

# Curvas de Bézier e Desenho de Fontes Tipográficas

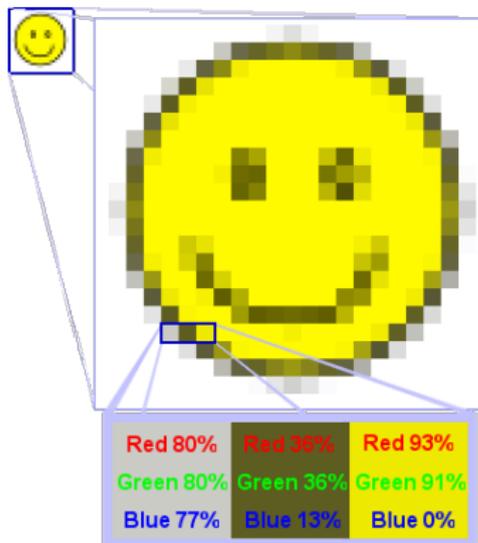
Fabício Caluza Machado  
fabcm1@gmail.com

Abril de 2013

# Conteúdo

- 1 Introdução
  - Tipos de imagens
  - O programa METAFONT
- 2 Curvas de Bézier
  - Definição e Propriedades
  - Exemplos
  - Bisseccção
- 3 METAFONT
  - O comando `draw`
  - Desenhando uma curva
  - Criando um caractere
- 4 Considerações Finais

## Imagens rasterizadas x vetoriais



Exemplo de imagem rasterizada

### Vantagens

1. É um formato adequado para impressão.

### Desvantagens

1. Dependente da resolução usada.
2. Imagem em alta resolução ocupa muito espaço.
3. Exige programas/algoritmos complexos para ser manipulada

## Imagens rasterizadas x vetoriais



Comparação entre imagens raster  
e vetorial  
fig/vetorial.svg

### Imagem vetorial

1. É gerada a partir de uma descrição geométrica das formas na figura.
2. Podem ser transformadas mais facilmente.
3. Ocupam menos espaço de armazenamento.
4. Apenas no momento de exibição/impressão são rasterizadas.



# Donald E. Knuth



- É considerado o pai da análise de algoritmos.
- Famoso pelo seu livro “The art of computer programming” .
- Desenvolveu um sistema que permite a produção de livros e textos de alta qualidade.

# O programa METAFONT

- É uma linguagem de programação usada para descrever as fontes usadas pelo  $\text{T}\text{E}\text{X}$
- O objetivo não é só desenhar uma fonte, mas criar uma descrição matemática desta em alto nível, de modo que novas fontes possam ser criadas apenas alterando alguns parâmetros e melhores resultados possam ser obtidos.
- Para isso, METAFONT usa as curvas de Bézier, que permitem que uma curva seja gerada a partir de poucos pontos de controle fornecidos pelo designer.

# Curvas de Bézier

# Definição

## Curva de Bézier

$$B(t) = \sum_{i=0}^n b_{i,n}(t)P_i = \sum_{i=0}^n \binom{n}{i} t^i (1-t)^{n-i} P_i, \quad t \in [0, 1]$$

Os  $P_i$  são chamados de *pontos de controle* e a curva é definida a partir da base formada pelos  $b_{i,n}(t)$ , chamados *polinômios de Bernstein*:

## Polinômios de Bernstein

$$b_{i,n}(t) = \binom{n}{i} t^i (1-t)^{n-i}, \quad i = 0, 1, \dots, n$$

## Relação Recursiva

$$\begin{aligned}
 B(t) &= \sum_{i=0}^n \binom{n}{i} t^i (1-t)^{n-i} P_i = \\
 &= \sum_{i=0}^{n-1} \binom{n-1}{i} t^i (1-t)^{n-i} P_i + \sum_{i=1}^n \binom{n-1}{i-1} t^i (1-t)^{n-i} P_i \\
 &= (1-t) \sum_{i=0}^{n-1} \binom{n-1}{i} t^i (1-t)^{n-1-i} P_i + t \sum_{i=1}^n \binom{n-1}{i-1} t^{i-1} (1-t)^{n-i} P_i \\
 &= (1-t) \sum_{i=0}^{n-1} \binom{n-1}{i} t^i (1-t)^{n-1-i} P_i + t \sum_{i=0}^{n-1} \binom{n-1}{i} t^i (1-t)^{n-1-i} P_{i+1}
 \end{aligned}$$

$$B_{P_0}(t) = P_0$$

$$B_{P_0 P_1 \dots P_n}(t) = (1-t) B_{P_0 P_1 \dots P_{n-1}}(t) + t B_{P_1 P_2 \dots P_n}(t)$$

## Relação Recursiva

$$\begin{aligned}
 B(t) &= \sum_{i=0}^n \binom{n}{i} t^i (1-t)^{n-i} P_i = \\
 &= \sum_{i=0}^{n-1} \binom{n-1}{i} t^i (1-t)^{n-i} P_i + \sum_{i=1}^n \binom{n-1}{i-1} t^i (1-t)^{n-i} P_i \\
 &= (1-t) \sum_{i=0}^{n-1} \binom{n-1}{i} t^i (1-t)^{n-1-i} P_i + t \sum_{i=1}^n \binom{n-1}{i-1} t^{i-1} (1-t)^{n-i} P_i \\
 &= (1-t) \sum_{i=0}^{n-1} \binom{n-1}{i} t^i (1-t)^{n-1-i} P_i + t \sum_{i=0}^{n-1} \binom{n-1}{i} t^i (1-t)^{n-1-i} P_{i+1}
 \end{aligned}$$

$$B(t) = \sum_{i=0}^{n-1} \binom{n-1}{i} t^i (1-t)^{n-1-i} \cdot ((1-t)P_i + tP_{i+1})$$

## Bézier de ordem 1

A curva de ordem 1 é simplesmente  $B(t) = (1 - t)P_0 + tP_1$ , um segmento de reta.

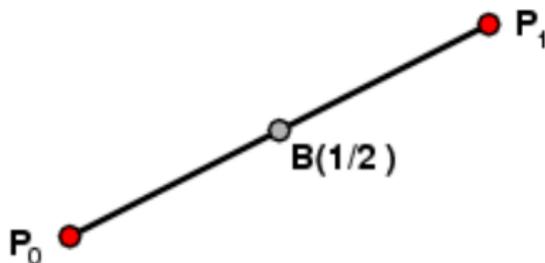


Figura : Curva de Bézier de ordem 1.

animação

## Bézier de ordem 2

A curva de ordem 2 é  $B(t) = (1-t)^2P_0 + 2(1-t)tP_1 + t^2P_2$ .  
Podemos interpretar o papel do ponto  $P_1$  derivando a curva:

$$\begin{aligned} B'(t) &= -2(1-t)P_0 + 2(1-t)P_1 - 2(1-t)tP_1 + 2tP_2 \\ &= 2(1-t)(P_1 - P_0) + 2t(P_2 - P_1) \end{aligned}$$

De modo que em  $t = 0$  a tangente da curva é paralela a  $P_1 - P_0$  e em  $t = 1$  a tangente é paralela a  $P_2 - P_1$ .

## Bézier de ordem 2

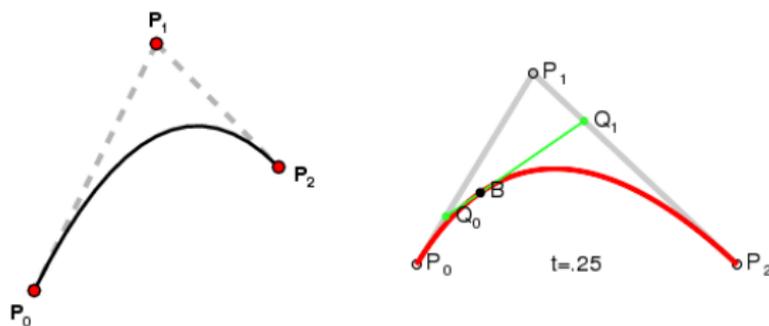


Figura : Curvas de Bézier de ordem 2. À direita, exemplo de sua construção recursiva.

animação

## Bézier de ordem 3

A curva de ordem 3 é:

$$B(t) = (1 - t)^3 P_0 + 3(1 - t)^2 t P_1 + 3(1 - t) t^2 P_2 + t^3 P_3$$

Também podemos interpretar o papel dos pontos  $P_1$  e  $P_2$  derivando a curva:

$$\begin{aligned} B'(t) &= -3(1 - t)^2 P_0 + 3(1 - t)^2 P_1 - 6(1 - t)t P_1 + 6(1 - t)t P_2 - 3t^2 P_2 + 3t^2 P_3 \\ &= 3(1 - t)^2 (P_1 - P_0) + 6(1 - t)t (P_2 - P_1) + 3t^2 (P_3 - P_2) \end{aligned}$$

Em  $t = 0$  a tangente da curva é paralela a  $P_1 - P_0$  e em  $t = 1$  a tangente é paralela a  $P_3 - P_2$ . Observe que com 4 pontos de controle somos capazes de determinar *independentemente* as direções de início e fim da curva. Essa característica é um dos motivos pelos quais a curva de Bézier cúbica é tão usada na prática.

## Bézier de ordem 3

A curva de ordem 3 é:

$$B(t) = (1 - t)^3 P_0 + 3(1 - t)^2 t P_1 + 3(1 - t) t^2 P_2 + t^3 P_3$$

Também podemos interpretar o papel dos pontos  $P_1$  e  $P_2$  derivando a curva:

$$\begin{aligned} B'(t) &= -3(1 - t)^2 P_0 + 3(1 - t)^2 P_1 - 6(1 - t)t P_1 + 6(1 - t)t P_2 - 3t^2 P_2 + 3t^2 P_3 \\ &= 3(1 - t)^2 (P_1 - P_0) + 6(1 - t)t (P_2 - P_1) + 3t^2 (P_3 - P_2) \end{aligned}$$

Observamos ainda que a curva de Bézier é o único polinômio de grau de 3 que satisfaz essas características. Quando definimos  $B(0)$ ,  $B'(0)$ ,  $B(1)$  e  $B'(1)$  estamos, na realidade, fazendo uma interpolação segundo Hermite em cada uma das coordenadas, de forma que a curva de Bézier é apenas uma forma conveniente de representar a interpolação segundo Hermite através dos pontos de controle.

## Bézier de ordem 3

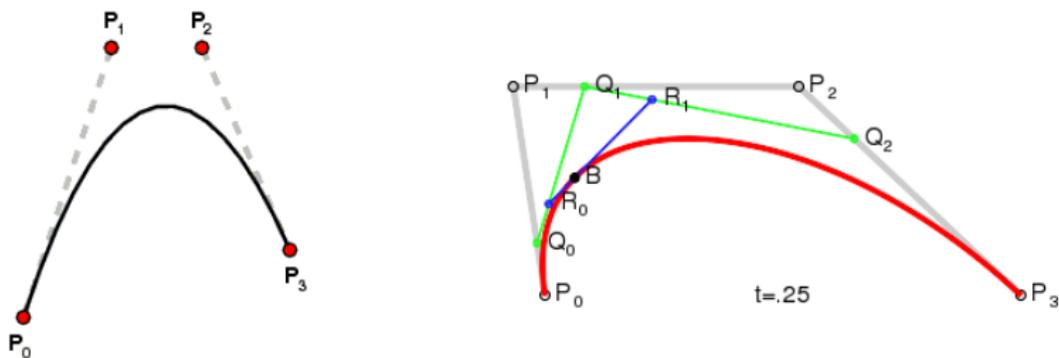


Figura : Curvas de Bézier de ordem 3. À direita, exemplo de sua construção recursiva.

animação interativo

# Bissecção

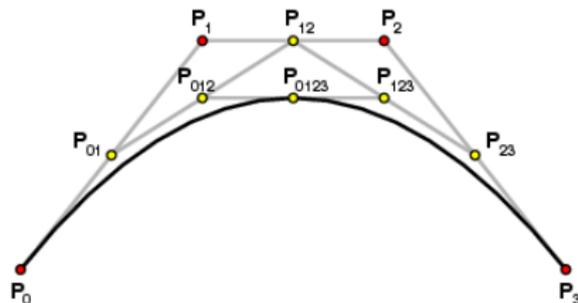


Figura : Bissecção de uma curva de Bézier.

Usando a notação apresentada na figura, vale a seguinte relação:

$$B_{P_0 P_1 P_2 P_3}(t) = B_{P_0 P_{01} P_{12} P_{0123}}(2t) = B_{P_{0123} P_{123} P_{23} P_3}(2t - 1)$$

# METAFONT

## O comando draw

Dados, digamos, 5 pontos  $z_0$ ,  $z_1$ ,  $z_2$ ,  $z_3$  e  $z_4$ , podemos desenhar uma linha que passe por esses pontos com o comando:

**draw** $z_0..z_1..z_2..z_3..z_4$

É possível ainda passar informações extras sobre a forma da curva, como a direção que ela deve ter ao passar por certo ponto:

**draw** $z_0..z_1..z_2\{(1, -1)\}..z_3\{(-1, 1)\}..z_4$



Figura : As duas curvas geradas pelo comando draw.

# Splines

```
drawz0..z1..z2{(1,-1)}..z3{(-1,1)}..z4..z6..z7;
```



Figura : 3 splines justapostas:  $z_0 - z_2$ ,  $z_2 - z_3$  e  $z_3 - z_7$ .

## Desenhando uma curva

Considerando  $B(z_0, z_1, z_2, z_3; t)$  a curva a ser calculada, esta corresponde a dois polinômios, um em cada coordenada:

$$x(t) = B(x_0, x_1, x_2, x_3; t) \text{ e } y(t) = B(y_0, y_1, y_2, y_3; t)$$

O objetivo é encontrar o conjunto de pontos inteiros:

$$P = \{(\lfloor x(t) \rfloor, \lfloor y(t) \rfloor) \mid 0 \leq t \leq 1\}.$$

## Desenhando uma curva

De forma simplificada, o algoritmo é:

1. Se  $[x_0] = [x_3]$ , dê  $[y_3] - [y_0]$  passos para cima.
2. Se  $[y_0] = [y_3]$ , dê  $[x_3] - [x_0]$  passos para a direita.
3. Caso contrário, bisseccione a curva e repita o processo nas duas metades.

## Exemplo

1. `u#:=4/8pt#;` Definição de um parâmetro de escala.
2. `define_pixels(u);`
3. `beginchar(66,14u#,17u#,4.5u#);"Letter beta";`
4. `x1=2u; x2=x3=3u;` Definição das coordenadas dos pontos.
5. `bot y1=-5u; y2=8u; y3=14u;`
6. `x4=6.5u; top y4=h;`
7. `z5=(10u,12u);`
8. `z6=(7.5u,7.5u); z8=z6;`
9. `z7=(4u,7.5u);`
10. `z9=(11.5u,2u);`
11. `z0=(5u,u);`
12. `penpos1(2u,10);` Definição da direção da caneta durante o traço.
13. `penpos2(u,20);`
14. `penpos3(u,-45);`

## Exemplo

```
16. penpos4(.8u,-90);
17. penpos5(1.5u,-180);
18. penpos6(.5u,120);
19. penpos7(.5u,0);
20. penpos8(.5u,210);
21. penpos9(1.5u,-180);
22. penpos0(.3u,20);
23. pickup pencircle;
24.
    penstroke z1e..z2e..z3e..z4e..z5e..z6e..{up}z7e..z8e..z9e..{up}z0e;
25. labels(range 0 thru 9);
26. endchar;
27. end
```

## O caractere

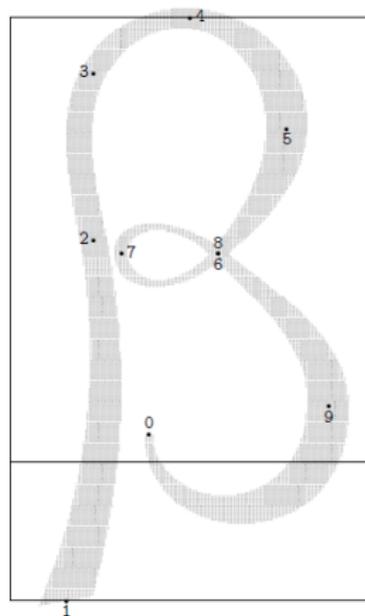


Figura : O caractere  $\beta$  criado pelo programa exibido.

# Considerações Finais

## Considerações Finais

Vejo várias possibilidades para se aprofundar o trabalho apresentado, entre elas:

- Um estudo das superfícies de Bézier.

## Considerações Finais

Vejo várias possibilidades para se aprofundar o trabalho apresentado, entre elas:

- Um estudo das superfícies de Bézier.
- Um estudo mais aprofundado dos polinômios de Bernstein e suas aplicações.

## Considerações Finais

Vejo várias possibilidades para se aprofundar o trabalho apresentado, entre elas:

- Um estudo das superfícies de Bézier.
- Um estudo mais aprofundado dos polinômios de Bernstein e suas aplicações.
- Um estudo mais aprofundado sobre o programa METAFONT, em especial aprender a usá-lo de forma eficiente para o desenvolvimento de uma fonte completa e compreender como tirar proveito de seu lado “META”, isto é, como especificar uma fonte em termos de parâmetros ajustáveis e as sutilezas relacionadas.

## Considerações Finais

Vejo várias possibilidades para se aprofundar o trabalho apresentado, entre elas:

- Um estudo das superfícies de Bézier.
- Um estudo mais aprofundado dos polinômios de Bernstein e suas aplicações.
- Um estudo mais aprofundado sobre o programa METAFONT, em especial aprender a usá-lo de forma eficiente para o desenvolvimento de uma fonte completa e compreender como tirar proveito de seu lado “META”, isto é, como especificar uma fonte em termos de parâmetros ajustáveis e as sutilezas relacionadas.
- Um estudo sobre os formatos usados na compressão e descrição de documentos hoje em dia. Em especial, as características dos arquivos PDF e Djvu.

## Considerações Finais

Vejo várias possibilidades para se aprofundar o trabalho apresentado, entre elas:

- Um estudo das superfícies de Bézier.
- Um estudo mais aprofundado dos polinômios de Bernstein e suas aplicações.
- Um estudo mais aprofundado sobre o programa METAFONT, em especial aprender a usá-lo de forma eficiente para o desenvolvimento de uma fonte completa e compreender como tirar proveito de seu lado “META”, isto é, como especificar uma fonte em termos de parâmetros ajustáveis e as sutilezas relacionadas.
- Um estudo sobre os formatos usados na compressão e descrição de documentos hoje em dia. Em especial, as características dos arquivos PDF e Djvu.
- Um estudo sobre o designer de fontes atual. Quais programas são usados hoje em dia e quais as sutilezas gráficas envolvidas (serifs, proporções, overshoot...).

-  KNUTH, Donald E. *The METAFONTbook*. Reading, Mass.; Wokingham: Addison-Wesley, c1986. (Computers & typesetting; v. C).
-  KNUTH, Donald E. *METAFONT: the program*. Reading, Mass.; Wokingham: Addison-Wesley, c1986. (Computers & typesetting; v. D).
-  Kaspar, Catherine. *Using Bezier Curves for Geometric Transformations* [monograph] MSM Creative Component, Fall 2009
-  *Bézier curve*. (19 de abril de 2013). Em Wikipedia, The Free Encyclopedia. Acessado em 22:08, 22 de abril de 2013, em [http://en.wikipedia.org/w/index.php?title=B%C3%A9zier\\_curve&oldid=551068111](http://en.wikipedia.org/w/index.php?title=B%C3%A9zier_curve&oldid=551068111)

-  *PostScript* (18 de abril de 2013). Em Wikipedia, The Free Encyclopedia. Acessado em 22:14, 22 de abril de 2013, em <http://en.wikipedia.org/w/index.php?title=PostScript&oldid=550998725>
-  Grandsire, Christophe. *The METAFONTtutorial*. (30 de dezembro de 2004) Em METAFONT Tutorial Page. Acessado em 22:33, 22 de abril de 2013, em <http://metafont.tutorial.free.fr/>
-  Casselman, Bill. *From Bézier to Bernstein*. (novembro de 2008) Em Feature Column, American Mathematical Society. Acessado em 22:42, 22 de abril de 2013, em <http://www.ams.org/samplings/feature-column/fcarc-bezier>

# Obrigado pela atenção!

Fabrício Caluza Machado

fabcm1@gmail.com

[www.ime.unicamp.br/~ra102174](http://www.ime.unicamp.br/~ra102174)

Instituto de Matemática, Estatística e Computação  
Científica - UNICAMP