

TIPOGRAFIA DIGITALE: CURVE E LINEE SOTTO GLI OCCHI DI TUTTI

L'uso massiccio del personal computer e delle stampanti ci pone quotidianamente di fronte a una serie di caratteri tipografici che spesso utilizziamo senza renderci bene conto da dove essi derivino e come siano stati progettati. In realtà, quando scriviamo sul PC, i caratteri sono già codificati nel software che adoperiamo, e forse ignoriamo che è stato necessario definire e progettare un algoritmo al fine di realizzare quella particolare figura geometrica che risponde al nome di una data lettera dell'alfabeto, di un numero o di un altro segno.

Sappiamo fin da ragazzi che mentre scriviamo la penna traccia linee di vario genere costituite essenzialmente da tratti rettilinei e da curve le cui caratteristiche distinguono un font da un altro. Generalmente le tecniche per realizzare tali curve consistono nel fissare un certo numero di punti e nel tracciare la curva «migliore» che possa interpolarli in base a opportuni criteri matematici che possono specificare la sua regolarità, la curvatura o altre caratteristiche di un dato carattere.

Nella moderna tipografia digitale viene introdotto il concetto di *metafont*: anziché costruire una volta per tutte un insieme di linee per ogni carattere tipografico, si sceglie di definire una curva come un oggetto che dipende da uno o più parametri, attraverso i quali si possono ottenere quelle variazioni che caratterizzano i vari tipi di font.

In questo lavoro cercheremo di introdurre tale concetto applicandolo alla costruzione della lettera *esse*, la cui forma particolarmente sinuosa ha fornito da sempre materia di studio dal punto di vista tipografico. A tal fine useremo concetti matematici abbastanza semplici, basati sulla geometria analitica e quindi alla portata di uno studente di scuola superiore, che potrà quindi vedere questa materia applicata alla determinazione di qualcosa di tangibile e magari provare a cimentarsi nella progettazione di nuove forme tipografiche.

1. LA ESSE DI FRANCESCO TORNIELLO

La nostra storia ha inizio alcuni secoli fa, quando nel 1517 il tipografo italiano Francesco Torniello pubblicò un alfabeto fornendo dei metodi geometrici per tracciare le lettere usando le tecniche e gli strumenti dell'epoca, cioè riga e compasso. È interessante tradurre la costruzione che egli realizzò per la lettera *esse*, nel linguaggio della geometria analitica [1].

Si comincia con il definire una griglia suddivisa in 9×9 quadrati unitari, i cui punti possono essere rappresentati con le usuali coordinate cartesiane (x, y) con $0 \leq x \leq 9$ e $0 \leq y \leq 9$. Nel lavoro di Torniello il contorno della *esse* è definito da 14 punti base

$P_i(x_i, y_i)$ attraverso i quali verranno tracciati archi di circonferenza e segmenti che andranno a delineare il contorno della figura. In tabella è descritta la procedura per delineare il contorno della figura. Si osservi che in qualche caso i punti base sono ricavati dalle particolari condizioni geometriche poste alla costruzione.

Punti base	Dati ausiliari	Procedura
$P_1(4.5, 9)$	Centro $C_1(4.5, 5.5)$, raggio $r_1 = 3.5$	Arco C_1, r_1 da P_1 fino a $P_2(6, y_2)$ Quindi $y_2 \cong 8.66$
$P_2(6, 8.66)$		
$P_3(6.5, 8.5)$	$C_2(6.5, 9)$, $r_2 = 0.5$, $Q(7, 9)$	Arco C_2, r_2 da P_3 a Q
$P_4(6, 7)$		Segmento P_4P_5 tangente all'arco C_2, r_2 in P_5 . Si determina P_5 (vedi problema)
$P_5(6.94, 8.76)$		
$P_6(4, 9)$	$C_3(4, 7)$, $r_3 = 2$	Arco C_3, r_3 da P_6 a $P_7(3, y_7)$ con $y_7 < 7$ Quindi $y_7 \cong 5.27$
$P_7(3, 5.27)$		
$P_8(5, 4)$		Segmento P_7P_8
$P_9(3.5, 6)$	$C_4(4.5, 7.12)$, $r_4 = C_4 - P_9 $	Arco C_4, r_4 da P_4 a P_9
$P_{10}(6, 4.5)$		Segmento P_9P_{10}
$P_{11}(3, 0.5)$	$C_5(4.5, 2.5)$, $r_5 = 2.5$	Arco C_5, r_5 da P_{10} a P_{11}
$P_{12}(1.87, y_{12})$	$C_6(2.5, y)$, $r_6 = 1$	Arco C_6, r_6 da P_{11} a P_{12} Quindi $y \cong -0.37$ $y_{12} = 0.41$
$P_{13}(4.5, y_{13})$	$C_7(4, y)$, $r_7 = 2$	Arco C_7, r_7 da P_8 a P_{13} Quindi $y = 2.27$ e $y_{13} = 0.33$
$P_{14}(x_{14}, 2)$	$C_8(4.5, y)$, $r_8 = 2$	Arco C_8, r_8 da P_{13} a P_{14} Quindi $y \cong 2.33$ e $x_{14} \cong 2.53$
$P_{14}(2.53, 2)$		Segmento $P_{12}P_{14}$

Dalla costruzione di Torniiello non si evince quale curva debba collegare P_1 a P_6 e P_2 a P_3 anche se in prima approssimazione si può tracciare un segmento in entrambi i casi. La *esse* così come realizzata da Torniiello è illustrata in Figura 1, mentre in Figura 2 è riportata quella da noi ottenuta con il software *Mathematica*.

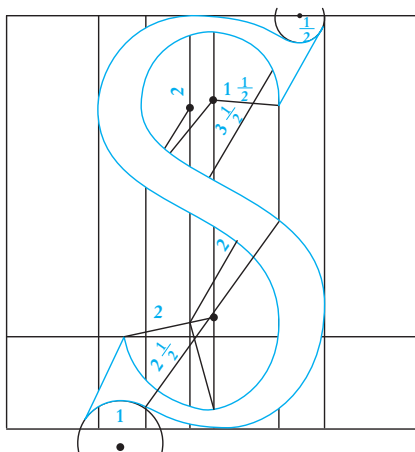


Figura 1 - La esse di Torricelli con i relativi centri e raggi dei vari archi

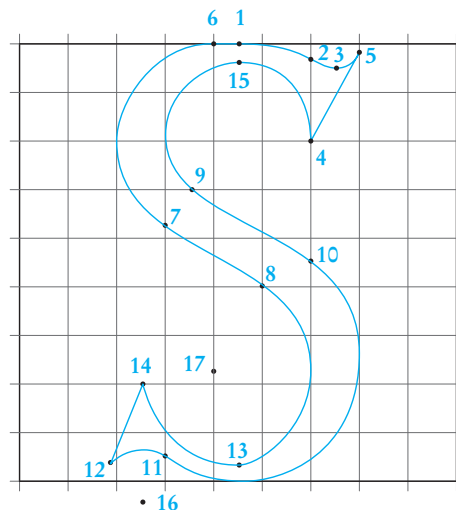


Figura 2 - La versione implementata con Mathematica con i 14 punti di controllo

La costruzione del punto P_5 può servire per discutere un problema di geometria analitica. Data una circonferenza di raggio r centrata nell'origine O degli assi e assegnato un punto esterno $P_4(-r, -b)$, chiediamo di determinare le coordinate del punto $P_5(x, y)$ in modo che il segmento P_4P_5 risulti tangente alla circonferenza in P_5 (Figura 3). Si osservi dalle Figure 1 e 2 che l'altro segmento di tangente risulta parallelo all'asse y , cosa che semplifica l'impostazione e la soluzione del problema. Tuttavia, in un contesto più generale, il punto P_4 può essere in una generica posizione.

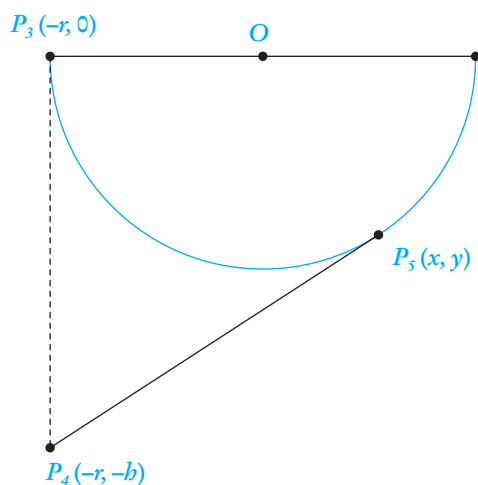


Figura 3

Uno dei tanti modi di procedere è osservare che, per una nota proprietà, i segmenti delle due tangenti condotte da P_4 alla circonferenza sono congruenti, quindi $P_4P_5 = |b|$. Il punto P_5 si ottiene pertanto come intersezione della circonferenza di centro O e raggio r e quella di centro P_4 e raggio $|b|$. Occorre risolvere allora il sistema

$$\begin{cases} x^2 + y^2 = r^2 \\ (x+r)^2 + (y+b)^2 = b^2 \end{cases} \quad (1)$$

le cui soluzioni danno le coordinate di $P_5 = \left(\frac{b^2r - r^3}{b^2 + r^2}, -\frac{2br^2}{b^2 + r^2} \right)$ e di $P_3 = (-r, 0)$. Le coordinate di P_5 nella tabella a pagina 116 si ottengono per traslazione delle precedenti con il vettore $OC_2(6.5, 9)$ e $r = r_2 = 0.5$.

La tabella a pagina 116 evidenzia come la costruzione di Tornielo si presti a essere descritta con un linguaggio di progetto detto *metafont*, utile per definire le forme dei caratteri tipografici a partire da alcuni parametri e facilmente implementabile con *Mathematica*, ottenendo come primo risultato la Figura 2. Particolare attenzione è stata posta al tracciamento degli archi, attraverso un'istruzione che ha codificato il centro C della circonferenza, il suo raggio r e gli angoli α_1 e α_2 che individuano i punti estremi dell'arco definendone il verso di percorrenza (orario o antiorario a seconda dei casi). Inoltre, poiché il software consente di variare il rapporto di aspetto, ovvero il rapporto altezza/larghezza della figura, è stato possibile fare qualche esperimento per valutare la gradevolezza del risultato finale. Qualche esempio è illustrato in Figura 4.

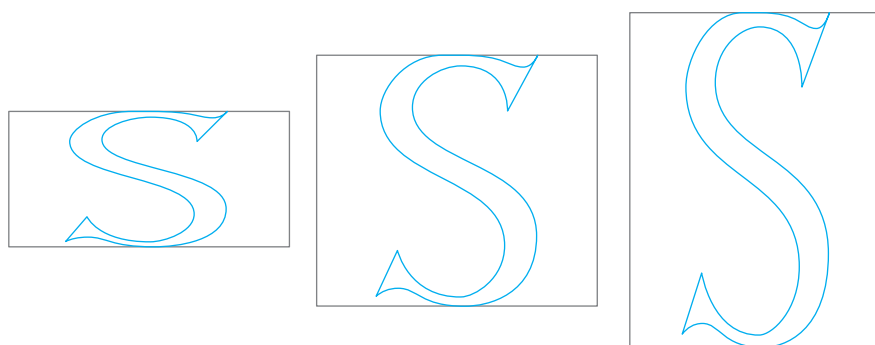


Figura 4 - Alcune esse ottenute con l'algoritmo descritto nella tabella a pagina 116 al variare del rapporto di aspetto (rispettivamente 0,5, 0,9, 1, 1)

La forma della *esse* così ottenuta, anche se differente da quella dei caratteri che solitamente usiamo dalla tastiera del nostro PC, soddisfa però abbastanza bene criteri estetici non particolarmente rigorosi o esigenti. In realtà un'analisi più attenta della Figura 2 mostra un difetto non del tutto trascurabile. Esso consiste in un cambiamento di direzione, ovvero in una discontinuità della derivata nel punto P_7 tra l'arco che unisce P_6 a P_7 e il segmento P_7P_8 . Difatti la direzione della tangente nel punto $P_7(3, 5.27)$ dell'arco di centro $C_3(4, 7)$ e raggio $r_3 = 2$ è data dall'inverso di $-(7 - 5.27) / (4 - 3)$ cioè -0.578 , mentre la direzione di P_7P_8 è $(5.27 - 4) / (3 - 5) = -0.635$. Questa angolosità, particolarmente evidente anche nei punti 9 e 13 oltre che in 8, 10 e 11, sta a significare che in questi punti la curva risulta non derivabile e quindi la lettera *esse* non è perfettamente liscia.

2. LA COSTRUZIONE DI UNA ESSE LISCIA

La costruzione di Torniello può essere generalizzata sostituendo gli archi di circonferenza con archi di ellisse che possono essere uniti a segmenti rettilinei. Il problema di evitare eventuali punti angolosi può essere posto in questi termini: immaginiamo di disegnare una *esse* partendo da un punto in alto $T(x_T, y_T)$ e di tracciare un arco di ellisse verso sinistra fino a un punto $L(x_L, y_L)$. Tale arco dovrà poi prolungarsi fino a congiungersi con un segmento AC , di assegnata direzione m e passante per $C(x_C, y_C)$, in modo da risultare tangente all'ellisse di centro $O(x_T, y_L)$ in $A(x, y)$, come illustrato in Figura 5.

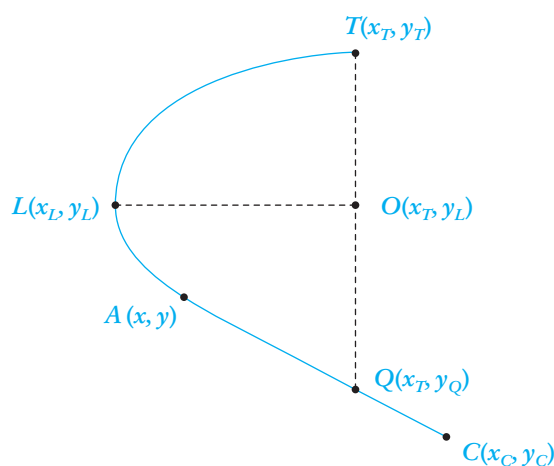


Figura 5

Il problema è quindi quello di determinare le coordinate x, y e y_L , note x_T, y_T, x_L, m, x_C e y_C . La sua soluzione si ottiene dalla soluzione del seguente sistema nelle incognite x, y e y_L :

$$\begin{cases} \left(\frac{x - x_T}{x_L - x_T} \right)^2 + \left(\frac{y - y_L}{y_T - y_L} \right)^2 = 1 \\ \frac{y_C - y}{x_C - x} = m \\ - \left(\frac{y_T - y_L}{x_L - x_T} \right)^2 \frac{x - x_T}{y - y_L} = m \end{cases} \quad (2)$$

La prima del sistema (2) è l'equazione canonica dell'ellisse, la seconda è la definizione del coefficiente angolare m , la terza si ricava differenziando la prima e ponendo la derivata $dy/dx = m$, in modo che la curva sia derivabile in A , eliminando pertanto eventuali angolosità della curva come illustrato in Figura 5. La soluzione del sistema (2) assume una forma più compatta e implementabile se consideriamo un punto $Q(x_T, y_Q)$ su AC in modo da avere $y_Q = y_C + m(x_T - x_C)$:

$$\begin{aligned} x &= x_T + \frac{2m(x_L - x_T)^2(y_T - y_Q)}{m^2(x_L - x_T)^2 + (y_T - y_Q)^2} \\ y &= y_Q + \frac{2m^2(x_L - x_T)^2(y_T - y_Q)}{m^2(x_L - x_T)^2 + (y_T - y_Q)^2} \\ y_L &= y_T - \frac{-m^2(x_L - x_T)^2 + (y_T - y_Q)^2}{2(y_T - y_Q)} \end{aligned} \quad (3)$$

L'algoritmo consente la scelta di sei parametri $(x_T, y_T, x_L, m, x_C, y_C)$ che definiscono le coordinate dei punti di controllo, quindi determina $Q(x_T, y_Q)$, calcola y_L , che definisce il punto *semilibero* L , e infine il punto di contatto $A(x, y)$. È possibile allora tracciare la prima parte della curva (la pancia di sinistra della *esse*); la seconda parte (la pancia di destra) è stata invece ricavata ripetendo la procedura ma con una scelta di soli tre parametri (x'_T, y'_T, x'_L) , avendo posto $m' = m$ e $C'(x'_C, y'_C) \equiv C(x_C, y_C)$. La scelta di altri nove parametri consente di disegnare una nuova sagoma di *esse* affiancata alla prima, in modo da poter tracciare una curva chiusa. Alcuni output del programma sono illustrati in Figura 6 al variare del rapporto di aspetto.

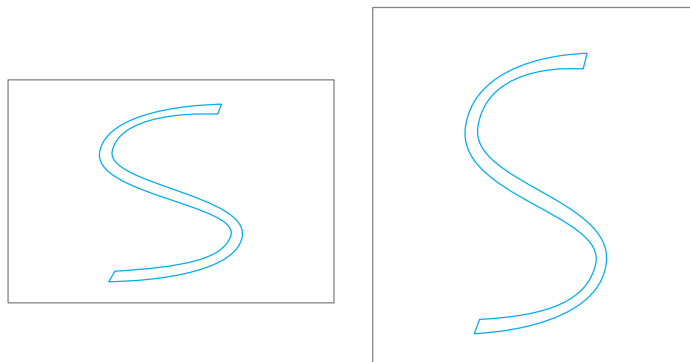


Figura 6 - Alcuni esperimenti grafici sulla *esse* costruita con archi di ellisse, usando 18 parametri di controllo e due diversi rapporti di aspetto

Questi risultati, anche se non pienamente convincenti sul piano estetico, mettono però in rilievo i vantaggi di un approccio parametrico alla modellizzazione di un dato carattere tipografico: il *metafont* infatti consente di effettuare svariati esperimenti grafici al fine di determinare i parametri più appropriati del carattere in questione.

3. CURVE PARAMETRICHE PIÙ GENERALI

Nei paragrafi precedenti abbiamo evidenziato alcune problematiche legate al tracciamento di un dato carattere tipografico la cui forma è definita da un insieme di curve che si raccordano in base a certi punti di controllo scelti dall'utente. Da questo punto di vista, una tecnica matematica particolarmente efficace consiste nel determinare le cosiddette curve interpolanti (*spline*) di Bézier, dal nome dell'ingegnere della Renault che per primo le introdusse nel 1960, al fine di trattare matematicamente il disegno delle scocche aerodinamiche delle autovetture. La curva di Bézier è una particolare forma di polinomi a tratti: per rappresentare una curva di elevata complessità attraverso una funzione matematica relativamente semplice, si deve suddividere questa curva in tanti tratti, ognuno dei quali sia sufficientemente regolare, in modo da individuarvi una funzione matematica che descriva la curva stessa.

Una cubica, per esempio, è una curva con al massimo due «pance» – proprio come la nostra *esse* – individuata da due punti estremi di ancoraggio più due punti esterni alla curva che permettono di tracciare la tangente alla curva all'attacco dei punti di ancoraggio. Questo permette tramite una «maniglia» a esso collegata di modificare la curvatura della linea da descrivere e la direzione della tangente, in modo da ricalcare qualsiasi curva a pancia singola o doppia che passi tra i due punti di ancoraggio.

Nel caso più semplice una curva B di Bézier è definita da quattro punti di controllo P_0, P_1, P_2, P_3 : due (P_0, P_3) di definizione (origine e destinazione della curva) e due (P_1, P_2) vincolati ai primi da opportune condizioni e non necessariamente appartenenti a B . In Figura 7 sono mostrati due esempi di curve di Bézier definite rispettiva-

mente da sette (a) e cinque punti di controllo (b). Esse richiamano alla mente delle rudimentali esse e generalizzano il procedimento di costruzione descritto nel paragrafo 2.

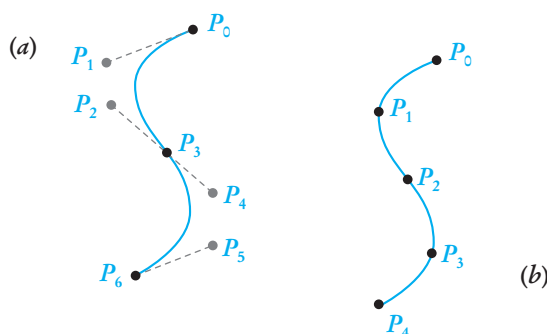


Figura 7

Con curve definite da quattro punti $P_i(x_i, y_i)$ con $i = 0, \dots, 3$, una condizione molto usata è quella di richiedere che i segmenti P_0P_1 e P_2P_3 siano tangenti a B nei punti P_0 e P_3 . La curva di Bézier è allora descritta da due polinomi cubici mediante le seguenti equazioni parametriche:

$$\begin{aligned} x(t) &= x_0(1-t)^3 + 3x_1t(1-t)^2 + 3x_2t^2(1-t) + x_3t^3 \\ y(t) &= y_0(1-t)^3 + 3y_1t(1-t)^2 + 3y_2t^2(1-t) + y_3t^3 \end{aligned} \quad (4)$$

dove $0 \leq t \leq 1$. Si osservi che, quando $t = 0$, abbiamo $(x, y) = (x_0, y_0)$, mentre $(x, y) = (x_3, y_3)$ quando $t = 1$; quindi la curva inizia in P_0 e termina in P_3 . È abbastanza semplice verificare le precedenti condizioni di tangenza ricordando ora che $dy/dx = (dy/dt) / (dx/dt)$ e ottenendo dalle equazioni (4):

$$\begin{aligned} \left(\frac{dy}{dx}\right)_{t=0} &= \frac{y_1 - y_0}{x_1 - x_0} = m_{P_1P_0} \\ \left(\frac{dy}{dx}\right)_{t=1} &= \frac{y_3 - y_2}{x_3 - x_2} = m_{P_2P_3} \end{aligned}$$

Linee più complicate possono essere ottenute concatenando due o più curve di Bézier. Per esempio, la Figura 7a può essere creata congiungendo una prima curva B_1 avente quattro punti di controllo $P_0P_1P_2P_3$ con una seconda B_2 definita dai punti $P_3P_4P_5P_6$. La regolarità è assicurata dal fatto che le tangenti in P_3 a B_1 e B_2 devono coincidere, quindi i punti $P_2P_3P_4$ devono appartenere alla tangente comune. I punti $P_0P_1P_2P_3P_4P_5P_6$ vanno a definire quello che si chiama poligono di controllo della curva; più la sua forma è complessa, più aumenterà il numero di vertici di questo poligono e quindi il grado del polinomio. Combinando opportunamente curve di pri-

mo, secondo e terzo grado, si possono rappresentare anche i più complessi tracciati bidimensionali. Ecco perché le curve Bézier sono alla base della grafica vettoriale.

Quando iniziò la fotocomposizione, uno degli ostacoli da superare era la descrizione del contorno delle lettere che risultava scalettato nei forti ingrandimenti, perché non si era ancora trovato un algoritmo che ne permettesse un disegno regolare. Per esempio, il linguaggio *PostScript* e anche programmi di modellizzazione geometrica come il *CAD* fanno uso delle curve di Bézier che, oltre a presentare un notevole grado di flessibilità e regolarità, sono anche alla base della grafica vettoriale di molti pacchetti applicativi che utilizzano il *PostScript*.

Il linguaggio *PostScript* si serve di questa rappresentazione per descrivere le lettere, che non sono altro che un insieme molto complesso di curve la cui unione forma il segno scritto. I caratteri non sono memorizzati come immagini definite per punti (*bitmapped*), ma piuttosto vengono implementate le equazioni delle curve che descrivono i caratteri stessi: nel linguaggio vettoriale viene descritto l'oggetto stesso e non la sua forma o immagine. Questa capacità di fornire istruzioni che ogni dispositivo di stampa è in grado di interpretare ne ha fatto il sistema preferito per la descrizione dei font, perché consente di ingrandire il carattere senza alterarne la nitidezza. Il linguaggio *PostScript* è un linguaggio vettoriale applicato alla scrittura ed è l'unico che riproduce la qualità del carattere e delle proporzioni della tipografia classica.

Franco Nuzzi

Liceo Q.O. Flacco, Bari
fnuzzi01@alice.it

Giuseppe Pontrelli

IAC-CNR, Roma
giuseppe.pontrelli@gmail.com

Bibliografia e sitografia

- [1] D.E. Knuth, *Digital Typography*, CSLI Publications, 1999.
- [2] D.E. Knuth, *Mathematical typography*, Bull. Am. Math. Soc., 1 (1979).
- [3] A. Syropoulos, A. Tsolomitis, N. Sofroniou, *Digital typography using LaTeX*, Springer Verlag, 2002.
- [4] T. Porter, G. Sue, *Manuale di tecniche grafiche*, Il Castello, 1996.
- [5] G. Fruendi, *Una soluzione moderna per un problema antico: costruzione di una curva che interpola un insieme finito di punti*, L'Insegnamento della Matematica e delle Scienze Integrate, vol. 23B (2000).
- [6] F. Schlöglhofer, Bézier curves in school, DERIVE Newsletter # 52 (2003).
- [7] http://en.wikipedia.org/wiki/Computer_font
- [8] http://en.wikipedia.org/wiki/Bézier_curve
- [9] <http://www.marcoc.it/corsops/indice.htm>
- [10] http://areeweb.polito.it/didattica/polymath/htmlS/Interventi/MatheMath/Ottobre_05/BEZIER.pdf