

Corso di Approssimazioni Numeriche, A.A. 2007/08

ESERCITAZIONE DI LABORATORIO SU:

Curve di Bézier, algoritmo di de Casteljau e blossoming

Prof. S. De Marchi - 30 gennaio 2008

1 Curve di Bézier e algoritmo di de Casteljau

1.1 Curve di Bézier

Definizione 1. Consideriamo n punti bidimensionali P_0, P_1, \dots, P_{n-1} e sia \mathcal{P} il corrispondente poligono di controllo.

a) La curva di Bézier associata a \mathcal{P} è la curva parametrica

$$B(t) = \sum_{i=0}^m P_i B_i^m(t), \quad t \in [\alpha, \beta]$$

dove $B_i^m(t) := \binom{m}{i} (1-t)^{m-i} t^i$, sono i polinomi di Bernstein di grado m .

b) La curva B-spline di ordine $m < n$ associata a \mathcal{P} è la curva parametrica

$$S(t) = \sum_{i=0}^{n-1} P_i B_{i,m}(t), \quad t \in [\alpha, \beta],$$

con $B_{i,m}(t)$ che indica la i -esima B-spline di ordine m .

1.2 Algoritmo di de Casteljau

L'algoritmo per la costruzione di curve di Bézier è l'**algoritmo di de Casteljau** (descritto ad esempio in Gerald Farin: *Curves and Surfaces for CAGD. A Practical Guide*, Morgan Kaufmann, 2006), anche nel libro di S. De Marchi *Funzioni spline univariate*, Forum Ed., 2001.

Algoritmo 1. Dato un insieme $\mathcal{B} = \{b_0, \dots, b_n\}$ di punti del piano e $t \in \mathbb{R}$ (usualmente $t \in [0, 1]$), il generico punto appartenente alla curva di Bézier si determina con i seguenti passi:

1. Inizializzazione

$$b_i^{(0)}(t) = b_i \quad (1)$$

2. Passo iterativo

$$b_i^{(r)}(t) = (1-t)b_i^{(r-1)}(t) + t b_{i+1}^{(r-1)}(t) \quad r = 1, \dots, n \quad i = 0, \dots, n-r \quad (2)$$

La curva di Bézier calcolata con l'algoritmo 1 è quindi ottenuta con combinazioni baricentriche ripetute. In figura 1 è descritto il funzionamento dell'algoritmo di de Casteljau.

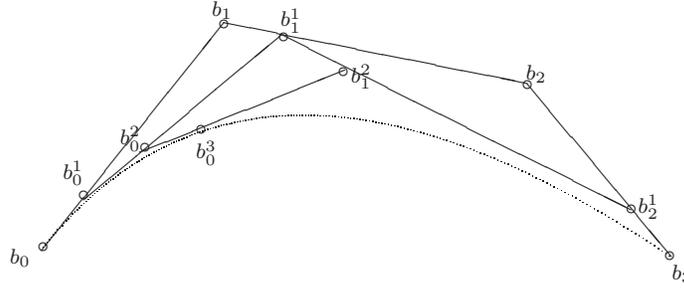


Figure 1: Costruzione di una curva di Bézier con l'algoritmo di de Casteljau.

Proposizione 1. I punti $b_i^{(r)}(t)$ possono essere espressi in termini di polinomi di Bernstein B_j^r di grado r risultando

$$b_i^{(r)}(t) = \sum_{j=0}^r b_{i+j} B_j^r(t) \quad i = 0, \dots, n-r \quad (3)$$

Dim. Induzione su r .

$$\begin{aligned} b_i^{(r)}(t) &\stackrel{(2)}{=} (1-t)b_i^{(r-1)}(t) + t b_{i+1}^{(r-1)}(t) \\ &\stackrel{(ricorr.)}{=} (1-t) \sum_{j=i}^{i+r-1} b_j B_{j-1}^{r-1}(t) + t \sum_{j=i}^{i+r} b_j B_{j-i-1}^{r-1}(t) \end{aligned}$$

Usiamo il fatto che $B_j^r(t) = 0$ se $j \notin \{0, \dots, n\}$. Riordinando gli indici otteniamo

$$\begin{aligned} &(1-t) \sum_{j=i}^{i+r} b_j B_{j-1}^{r-1}(t) + t \sum_{j=i}^{i+r} b_j B_{j-i-1}^{r-1}(t) = \\ &\sum_{j=i}^{i+r} b_j \left[\underbrace{(1-t)B_{j-i}^{r-1}(t) + tB_{j-i-1}^{r-1}(t)}_{B_{j-i}^r(t)} \right] = \sum_{j=i}^{i+r} b_j B_{j-i}^r(t) \end{aligned}$$

Questo conclude la dimostrazione □

1.3 Blossoming

Abbiamo appena provato che l'algoritmo di de Casteljau è essenzialmente equivalente a una curva di Bézier che è una curva parametrica che dipende da un parametro t . Ora, se ad ogni passo r dell'algoritmo di de Casteljau, cambiamo il valore del parametro su cui valutare la curva, otterremo non più il valore sulla curva di Bézier, ma sulle *forme polari* corrispondenti.

Ad esempio, se abbiamo un poligono di controllo formato da $n = 4$ punti, con $b_0^4(t)$ finora abbiamo indicato il valore sulla curva di Bézier, che potremo anche indicare con $b_0^4[t^{(4)}]$ per sottolineare che è ottenuto usando 4 volte lo stesso parametro t . Se al primo passo dell'algoritmo di De Casteljau, prendessimo un valore differente, t_1 , il valore $b_0^4[t_1, t^{(3)}]$ indicherebbe il valore sulla *prima polare*. Così via per le altre tre polari. In particolare $b_0^4[t_1, t_2, t_3, t_4]$ indica il *blossom* della curva di Bézier.

Per completezza, ricordiamo che $b[0^{(n)}], b[0^{(n-1)}, 1], \dots, b[1^{(n)}]$ indicano i vertici del poligono di controllo.

2 Esercitazione proposta

1. Preso un insieme di punti sul piano, implementare l'algoritmo di De Casteljau.
2. Implementare quindi l'algoritmo di De Casteljau per il calcolo di tutte le forme polari associate.

Come esempio, riportiamo il codice per rappresentare la curva di Bézier e della sua prima polare ottenuta scegliendo $t_1 = .2$.

```
clear;

s=[1 2 4 10; 0 2 .8 -1];
plot(s(1,:),s(2,:),'bo-.');
p=s;
t=0:.01:1; t1=.2; tt=[t;t;t]; n=size(p,2);

% Curva di Bezier
for h=1:length(t),
for i=1:n-1,
    for j=1:n-i,
        p(:,j)=(1-tt(i,h))*p(:,j)+tt(i,h)*p(:,j+1);
    end;
end;
bb(:,h)=p(:,j);
end;

% Prima polare
p=s; tt=[t;t;t1*ones(1,length(t))];
for h=1:length(t),
for i=1:n-1,
    for j=1:n-i,
        p(:,j)=(1-tt(i,h))*p(:,j)+tt(i,h)*p(:,j+1);
    end;
end;
bb1(:,h)=p(:,j);
end;
```

```
hold on;  
plot(bb(1,:),bb(2,:),'r-',bb1(1,:),bb1(2,:),'g+-'); hold off;
```

Tempo assegnato: 2 ore.

3 Approfondimenti

Indichiamo alcuni importanti riferimenti per poter approfondire e ampliare la conoscenza degli argomenti trattati nell'esercitazione.

3.1 Curve di Bézier e geometria computazionale

A parer mio, il libro di Gerald Farin: *Curves and Surfaces for CAGD. A Practical Guide*, Morgan Kauffmann, (2006) è senza dubbio quello che affronta in maniera sistematica e matematicamente leggibile il settore della geometria denominato della *geometria computazionale*. I vari capitoli, che spaziano dall'1D al 3D, sono davvero un manuale di riferimento per chiunque voglia comprendere come applicare il computer per la soluzione di problemi geometrici.

3.2 Blossoming

L'articolo più importante che collega le forme polari e il blossom di una curva parametrica, è l'articolo di Lyle Ramshaw, *Blossoms are polar forms* (1989), che si può scaricare al link <http://www.hpl.hp.com/techreports/Compaq-DEC/SRC-RR-34.html>. Consiglio anche il seguente interessante lavoro di Hans-Peter Seidel, *An introduction to Polar Forms*, anche questo disponibile in rete al link www.cs.uiuc.edu/class/sp05/cs419/polarforms.pdf.