

# Curve di Bézier e Grafica Vettoriale

*Enlarged PNG image*



*Enlarged SVG image*



# Funzioni Vettoriali o Curve in forma parametrica

Consideriamo un punto  $P$  che si muove nel piano  $x y$  in un intervallo di tempo  $a \leq t \leq b$ . Le due coordinate di  $P$  saranno entrambe funzioni reali di  $t$ ,

$$x = f(t), \quad y = g(t), \quad t \in [a, b].$$

Quindi al variare del tempo  $t$  in  $[a, b]$  le due coordinate del punto  $P = [f(t), g(t)]$  descriveranno nel piano Euclideo "una traiettoria (curva)" che indicheremo con  $C$ .

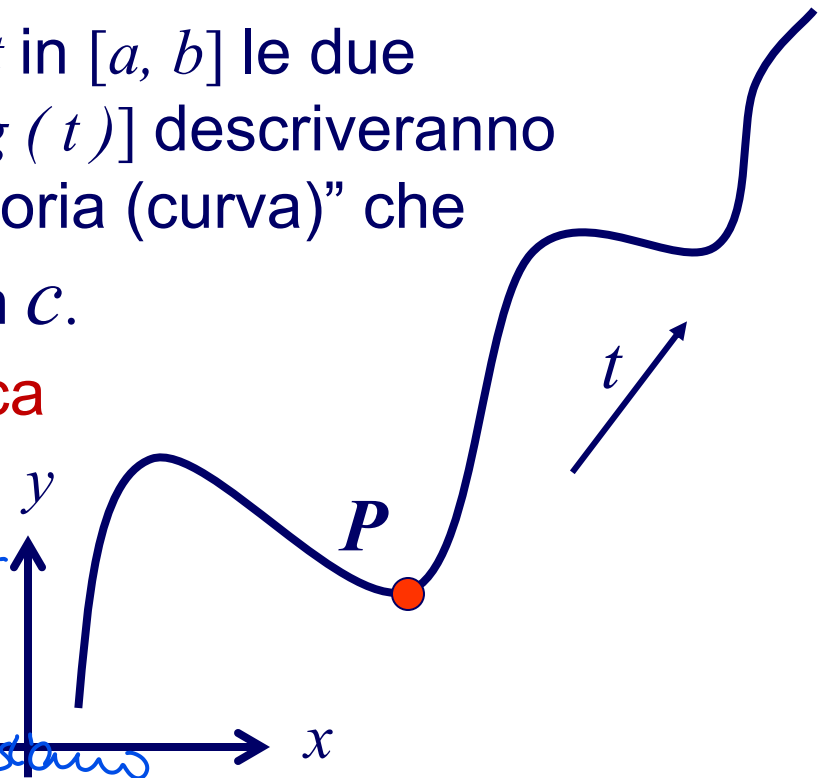
Curva piana in forma parametrica

$$c(t) = [f(t), g(t)]$$

$$t \in [a, b]$$



Coppie di  
valor; nel  
piano cartesiano



# Rappresentazione di Forme/Disegni 2D su un computer

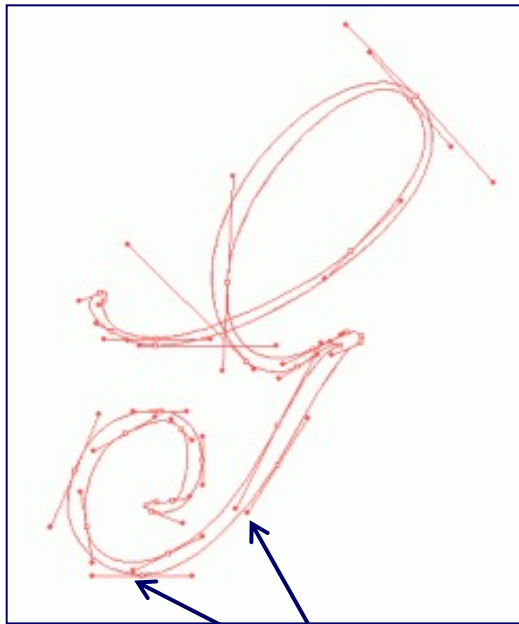
**Problema:** rappresentare matematicamente forme 2D (disegni) su computer

**Soluzione:** si usa una curva piana polinomiale che ne definisce il contorno.

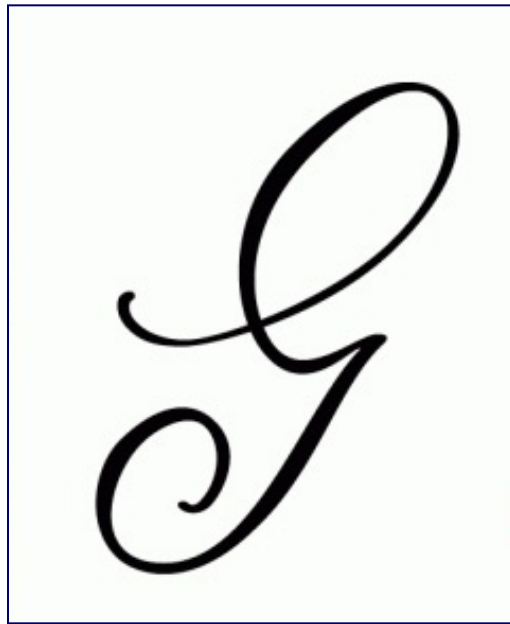
**ma come?**

1. Si specifichi una sequenza di punti  $P_i$ ,  $i = 0, \dots, n$ , (detti *punti di controllo*) nel piano;
2. Si definisca una parametrizzazione cioè una funzione vettoriale (o curva piana polinomiale in forma parametrica, cioè  $f(t)$  e  $g(t)$  sono funzioni polinomiali), che abbia la forma dei punti  $P_i$ ,  $i = 0, \dots, n$  (*interpolazione/approssimazione* di punti).

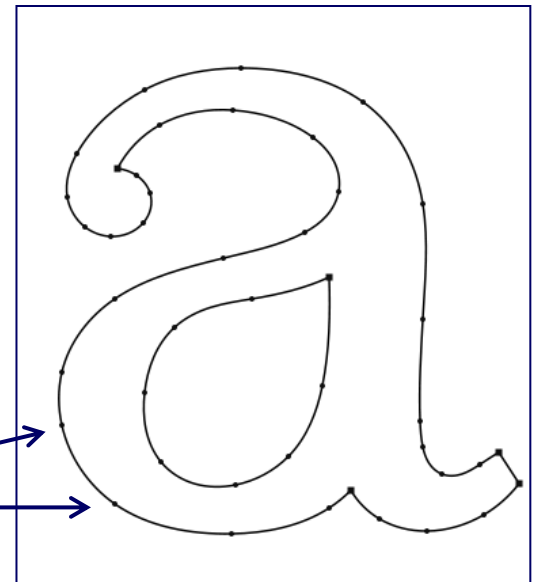
# Esempi



Punti di Controllo



Interpolazione  
di Punti



# Curva di Bézier

Una curva di Bézier di grado  $n$ , nel piano, è una curva definita a partire da un insieme di punti  $P_i = (x_i, y_i)$ ,  $i=0, \dots, n$  del piano detti **punti di controllo** ed è definita da:

$$\begin{aligned} c(t) &= \sum_{i=0}^n P_i B_{i,n}(t) & t \in [0,1] \\ &= \left[ \sum_{i=0}^n x_i B_{i,n}(t) , \sum_{i=0}^n y_i B_{i,n}(t) \right] \end{aligned}$$

dove ogni componente è una

**funzione polinomiale nella base di Bernstein.**

Si tratta di una curva in forma parametrica definita in termini del parametro  $t$ .

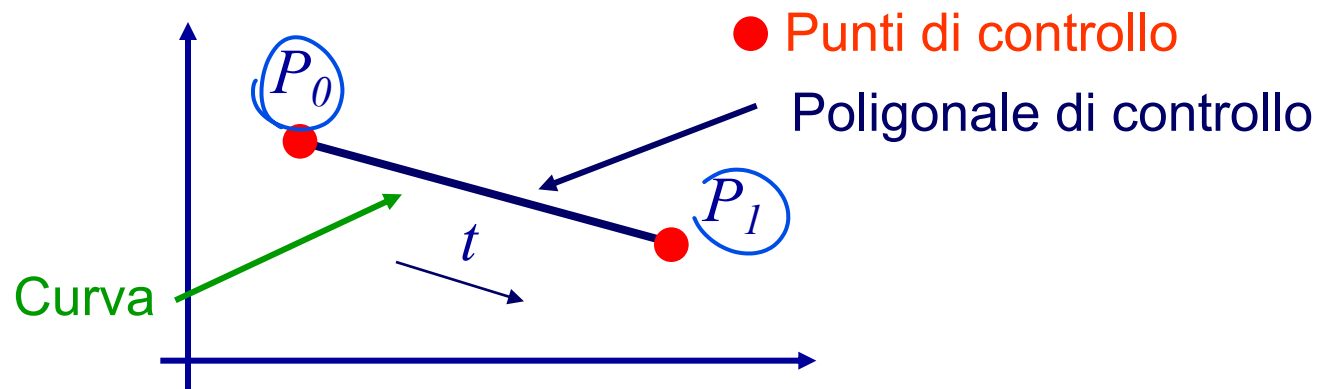
Al variare di  $t$  nell' intervallo  $[0,1]$  vengono definiti i punti della curva.

# Le Curve di Bézier

Esempi di curve di Bézier di grado  $n$ :

Caso  $n=1$

$$c(t) = \sum_{i=0}^1 P_i B_{i,1}(t) = P_0(1-t) + P_1 t \quad t \in [0,1]$$



Se  $P_0 = (x_0, y_0)$  e  $P_1 = (x_1, y_1)$ , in forma esplicita sarà:

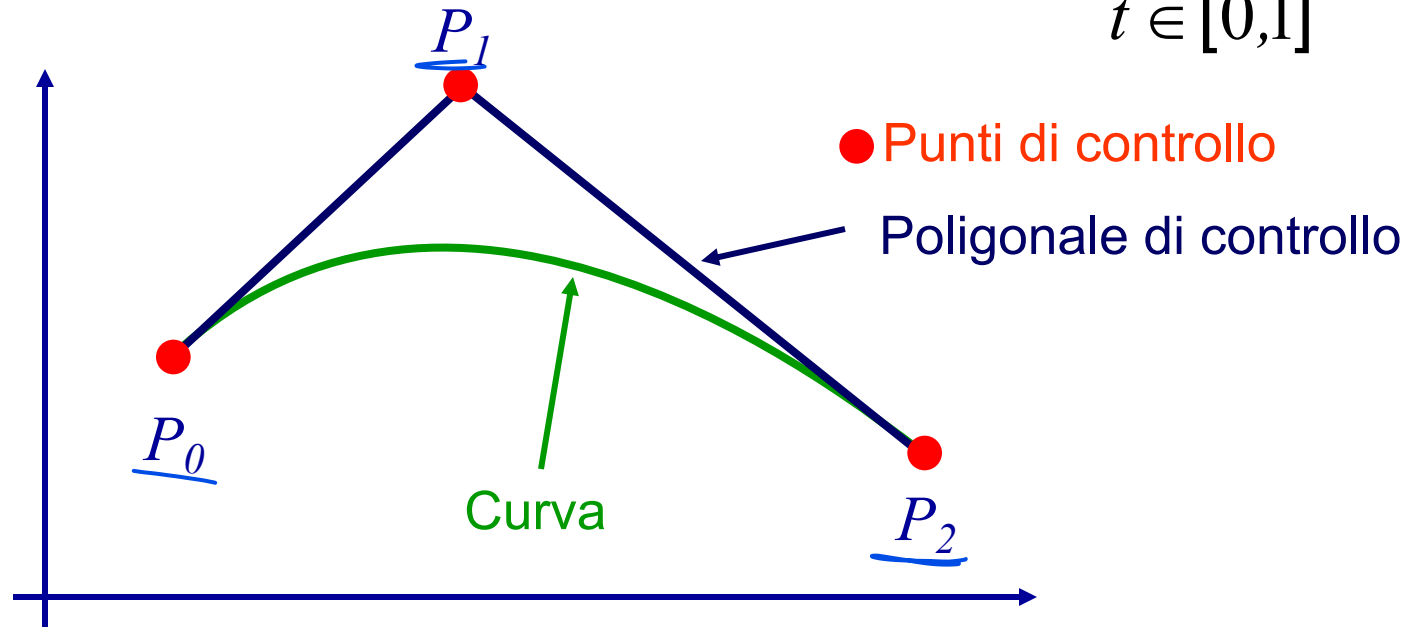
$$\begin{cases} x(t) = x_0(1-t) + x_1 t \\ y(t) = y_0(1-t) + y_1 t \end{cases} \quad t \in [0, 1]$$

# Le Curve di Bézier

Caso  $n=2$

$$c(t) = \sum_{i=0}^2 P_i B_{i,2}(t) = P_0(1-t)^2 + 2P_1(1-t)t + P_2t^2$$

$t \in [0,1]$

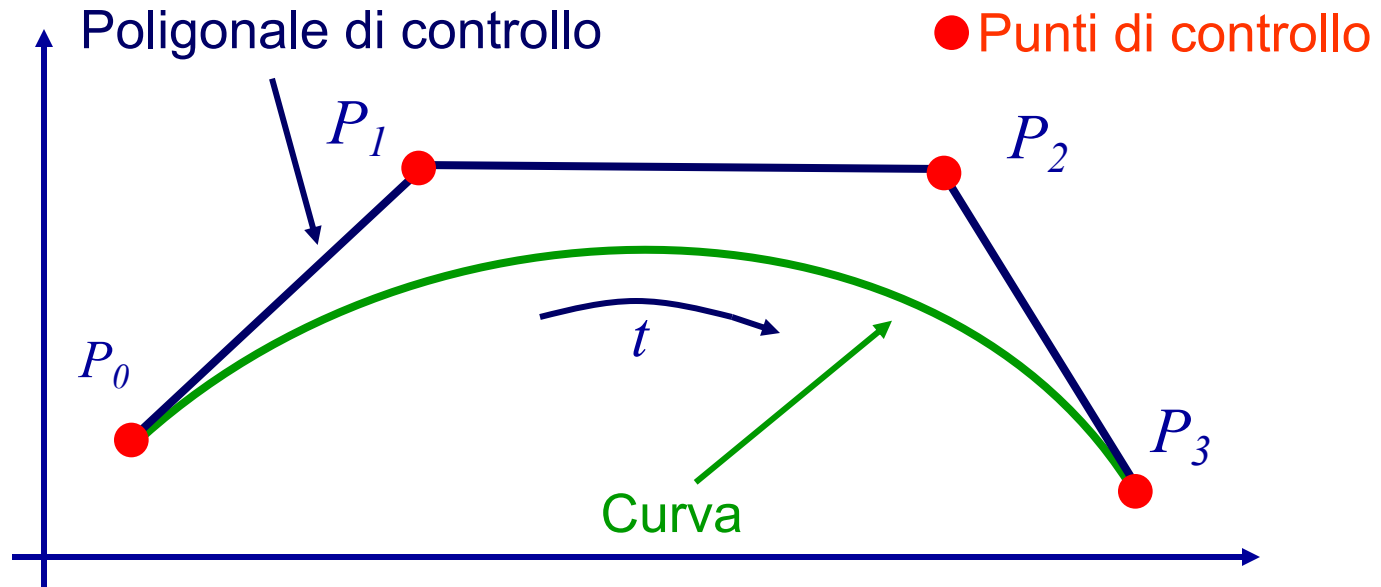


# Le Curve di Bézier

Caso  $n=3$

$$c(t) = \sum_{i=0}^3 P_i B_{i,3}(t) = P_0(1-t)^3 + 3P_1(1-t)^2t + 3P_2(1-t)t^2 + P_3t^3$$

$t \in [0,1]$





# Le Curve di Bézier e di de Casteljau

Il matematico francese de Casteljau, negli anni '60, diede una definizione di curva polinomiale basata su “corner cutting” successivi:

$$P_i^{[k]}(t) = (1-t)P_i^{[k-1]}(t) + tP_{i+1}^{[k-1]}(t) \quad t \in [0,1]$$

dove

$$k = 1, \dots, n$$
$$i = 0, \dots, n-k$$

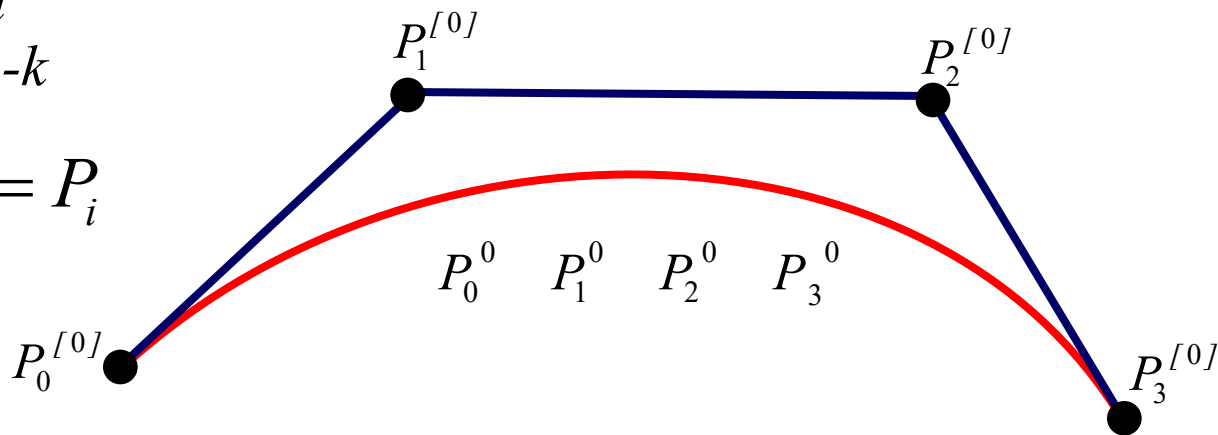
con

$$P_i^{[0]}(t) = P_i$$
$$i = 0, \dots, n$$

e

$$c(t) := P_0^{[n]}(t)$$

Es.  $n=3$ ,  $k=0$



# Le Curve di Bézier e di de Casteljau

Il matematico francese de Casteljau, negli anni '60, diede una definizione di curva di Bézier basata su “corner cutting” successivi:

$$P_i^{[k]}(t) = (1-t)P_i^{[k-1]}(t) + tP_{i+1}^{[k-1]}(t) \quad t \in [0,1]$$

dove

$$k = 1, \dots, n$$
$$i = 0, \dots, n-k$$

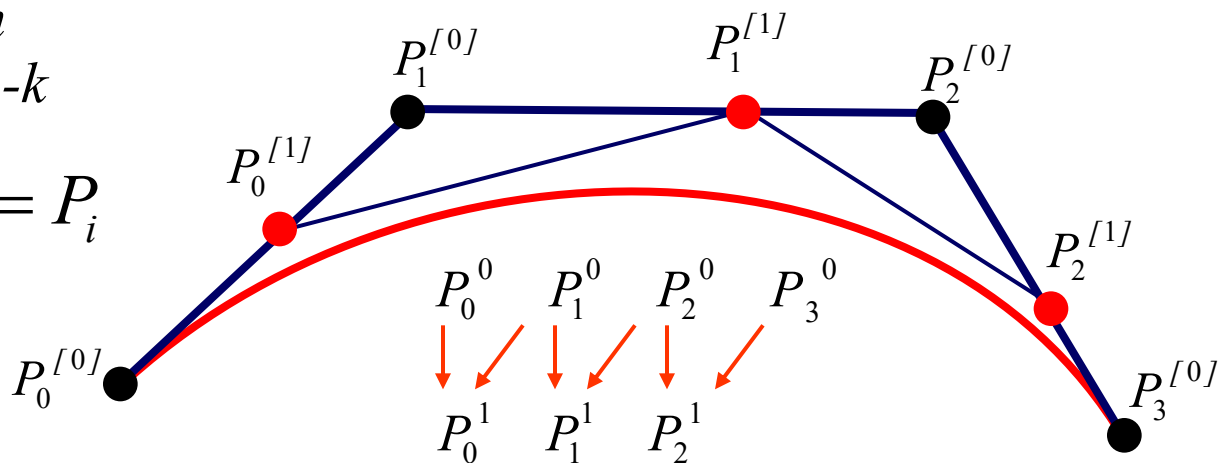
con

$$P_i^{[0]}(t) = P_i$$
$$i = 0, \dots, n$$

e

$$c(t) := P_0^{[n]}(t)$$

Es.  $n=3, k=1$



# Le Curve di Bézier e di de Casteljau

Il matematico francese de Casteljau, negli anni '60, diede una definizione di curva di Bézier basata su “corner cutting” successivi:

$$P_i^{[k]}(t) = (1-t)P_i^{[k-1]}(t) + tP_{i+1}^{[k-1]}(t) \quad t \in [0,1]$$

dove

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

$$i = 0, \dots, n-k$$

con

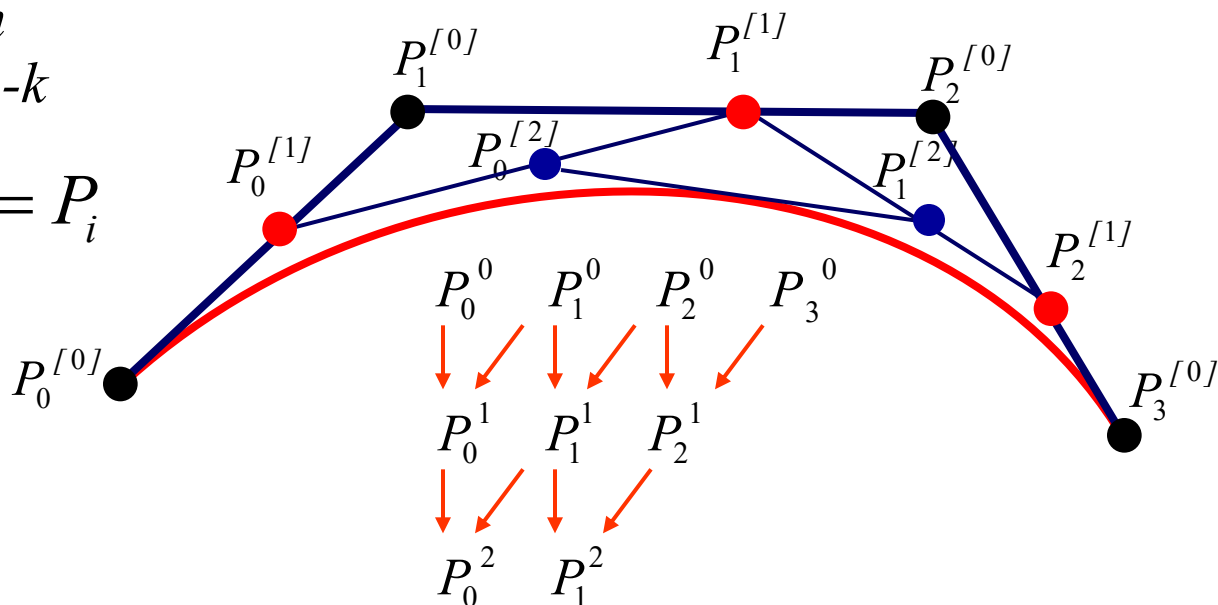
$$P_i^{[0]}(t) = P_i$$

$$i = 0, \dots, n$$

e

$$c(t) := P_0^{[n]}(t)$$

Es.  $n=3, k=2$



# Le Curve di Bézier e di de Casteljau

Il matematico francese de Casteljau, negli anni '60, diede una definizione di curva di Bézier basata su “corner cutting” successivi:

$$P_i^{[k]}(t) = (1-t)P_i^{[k-1]}(t) + tP_{i+1}^{[k-1]}(t) \quad t \in [0,1]$$

dove

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

$$i = 0, \dots, n-k$$

con

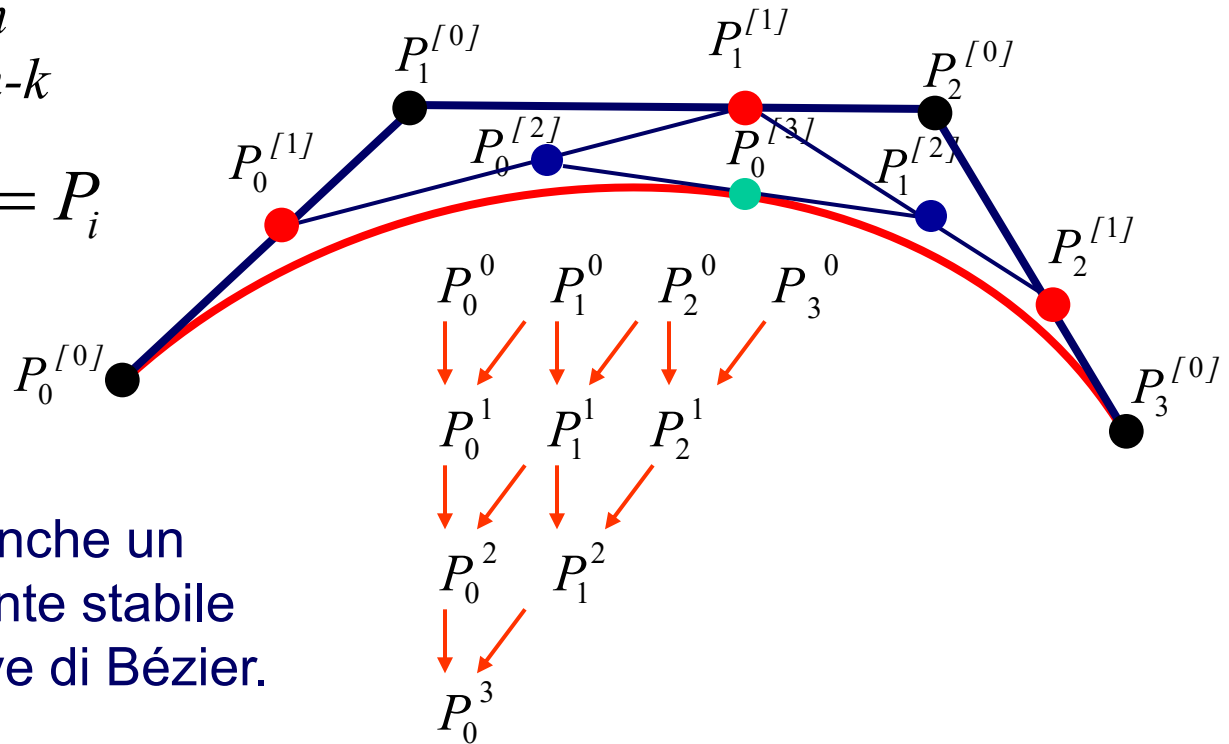
$$P_i^{[0]}(t) = P_i$$

$$i = 0, \dots, n$$

e

$$c(t) := P_0^{[3]}(t)$$

Es.  $n=3, k=3$

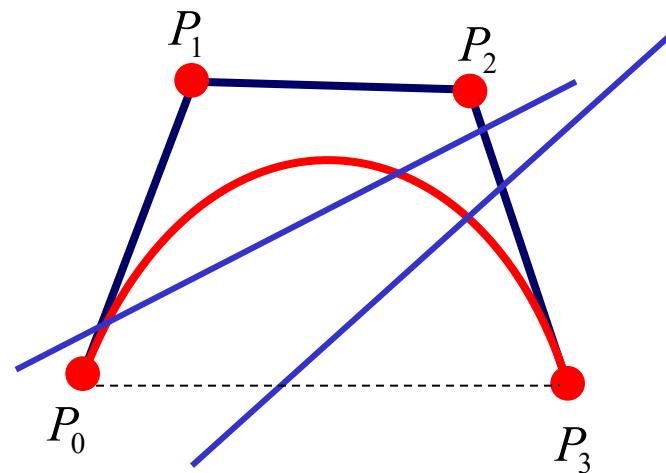
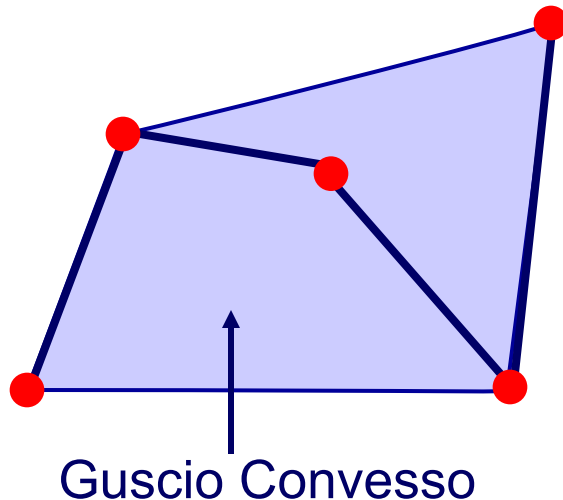


Questa definizione è anche un algoritmo numericamente stabile per il calcolo delle curve di Bézier.

# Le Curve di Bézier e di de Casteljau

Proprietà:

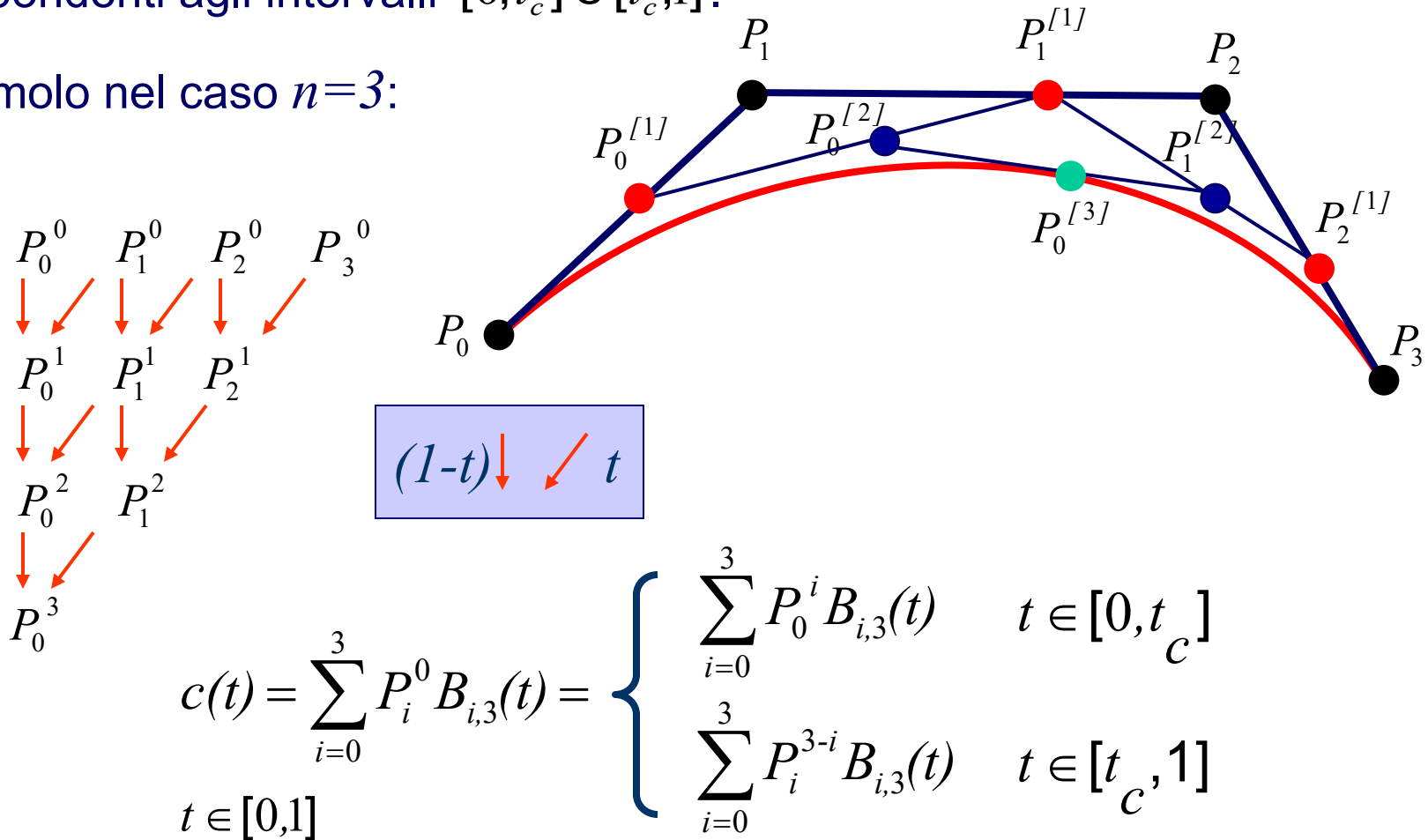
1.  $c(t)$   $t \in [0,1]$  giace nel guscio convesso definito dai suoi punti di controllo;
2.  $c(0)=P_0$  e  $c(1)=P_n$ ;
3.  $c'(0)=n(P_1-P_0)$  e  $c'(1)=n(P_n-P_{n-1})$ ;
4.  $c(t)$  è invariante per trasformazioni affini; in particolare per traslazione, scala, rotazione e deformazione lineare (shear);
5.  $c(t)$  è approssimante in forma della poligonale di controllo.



# Le Curve di Bézier e la Suddivisione

La definizione o algoritmo di valutazione di de Casteljau di una curva di Bézier fornisce anche i punti di controllo delle curve di Bézier corrispondenti agli intervalli  $[0, t_c]$  e  $[t_c, 1]$ .

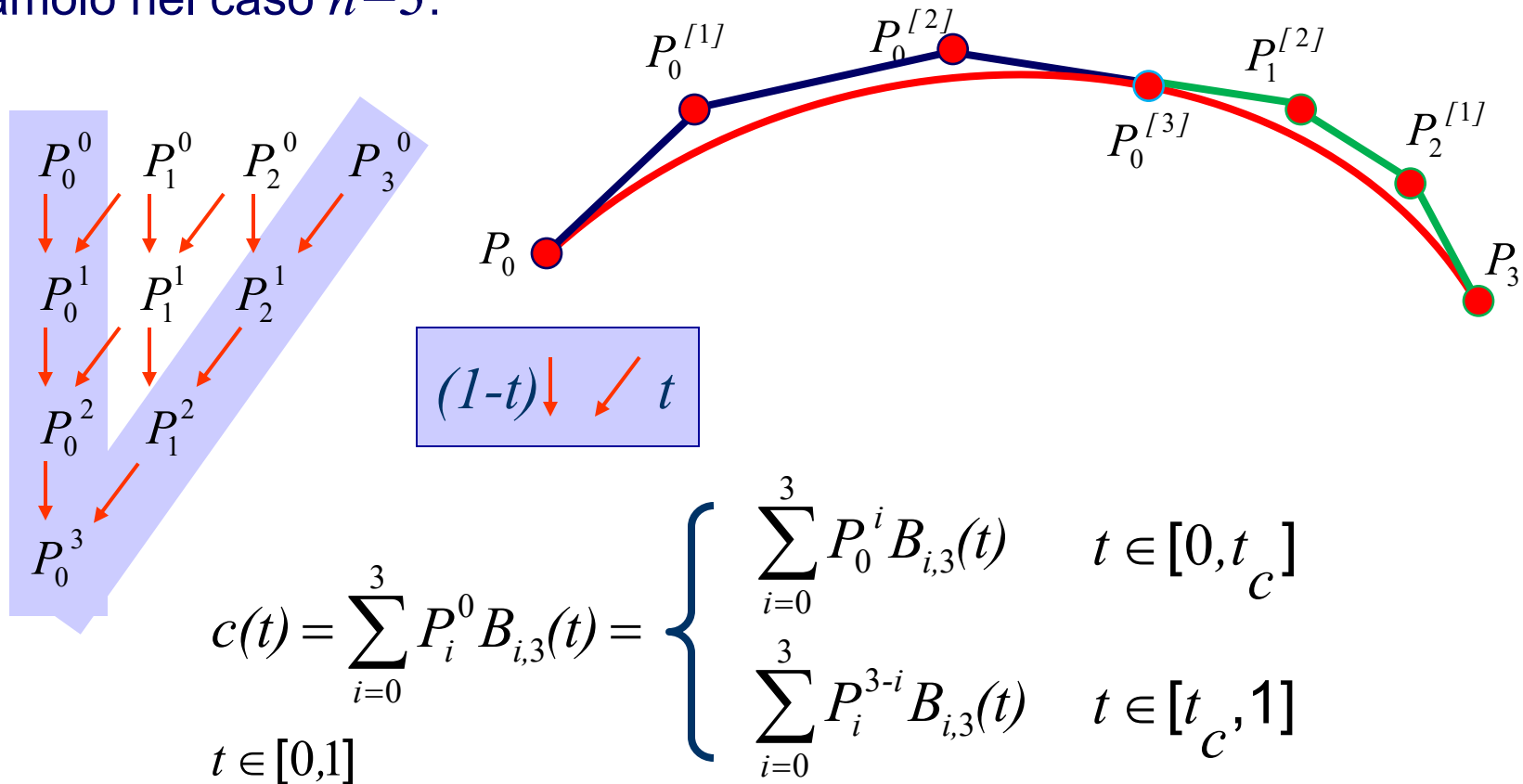
Vediamolo nel caso  $n=3$ :



# Le Curve di Bézier e la Suddivisione

La definizione o algoritmo di valutazione di de Casteljau di una curva di Bézier fornisce anche i punti di controllo delle curve di Bézier corrispondenti agli intervalli  $[0, t_c]$  e  $[t_c, 1]$ .

Vediamolo nel caso  $n=3$ :



# Curve Complesse

Una singola curva di Bézier può rappresentare solo una parte di una forma 2D complessa

Una soluzione potrebbe essere aumentare il grado

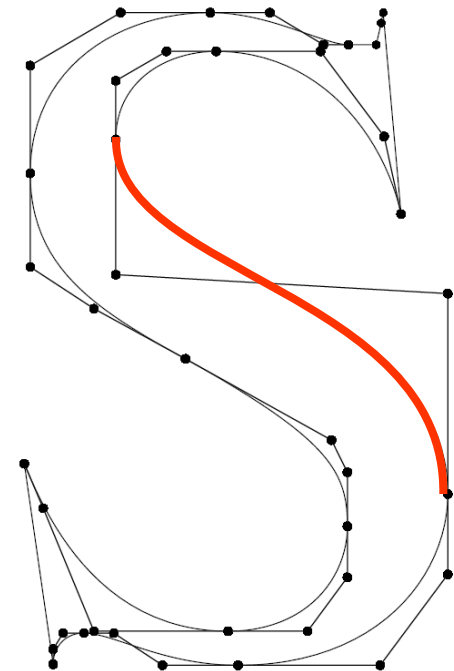
- questo aumenta le possibilità, ma al costo di più punti di controllo e polinomi di grado maggiore
- il controllo è globale, cioè un punto di controllo influenza l'intera curva



# Curve Complesse

In alternativa, la soluzione più comune è unire insieme più curve di Bézier di grado basso in una *piecewise curve* (*curva di Bézier a tratti*)

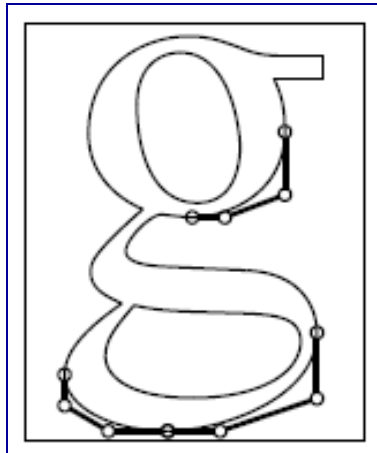
- una curva complessa in forma, può essere pensata in più tratti, ciascuno dei quali rappresentabile con una curva di Bézier di grado basso (per es. grado 3)
- *Controllo Locale*: ogni punto di controllo influenza solo una parte limitata della curva
- L'interazione e la modellazione sono più semplici



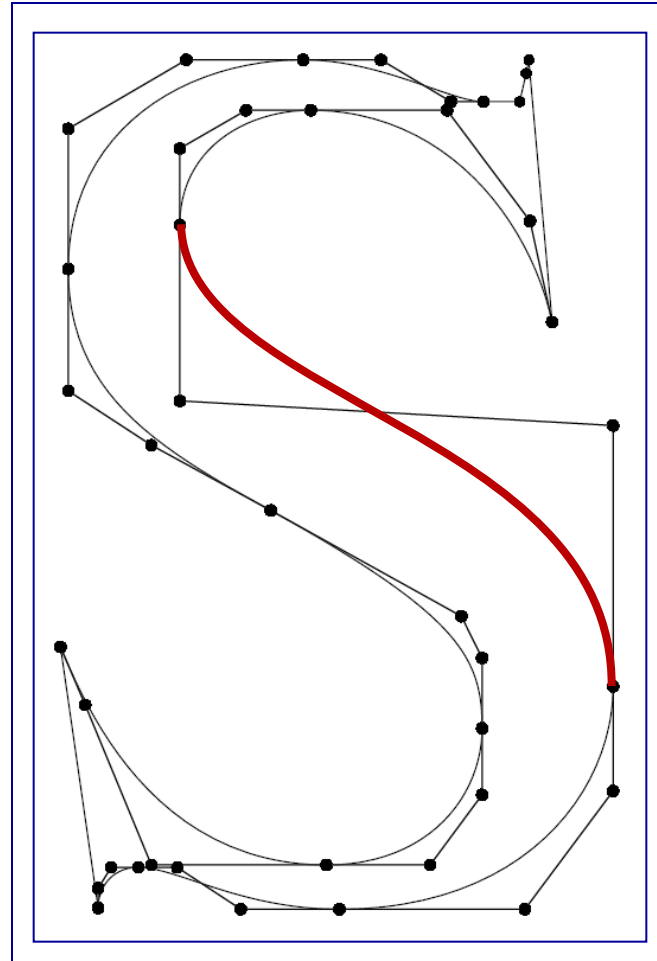
# Curve di Bézier e Bézier a tratti

Per il disegno di curve 2D lo standard de facto sono le curve di Bézier a tratti; ogni tratto è una **curva di Bézier**.

g

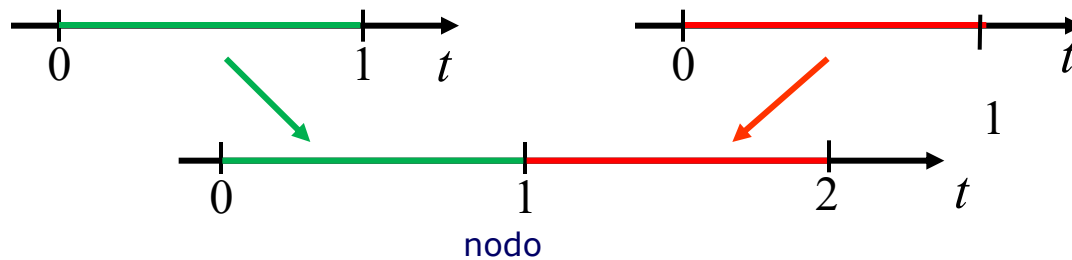
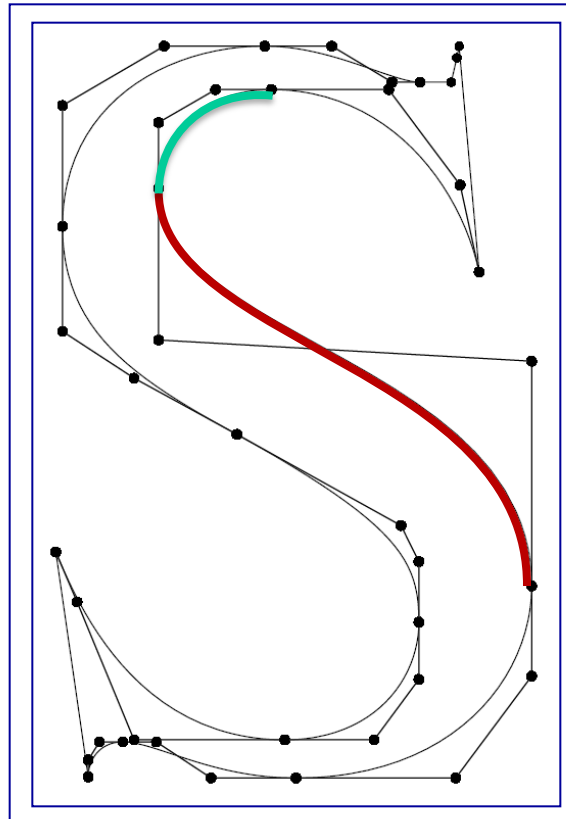


Font Times New Roman



S

# Curva di Bézier a tratti



# Continuità

- Quando due curve vengono unite, solitamente si vuole un certo raccordo negli estremi (**ordine di continuità**):
  - $C^0$ , "C-zero", continuità posizionale, le curve condividono lo stesso punto dove si uniscono
  - $C^1$ , "C-uno", continuità della derivata, le curve hanno lo stesso vettore tangente dove si uniscono
  - $G^1$ , "G-uno", le curve hanno la stessa retta tangente e verso del vettore tangente, ma non il modulo
  - $C^2$ , "C-due", continuità della derivata seconda, le curve hanno la stessa derivata seconda dove si uniscono

# Grafica 2D al Calcolatore

Ci sono due modi per definire un'immagine su un computer:

**modalità Raster:** cioè una matrice di valori interi di intensità, associata alla matrice dei pixel che costituiscono l'immagine;

**modalità Vector:** descrizione matematica delle **curve** che separano regioni di differente colore (outline);



Raster



Vector

Le immagini vettoriali sono **scalabili**, le immagini raster no.

# Vector vs Raster

I principali vantaggi della **grafica vector** rispetto alla **grafica raster** sono la qualità ad ogni risoluzione, la maggior compressione dei dati e la più facile gestione delle eventuali modifiche.

Le **immagini raster** sono ideali per rappresentare foto (real life images), o per simulare il colore di materiali (textures);

Le **immagini vettoriali** sono migliori per tutti gli altri scopi.

# Una dimostrazione pratica

PDF format, Adobe Systems 1993

Dimostrazione con

Acrobat Reader



# Vector vs Raster

La maggior parte dei dispositivi collegati ad un computer, come monitor, stampanti a matrici di punti e stampanti laser, sono dispositivi **raster**.

Ciò significa che tutti gli elementi prima di essere inviati a tali dispositivi (disegnati) devono essere **trasformati in raster**.

Il procedimento di conversione di un'immagine vector in una raster è detto **rasterizing**.

Con **Immagini Digitali** (cioè discrete) ci si riferisce sia a quelle raster che vector; l'unità elementare di un'immagine raster è il **pixel** (picture element).



# Un esempio: le bitmap

Le **bitmap** sono immagini in bianco e nero tipicamente memorizzate e rappresentate in modalità raster.

Per quanto detto sui vantaggi della grafica vector, spesso è utile convertire una bitmap in modalità vector.

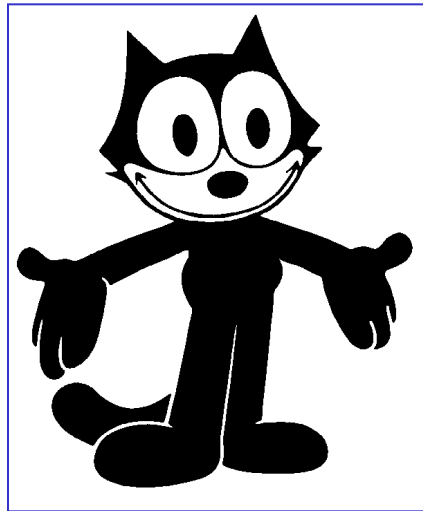
Il procedimento di conversione di un'immagine raster in una vector è detto **tracing** (o vectorizing).



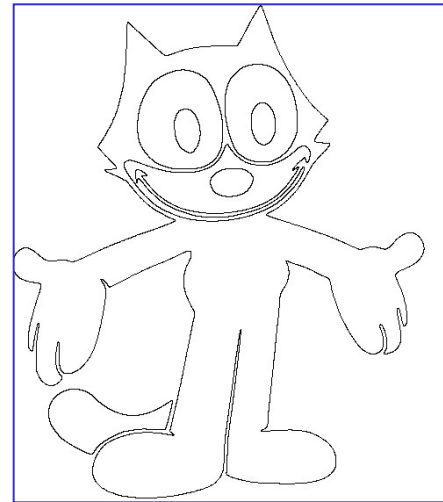
La Linea

# Esempio: da bitmap a vector

Vettorizzazione con Curve di Bézier a tratti



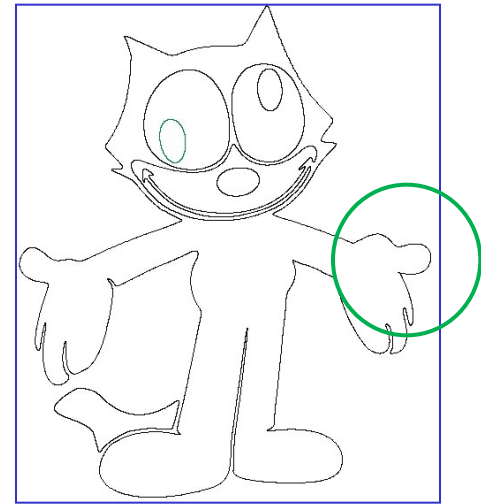
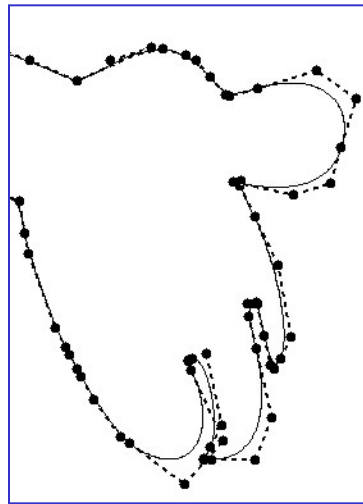
*Bitmap: 672x777 pixel*



*Conversione vector*

# Modifica di Curve di Bézier a tratti

Modellazione mediante modifica dei Punti di Controllo



# Esempio: da vector a bitmap:

Rasterizzazione di Curve di Bézier a tratti



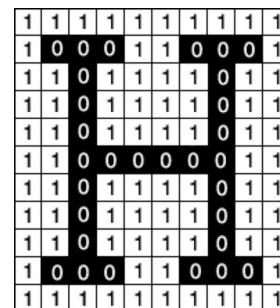
*Bitmap: 672x777 pixel*

# Un esempio: i font digitali

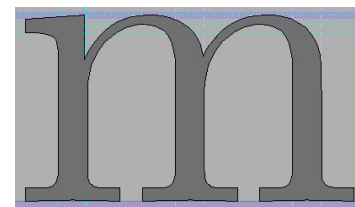
Un **font**, ovvero un tipo di carattere informatico, è una collezione indicizzata di glifi contenente informazioni su come associarvi un particolare codice, visualizzarli in differenti dimensioni e stamparli correttamente.

Ci sono due tipologie di font digitali:

**Bitmap font**: consiste in una matrice di pixel rappresentante l'immagine di un glifo;



**Outline font**: consiste in una descrizione delle curve che racchiudono lo spazio di un glifo;



Formati di font: type1 (Linux),

true type (Apple),

open type (Adobe e Microsoft).

# Esempi di font

Cunningham Singleton à € di Perspective Studio 

*Cunningham Singleton*

Ramadhan Mubarak à € di Andrimada Creative 

*Ramadhan Mubarak*

Richtive Script à € di Mans Greback 

*Richtive Script*

Bellaty di SheillaType 

*Bellaty*

Creepy Pumkin à € di Letter Art Studio

CREEPY PUMKIN

# Esempi di font

Eagle di Woodcutter [↗](#)



Snake Mix di Woodcutter [↗](#)



Kitty Cats TFB di zanatlíja



Lions di Woodcutter [↗](#)



Animal Tracks di Andrew D. Taylor



Deers di Woodcutter [↗](#)



# Software e Formati comuni

LaTeX e METAFONT

Postscript (ps, eps) GhostView

Adobe Acrobat Reader e Adobe Illustrator (pdf)

Adobe Flash (ex Macromedia Flash) grafica su web

PowerPoint, OpenOffice, LibreOffice

**Inkscape**, CorelDRAW

Gimp, Adobe Photoshop

Xfig (Linux)

**Font Forge** (Linux e Mac)

**SVG** (Scalable Vector Graphics)

Html5 (linguaggio di markup per il Web)

OpenGL, WebGL (librerie grafiche 3D)

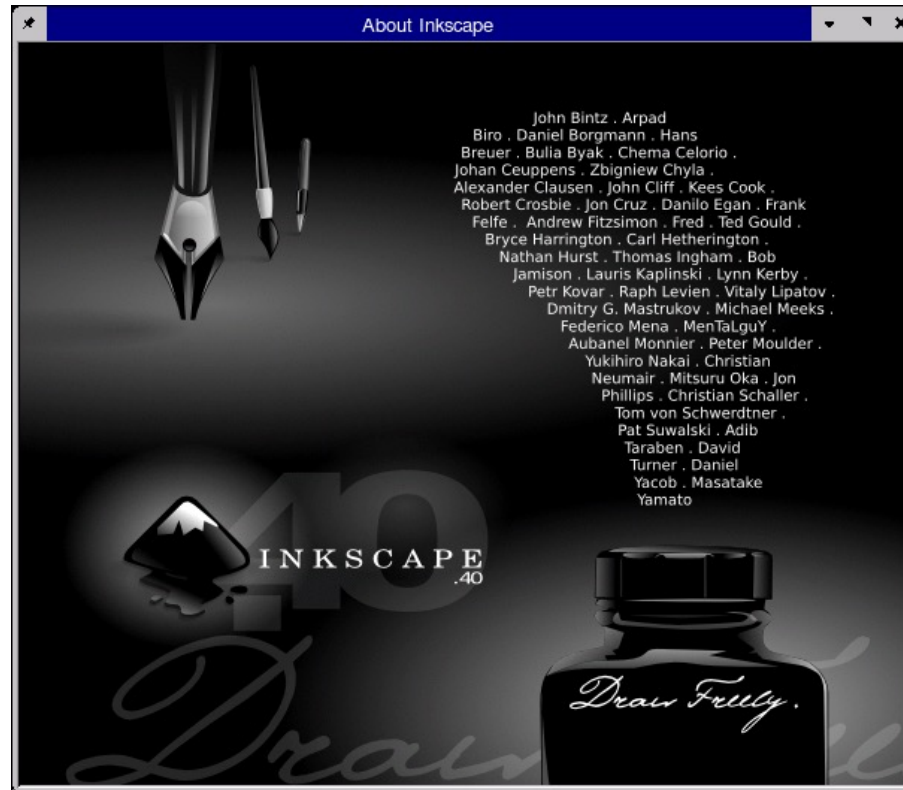
Direct3D (libreria grafica 3D)

...



# InkScape (Windows, Linux, Mac)

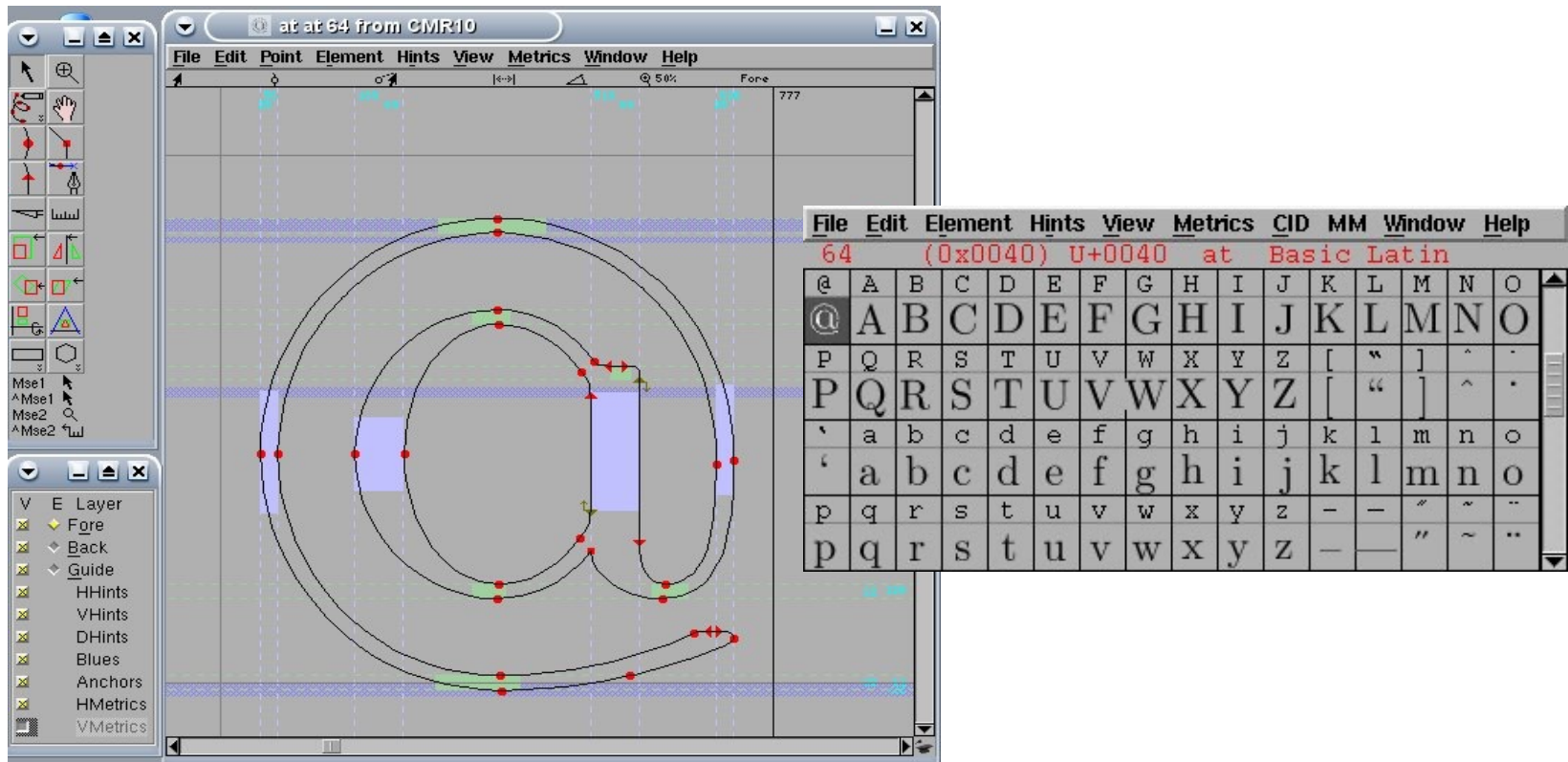
InkScape è un software libero OpenSource e licenza GPL per il disegno vettoriale basato sul formato SVG.



<http://www.inkscape.org>

# Font Forge (Linux, Mac)

FontForge è un software libero OpenSource e licenza GPL che permette la creazione e modifica di font in molti formati standard; è un vero e proprio CAD 2D.



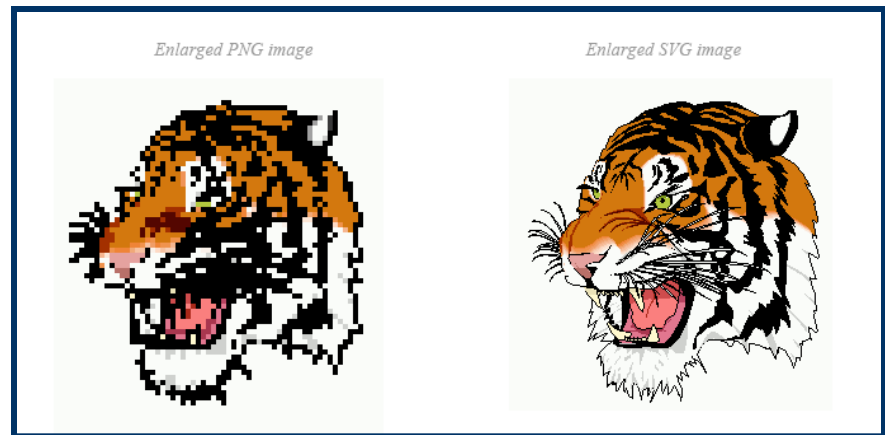
# SVG (Scalable Vector Graphics)

**SVG** è un formato di grafica vettoriale, prodotto dal W3C (World Wide Web Consortium) consorzio nonprofit che si occupa degli standard WEB.

Il sorgente di un file SVG è puro testo XML, ed è quindi componibile modularmente con qualsiasi altra applicazione XML.

Come linguaggio XML può essere processato con i tradizionali tool XML come parser, validatori, editor, e browser (SVG è supportato dalle attuali versioni di browser in modo nativo o mediante appositi plug-in).

SVG viene anche utilizzato per cellulari, smartphone, tablet, ecc.



SVG (Scalable Vector Graphics)

[http://it.wikipedia.org/wiki/Scalable\\_Vector\\_Graphics](http://it.wikipedia.org/wiki/Scalable_Vector_Graphics)