$$

E se riscriviamo le funzioni base di Bézier nella direzione v come un vettore colonna:

$$\begin{bmatrix}
 B_{n,0}(v) \\
 B_{n,1}(v) \\
 \cdot \\
 \cdot \\
 \cdot \\
 B_{n,n}(v)
 \end{bmatrix}$$

Allora possiamo riscrivere il risultato della grande parentesi come una matrice prodotto:

$$\begin{bmatrix} p_{0,0} & p_{0,1} & \cdot & \cdot & p_{0,n} \\ p_{1,0} & p_{1,1} & \cdot & \cdot & p_{1,n} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ p_{m,0} & p_{m,1} & \cdot & \cdot & p_{m,n} \end{bmatrix} * \begin{bmatrix} B_{n,0}(v) \\ B_{n,1}(v) \\ \cdot \\ \cdot \\ \cdot \\ B_{n,n}(v) \end{bmatrix}$$

Rappresentata in questa forma matricale, il risultato è ovviamente una matrice colonna di m+1 elementi:

$$[B_{m,0}(u) \quad B_{m,1}(u) \quad \cdot \quad \cdot \quad \cdot \quad B_{m,m}(u)]$$

L'equazione della curva di Bézier diviene allora un prodotto di tre matrici come segue:

$$p(u, v) = [B_{m,0}(u) \quad B_{m,1}(u) \quad \cdot \quad \cdot \quad \cdot \quad B_{m,m}(u)] * \begin{bmatrix} p_{0,0} & p_{0,1} & \cdot & \cdot & p_{0,n} \\ p_{1,0} & p_{1,1} & \cdot & \cdot & p_{1,n} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ p_{m,0} & p_{m,1} & \cdot & \cdot & p_{m,n} \end{bmatrix} * \begin{bmatrix} B_{n,0}(v) \\ B_{n,1}(v) \\ \cdot \\ \cdot \\ \cdot \\ B_{n,n}(v) \end{bmatrix}$$

Cioè, siamo riusciti a convertire la definizione di una superficie di Bézier in un prodotto. Visto che gli elementi della matrice dei control points sono matrici (vettore colonna), abbiamo ottenuto quello che in matematica viene chiamato **prodotto tensoriale**.

Come conseguenza, possiamo affermare che le superfici di Bézier sono **superfici prodotto tensore**.

4.6 Superfici B-Spline

Come per il caso delle curve, anche per le superfici c'è stato il bisogno di migliorare le prestazioni delle funzioni di Bezier (mancato controllo locale) e per fare questo ci si è indirizzati verso le funzioni B-Spline.

4.6.1 Definizione superfici B-Spline

Una curva B-Spline è definita in funzione della coordinata $x(u)$ del piano

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

perché deve essere disegnata su un piano, allo stesso modo una superficie B-Spline è definita in funzione di 2 coordinate x e y (u e v)

$$P(u, v) = \sum_{i=0}^m \sum_{j=0}^n N_{i,p}(u)N_{j,q}(v)P_{i,j}$$

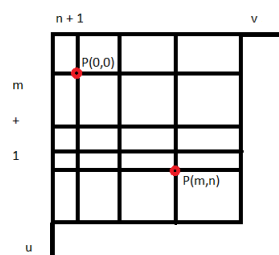
dello spazio perché deve essere disegnata in uno spazio tridimensionale.

Per comodità di calcolo u e v apparterranno all'intervallo $[0, 1]$.

Iniziamo a descrivere gli elementi della formula appena proposta:

$P(u, v)$ sarà l'insieme delle coordinate z che permetterà di disegnare la superficie sullo spazio.

Il piano che rappresenta le coordinate u e v è scandito da $(n + 1) \times (m + 1)$ punti di controllo (x rappresenta il prodotto cartesiano), quindi i punti $P_{i,j}$ possono andare da $(0,0)$ a (m,n) .



Le coordinate (direzioni) u e v sono soggette anche ad un'altra suddivisione, infatti esse sono ripartite rispettivamente in $(h + 1)$ knots (nodi) la direzione u che formano il vettore $U = \{ u_0, u_1, \dots, u_h \}$, e in $(k + 1)$ knots la direzione v che formano il vettore $V = \{ v_0, v_1, \dots, v_k \}$; essi rappresentano il vettore detto knots span caratteristico appunto delle curve B-Spline.

Oltre a queste suddivisioni, ad ognuna delle 2 direzioni è associato un intero, (p per u) e (q per v), che rappresenterà il grado che le curve B-Spline avranno sulle 2 direzioni (il valore massimo che potranno raggiungere sia q che p è 3). La relazione che lega h ad m e k ad n è ($h = m + p + 1$) e ($k = n + q + 1$) proprio come per le curve B-Spline.

In definitiva possiamo dire che una superficie B-Spline è descritta dal prodotto tensoriale di 2 curve B-Spline, cioè la possiamo decomporre in 2 curve B-Spline, una sulle x e una sulle y messe a prodotto fra loro.

4.6.2 Proprietà delle superfici B-Spline

1. Basso Grado massimo:

Il grado delle 2 curve è al massimo 3, cioè quello di q e p , quindi i calcoli da fare con polinomi di grado 3 non sono particolarmente onerosi.

2. Non negatività:

Per ogni j, i, p, q, v e u , $N_{i,p}(u) * N_{j,q}(u)$ è non negativa nell'intervallo tra 0 e 1. Questa proprietà garantisce che la superficie sarà contenuta nel dominio convesso del control net.

3. Partizione dell'unità

La superficie dipende unicamente dalla scelta dei control points e non dalla scelta delle origini delle coordinate.

4. Non interpolazione agli estremi

Le superfici B-Spline non passano per default agli estremi del control net, per farcele passare bisogna aumentare la molteplicità geometrica di quei punti (come per un qualsiasi altro punto). E' lo stesso comportamento delle curve da cui sono formate.

5. Invarianza affine

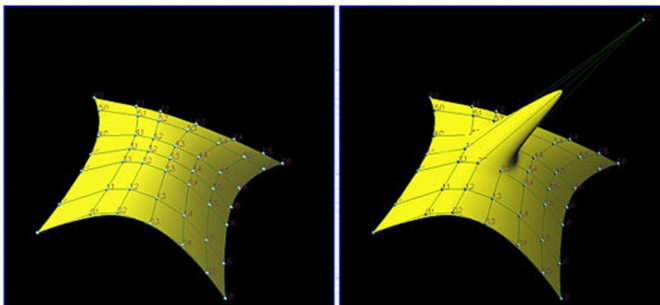
Le superfici B-Spline sono invarianti per trasformazioni affini. Questo implica che per fare trasformazioni su una superficie B-Spline è possibile applicare la trasformazione ai soli punti di controllo.

6. Strong convex hull

Se (u,v) è in $[u_i, u_{i+1}) * [v_j, v_{j+1})$, allora $p_{u,v}$ giace nel dominio convesso definito dai control points $p_{h,k}$ dove $i-p \leq h \leq i$, e $j-q \leq h \leq j$. Questa proprietà deriva direttamente dalla proprietà dello strong convex hull per le curve B-Spline. Per la direzione u , se u è in $[u_i, u_{i+1})$ allora ci sono al più $p+1$ funzioni base diverse da zero, su questo intervallo dei knot, etichettate $N_{j,q}(v), N_{j-1,q}(v), \dots, N_{j-q,q}(v)$. Così solo in control points dalla colonna $j-q$ alla colonna j hanno funzioni base diverse da zero nella direzione v . Combinando questi due fatti insieme, solamente i control points dell'intervallo dalla riga $i-p$ alla riga i e le colonne da $j-q$ a j hanno funzioni base diverse da zero. Dal momento che queste funzioni sono maggiori di zero e la loro somma è 1, $p_{u,v}$ giace nel dominio convesso definito da questi control points. Chiaramente, la superficie definita dal rettangolo $[u_i, u_{i+1}) * [v_j, v_{j+1})$ giace completamente all'interno nello stesso dominio convesso.

7. Schema di modifica locale

$N_{i,p}(u), N_{j,q}(v) = 0$, se (u,v) è fuori dal rettangolo $[u_i, u_{i+1}) * [v_j, v_{j+1})$. Dalla proprietà del controllo locale sappiamo che nella direzione u , $N_{i,p}(u)$ è diversa da zero in $[u_i, u_{i+1})$ ed è zero altrove. Se il control point $p_{3,2}$ è spostato in una nuova locazione, solo l'area di vicinato subisce una deformazione e altrove la forma rimane invariata.



8. Le superfici di Bezier sono un caso particolare delle B-Spline

Se $m = p$, $n = q$, e $U = \{0,0,\dots,1,1,\dots,1\}$ allora la superficie B-Spline diventa una superficie di Bezier.

4.6.3 Algoritmo per ricavare la superficie

Come per le superfici di Bezier, la formula per le superfici B-Spline

$$\mathbf{p}(u, v) = \sum_{i=0}^m \sum_{j=0}^n N_{i,p}(u) N_{j,q}(v) \mathbf{p}_{i,j}$$

si semplifica inserendo $q_i(v) = \sum_j N_{j,q}(v) \mathbf{p}_{i,j}$

In modo tale che la definizione diventi $\mathbf{P}_{u,v} = \sum_i N_{i,p}(u) q_i(v)$

E, così come si è fatto per le superfici di Bezier, anche per le B-Spline si applica n volte l'algoritmo di de Boor per calcolare $q_i(v)$ per ogni i e poi si userà altre m volte per calcolare $\mathbf{P}_{u,v}$ per ogni j .

5 Sistemi di proiezione

I sistemi di proiezione sono largamente diffusi in informatica, specialmente nell'ambito dei videogiochi e dei loro principi di movimento (funzionamento delle schede grafiche).

In pratica si hanno a disposizione dispositivi bidimensionali, come gli schermi, che devono riprodurre scene tridimensionali, perciò sono indispensabili dei sistemi di proiezione.

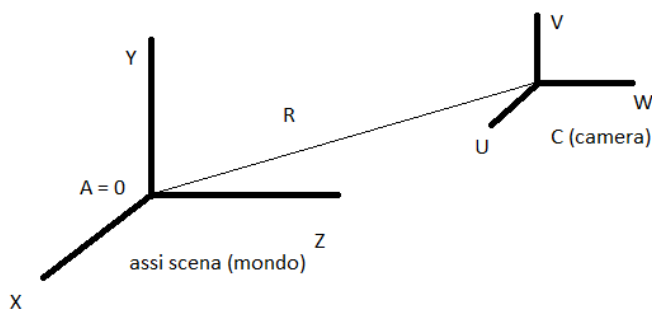
Per usufruire di questi sistemi è necessaria molta matematica.

Le due componenti fondamentali sono: la scena (3D) che sarebbe il mondo da rappresentare, e l'osservatore, cioè la macchina fotografica (2D) che si muova all'interno della scena.

5.1 Sistemi di riferimento

Si definisce un sistema di riferimento come 3 assi ortogonali e un'origine, che sarebbe il punto di incontro dei 3 assi.

Per la nostra scena servono 2 sistemi di riferimento, uno per la scena e uno per l'osservatore.



Per la fotocamera serve sia la posizione, sia la direzione di puntamento

Ipotesi: ovunque sia la camera, essa punta verso l'origine delle coordinate del mondo

W è messo sulla direzione CA ma in verso opposto

U è messo perpendicolare al piano YW

V è scelto di conseguenza

(queste 3 direzioni formano un sistema di riferimento variabile, mentre X,Y,Z sono fisse)

5.1.1 Definizione punti

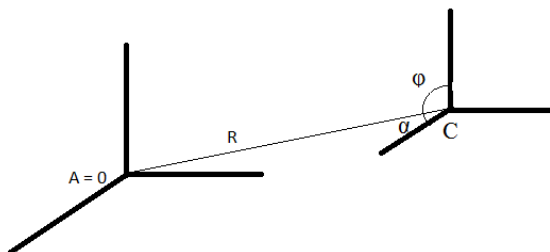
Ogni punto è definito come una tripletta (x,y,z) nel sistema mondo mentre è definito come (u,v,w) nel sistema camera.

Il nostro obiettivo è: data una tripletta nel sistema mondo, trovarne una nel sistema camera.

5.2 Coordinate camera

Troviamo, per iniziare, le coordinate della camera (coordinate polari).

$$C = (R\cos(\varphi)\cos(\alpha) , R\sin(\varphi) , R\cos(\varphi)\sin(\alpha))$$



φ (alto, basso)

α (destra, sinistra)

R (distanza)

(il simbolo “ $\vec{\quad}$ ” significa vettore)

\vec{W} (essendo un versore, si divide per la norma) = $CA / |CA| =$

$$\underline{(R\cos(\varphi)\cos(\alpha) , R\sin(\varphi) , R\cos(\varphi)\sin(\alpha)) / (R^2\cos^2(\varphi)\cos^2(\alpha) + R^2\sin^2(\varphi) + R^2\cos^2(\varphi)\sin^2(\alpha))^{1/2}}$$

Sistemi di proiezione

Semplifichiamo il denominatore

$$(R^2 \cos^2(\varphi)(\cos^2(\alpha) + \sin^2(\alpha)) + R^2 \sin^2(\varphi))^{1/2}$$
$$\cos^2(\alpha) + \sin^2(\alpha) = 1$$
$$\text{rimane } (R^2 \cos^2(\varphi) + R^2 \sin^2(\varphi))^{1/2} = (R^2)^{1/2} = R$$

Quindi:

$$\underline{W^-} = (\cos(\varphi)\cos(\alpha), \sin(\varphi), \cos(\varphi)\sin(\alpha))$$

U (perpendicolare al piano WY)

Per trovare un vettore perpendicolare ad altri 2 si usa il prodotto vettoriale

$$\underline{a} \times \underline{b} = (a_2b_3 - a_3b_2, a_1b_3 - b_1a_3, a_1b_2 - b_1a_2)$$

$$\underline{U^-} = (\underline{Y^-} \times \underline{W^-}) / (|\underline{Y^-} \times \underline{W^-}|) =$$

$$(0, 1, 0) \times (\cos(\varphi)\cos(\alpha), \sin(\varphi), \cos(\varphi)\sin(\alpha)) / (|\dots\dots\dots|) =$$

$$(\cos(\varphi)\sin(\alpha), 0, -\cos(\varphi)\cos(\alpha)) / (\cos(\varphi)^2\sin(\alpha)^2 + 0 + -\cos(\varphi)^2\cos(\alpha)^2)^{1/2} =$$

$$(\cos(\varphi)\sin(\alpha), 0, -\cos(\varphi)\cos(\alpha)) / (\cos(\varphi)^2(\sin(\alpha)^2 + \cos(\alpha)^2))^{1/2} =$$

$$(\cos(\varphi)\sin(\alpha), 0, -\cos(\varphi)\cos(\alpha)) / (\cos(\varphi)^2)^{1/2} =$$

$$(\sin(\alpha), 0, -\cos(\alpha))$$

$$\underline{U^-} = (\sin(\alpha), 0, -\cos(\alpha))$$

$\underline{V^-}$ (è detreminato)

$$\underline{V^-} = (\underline{W^-} \times \underline{U^-}) / (|\underline{W^-} \times \underline{U^-}|) = (\text{la norma è 1 perché W e U sono perpendicolari})$$

$$(\cos(\varphi)\cos(\alpha), \sin(\varphi), \cos(\varphi)\sin(\alpha)) \times (\sin(\alpha), 0, -\cos(\alpha)) =$$

$$\underline{V^-} = (-\sin(\varphi)\cos(\alpha), -\cos(\varphi), -\sin(\varphi)\sin(\alpha))$$

Ora abbiamo trovato C rispetto alla scena (mondo)

5.3 Trasformazione punti del mondo nel sistema camera

Prendiamo I punti:

$\langle 1,0,0 \rangle$ (per le trasformazioni è sempre bene

$\langle 0,1,0 \rangle$ usare le coordinate omogenee)

$\langle 0,0,1 \rangle$

$\langle 0,0,0 \rangle$

Un punto generico p in un sistema di riferimento si denota come:

$$p = (x,y,z) = (x^*X^- + y^*Y^- + z^*Z^-)$$

$$\left\{ \begin{array}{l} \langle 1,0,0 \rangle = l_{11}^*U^- + l_{12}^*V^- + l_{13}^*W^- \\ \langle 0,1,0 \rangle = l_{21}^*U^- + l_{22}^*V^- + l_{23}^*W^- \\ \langle 0,0,1 \rangle = l_{31}^*U^- + l_{32}^*V^- + l_{33}^*W^- \\ \langle 0,0,0 \rangle = l_{41}^*U^- + l_{42}^*V^- + l_{43}^*W^- + C^- \text{ (distanza da C)} \end{array} \right.$$

(la distanza da C serve per la traslazione della camera)

5.3.1 Soluzioni del sistema

$$\left\{ \begin{array}{l} l_{11} = U_x \quad l_{12} = V_x \quad l_{13} = W_x \\ l_{21} = U_x \quad l_{22} = V_x \quad l_{23} = W_x \\ l_{31} = U_x \quad l_{32} = V_x \quad l_{33} = W_x \\ l_{41} = 0 \quad l_{42} = R \sin(2\varphi) \quad l_{43} = -R \end{array} \right.$$

Quindi la relazione:

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} = \mathbf{M} \begin{pmatrix} U^- \\ V^- \\ W^- \\ C^- \end{pmatrix}$$

Identifica la matrice di rotazione \mathbf{M}

Un punto nello spazio si può scrivere:

$$p = (X, Y, Z, 1) * \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} = (U, V, W, 1) * \begin{pmatrix} U \\ V \\ W \\ C \end{pmatrix}$$

$$\text{Quindi } (x, y, z, 1) * \mathbf{M} = (U, V, W, 1)$$

In definitiva il problema si risolve semplicemente con un prodotto fra il punto da traslare-rotare per la matrice di rotazione.

Per tanto la scheda grafica è banalmente uno strumento Hardware capace di effettuare prodotti vettore-matrice a velocità molto elevate (non a caso il coprocessore matematico dei computer moderni a smesso di farsi le moltiplicazioni da se ma le delega alla scheda grafica).

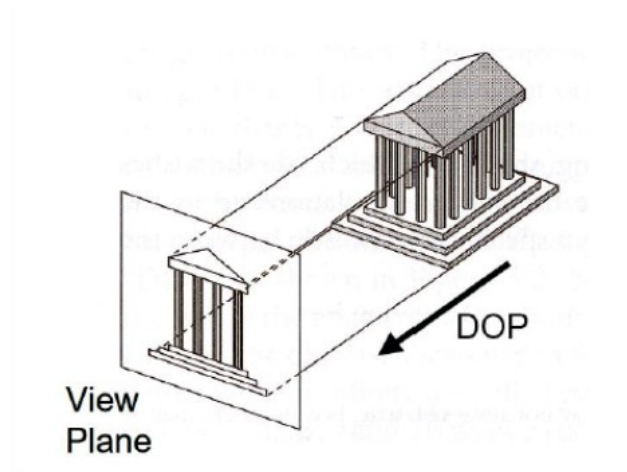
5.4 Scelta della prospettiva

L'ultimo problema su cui decidere è la scelta della prospettiva da utilizzare

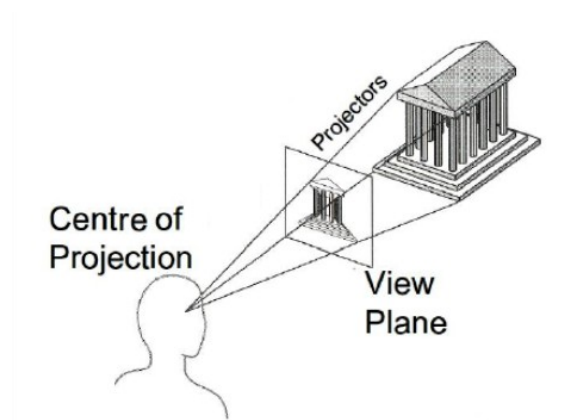
Ce ne sono due tipi: la prospettiva Antica (o parallela) e quella moderna

Sistemi di proiezione

La prospettiva Antica non tiene conto della distanza e riporta gli oggetti sulla tela (o schermo) (o camera) con le loro dimensioni reali



La prospettiva Moderna invece tiene conto delle distanze dei vari oggetti dalla camera



5.5 Progetto

Per il progetto sulle curve di Bezier e B-Spline è sufficiente usare la prospettiva Antica, visto che non ci saranno oggetti a diverse distanze, ed essendo oggetti bidimensionali l'equazione di trasformazione dei punti dal mondo alla camera si riduce a $(U,V) = (X,Y,Z,1) * M$

Capitolo 6

Simulazione

6.1 Simulazione

Lo sviluppo e la ricerca in campo scientifico-tecnologico di questi ultimi anni hanno portato l'uomo a considerare l'opportunità di progetti ambiziosi e di portata infinitamente grande dal punto di vista delle difficoltà tecniche e di progettazione. Esempi intuitivi e più comuni sono la costruzione di grandi edifici come stadi e grattacieli, missioni esplorative spaziali o ancora sistemi di comunicazione su larga scala come la rete internet.



Figura 6.1: Simulazione del ponte dello stretto di Messina

Ma esistono situazioni più sottili o progetti più sofisticati come la nuova chirurgia basata sul laser, le ricerche di massa su campioni di microparticelle, gli studi sugli ecosistemi a livello micro e macroscopico, oppure l'ortopedia con le nuove protesi. In ultima analisi esistono progetti specifici i cui benefici fanno parte della nostra vita quotidiana quali la disposizione ottima degli airbag relativamente all'anatomia delle auto, la definizione della posizione perfetta delle uscite di sicurezza in un locale pubblico o gli studi di viabilità in relazione alla posizione e ad i periodi dei semafori.

Tutti questi lavori hanno in comune un unico denominatore: la necessità di effettuare un "primo" tentativo che deve essere il più vicino possibile alla soluzione finale ottima. In altre parole pensate ad un chirurgo che, operando con il laser sulla retina dei suoi pazienti, prima di determinare l'esatta intensità del raggio di luce debba effettuare una serie di prove con il rischio di rendere cieche alcune persone. Altro esempio può essere quello di un cinema che organizza le sue uscite di sicurezza senza badare al fatto che i percorsi previsti non sono attraversabili dai diversamente abili... Non è pensabile che il cinema si doti di un sistema di evacuazione adattabile solo dopo che è andato a fuoco un paio di volte e magari coinvolgendo nell'incendio persone in sedia a rotelle. La risposta a questa necessità è stata fornita dall'informatica sottoforma di quelle che vengono definite simulazioni.

6.2 Cosa vuol dire simulazione?

Un'applicazione di simulazione ha come obiettivo quello di far risultare all'utente l'esperienza simulata quanto più simile all'esperienza reale. Simulare è quindi un processo che coinvolge l'utente o l'analista in prima persona come supervisore di un sistema che si sviluppa e modifica nel tempo; il sistema ha delle caratteristiche che ricordano e riproducono il più fedelmente possibile quelle di una situazione di vita che l'utente potrebbe dover affrontare o che ha già affrontato. I principali vantaggi dell'utilizzo della simulazione sono quelli di:

- valutare: attraverso la simulazione l'utente/analista è in grado di trarre le sue considerazioni e conclusioni riguardo il progetto simulato.
- prevedere: gli effetti del progetto specifico possono essere riscontrati e valutati prima che il progetto stesso venga intrapreso. Questa è forse la caratteristica più importante della simulazione, molto spesso infatti accade che i risultati della simulazioni siano diversi dalle aspettative indotte dall'intuizione: si è sperimentato per esempi attraverso la simulazione che il posizionamento di un pilastro a breve distanza dall'uscita di sicurezza di un cinema sia vantaggioso per la viabilità in caso di incendi, fattore che ad un primo esame sembrerebbe invece dannoso.
- "far risparmiare": i costi, il superamento di difficoltà tecniche, il rischio di errore o la sperimentazione graduale di un lavoro possono essere completamente risparmiati attraverso una simulazione adeguata: una compagnia telefonica americana ha risparmiato svariati milioni di dollari simulando le conseguenze di un nuovo piano tariffario sperimentale per cellulari: la compagnia notò che attraverso questa nuova tariffa il numero di nuovi clienti stimati non sarebbe stato sufficiente a coprire il capitale perso per gli sconti ai clienti della nuova tariffa.

Per comprendere a fondo in cosa consiste la simulazione vediamo ora le applicazioni più riuscite della stessa:

- Sicurezza: si considerino per esempio i cosiddetti crash test, prima erano dei dispendiosissimi scontri reali tra macchine vere contro muri appositamente progettati, adesso sono una semplice immagine che si muove su un computer. Questa immagine simula perfettamente, cioè riproduce con un ottimo livello di fedeltà, quello che succederebbe se facessimo schiantare una macchina contro un muro reale.

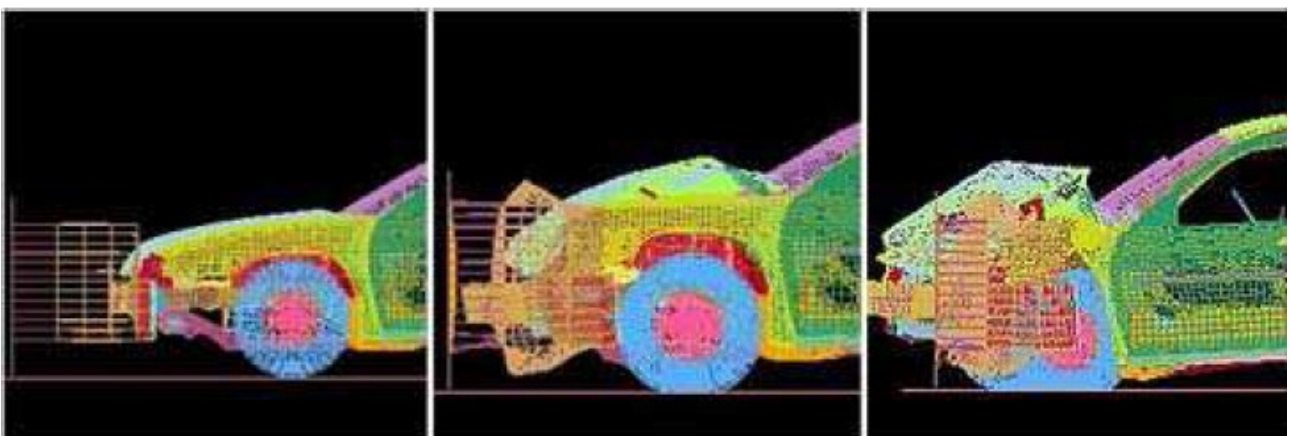


Figura 6.2: Simulazione Crash-Test

- **Efficienza:** possiamo pensare agli studi di aerodinamica. Immaginare di dover produrre dieci diversi tipi di alettoni per macchine da corsa, testarli uno per uno e poi sceglierne solo quello che risulta più adatto degli altri è evidentemente una procedura dispendiosa e con un altissimo livello di spreco. Ad oggi gli alettoni, le chiglie delle navi, i profili delle auto e delle ali degli aeroplani, gli impennaggi delle frecce da gara e molti altri elementi degli attrezzi sportivi sono tutti prima progettati tramite la simulazione e poi realizzati fisicamente.
- **Apprendimento:** software simulatori di volo sono diventati ormai parte integrante nello studio di un giovane apprendista pilota, le meccaniche e i procedimenti di decollo, atterraggio e manovre vengono perfettamente assimilate attraverso l'uso del simulatore.
- **Intrattenimento ludico:** settore sicuramente più giovane è alla ricerca di simulazioni il più realistico possibile a scopo di intrattenimento dell'utente, l'esempio più eclatante è quello di "Second Life", un simulatore di vita reale disponibile on-line. La maggior parte dei videogiochi sono ottimi esempi di questa applicazione.



Figura 6.3: Second Life

- **Mercato/economia:** una determinata azienda potrebbe testare la riuscita del suo nuovo prodotto sul mercato attraverso una simulazione computerizzata. Abbiamo visto in precedenza l'esempio della compagnia telefonica americana.
- **Situazioni particolari:** le camere per la simulazione di condizioni a gravità 0 hanno fatto straparlare i nostri programmi televisivi per la divulgazione scientifica ed il film Apollo 13 ha egregiamente pubblicizzato l'esistenza di stanza per la simulazione dell'attività all'interno di shuttle e basi orbitanti. Andando avanti possiamo pensare all'esplosione che le biotecnologie hanno avuto in questi anni elaborando applicazioni in grado di testare i dettagli di una protesi interna a partire dalla semplice immagine di una radiografia.

In conclusione la simulazione è un procedimento che ha applicazioni nei più svariati campi e nelle più diverse discipline e che si è fatta strumento fondamentale nel progresso scientifico-tecnologico della nostra società. Oggi possiamo immaginare di simulare praticamente qualunque cosa, i più fantasiosi racconti di fantascienza parlano di robot antropomorfi in grado di simulare le emozioni umane; ovviamente la contraddizione giace nella definizione stessa di simulazione: un processo che riproduce con un certo grado di fedeltà un sistema o una situazione della vita reale. Questo è dovuto

al fatto che per effettuare una simulazione vengono definite e codificate una serie di regole che insieme prendono il nome di modello.



Figura 6.4: Camera di simulazione a gravità

6.3 Cosa vuol dire modello?

Un modello è un insieme di regole applicate ad un dominio tale che, dato uno stato iniziale, è possibile prevedere, dopo un certo numero di transazioni temporali, quali saranno le caratteristiche del nuovo stato del dominio. In poche parole, se il nostro sistema è costituito da un grattacielo con un determinata resistenza R ed un vento che soffia sempre dalla stessa direzione con una determinata forza V e le regole imposte sono che ogni secondo la forza del vento aumenta di 1 ed il grattacielo crolla se la forza de vento è superiore alla sua resistenza, dati $R1$ e $V1$ posso con certezza assoluta dire che il grattacielo della simulazione crollerà dopo $V1-R1$ secondi. Se il modello per la simulazione che ho creato è sufficientemente veritiero e quindi rappresenta abbastanza fedelmente la situazione di un vero grattacielo sottoposto all'azione di un vero vento che aumenta costantemente di potenza, c'è un'ottima probabilità che il vero grattacielo crolli se sottoposto per $V1-R1$ secondi al vero vento. Quello appena presentato è evidentemente un esempio banale e poco veritiero, possiamo però riprendere alcuni degli esempi presentati in precedenza e cercare di abbozzare un modello per alcuni di questi. Banalmente possiamo generare un applicazione che ci permetta di definire un numero di posti a sedere P , un numero di servizi sanitari S , ed un coefficiente di bisogno dell'utilizzo dei servizi B . Possiamo immaginare che il sistema invierà una segnalazione di errore ogni volta che cercheremo di aggiungere un posto a sedere che faccia in modo che il rapporto tra i gabinetti e i possibili utenti della struttura sia inferiore a B , in questo modo simuliamo la necessità di un certo numero di servizi per ogni persona che probabilmente entrerà nel nostro edificio. Aggiungendo un livello di complessità alla nostra precedente applicazione immaginiamo di stabilire un coefficiente di difficoltà D per ogni zona di edificio percorribile che possiamo definire all'interno del nostro edificio. Definiamo poi dei coefficienti di abilità A assegnati ad ogni possibile utente della struttura. Il programma dovrà segnalare errore ogni volta che tentiamo di costruire un edificio che permetta l'entrata con un coefficiente di difficoltà D minore di quello dell'uscita o dei percorsi per raggiungere i sanitari.

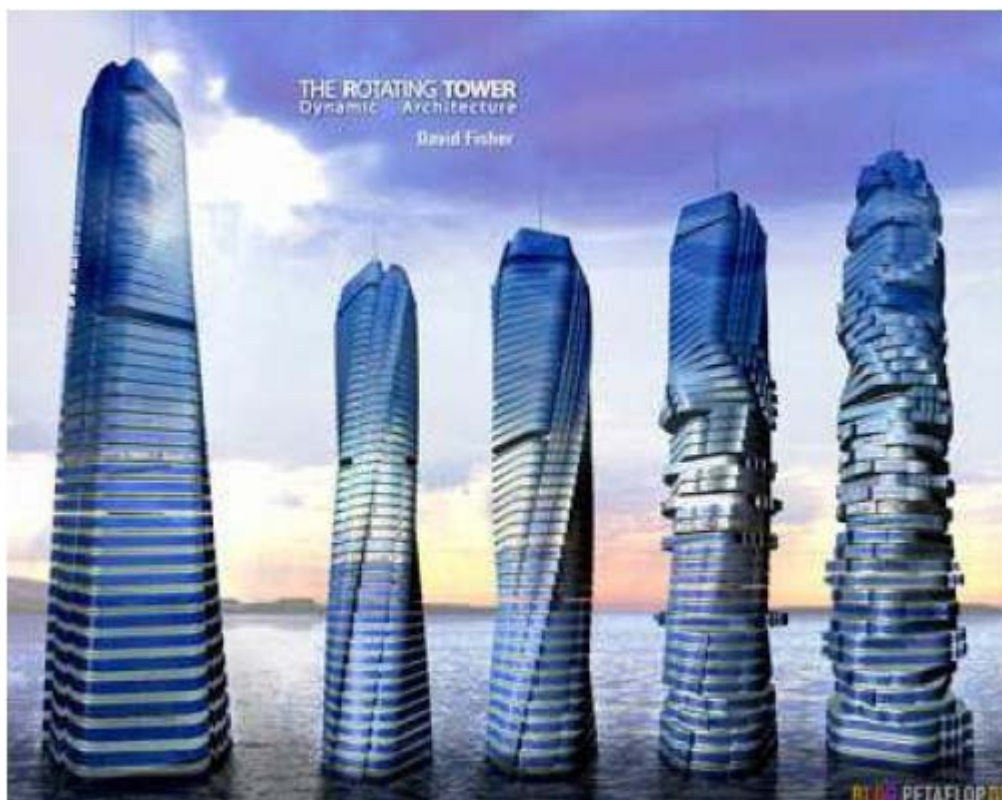


Figura 6.5: Torri rotanti - esempio di architettura dinamica

In questo modo saremo sicuri che l'edificio che andremo a costruire garantirà a chi è in grado di entrarci la possibilità di accedere ai servizi e alle uscite. Ancora, ipotizziamo di avere due strade che vengono imboccate da un certo numero di macchine che dipende dal momento della giornata e che queste strade formino un incrocio regolato da un semaforo. I periodi del semaforo dovranno essere definiti dal rapporto tra il numero di macchine in attesa e quelle che occupano l'incrocio tenendo conto del fatto che questo rapporto varia a seconda dell'ora corrente. Questi sono semplici esempi di quello che un modello può rappresentare. Al giorno d'oggi esistono modelli complessi utilizzati in economia, statistica, demografia ed un'infinità di altri campi di ricerca.

6.4 Complessità e accuratezza: la relazione tra simulazione e modello

Nel paragrafo precedente si è parlato di modelli complessi e di accuratezza di un modello analizzando più esempi, si è quindi introdotto a livello intuitivo il concetto di complessità di un modello e il concetto di accuratezza di un modello. Entrambi i concetti costituiscono il punto di contatto tra una simulazione ed il modello sottostante ad essa: in particolare la complessità di un modello e quindi di una simulazione è rappresentata dal numero di elementi che la simulazione prende in considerazione, dalla grandezza del dominio in cui questi elementi sussistono ed infine dal numero di possibili diverse interazioni tra elementi distinti. Generalmente un modello tende a non essere mai troppo complicato e a tenere conto solamente degli aspetti principali e caratteristici dell'oggetto da simulare, il progetto viene astratto e semplificato il più possibile in modo da ottenere un modello veritiero che risulti comunque facile da maneggiare: è infatti impensabile cimentarsi con complicati sistemi di equazioni differenziali quando la maggior parte delle variabili che entrano in gioco sono poco rilevanti ai fini della valutazione della simulazione stessa. I vantaggi dietro alla simulazione sono proprio quelli di studiare il fenomeno simulato attraverso un modello semplice e gestibile; esagerare nei dettagli e negli elementi da prendere in considerazione diminuirebbe i benefici che traiamo dal concetto stesso di simulazione. L'accuratezza di una simulazione è quindi indicativa della frequenza di eventi che agiscono sul nostro sistema reale che non sono presi in

considerazione nella nostra simulazione. Prendendo l'esempio del nostro grattacielo sottoposto alla forza del vento possiamo definirlo un sistema semplice perché prende in considerazione due dati di cui uno solo variabile in N e una sola relazione banale (maggioranza) tra i due elementi. Per quanto riguarda l'accuratezza di questo sistema è un sistema quanto mai poco accurato in quanto dato un generico grattacielo è quanto mai improbabile che esista un vento che soffia con potenza costantemente in aumento sempre dalla stessa direzione contro di esso, e se anche questo fosse vero ci sarebbero sicuramente un gran numero di altri elementi che entrerebbero in gioco, quali ad esempio l'effetto che un vento in grado di spezzare un grattacielo avrebbe sull'intorno del sistema che stiamo prendendo in considerazione. Il nostro sistema sarebbe già molto più accurato se la potenza del vento non fosse in costante aumento ma semplicemente variabile in maniera casuale nel tempo; il sistema risulterebbe ancora più accurato, nonché più semplice se la forza del vento avesse un valore massimo che può raggiungere. Aumenteremmo l'accuratezza del sistema ma anche la sua complessità inserendo due fattori di oscillazione, uno che rappresenta lo stato di movimento del grattacielo e uno lo stato di variazione della potenza del vento. A questo punto avremmo però bisogno di aumentare il numero di regole che determinano le relazioni tra i vari coefficienti del sistema e quindi il sistema si complicherebbe aumentando così in accuratezza. E' quindi intuitivo e facilmente riscontrabile come i concetti di complessità e accuratezza siano in realtà molto legati tra di loro: in generale è vero che una simulazione è tanto più accurata quanto è più complessa e quindi è complesso il suo modello, anche se in realtà non esiste una regola definita sulla relazione tra questi due fattori. Il livello di dettaglio e accuratezza vengono decisi a priori dall'analista e sono ancora un oggetto di studio e di ricerca di grande importanza.

6.5 Limiti dei modelli per la simulazione

La simulazione, come ogni attività svolta sfruttando le capacità di un calcolatore elettronico, richiede un certo quantitativo di risorse logiche e fisiche. I limiti di questa disciplina sono costituiti esattamente da questa condizione, unitamente alla durata del processo di codifica delle regole del modello. Supponiamo per un istante di voler simulare il sistema innegabilmente più complicato che esiste in natura: l'essere umano. Riduciamo drasticamente le situazioni di vita che questa nostra simulazione vuole rappresentare ed immaginiamo che il nostro automa abbia semplicemente il compito di decidere quale vestito indossare per una indefinita giornata. Dovremmo scrivere delle regole per far sì che l'automa scelga dei capi di abbigliamento:

- della sua taglia
- adatti al lavoro che deve svolgere
- utili rispetto agli elementi con cui dovrà interagire
- che si adattino al gusto personale simulato dal nostro automa

In pratica dobbiamo tenere in considerazione se il vestito è troppo grande, se è troppo piccolo, se deve essere elegante, se deve essere comodo, se deve potersi sporcare, se deve potersi bagnare, se deve essere a norma di legge, se ha una forma piacevole ecc.. Immaginate adesso di dover combinare tutti i possibili accoppiamenti tra biancheria intima, calzini, magliette, pantaloni, camice, giacche, felpe, cappelli, cinte, cravatte, orologi, scarpe, borse e quant'altro possa esistere nel guardaroba di un essere umano. Si tratta di un lavoro enorme che può essere ottimizzato tramite un altro lavoro enorme e si consideri che quanto presentato sopra è esclusivamente il sistema di relazione tra un essere umano e un numero ristretto di oggetti. Quando poi si va a tentare di stabilire una serie di regole per l'interazione tra due esseri umani allora il compito è semplicemente

impossibile, anche perché gli esseri umani sono in grado a loro volta di simulare, quindi si entrerebbe in un discorso che ha le caratteristiche di una ricorsione senza una fine. Ci troviamo di fronte a dei colli di bottiglia di tipo tecnico nei mezzi a disposizione per effettuare la nostra simulazione: anche ammettendo di avere un computer con una memoria capace di processare dei file contenenti tutti i possibili comportamenti umani, ci vorrebbe almeno qualche decina di anni per raccogliere e codificare tutti i dati necessari ad una simulazione pressochè veritiera. Il rapporto tra **complessità** ed **accuratezza** costituisce quindi il **limite stesso della simulazione** e quindi dei modelli su cui questa si basa.



Figura 6.6: Una lince (predatore) e un coniglio (preda)

6.6 Esempi di modello

6.6.1 Il modello preda predatore

La potenza della scienza dei modelli è costituita dalla generalità che un modello dà ad un problema: riprendendo l'esempio banale del grattacielo e del vento, il modello prevede, una forza, una resistenza e un punto di rottura determinato dalla relazione di maggioranza dei due parametri. Questo stesso modello può essere adottato anche su sistemi diversi da quello di un grattacielo esposto al vento: immaginiamo un'altalena ed il peso del bambino che sale sopra, la molla di una bilancia ed il peso massimo che questa può sostenere, una giostra e la forza centrifuga a cui sono sottoposti i suoi seggiolini, e via dicendo. Restringiamo adesso il nostro campo di studi ad uno specifico modello che è stato oggetto di studi e al quale sono state applicate nel corso degli anni una notevole quantità di variazioni per ottimizzarlo alla situazione contingente. Il modello in questione prende il nome di **Modello Preda Predatore** (predator-prey) ed ha come obiettivo quello di simulare l'andamento demografico di due popolazioni immaginando che una tragga nutrimento dall'altra. Indipendentemente dal fatto che si tratti di carote e conigli, lupi o agnelli, vampiri ed umani, nobiltà e plebe, dirigenti di azienda e operai o qualunque altro caso assimilabile a questi, l'obbiettivo per il modello è quello di tenere traccia del cambiamento dei valori di alcuni fattori allo scorrere del tempo. Abbiamo pertanto già definito alcuni aspetti del nostro modello che sono: una popolazione di prede, una popolazione di predatori, e lo scorrere del tempo. Vediamo ora come è stato affrontato il problema da due matematici che hanno vissuto a cavallo tra il 19° e 20° secolo.

6.6.2 Il modello Lotka - Volterra

Alfred James Lotka e Vito Volterra sono due scienziati che nel corso della loro vita si sono dedicati a diversi aspetti della matematica, della fisica, della chimica e della statistica. Pur non collaborando tra loro svilupparono nel 1924 Lotka, e nel 1926 Volterra lo stesso modello per la soluzione della dinamica preda-predatore, per questo si è scelto di attribuire l'origine del modello ad entrambi gli autori. La loro idea fu che le caratteristiche che assicurano la sopravvivenza alle prede e ai predatori



Figura 6.7: A sinistra Alfred J. Lotka e a destra Vito Volterra

sono incompatibili ed è l'interazione dovuta alla lotta per la sopravvivenza a influenzare la dinamica delle popolazioni delle prede e dei predatori. Entriamo ora in dettaglio: la soluzione proposta è costituita da un sistema di equazioni differenziali non lineari del primo ordine:

$$\begin{cases} \frac{dx(t)}{dt} = n_x x(t) - m_x y(t)x(t) \\ \frac{dy(t)}{dt} = n_y x(t)y(t) - m_y y(t) \end{cases}$$

Abbiamo quindi due funzioni che rappresentano il parametro demografico delle due popolazioni al variare del tempo: chiameremo $x(t)$ la popolazione delle prede e $y(t)$ quella dei predatori. Ogni preda ha la facoltà di riprodursi fino ad una volta per ogni unità di tempo, pertanto indicheremo con n_x il tasso di natalità della popolazione delle prede. Ogni predatore ha la facoltà di nutrirsi massimo di una preda per unità di tempo, indichiamo quindi con m_x il tasso di nutrimento che la popolazione dei predatori impone alla popolazione delle prede ossia il tasso di mortalità della popolazione delle prede. Ogni predatore ha la capacità di riprodursi solo se le condizioni circostanti sono favorevoli, cioè se ci sono abbastanza prede per sfamare tutta la popolazione dei predatori, chiamiamo quindi n_y il tasso di natalità dei predatori. Si assume che un predatore, non avendo nemici naturali all'interno del sistema, raggiunga senza difficoltà il termine del suo ciclo di vita e muoia quindi di vecchiaia, chiamiamo m_y il tasso di mortalità della popolazione dei predatori. $dy(t) / dt$ (da ora in poi utilizzeremo la notazione $x'(t)$) rappresenta quindi il tasso di crescita della popolazione delle prede mentre $y'(x)$, quello dei predatori. Riepilogando:

- $x(t)$ = popolazione delle prede al tempo t

- $y(t)$ = popolazione dei predatori al tempo t
- n_x = tasso di natalità delle prede
- n_y = tasso di natalità dei predatori
- m_x = tasso di mortalità delle prede
- m_y = tasso di mortalità dei predatori

Cerchiamo ora di approfondire i nostri studi sull'argomento analizzando il sistema con i tradizionali metodi matematici: Immaginiamo alcune situazioni limite per cercare di capire cosa succederebbe ad esempio se come parametri iniziali una delle due popolazioni fosse di 0 individui, ecco due casi interessanti:

Primo caso limite

Ipotizziamo per esempio un sistema in cui non esiste la specie "predatori", per tanto $y(t) = 0$ per ogni t appartenente a \mathbb{R} . Le nostre equazioni risulterebbero:

$$\begin{cases} \frac{dx(t)}{dt} = n_x x(t) - m_x 0 x(t) \\ \frac{dy(t)}{dt} = n_y x(t) 0 - m_y 0 \end{cases}$$

Pertanto avremmo:

- $x'(t) = n_x x(t)$
- $y'(t) = 0$

da questi due risultati deduciamo che la popolazione di prede, in assenza di predatori crescerà linearmente all'infinito secondo il suo tasso di natalità, situazione che almeno a livello intuitivo rispecchierebbe il vero: la preda non avendo motivo di morire se non quello dell'età ha libertà di riprodursi.

Secondo caso limite

Nel caso opposto, cioè con una popolazione di prede inesistente e quindi $x(t) = 0$ per ogni t appartenente ad \mathbb{R} , otterremmo:

$$\begin{cases} \frac{dx(t)}{dt} = n_x 0 - m_x y(t) 0 \\ \frac{dy(t)}{dt} = n_y 0 y(t) 0 - m_y y(t) \end{cases}$$

Pertanto risulta:

- $x'(t) = 0$
- $y'(t) = -m_y y(t)$

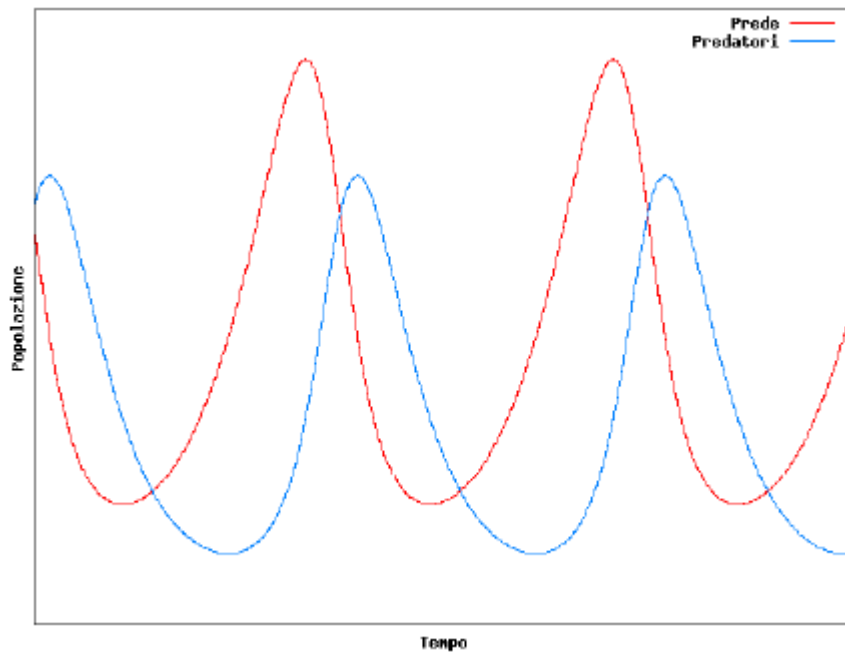
Da cui si deduce facilmente che la popolazione dei predatori si estinguerà a sua volta in maniera lineare secondo il suo fattore di mortalità: il predatore trovandosi a meno della sua principale se non unica fonte di cibo non riesce a sopravvivere. Un altro caso di interesse rilevante è quello dei punti stazionari del sistema, cioè quei punti nei quali entrambe le variazioni demografiche risultano perfettamente uguali a 0:

$$\begin{cases} 0 = n_x x(t) - m_x y(t)x(t) \\ 0 = n_y x(t)y(t) - m_y y(t) \end{cases} \rightarrow \begin{cases} 0 = x(t)(n_x - m_x y(t)) \\ 0 = y(t)(n_y x(t) - m_y) \end{cases}$$

Osservando la forma in cui abbiamo riscritto il sistema è facile intuire che esistono due punti stazionari:

$$\begin{cases} x(t) = 0 \\ y(t) = 0 \end{cases} \rightarrow \begin{cases} x(t) = \frac{m_y}{n_y} \\ y(t) = \frac{n_x}{m_x} \end{cases}$$

Da questo possiamo concludere che le uniche due situazioni in cui il sistema resta invariato nel tempo sono quando otteniamo la completa estinzione oppure quando il numero della nascita delle prede è perfettamente coincidente al numero di prede mangiate dai predatori. Si trascura di dimostrare che il primo di questi due punti (0, 0) non è un punto dall'equilibrio stabile, mentre il secondo è un punto stabile. Da queste affermazioni trarremo soltanto alcune conclusioni senza spendere troppo tempo nella dimostrazione matematica. Un punto stazionario dall'equilibrio instabile è un punto verso cui i valori negli altri punti difficilmente saranno attratti, questo ci garantisce che la coesistenza delle due specie, data una situazione di partenza ragionevolmente distante dall'estremo, difficilmente porterà alla reciproca estinzione. Al contrario la stabilità dell'equilibrio del secondo punto stazionario ci garantisce che il sistema oscillerà periodicamente intorno a questo valore. Scrivere le formule trigonometriche di questa oscillazione non è cosa semplice a livello matematico. Limitiamoci a confermare le nostre conclusioni osservando il grafico dell'andamento del coefficiente demografico delle due specie.



Dal grafico si nota come paradossalmente la situazione stazionaria sia una situazione di tipo oscillatorio: il numero delle prede e dei predatori varierà sempre, mantenendo però il sistema stabile e permetterà alle due specie di coesistere nello stesso ambiente per un periodo di tempo indefinito.

Limiti del modello:

I limiti del modello elaborato da Lotka e Volterra in realtà è intrinseco alla definizione stessa di modello che abbiamo cercato di dare nei paragrafi precedenti: ovvero la semplificazione del sistema ignorandone alcuni fattori. In questo caso le equazioni non tengono conto che:

1. La specie predatori non può consumare un infinita quantità di prede
2. Vengono ignorate le condizioni ambientali, fattore invece fondamentale

Capitolo 7

Automati cellulari

7.1 Introduzione

Abbiamo analizzato a tutti gli effetti il significato di simulazione e affrontato il concetto di modello. Quello che ora vogliamo fare è prendere in considerazione un tipo particolare di modello e studiarne le caratteristiche per avere una concezione e una coscienza più profonda del significato e soprattutto del ruolo che i modelli hanno nello studio delle scienze. La seguente presentazione ha lo scopo di introdurre quelli che vengono definiti gli automi cellulari, un tipo di modello piuttosto recente le cui sorprendenti potenzialità continuano ad essere studiate e ad essere oggetto di ricerca. Ci soffermeremo soprattutto sugli utilizzi che i suddetti automi cellulari hanno per rendere l'idea della vasta gamma di applicazioni nelle più diverse discipline che un modello in genere può avere.

7.2 Definizione

Un automa cellulare (cellular automaton) è un modello discreto che consiste in uno spazio cartesiano di una o più dimensioni (solitamente una griglia) di elementi detti celle. Ogni cella può trovarsi ad ogni istante di tempo t , in un uno ed un unico di diversi stati. Lo stato di una cella varia secondo una o più regole di transizione uguali per tutte le celle: ogni cella cioè viene trattata allo stesso modo. Le regole di cambiamento di stato dipendono dalle altre celle "vicine".

7.2.1 Definizione formale

Un automa cellulare è una quadrupla (D, S, r, R) dove:

- D è la dimensione dell'automa in Z^D . Esso può essere:
 - a 1 dimensione (automa vettore)
 - a 2 dimensioni (griglia, triangolo esagono ecc..)
 - a 3 dimensioni (cubo ecc..)
 - a n dimensioni, con n intero finito
- S è l'insieme degli stati che una cella può assumere per istanza di tempo.
- r è il "raggio di vicinato", ovvero il raggio entro la quale ogni cella si "accorge" dello stato dei proprio vicini: con $r = 1$ in una griglia una cella sarà a conoscenza dello stato delle 8 ad essa adiacente.
- R è l'insieme delle regole che determinano il passaggio da uno stato all'altro (le regole sono le stesse per ogni cella)

7.2.2 Parole chiave

- generazione: ogni volta che le regole vengono applicate e si passa dall'istante di tempo t_i all'istante t_{i+1} viene creata una nuova generazione.
- configurazione: l'insieme degli stati delle celle all'istante t_i

7.3 Le origini del modello

Nei primi anni del 1940 Stanislaw Ulam lavorava presso il Los Alamos National Laboratory sulla crescita della struttura dei cristalli servendosi di una semplice griglia come modello. Contemporaneamente Von Neumann (che allora era suo collega a Los Alamos) stava lavorando su un problema assai più arduo: i "sistemi autoreplicanti": la sua idea originale era quella di un "robot" in grado di poter ricreare un altro robot uguale a se stesso. La collaborazione tra i due nacque quando Ulam si accorse che il collega avrebbe potuto utilizzare i modelli a griglia su cui stava lavorando come modello matematico astratto per i suoi sistemi autoreplicanti. Nacque così il primo sistema integrato con automi cellulari : un autoreplicatore e costruttore basato su automi dal ristretto raggio (solo le celle adiacenti erano considerati i vicini) con 29 stati per cella. Questo sistema è passato alla storia come "universal constructor". Nel 1969 il tedesco Konrad Zuse pubblicò "Calculating System", libro nel quale per la prima volta venne introdotta una visione della fisica discreta come un gigantesco automa deterministico computazionale che lavorava a stati discreti. Questo fu il primo trattato di quella che oggi viene chiamata fisica digitale. Negli anni 70 ritroviamo gli automi cellulari nella rivista Scientific American in un articolo di Martin Gardner: Game of Life. Inventato da John Conway il "gioco" non era altro che un automa cellulare su griglia a 2 stati: vedremo in seguito in dettaglio in cosa consisteva ed il successo che ebbe. Nel 1983 Stephen Wolfram pubblicò il primo di una serie di articoli che investigavano le proprietà nascoste



Figura 7.1: A destra John Von Neumann e a sinistra Stanislaw Ulam

di una nuova classe di automi cellulari ancora sconosciuti che lui chiamò elementary cellular automata. L'inaspettata complessità che si celava dietro alle semplici regole di questi modelli portò Wolfram a pensare che la complessità della natura potesse essere dovuta a meccanismi simili. Lasciata l'accademia Wolfram si dedicò completamente allo studio degli automi cellulari che culminò nella pubblicazione di un nuovo libro: A new kind of science, un trattato di 1280 pagine in cui vengono discusse tutte le scoperte e risultati ottenuti e in cui vengono mostrati i collegamenti che gli automi hanno con le altre discipline. Il libro fu rivisto, riadattato e aggiornato nel 2005 da Rudy Rucker con il nome di "The Lifebox".

7.4 Dettagli e caratteristiche

L'automa cellulare sicuramente più gettonato è quello a 2 dimensioni. Il modo più efficace per rappresentarlo è una griglia bidimensionale su un immaginario foglio infinito con un set di regole da definire. I vicini di una cella sono gli 8 "quadrati" che la toccano: per ogni cella e i suoi corrispondenti vicini abbiamo così 2×2^8 configurazioni possibili: per ognuna di queste possibili configurazioni le nostre regole ci indicheranno come sarà lo stato della cella di mezzo all'istante di tempo successivo. Nella realtà però, (come per i famosi "nastri" infiniti della macchina di Turing) accade che non abbiamo a disposizione un foglio di dimensioni infinite, quindi la nostra griglia sarà finita. Nasce allora il problema di come gestire le celle ai bordi della griglia dato che queste ultime non possiedono 8 vicini come le altre. Una possibile soluzione è quella di lasciare che gli stati delle celle ai bordi siano sempre gli stessi della configurazione di partenza, ma così facendo si andrebbe anche a influenzare l'andamento delle celle ad esse adiacenti. Si potrebbe invece definire queste ultime come celle speciali alle quali attribuire regole diverse, ma questo andrebbe a violare la definizione di automa cellulare stesso che prevede che tutte le cellule siano equivalenti (e quindi

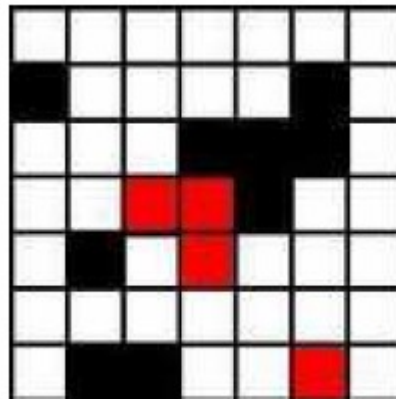


Figura 7.2: Automa cellulare a 2 dimensioni e 3 stati

anche soggette alle stesse regole di transizione). Il metodo più usato è quello del "toro", ovvero considerare la griglia come se fosse mappata su una superficie toroidale (a ciambella) in modo che si finisca alla corrispondente posizione nella griglia in basso se si proviene dall'alto, alla corrispondente posizione nella griglia a destra se si proviene da sinistra. Con questo "trucco" abbiamo risolto il problema delle celle isolate ai bordi costruendo una griglia di dimensioni artificialmente infinite.

7.4.1 Possibilità di un automa cellulare

- ammettere una configurazione stabile: ovvero quella configurazione che resta uguale a se stessa all'istante di tempo successivo (e quindi si stabilizzerà così per sempre)
- essere periodico: si dice periodico di ciclo n quell'automa che dopo n configurazioni consecutive ritorna ad una configurazione particolare C_i e così via, la configurazione stabile è considerata come ciclo di lunghezza 1.
- essere reversibile: se per ogni possibile configurazione all'istante t_i ne esiste una ed una sola precedente. Immaginando un automa cellulare come una funzione che mappa una configurazione in un'altra, la reversibilità può essere intesa come la garanzia che la funzione sia biettiva. Questa caratteristica può essere riscontrata attraverso un algoritmo conosciuto negli automi vettoriali, ma è stato dimostrato che non esiste algoritmo in grado di decidere

con sicurezza se, dato un insieme di regole l'automa corrispondente sia reversibile. In generale la reversibilità è indecidibile per sistemi a 2 o più dimensioni. In alcuni casi di non reversibilità esistono delle particolari configurazioni per le quali non esistono configurazioni precedenti: esse vengono chiamate "Garden of Eden Patterns", in altre parole, non esistono configurazioni che si evolveranno nel GoE pattern.

- essere probabilistico: se per ogni cella la regola ad ogni istanza di tempo si applica con probabilità fissata p ci troviamo di fronte a delle regole non deterministiche.
- essere continuo: viene definito continuo l'automa per il quale vengono utilizzate funzioni continue invece che regole deterministiche per la transizione da uno stato all'altro. Gli stati di ogni cella possono essere per esempio un sottointervallo dei numeri reali (S in $[0,1]$).
- cambiare le sue regole a seconda del tempo o dello spazio: per esempio alcune zone della griglia potrebbero essere soggette a regole diverse o queste ultime potrebbero cambiare dopo un intervallo di tempo prestabilito o anche scelto a caso con probabilità p

7.5 Utilizzi

Il passo in avanti più importante nell'utilizzo degli automi cellulari dopo la loro creazione da parte di Von Neumann e Ulman è stato quello degli studi di Wolfram, che nel corso degli anni 80 si dedicò allo studio delle loro proprietà e con la pubblicazione del libro "A new kind of science" impressionò la comunità della scienza con la dimostrazione dell'utilità e potenzialità dei suddetti modelli. Gli automi cellulari forniscono una rappresentazione immediata (e, in un certo senso, semplice) di fenomeni in cui l'evoluzione globale dipende da leggi locali: la bellezza che si cela dietro ad alcuni sistemi complessi presenti in natura è che le azioni di semplici componenti con comunicazione e informazioni locali danno contributo e coordinazione al processo di evoluzione globale del fenomeno.

7.5.1 Automi e algoritmi decentralizzati

Recentemente è nato un grande interesse nel campo della ricerca nello studiare sistemi decentralizzati come i network di sensori o più sofisticati sistemi microscopici che lavorino su informazioni locali. M. Mitchell, professore in CS alla Portland State University sta studiando attraverso gli automi cellulari algoritmi genetici decentralizzati. Quali sono le caratteristiche affinché un algoritmo sia decentralizzato? Se prendiamo, per esempio, il caso di un grafo, un algoritmo decentralizzato è quell'algoritmo che cerca il cammino tra un vertice sorgente ed uno destinatario che gode di queste proprietà:

- ogni vertice "conosce" la sorgente e il destinatario
- ad ogni passo ogni vertice conosce la lista dei vertici che sono stati passati
- ogni vertice conosce solamente i propri vicini

E' osservando l'ultima proprietà che saltano in mente gli automi cellulari: la località. Mitchell e i suoi collaboratori applicano così algoritmi decentralizzati per evolvere la struttura di un automa cellulare: il suo gruppo di lavoro ha utilizzato un singolo array bidimensionale in cui ogni cella aveva solo 2 stati e 2 vicini (quello destro e quello sinistro). L'array può essere pensato come un cerchio in cui l'ultima cella e la prima sono vicine. L'evoluzione dell'array è stata tracciata attraverso la

variazione di 0 e di 1 (i 2 stati) dopo ogni iterazione: la ricerca si è interessata di conoscere le regioni con più o meno densità di 0 e 1. La motivazione di questo approccio nasce dalla necessità di capire i sistemi naturali e di poter costruire sistemi decentralizzati computazionali. Il gruppo di Mitchell ha mostrato 3 livelli di processazione dell'informazione durante l'iterazione dell'automa:

- il primo livello è quello della trasmissione e "salvataggio" delle informazioni dovuto dalle celle
- il secondo è comprendere la geometria delle subroutines in regioni di media scala come per esempio la grandezza di regioni ad alta o bassa densità di 0 o 1
- L'ultimo livello è la computazione globale del sistema vista "dall'alto".

7.5.2 Automi e crittografia

L'idea del loro utilizzo in crittografia nasce dal fatto che, date certe regole ed una configurazione, è immediato passare da una configurazione all'altra ma non lo è affatto il contrario. E' inoltre molto difficile data una successione di configurazioni trovare le regole che passino da una alla successiva. Questo è un campo di ricerca recentissimo e le cui potenzialità sono ancora sconosciute.

7.5.3 Automi e sistemi biotici naturali

Alcuni oggetti presenti in natura utilizzano automi cellulari nel loro funzionamento. Schemi come quelli di alcune conchiglie di mare, come quelli del *Conus* e della *Cymbiola* sono generati da naturali automi. I pigmenti di ogni regione interagiscono con l'attivazione/inibizione dei pigmenti delle regioni limitrofe obbedendo a regole matematiche precise. Per esempio, i "disegni" della specie *Conus textile* si comportano come un automa cellulare a due dimensioni che segue le "Rules 30".



Figura 7.3: *Conus textile*

7.5.4 Automi e chimica

Negli anni 50 il chimico Zhabotinsky scoprì una particolare miscela facendo reagire bromato di potassio, solfato di cerio (IV) e acido citrico in soluzione di acido solforico diluito. La reazione

dava origine a una spettacolare sequenza geometrica con cerchi concentrici e spirali che si propagavano sulla superficie della miscela chiamata reazione oscillatoria che poteva perdurare per ore. Nel 1998 il professore della Scientific American A. K. Dewdney propose un automa cellulare la cui evoluzione assomigliava veramente alla reazione di BelousovZhabotiksky. Anche il comportamento dei gas perfetti, o il movimento di filamenti di DNA all'interno di una soluzione sono altri esempi di come questi automi entrano in gioco in sistemi di tipo chimico.

7.5.5 Automi e processori nei computer

Alcuni processori odierni (basati sull'architettura di Von Neumann) utilizzano parti integrate che funzionano come automi cellulari in cui le celle comunicano con i vicini attraverso cariche elettriche, vibrazioni o cariche magnetiche.

8 Modelli basati su agenti

8.1 Introduzione

Nei modelli basati su agente (Agent-Based Model = ABM), un sistema è model-lato come una collezione di entità autonome capaci di prendere delle decisioni, queste entità prendono il nome di agenti. Ogni agente valuta la propria situazione e prende delle decisioni in base allo schema di regole che caratterizza il sistema in cui l'agente è inserito. Ogni agente può così eseguire diverse funzioni appropriate all'elemento della vita reale che esso rappresenta, come per esempio produrre, consumare o vendere. Le ripetute interazioni tra gli agenti di un sistema sono l'oggetto principale della ricerca sviluppata tramite la simulazione basata su agente, questa ricerca fa affidamento alla potenza dei calcolatori per analizzare le dinamiche che non sono alla portata dei modelli matematici. Il più semplice fra tutti gli ABM è quello che al suo interno prevede la presenza di agenti tutti dello stesso tipo e delle relazioni che intercorrono tra questi. Anche in un semplice modello basato su agente possiamo notare interessanti sequenze di comportamenti che rivelano informazioni interessanti riguardo le dinamiche reali del sistema che stiamo simulando. Inoltre gli agenti possono essere in grado di evolversi, ossia di mostrare comportamenti imprevedibili. I modelli basati su agente più sofisticati spesso includono al loro interno reti neurali, algoritmi evolutivi o altre tecniche di apprendimento per permettere una simulazione più realistica delle dinamiche di adattamento. Più che uno strumento tecnico i modelli basati su agente sono una "forma mentis" che consiste nel descrivere un sistema dalla prospettiva dei suoi elementi costituenti.

Un certo numero di ricercatori afferma che i classici modelli ad equazioni differenziali siano un'alternativa ai modelli basati su agente, ma questo non è esatto, così come non è esatto che un insieme di equazioni differenziali che rappresentino ognuna il comportamento di un singolo agente sono un modello basato su agente. Un sinonimo di modello basato su agente può essere modello microscopico, che rappresenta un Large-Scale model e può contenere fino a parecchi milioni di agenti, ed una valida alternativa ad esso può essere un modello macroscopico, che spesso è fatto tramite un Desktop-Based model che è più semplice (contiene fino a qualche migliaia di agenti) e può stare in un computer singolo. Poiché ad oggi le simulazioni basate su agente hanno acquisito una certa popolarità, è bene sapere quali sono le loro proprietà ed i loro punti di forza e quali sono le situazioni in cui è il caso di applicarle. Più avanti in queste pagine risponderemo a questa esigenza prima classificando e discutendo dei vantaggi che portano in modelli basati su agente nella simulazione, e poi analizzando diversi esempi. Ma ora vediamo di soffermarci sull'aspetto più importante e di darne una definizione precisa: l'agente, il vero e proprio cuore del modello.

8.2 Cos'è un agente?

Sebbene non esista un consenso definitivo sulle definizioni esatte del termine agente sono più le caratteristiche su cui si è d'accordo che quelle su cui si è in disaccordo. Alcuni modellatori considerano qualsiasi tipo di componente indipendente (software, modello, individuo) essere un agente, altri insistono che il "comportamento" di un agente debba essere adattivo per essere considerato un agente:

L'etichetta agente dovrebbe essere quindi riservata a componenti che in qualche modo imparino dalle loro esperienze e modifichino il loro comportamento in base ad esse. Nel 1997 Casti affermò che un agente dovrebbe contenere entrambe le regole di basso livello che quelle di alto livello per modificare il proprio comportamento ("rules that change the rules"). Da un punto di vista più pratico consideriamo però tutti gli agenti aventi certe caratteristiche:

- 1) un agente è identificabile, un individuo discreto con un set di caratteristiche e regole che ne governino il comportamento e che gode della capacità di fare decisioni.
- 2) un agente è localizzato, esso vive in un sistema in cui interagisce con altri agenti. Gli agenti hanno protocolli di comunicazione e la capacità di rispondere a stimolazioni esterne.
- 3) un agente è "Goal-directed", ovvero ha obiettivi da raggiungere rispettando le regole del suo comportamento
- 4) un agente è autonomo, deve essere in grado di sopravvivere indipendentemente dal contesto e dall'interazione con altri agenti, almeno in specifiche situazioni.
- 5) un agente è flessibile, e può avere la capacità di imparare dalle esperienze modificando le proprie regole di comportamento

Differentemente da quanto abbiamo visto per gli automi cellulari, che sono un caso specifico dei sistemi basati su agente, gli agenti di questi ultimi sono diversi, eterogenei e dinamici negli attributi e nelle regole di comportamento.

8.3 Background degli ABM

I modelli basati su agenti hanno molte connessioni con altri campi, per esempio: le scienze della complessità, le scienze dei sistemi, la dinamica dei sistemi, la computer science, le scienze sociali e molte altre. Gli ABM hanno come radice storica i sistemi complessi di adattamento (CAS) che sfruttavano il modello bottom-up, cioè la vista totale costruita tramite l'unione dei singoli elementi (nel nostro caso gli agenti). Il campo dei CAS era la capacità di adattamento di sistemi biologici in determinate situazioni di emergenza, e studiavano la capacità di auto organizzarsi singolarmente e nell'insieme.

Le principali proprietà di un CAS sono:

- 1) Aggregazione: formazione di gruppi
- 2) Non Linearità: non riducibili a equazioni
- 3) Flussi: c'è la possibilità di trasformazione
- 4) Diversità: i vari agenti sono diversi fra loro

I meccanismi sfruttati da un CAS sono:

- 1) Etichettatura: ogni agente è riconoscibile dagli altri
- 2) Modelli interni: permette agli agenti di ragionare sul mondo che li circonda
- 3) Blocchi costruttivi: permette ai componenti e al mondo di essere costruito su vari livelli.

8.4 ABM vs modelli differenziali

Come abbiamo visto spesso il concetto di modello basato su agente non viene ben compreso e si pensa che sia possibile sostituire gli ABM con i modelli differenziali. Vediamo perché questo non è possibile:

- 1) i modelli differenziali tengono in maniera intrinseca nella loro forma (l'equazione) il comportamento generale del sistema nella sua globalità, mentre gli ABM utilizzano la prospettiva dell'agente, osservando il globale dall'evoluzione del locale. Il modello differenziale ignora il singolo ed è interessato solo al risultato finale.
- 2) se il comportamento e le caratteristiche dei singoli sono sufficientemente diversi tra loro un'equazione differenziali diventerebbe troppo complessa e si dovrebbe servire di un numero di variabili troppo grande per essere utile da studiare, questo non succede con un modello basato su agente.
- 3) un ABM è di tipo bottom-up, il modello differenziale è di tipo top-down

8.5 Perché utilizzare i modelli basati su agente?

Una delle ragioni che ha reso i modelli basati su agente così popolari è che sono di facile implementazione: in pratica, una volta che se ne è sentito parlare, si può programmare un modello basato su agente. Il fatto che la tecnica sia facile da utilizzare non vuol dire però che i concetti siano facili da padroneggiare: nonostante i modelli basati su agente siano tecnicamente semplici risultano essere concettualmente profondi.

Questa combinazione inusuale porta spesso ad un uso improprio della simulazione basata su agente

8.5.1 Benefici dei modelli basati su agente

I principali vantaggi possono essere catalogati in tre gruppi:

- 1) acquisizione di informazioni riguardo fenomeni emergenti
- 2) naturale inclinazione alla descrizione di un sistema
- 3) flessibilità

E' ragionevolmente chiaro, comunque, che la capacità dei modelli basati su agente di interagire con fenomeni imprevisti è quello che determina poi tutti gli altri vantaggi.

1) acquisizione di informazioni riguardo fenomeni emergenti

I fenomeni emergenti sono tutti quei comportamenti che si manifestano a causa delle interazioni tra gli agenti. Per definizione non possiamo ridurre questi fenomeni ad una parte del sistema, in quanto risulta chiaro che il tutto è maggiore della somma delle sue parti proprio a causa delle interazioni che scaturiscono tra le parti stesse. Un fenomeno emergente potrebbe avere proprietà decuplicate rispetto alle proprietà di una singola parte. Per esempio un ingorgo stradale, che è il risultato del comportamento e delle interazioni di diversi autisti, può muoversi nella direzione opposta a quella in cui si muovono le macchina che lo causano.

Queste caratteristiche dei fenomeni emergenti li rendono difficili da comprendere e quasi impossibili da predire, anzi spesso portano a conclusioni intuitivamente inaccettabili. Molti esempi di questo tipo saranno descritti in seguito. I modelli basati su agente sono per loro natura l'approccio canonico allo studio dei fenomeni emergenti: nella simulazione basata su agente si modellano e simulano i comportamenti delle unità costituenti del sistema (gli agenti) e le loro interazioni, in modo tale da poter analizzare i fenomeni dalla loro nascita fino al loro termine mentre la simulazione procede. Il comportamento collettivo di un gruppo è un fenomeno emergente. Sarà utile quindi sfruttare i modelli basati su agente lì dove c'è una buona possibilità che si presentino fenomeni emergenti, come ad esempio:

- 1) Quando il comportamento individuale non è lineare e può essere caratterizzato da soglie, regole if-then, o accoppiamenti non lineari. Descrivere le discontinuità nei comportamenti individuali tramite un'equazione differenziale è difficile.
- 2) Quando i comportamenti individuali mostrano processi legati alla memoria, alla dipendenza dai percorsi, all'isteresi, ai comportamenti non markoviani, o a correlazioni temporali incluso l'apprendimento e l'adattamento l'uso degli ABM è di molto vantaggioso.
- 3) Quando le interazioni tra agenti sono eterogenee e possono generare effetti a catena. Si può presumere di garantire una mescolanza omogenea e globale tra le interazioni grazie a delle equazioni di flusso aggregate, ma la topologia della rete di interazioni può condurre a significative deviazioni dai comportamenti di aggregazione previsti.

- 4) Quando la media non è significativa. Le equazioni differenziali aggregate tendono ad addolcire le fluttuazioni, i modelli ad agente no; questo è importante perché sotto certe condizioni le fluttuazioni possono essere amplificate: il sistema è lineare e stabile ma non per perturbazioni oltre una certa larghezza

Poiché la simulazione basata su agente genera fenomeni emergenti che possono essere analizzati dal principio al termine, è di crescente interesse stabilire cosa costituisce una spiegazione di questo tipo di fenomeni.

2) naturale inclinazione alla descrizione di un sistema

In molti casi gli ABM sono la più naturale forma per descrivere e simulare un sistema composto da entità dotate di comportamenti. Ogniqualevolta si cerca di descrivere una situazione di traffico, di flusso in un mercato o come funzionano organizzazioni sociali gli ABM sono decisamente vicini alla realtà. Per esempio viene più naturale servirsi di questi modelli per descrivere come delle persone si muovono per fare shopping in un supermercato piuttosto che con un equazione che governi le dinamiche della densità delle stesse. La differenza fra il processo di business e le sue attività è un altro naturale esempio per capire quanto sono più naturali a volte gli ABM. Un processo di business è un astrazione, talvolta utile, la quale spesso è difficile da gestire e prevedere per le persone all'interno dell'organizzazione. Il modello basato su agente guarda all'organizzazione dal punto di vista non del business ma da quello delle sue attività, che sono poi quelle che le persone nella realtà svolgono.

3) flessibilità

La flessibilità risiede nel fatto che è facile e veloce cambiare in qualunque momento il numero, il set di attributi/caratteristiche e le regole di comportamento degli agenti.

8.6 Topologia alla base delle interazioni sociali

Gli automi cellulari (semplici modelli basati su agenti) rappresentano molto bene i percorsi di interazione degli agenti con i loro vicini presi come caselle adiacenti all'interno di una griglia, o in altre maniere più sofisticate nel caso dei network come modello base. Lo studio dei network sociali è da molto tempo un campo che ha attratto studiosi di ABM, inizialmente sui network statici, poi con l'avvento del World Wide Web si è passati allo studio di network dinamici, con possibilità di host (agenti) che appaiono e scompaiono dalla rete. Tali progressi sono risultati positivi anche per il passaggio allo studio della formazione dell'opinione, basata sull'interazione con il vicinato. I comportamenti chiave di tale modello sono le repentine transizioni di fase che possono accadere senza preavviso e che stanno a significare un veloce cambio di opinione generale.

8.7 ABM, alcune applicazioni

Vediamo adesso alcuni esempi pratici di come sono stati utilizzati questi modelli.

8.7.1 Evacuazioni

Fughe precipitose di massa indotte dal panico spesso conducono a fatalità per persone che vengono buttate a terra e travolte dalla folla. Questi fenomeni capitano spesso in situazioni come incendi in strutture o risse in luoghi chiusi e talvolta apparentemente accadono senza una ragione ovvia. Esempi recenti includono il panico nello Zimbabwe e al Roskilde Rock concert in Danimarca. La frequenza di queste sventure sembra aumentare di continuo con la crescita della densità di popolazione combinata con le caratteristiche della società moderna in cui eventi a cui partecipano migliaia di persone come concerti pop, eventi sportivi o dimostrazioni sono sempre più frequenti. Le persone colpite dal panico sono ossessionate da impulsi incontrollati individualisti indipendenti da cultura o società. La riduzione di attenzione in questi casi causa spesso che le uscite di sicurezza o quelle meno visibili vengano del tutto ignorate. In aggiunta a questo avviene una sorta di contagio sociale che si trasforma in una sorta di patologia di massa che si dilaga attraverso le azioni delle persone. Questi comportamenti spesso portano a fatalità che devono essere assolutamente evitate.

In termini di agenti, il panico collettivo è un fenomeno emergente risultato dall'interazione delle persone con lo stesso obiettivo a breve termine di uscire dal posto in pericolo nel quale si trovano. Per esempio, consideriamo il caso di un'evacuazione da un edificio chiuso in caso di incendio e assumiamo che esista solo un'uscita. Come si potrebbe aumentare il flusso di persone che esce dall'edificio? Uno si potrebbe anche chiedere: quale sarebbe l'effetto di piazzare una colonna proprio di fronte all'uscita, a un metro dalla porta? Intuitivamente uno potrebbe pensare che la colonna porti ad una diminuzione del flusso di uscita, ma, attraverso gli ABM e ad esperimenti reali si è verificato che la colonna aumenta invece che diminuire il flusso e diminuisce invece il numero di incidenti come persone travolte e calpestate. Eppure evidentemente la colonna gioca un ruolo di regolazione dividendo il flusso in due "sottoflussi" ordinati e costringendo le persone a prestare un'attenzione maggiore per evitare di sbattere contro la colonna che risulterà utile ai fini di controllare il panico.

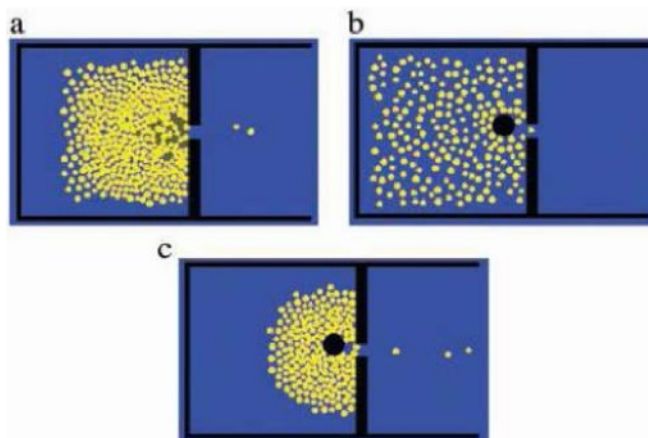
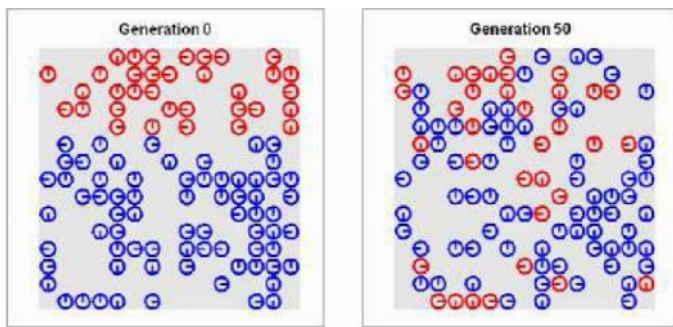


Figura 8.1: Simulazione di evacuazione senza colonna (a) e con colonna (b,c)

8.7.2 Interazioni sociali

In questo esempio ci sono due tipi di agenti: gli agenti blu e quelli rossi. Inizialmente ci sono un numero di agenti blu doppio rispetto a quello di agenti rossi. Ogni parte cerca di convincere l'altra che la sua posizione è migliore ma c'è una clausola: ogni agente rosso ha una capacità di convincere un agente blu doppia rispetto a quella blu. Un agente ha bisogno solo di essere circondato da metà agenti rossi per adottare la posizione di un agente rosso. La simulazione vedrà dominante una parte o si verificherà una sorta di compromesso?



8.7.3 Controllo dei flussi

Un'ovvia applicazione degli ABM nel controllo dei flussi è la gestione del traffico. Uno dei più ambiziosi progetti in questa area è stato intrapreso per molti anni al Los Alamos National Laboratory. Un team della sezione Technology and Safety Assessment ha rilasciato una software di simulazione del traffico metropolitano che possa essere usato dalle agenzie per la pianificazione dei servizi metropolitani.

Il TRansportation ANalysis SIMulation System (TRAN-SIMS) è dotato di una popolazione virtuale che percorre percorsi giornalieri (va a lavoro, per negozi, a scuola ecc..) e crea una regione metropolitana con una completa rappresentazione degli individui della regione, le loro attività e tutte le infrastrutture. Questo "mondo" virtuale imita i comportamenti usuali dei viaggiatori di quelle specifiche regioni. Il software permette determinati tipi di controlli come:

- 1) quando e dove rendere fuori servizio una particolare corsa
- 2) come distribuire le corse attraverso lo spazio disponibile
- 3) quale è la tolleranza nel tempo d'attesa dei viaggiatori
- 4) se e quando estendere le ore di servizio

Nella simulazione gli agenti rappresentano un realistico e modificabile mix di servizi (attrazioni, negozi ecc..) ed esigenze (viaggiatori con differenti preferenze). Inserendo risorse e dati presi dalla realtà come i servizi, i tempi di attesa, il numero di persone, il numero di veicolo e via dicendo il modello genera informazioni sul flusso del traffico metropolitano. L'utente può pianificare ed osservare un numero infinito di scenari per studiare le dinamiche del luogo e i suoi effettivi eventuali accorgimenti. Gli ABM si rivelano un'altra volta uno strumento molto potente perché mappare le preferenze e comportamenti degli agenti da una parte e le caratteristiche dell'ambiente dall'altra sarebbe troppo complesso con un modello matematico.

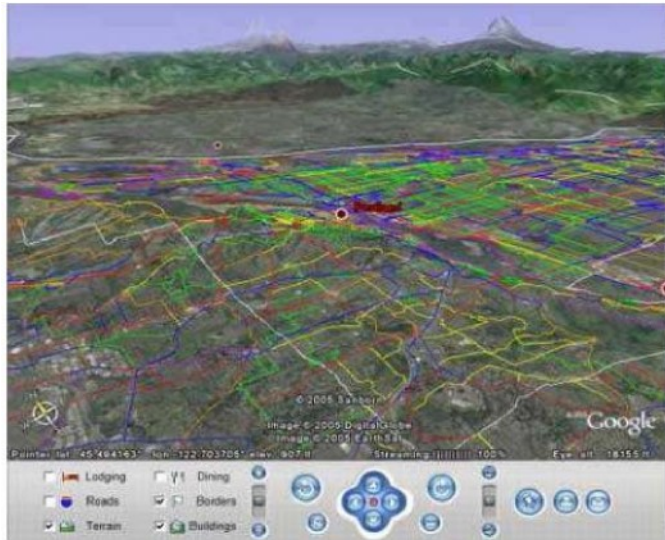


Figura 8.2: Servizi metropolitani dell'Oregon

8.7.4 Organizzazioni

Un'area promettente dell'applicazione degli ABM è quella della simulazione organizzativa. E' certamente possibile modellare il comportamento emergente di un'organizzazione o di una parte di essa in un determinato contesto. Un'organizzazione umana è spesso soggetta a rischi operazionali. Consideriamo le istituzioni finanziarie. I rischi comprendono forme di sistemi informativi in-adequati, problemi nei controlli interni, frodi, perdite di capitale ecc.. Sebbene molte banche si siano dotate spesso di sofisticati sistemi per affrontare rischi di mercato questo non è sempre stato sufficiente e non ha impedito che vere e proprie catastrofi avvenissero. Il problema è che molte banche mancano di una "traccia storica" delle loro operazioni fallimentari e delle cause delle loro perdite. Diversamente dai rischi di mercato, i rischi organizzativi e operazionali sono interni all'organizzazione e una connessione matematica o statistica definita tra i fattori di rischio e la frequenza di operazioni dannose non esiste. C'è una pressione in continua crescita da parte delle istituzioni finanziarie di quantificare i rischi operazionali in modo da convivere entrambi gli investitori e le entità regolatrici.

Più precisamente un'istituzione finanziaria deve essere in grado di quantificare i rischi e tenerli sotto controllo e ottimizzare il capitale economico. Date le caratteristiche dei rischi operazionali una simulazione bottom-up sembra l'approccio ideale. Ernst e Young hanno applicato gli ABM per misurare i rischi manageriali alla società Società' Generale Asset Management (SGAM). Hanno così iniziato a modellare processi di business servendosi delle sezioni di lavoro come agenti identificandone le caratteristiche nelle attività, nelle interazioni fra esse e nella percentuale di operazioni pericolose affidate alle stesse. In altre parole gli ABM risultano perfetti non solo per stimare i rischi operazionali nelle istituzioni finanziarie ma per modellare i rischi di un'organizzazione in generale.

8.7.5 Catena di servizi

Questo modello rappresenta il ciclo di vita dei prodotti commerciali, dalla creazione fino all'acquisto dell'utente finale. Si basa su 5 tipi di agenti:

- 1) Cliente
- 2) Rivenditore
- 3) Grossista
- 4) Distributore
- 5) Azienda

I rapporti fra questi agenti sono: il cliente fa un ordine al rivenditore; il rivenditore dà la merce se la ha in magazzino, altrimenti mette in coda la richiesta del cliente, poi a seconda degli ordini ricevuti fa a sua volta un ordine al grossista e tutto il ciclo si ripete fino all'azienda produttrice che sceglie la quantità di materiale da produrre in base alle richieste che le arrivano.

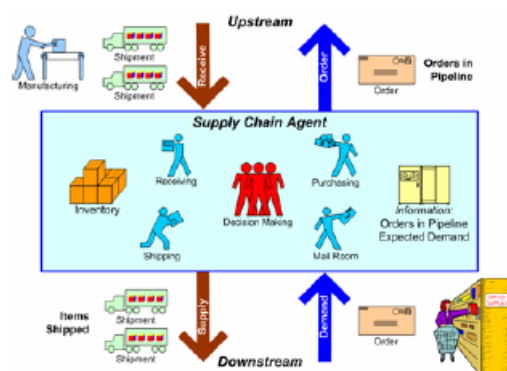


Figure 6: The World of the Supply Chain Agent

L'obiettivo di ogni agente è di gestire i propri magazzini in modo da non pagare troppo lo spazio che deve contenere le proprie merci e nel frattempo non far aspettare troppo a lungo i propri clienti.

In questo esempio di modello ogni agente ha accesso solo a informazioni locali, quindi dovrà fare le sue scelte non conoscendo la situazione globale dell'ambiente circostante, tuttavia, se pur semplice questo modello è un valido fondamento da cui partire per creare modelli più complessi di catene di servizi.

8.7.6 Stormi

Questa simulazione è un buon esempio di come semplici regole possano portare a complessi meccanismi di comportamento.

Le regole sono 3 e sono molto semplici:

- 1) Coesione: ogni volatile cerca di stare nel punto centrale rispetto ai suoi compagni
- 2) Separazione: ogni volatile cerca di non scontrarsi e quindi di non stare troppo attaccato ai suoi compagni
- 3) Allineamento: ogni volatile cerca di trovarsi dietro alla traiettoria di chi guida il gruppo

Date quindi queste semplici regole per ogni agente, regole che si basano sulla situazione dei vicini e non sulla totalità del gruppo, si vede che il comportamento globale sembra coordinato e anche che emergono volatili leader, cioè che guidano il gruppo. Questo semplice modello è anche usato per lo studio del comportamento dei banchi di sardine, oppure (con alcune varianti) di api e formiche.



(a) Initial random configuration



(b) After 500 updates

Figure 3: Boids Simulation

8.7.7 Mercato dell'energia Elettrica

L'EMACS(Sistema di adattamento complessivo del mercato dell'elettricità) è un modello di simulazione basato su agente designato per investigare il ristrutturamento a la deregolamentazione del mercato dell'elettricità, con lo scopo di capire le implicazioni di un mercato competitivo rispetto ai prezzi, disponibilità e affidabilità del servizio. Questo modello rappresenta i partecipanti del mercato come agenti. Tali agenti sono:

- 1) Compagnie generatrici
- 2) Società di domanda
- 3) Compagnie di trasmissione
- 4) Compagnie di distribuzione
- 5) Consumatori
- 6) Regolatori

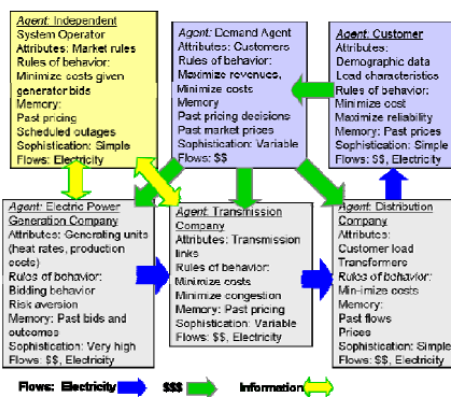


Figure 5: Agents in EMCAS Electric Power Market Model

9 Game of life

L'esempio più conosciuto di automa cellulare è quello di "Game of Life", conosciuto più semplicemente con il nome di Life, ideato dal matematico britannico John H. Conway nel 1970.

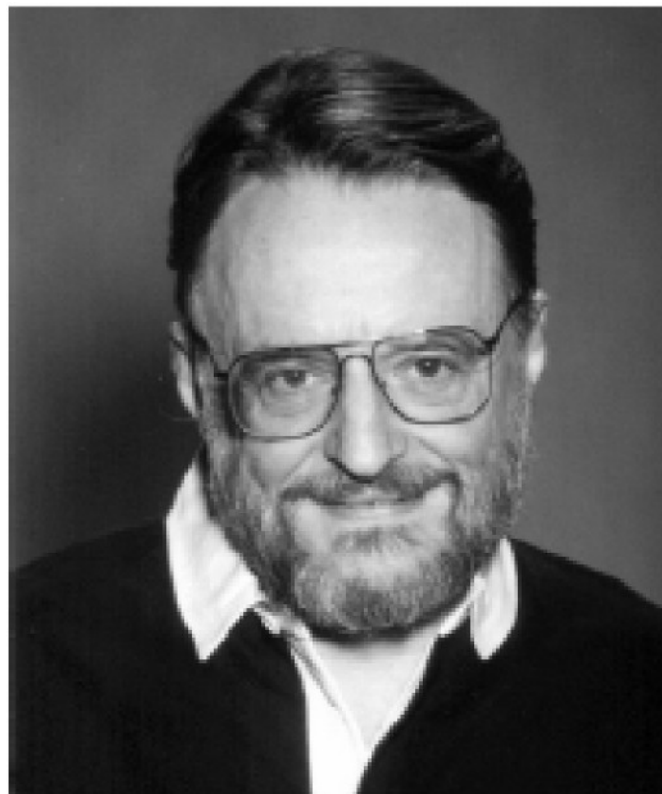


Figura 9.1: John Horton Conway

del gioco è determinata dal suo stato iniziale e non richiede interazioni con un utente. L'unica interazione che si ha con Life è quella di creare una configurazione iniziale ed osservarne l'evoluzione. Il gioco non è altro che un automa bidimensionale sviluppato su griglia ortogonale, nella quale ogni cella può avere uno di due possibili stati: vita o morte. Dichiarando viva la cella colorata di nero e morta la cella non colorata, ognuna di esse interagisce con i propri 8 vicini seguendo queste regole:

- 1) Una cella viva muore se all'istante t ha meno di 2 vicini o più di 3 vicini vivi.
- 2) Una cella viva con esattamente 2 o 3 vicini vivi sopravvive alla prossima generazione.
- 3) Una cella "nasce" (passa da morta a viva) se è circondata da esattamente 3 celle vive.

Gli stati delle celle vengono controllati contemporaneamente per tutte le celle ad ogni istanza di tempo. L'evoluzione del gioco dipende quindi unicamente dallo stato iniziale dell'automa.

9.1 Le origini del gioco

Game of Life nacque da una semplificazione da parte del matematico Conway del modello che Von Neumann aveva utilizzato negli anni 1940 per il suo costruttore universale.






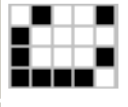
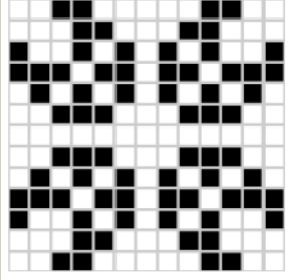
Il successo del gioco avvenne con la sua pubblicazione nella rivista di divulgazione scientifica *American Scientific* nell'ottobre 1970 nella sezione Giochi Matematici di Martin Gardner. Come scrisse lo stesso Gardner: "Il gioco ideato da Conway divenne istantaneamente famoso, ed aprì inoltre un nuovo campo nella ricerca della matematica: il campo degli automi cellulari..." Anche molto tempo dopo la sua pubblicazione il gioco suscitava grandi attenzioni a causa delle sorprendenti vie nelle quali le configurazioni si evolvevano. Il gioco fece quindi da pioniere nella la scoperta dell'importanza degli automi cellulari. E' da sottolineare che quando il gioco fu pubblicato l'evoluzione dell'automa era tracciata a mano su carta (rendendo il lavoro es-tenuante) ma la fortuna di Conway fu proprio l'introduzione nel mercato di una nuova generazione di minicomputer economici che permise a chiunque di farlo girare per ore sulle proprie macchine.

9.2 Esempi di configurazione

Esistono molti tipi di configurazioni possibili in Life, che includono configurazioni stabili ("still lives"), periodiche/oscillatorie, e configurazioni che si autotraslano (Spaceships) attraverso lo spazio della griglia restando invariate.

Modelli basati su agente

Il numero di configurazioni stabili di n celle con $n = 1, 2, 3, \dots$ sono
0, 0, 0,

Block (stabile)	
Boat (stabile)	
Blinker (oscillatore a 2 fasi)	
Toad (oscillatore a 2 fasi)	
Glider (spaceship)	
Lightweight spaceship (spaceship)	
Pulsar (oscillatore a 2 fasi)	

2, 1, 5, 4, 9, 10, 25, 46, 240, 619, 1353... I più diffusi sistemi oscillatori sono di periodo 2, ma si sono trovati casi di oscillatori di periodo 4, 8, 14, 15, 30 e altri più rari. La configurazione "Diehard" si evolve in maniera casuale fino a

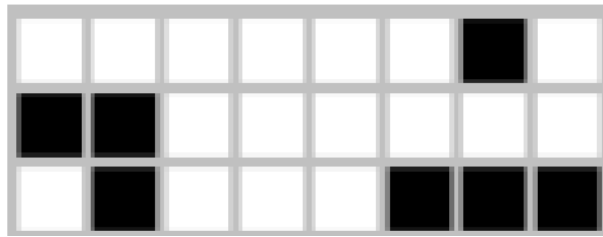


Figura 9.2: Diehard

svanire del tutto dopo ben 130 generazioni. "Acorn" impiega 5206 generazioni per stabilizzarsi in un sistema di 25 Spaceship e altrettanti oscillatori. Questi e altri sono esempi di configurazioni e proprietà intrinseche del gioco ancora oggetto di ricerca. Conway originariamente congetturò che nessuna configurazione iniziale potesse crescere indefinitivamente all'infinito: in altre parole nessun pattern in input con un numero finito di celle vive potesse crescere al disopra di un limite superiore finito. Egli offrì un premio di 50 dollari a chi riuscisse a dimostrare che questa congettura fosse falsa entro la fine del 1970. Un modo per dimostrare la falsità della sua congettura era quello di trovare patterns che "aggiungessero" continuamente altri pattern alla griglia: un "gun" (fucile) vorrebbe essere quella configurazione che "spara" ripetutamente oggetti che si muovono per la griglia (spaceship) mentre un "puffer train" (treno a vapore) la configurazione che lascia dietro di se "fumo" o tracce di detriti persistenti. La sfida fu vinta nel novembre dello stesso anno da un gruppo di lavoro del MIT condotto da Bill Gosper: il Gosper Gun produceva la sua prima struttura autotraslante spaceship dopo 15 iterazioni e un'altra nuova ogni 30. Il Gosper Gun è tutt'oggi la configurazione conosciuta di tipo "gun" più piccola. I pattern che emergono da regole semplici sono considerate una forma di bellezza.

Piccoli sub-pattern isolati senza alcuna simmetria iniziale hanno una naturale tendenza intrinseca a diventare simmetrici, una volta che ciò accade la simmetria tende a diventare sempre più ricca e non verrà mai persa se non per il contatto con altri sub-pattern vicini. Che sia per puro caso fortuito o per intuito, il gioco che nacque dalla mente di Conway si rivelò un vero e proprio successo ed è forse grazie ad esso se si sono introdotti con successo modelli cellulari come strumento per la simulazione.

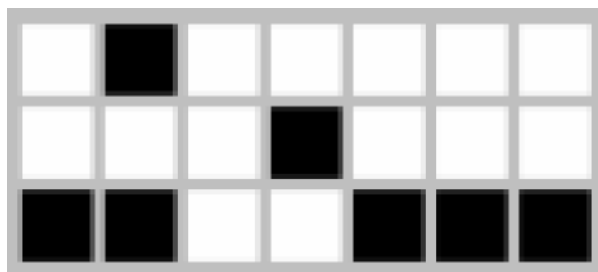


Figura 9.3: Acorn

Domande frequenti per l'esame:

Curve di Bèzier (motivazione, definizione, costruzione, proprietà positive e negative, particolarità, ecc...)

Algoritmo di de Casteljou

Algoritmo del degree elevation

Funzioni B-Spline (def, costruzione, proprietà)

Curve B-Spline (motivazione, definizione, costruzione, proprietà positive e negative, particolarità, ecc...)

Differenza tra la derivata di una curva di Bèzier e la derivata di una curva B-Spline

Trasformazioni euclidee (traslazione e rotazione)

Trasformazioni affini

Trasformazioni proiettive

NURBS (motivazione, definizione, costruzione, proprietà positive e negative, particolarità, ecc..)

Superfici (Bèzier e B-Spline, definizione, costruzione, algoritmi ad esse applicati, ecc...)

G1 continuità (cos'è, quali sono i vantaggi, ecc...)

Conclusioni:

Queste dispense sono state fatte da studenti basandosi sugli appunti presi a lezione e sulle dispense precedentemente disponibili in rete. Per questo possono essere soggette ad errori (speriamo soltanto marginali) anche se sono state ricontrollate più volte.

Si raccomanda quindi una lettura critica e attiva che non dia nulla per scontato, che non si limiti ad imparare a memoria ciò che qui è scritto (cosa che è comunque sempre deprecabile e sconsigliata).

Detto questo, se in futuro, ad altri studenti, venga chiesto nuovamente di modificare tali dispense e quindi si presentasse il bisogno del materiale con cui è stato costruito questo pdf (appunti in .doc , .pub, ecc...) vi invitiamo a contattare gli studenti che hanno collaborato alla stesura (i contatti sono in copertina).

Buono studio a tutti!