

## Curvas

A representação de uma curva como uma sucessão de trechos retos pode ser suficiente para várias aplicações. No entanto, curvas e superfícies complexas normalmente demandam uma maneira mais eficiente de representação. Tal representação é normalmente mais compacta que as formas discretas.

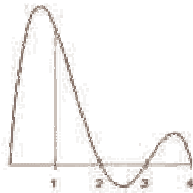
Definir uma curva que passe por um conjunto determinado de pontos é um problema de **interpolação**, enquanto a definição de uma curva que passe próximo a um conjunto determinado de pontos é um problema de **aproximação**. Neste capítulo usaremos o termo **ajustamento** de curva para ambos os casos pois frequentemente, os dois problemas acontecem de forma combinada.

O número de pontos providos para a determinação de uma curva normalmente não é alto. O problema de ajustamento se resume a conectar estes pontos com curvas e exibi-los em uma densidade adequada ao meio de representação. Outra aplicação comum envolve a representação de dados experimentais, para exibição ou para reconhecimento automático de padrões [PAVLIDIS, 1984]. No reconhecimento de padrões, a descrição matemática do contorno de um objeto pode prover informação sobre uma classe à qual o objeto pertence. Uma primeira escolha para funções de ajuste de curva recai sobre funções polinomiais.

Seja  $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$  uma sucessão de pontos no plano, o polinômio de grau  $(n-1)$  que gera uma reta que passa por todos os pontos pode ser facilmente obtido por:

$$f(x) = y_1 \frac{(x-x_2)(x-x_3)\dots(x-x_n)}{(x_1-x_2)(x_1-x_3)\dots(x_1-x_n)} + y_2 \frac{(x-x_1)(x-x_3)\dots(x-x_n)}{(x_2-x_1)(x_2-x_3)\dots(x_2-x_n)} + y_3 \frac{(x-x_1)(x-x_2)\dots(x-x_n)}{(x_3-x_1)(x_3-x_2)\dots(x_3-x_n)}$$

A principal desvantagem dos polinômios reside na sua globalidade, ou seja, no fato de que alterações em qualquer um dos pontos afeta toda a curva bem como sua grande tendência a oscilações. A Figura 1 ilustra este efeito oscilatório no ajuste de uma curva gerada por polinômio para 5 pontos definidos por:  $(0,0)$ ,  $(1,3)$ ,  $(2,0)$ ,  $(3,0)$ , e  $(4,0)$ .



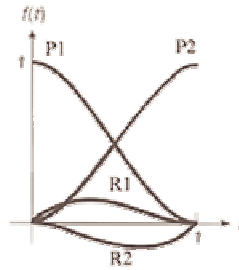
**Figura 1 - Efeito oscilatório dos polinômios: [PAVLIDIS, 1984]**

Polinômios tem geralmente uma aplicação bastante específica, sendo o seu uso prático restrito a pequenos intervalos e a um número relativamente baixo de pontos de controle.

Funções de ajuste de curva com atuação local são mais práticas, uma vez que alterações em pontos de controle não se propagam por toda a curva. Ajustadores locais são também conhecidos como **curvas paramétricas**. Curvas paramétricas de segunda ordem apresentam geralmente pouca utilidade pois as tangentes nos pontos de entrada e saída da função devem ser obrigatoriamente iguais. Curvas paramétricas de terceira ordem, ou cúbicas são, no entanto, largamente utilizadas resultando em vários nomes familiares aos usuários de computação gráfica, tais como: Spline, Bezier, B-Spline, Hermite e Catmull. Neste capítulo abordaremos algumas destas funções de ajuste de curva.

## HERMITE

O uso de polinômios de terceira ordem para ajuste de curvas foi extensamente descrito por Hermite. Sua aplicação é básica para o entendimento dos demais polinômios de ajuste de curvas. A geometria proposta por Hermite propõe um interpolador local controlado por 4 fatores a cada intervalo de 2 pontos: os próprios pontos de entrada e saída da curva (P1, P2) e os vetores tangentes à curva neste pontos, definidos como (T1, T2). A Figura 4-2 ilustra a geometria de mistura proposta por Hermite para controlar a influência de cada um dos quatro fatores de controle de suas curvas.



**Figura 2 - Geometria de composição da função de Hermite: [FOLEY, 1993]**

Uma interpretação rápida da Figura 2 nos mostra que para um dado instante ( $t$ ) qualquer, o fator de ponderação das coordenadas de um ponto para a geração da curva Hermite é resultado da soma de cada um dos fatores de controle (P1, P2, T1, T2) ponderados pelas curvas da geometria de Hermite. As curvas de ponderação da geometria de Hermite são definidas por polinômios de terceira ordem com a seguinte forma geral:

$$x(t) = P_x = a_x t^3 + b_x t^2 + c_x t + d_x; \quad (1)$$

Assim sendo, tem-se que a solução da geometria de Hermite consiste na determinação dos valores de ( $a$ ,  $b$ ,  $c$ ,  $d$ ) para cada uma das curvas de ponderação (P1, P2, T1, T2). Sabe-se, em função da geometria proposta, que para ( $t = 0$ ) tem-se:

$$P1_x = a_x 0^3 + b_x 0^2 + c_x 0 + d_x; \text{ ou na forma matricial; } M_{Hx} = [0 \ 0 \ 0 \ 1]^T; \quad (2)$$

Sabe-se, em função da geometria proposta, que para ( $t = 1$ ) tem-se:

$$P2_x = a_x 1^3 + b_x 1^2 + c_x 1 + d_x; \text{ ou na forma matricial; } M_{Hx} = [1 \ 1 \ 1 \ 1]^T; \quad (3)$$

Sabe-se que as tangentes em P1 e P2 podem ser obtidas pela derivada da equação da curva no ponto ( $x(t) = P_x = a_x t^3 + b_x t^2 + c_x t + d_x$ ), como se segue:

$$\text{Para } (t = 0): T1_x = 3a_x 0^2 + 2b_x 0 + c_x 0 + 0; \text{ ou na forma matricial; } M_{Hx} = [0 \ 0 \ 1 \ 0]^T; \quad (4)$$

$$\text{para } (t = 1): T2_x = 3a_x 1^2 + 2b_x 1 + c_x 1 + 0; \text{ ou na forma matricial; } M_{Hx} = [3 \ 2 \ 1 \ 0]^T; \quad (5)$$

Para que a geometria de Hermite seja válida é necessário então que a relação expressa em (6) seja verdadeira, onde a  $M_H$  é a incógnita do problema:

$$[P1 \ P2 \ T1 \ T2] = G_H - G_H * M_H * \begin{bmatrix} 0 & 1 & 0 & 3 \\ 0 & 1 & 0 & 2 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix} \quad (6)$$

Para que a relação expressa em 6 seja válida é necessário que  $M_H$  seja a matriz inversa da matriz acima, anulando o produto com a matriz identidade resultante, ou seja,  $M_H$  (matriz de Hermite) é definida por:

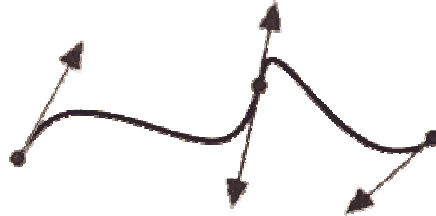
$$M_H^{-1} = \begin{bmatrix} 0 & 1 & 0 & 3 \\ 0 & 1 & 0 & 2 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix}^{-1} = \begin{bmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix} \quad (7)$$

A função geradora da curva de Hermite encontra-se detalhada no código fonte provido pelo autor sendo esta a aplicação direta da matriz  $M_H$  e possuindo a seguinte forma geral (o valores nulos foram incluídos no algoritmo com a finalidade de facilitar a compreensão do leitor):

```
WG=0.5; // Fator de atenuação das tangentes T1 e T2
for(t=0;t<=1;t+=Step)
{
x=(( 2*pow(t,3) -3*pow(t,2) +0*t +1)*X[i]+
(-2*pow(t,3) +3*pow(t,2) +0*t +0)*X[i+1]+
( 1*pow(t,3) -2*pow(t,2) +1*t +0)*WG*T1X+
( 1*pow(t,3) -1*pow(t,2) +0*t +0)*WG*T2X);

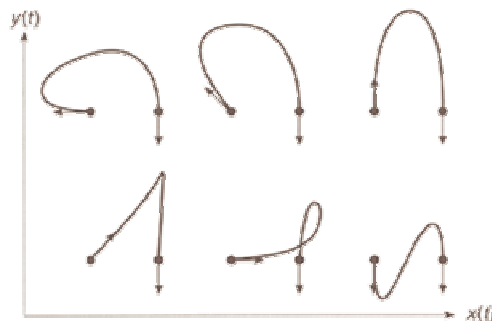
y=(( 2*pow(t,3) -3*pow(t,2) +0*t +1)*Y[i]+
(-2*pow(t,3) +3*pow(t,2) +0*t +0)*Y[i+1]+
( 1*pow(t,3) -2*pow(t,2) +1*t +0)*WG*T1Y+
( 1*pow(t,3) -1*pow(t,2) +0*t +0)*WG*T2Y);
}
```

O controle preciso das tangentes de entrada e de saída da curva é essencial para a "suavização" da curva total. Diz-se que uma curva é contínua à outra se as suas respectivas tangentes de saída e entrada possuem a mesma direção. A Figura 3 ilustra a continuidade da curva obtida pelo alinhamento das tangentes no ponto de união.



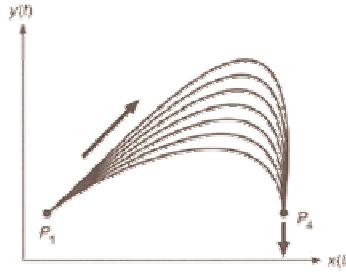
**Figura 3 - Continuidade dos segmentos de curva em função da direção da tangente: [FOLEY, 1993]**

As a direção das retas tangentes utilizadas na geração da curva de Hermite permitem introduzir modificações significativas na curva produzida pela técnica de Hermite. A Figura 4 ilustra os diferentes resultados obtidos pela simples alteração na direção da tangente inicial da curva de Hermite.



**Figura 4 - Efeitos da variação na direção da tangente na função de Hermite: [FOLEY, 1993]**

Variações no módulo do vetor tangente afetam a "agressividade" da curva. O módulo do vetor tangente pode ser interpretado como a velocidade no ponto tangente [FOLEY, 1993]. A Figura 5 ilustra os diferentes resultados obtidos pela variação do módulo da primeira tangente utilizada no cálculo da curva de Hermite.



**Figura 5 - Efeitos da variação no módulo da tangente na função de Hermite: [FOLEY, 1993]**

O controle individual das retas tangentes nem sempre é intuitivo e pode ser pouco adequado quando uma interação manual é esperada no controle da curva. Uma maneira "automática" de controle das retas tangentes, necessárias ao cálculo da função de Hermite, pode ser então adequada. No exemplo em C++, provido juntamente com este texto, o controle das retas tangentes encontra-se oculto do operador e seu cálculo é estabelecido objetivando a continuidade da curva total (tangentes de mesma direção nos pontos de continuidade).

Dados 4 pontos definidos por  $P1(X1, Y1)$ ,  $P2(X2, Y2)$ ,  $P3(X3, Y3)$  e  $P4(X4, Y4)$  são as geradas 4 tangentes necessárias à geração de 3 segmentos de curva contínuos como se segue:

a) Tangente em P1:

$$\text{Tangente em } P1(X1, Y1) = (T1X, T1Y) \quad \text{Onde: } T1X = X2 - X1; \quad T1Y = Y2 - Y1; \quad (8)$$

b) Tangente em P2:

$$\text{Tangente em } P2(X2, Y2) = (T2X, T2Y) \quad \text{Onde: } T2X = X3 - X1; \quad T2Y = Y3 - Y1; \quad (9)$$

c) Tangente em P3:

$$\text{Tangente em } P3(X3, Y3) = (T3X, T3Y) \quad \text{Onde: } T3X = X4 - X2; \quad T3Y = Y4 - Y2; \quad (10)$$

d) Tangente em P4:

$$\text{Tangente em } P4(X4, Y4) = (T4X, T4Y) \quad \text{Onde: } T4X = X4 - X3; \quad T4Y = Y4 - Y3;$$

Um fator de correção para o "peso" da tangente na composição da curva pode ser aplicado para tornar a curva mais "agressiva". A Figura 6 ilustra a curva obtida com peso de 50% para as tangentes e utilizando-se a técnica descrita .

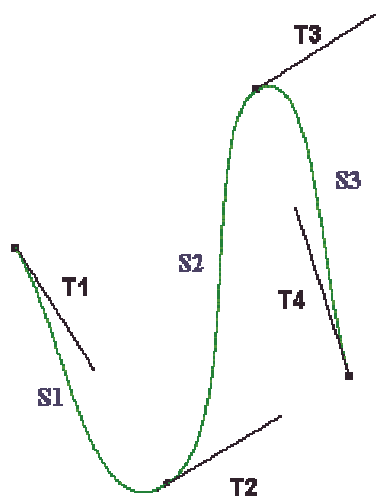


Figura 6 - Cálculo automático da tangente para composição de Hermite

## BEZIER

A curva de Bézier foi desenvolvida por Pierre Bézier durante seus trabalhos em projetos de automóveis para a Renault francesa no início da década de 1970. Bézier baseou sua curva nos princípios descritos por Hermite, adicionando à definição original uma metodologia para a determinação das tangentes em P1 e P2, pelo uso de dois pontos de controle T1 e T2, definidos conforme se segue:

a) Tangente em P1:

$$\text{Tangente em } P1(X1, Y1) = (T1X, T1Y) \quad \text{Onde: } T1X = 3*(X2 - X1); \quad T1Y = 3*(Y2 - Y1); \quad (1)$$

b) Tangente em P2:

$$\text{Tangente em } P2(X2, Y2) = (T2X, T2Y) \quad \text{Onde: } T2X = 3*(X3 - X1); \quad T2Y = 3*(Y3 - Y1); \quad (2)$$

Assim como em Hermite, a curva de Bézier utiliza 4 pontos de controle e 4 polinômios de terceira ordem (Bernstein) para definir geometria similar onde os polinômios ponderam a participação dos pontos de controle na geração da curva. A Figura 7 ilustra a geometria de mistura proposta por Bézier para controlar a influência de cada um dos quatro fatores de controle de suas curvas.

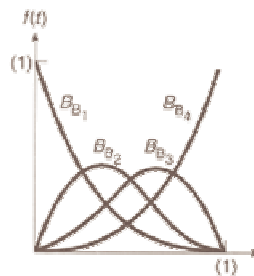


Figura 7 - Geometria de composição da função de Bezier (4 pontos): [FOLEY, 1993]

Para intervalos maiores que 4 pontos, no entanto, a função de Bézier expande o conceito de "mistura" proposto por Hermite e assume um caráter polinomial, ou seja, não local, onde todos os pontos participam da formação ponderada da curva. A Figura 8 ilustra a geometria proposta por Bézier para curvas com 6 pontos de controle.

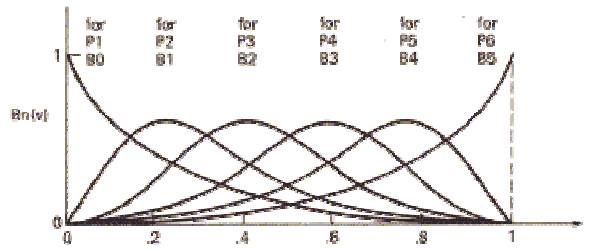


Figura 8 - Geometria de composição da função de Bezier (6 pontos): [ARTWICK, 1984]

A geometria proposta por Bézier baseia-se na divisão recursiva dos segmento medianos de reta criados a partir da união dos seus pontos de controle da curva. A Figura 9 ilustra esta divisão recursiva que se levada ao seu limite irá gerar a curva de Bézier.

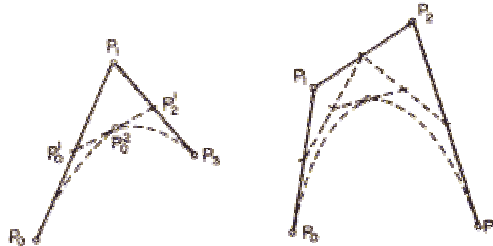


Figura 9 - Princípio geométrico da curva de Bézier: [ARTWICK, 1984]

A matriz de Bézier, que descreve as curvas de mistura de sua geometria, são definidas como se segue:

$$M_B = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \quad (3)$$

A função geradora da curva de Bézier, para intervalos de 4 pontos, encontra-se detalhada no código fonte provido pelo autor, sendo este a aplicação direta da matriz  $M_B$  e possuindo a seguinte forma geral (o valores nulos foram incluídos no algoritmo com a finalidade de facilitar a compreensão do leitor):

```
for(t=0;t<=1;t+=Step)
{
x=((-1*pow(t,3) + 3*pow(t,2) - 3*t + 1)*X[i] +
( 3*pow(t,3) - 6*pow(t,2) + 3*t + 0)*X[i+1] +
(-3*pow(t,3) + 3*pow(t,2) + 0*t + 0)*X[i+2] +
( 1*pow(t,3) + 0*pow(t,2) + 0*t + 0)*X[i+3]);

y((-1*pow(t,3) + 3*pow(t,2) - 3*t + 1)*Y[i] +
( 3*pow(t,3) - 6*pow(t,2) + 3*t + 0)*Y[i+1] +
(-3*pow(t,3) + 3*pow(t,2) + 0*t + 0)*Y[i+2] +
( 1*pow(t,3) + 0*pow(t,2) + 0*t + 0)*Y[i+3]);
}
```

A Figura 10 e a Figura 11 ilustram alguns exemplos de curvas geradas pelo método de Bézier. Note-se que a curva sempre está contida no polígono convexo determinado por seus pontos de controle.

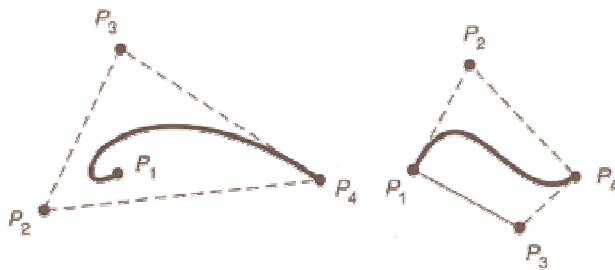


Figura 10 - Curvas de Bezier não extrapolam os limites do polígono definido por seus pontos de controle: [FOLEY, 1993]

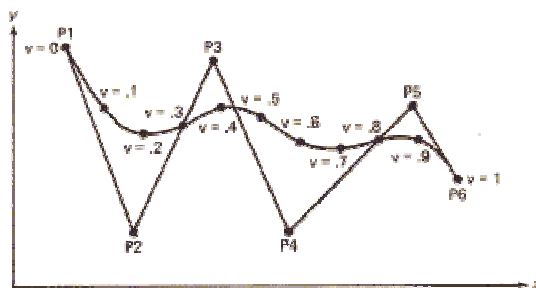


Figura 11 - Curvas de Bézier são funções de aproximação para P2, P3, P4 e P5 e de interpolação para P1 e P6 (6 pontos): [ARTWICK, 1984]

## B-SPLINE

O nome B-Spline faz alusão ao termo da língua inglesa utilizado para definir a curvatura natural assumida por uma barra ou fita metálica quando fletida por cargas distribuídas entre apoios. A expressão matemática que descreve esta curvatura é denominada Spline Cúbica Natural. A Spline natural é no entanto um polinômio cúbico contínuo, ou seja, é gerado em função de todos os pontos de controle da curva. A alteração em qualquer um deles provoca alteração em toda a curva. Este comportamento é, na maioria das vezes, pouco apropriado para aplicações de ajuste de curva interativas. A Figura 12 ilustra um Spline cúbica natural criada a partir de 7 pontos de controle.

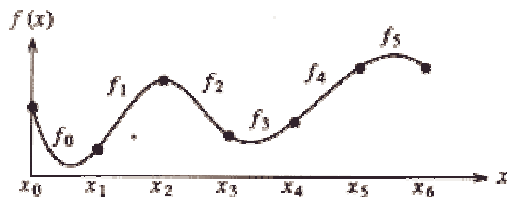


Figura 12 - Spline Cúbica Natural: [WOLBERG, 1990]

O polinômio para geração de Splines Naturais tem a seguinte forma geral:

$$f_k(x) = a_3(x-x_k)^3 + a_2(x-x_k)^2 + a_1(x-x_k) + a_0 \quad (4.3.1)$$

George Wolberg, dedica todo o Apêndice 2 de seu livro "Digital Image Warping" à demonstração da obtenção dos coeficientes do polinômio para a Spline natural.

B-Spline é uma versão simplificada da Spline Natural que implementa o controle local da curva. Alterações nos pontos de controle da B-Spline apenas se propagam para os vizinhos mais próximos, em função da sua ordem de continuidade. A função B-Spline, no entanto, é um ajustador de aproximação, pois a curva gerada não passa pelos pontos de controle, ao contrário do ajustador de Hermite, que efetua a interpolação dos pontos de controle (os pontos de controle pertencem à curva).

A B-Spline utiliza o mesmo conceito de curvas de ponderação para a mistura dos pontos de controle. Os ajustadores B-Spline são classificados de acordo com o número de pontos envolvidos na sua geração e pela área de influência de cada ponto na curva. A Figura 13 ilustra a função de mistura para uma B-Spline com 6 pontos de controle e fator de continuidade 3 (cada trecho da curva é afetado por 3 pontos).

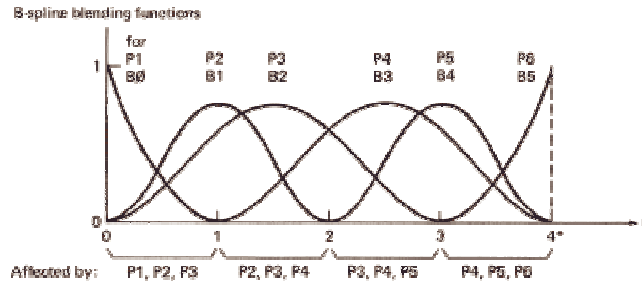


Figura 13 - Função de mistura para B-Spline (6x3): [WOLBERG, 1990]

A Figura 14 ilustra os diversos resultados obtidos quando se varia o grau de continuidade de uma B-Spline, maiores graus de continuidade produzem curvas menos “agressivas”.

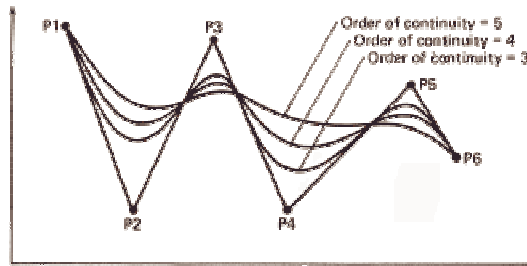


Figura 14 - Curva obtida com função de mistura em diversas ordens de continuidade: [ARTWICK, 1984]

Muito embora o conceito de mistura da B-Spline seja bastante simples, a matemática necessária para sua implementação é bastante complexa. Assim como em Hermite, um ajustador B-Spline utilizando 4 pontos de controle e fator de continuidade 4 (4 polinômios de terceira ordem) constitui um bom equilíbrio entre funcionalidade e complexidade. A Figura 15 ilustra a geometria proposta para controlar a influência de cada um dos quatro fatores de controle da curvas.

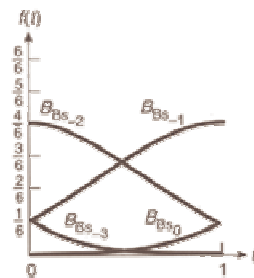


Figura 15 - Geometria de composição da função B-Spline (4x4): [FOLEY, 1993]

A matriz para ajustadores B-Spline, que descreve as curvas de mistura de sua geometria, são definidas como se segue:

$$M_{Bs} = \frac{1}{6} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 0 & 4 \\ -3 & 3 & 3 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \quad (2)$$

A função geradora da curva B-Spline, utilizando 4 pontos de controle e fator de continuidade 4, encontra-se detalhada no código fonte provido pelo autor, sendo esta a aplicação direta da matriz  $M_{Bs}$ .

e possuindo a seguinte forma geral (o valores nulos foram incluídos no algoritmo com a finalidade de facilitar a compreensão do leitor):

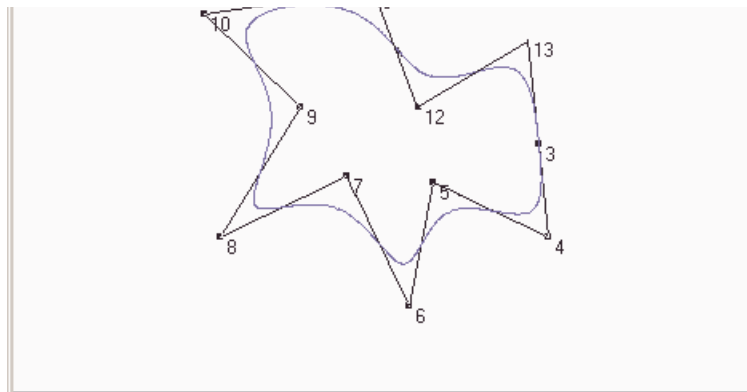
```

for(t=0;t<=1;t+=Step)
{
x=((-1*pow(t,3) +3*pow(t,2) -3*t +1)*X[i] +
( 3*pow(t,3) -6*pow(t,2)+0*t +4)*X[i+1] +
(-3*pow(t,3) +3*pow(t,2)+3*t +1)*X[i+2] +
( 1*pow(t,3) +0*pow(t,2)+0*t +0)*X[i+3])/6);

y=((-1*pow(t,3) +3*pow(t,2) -3*t +1)*Y[i] +
( 3*pow(t,3) -6*pow(t,2) +0*t +4)*Y[i+1] +
(-3*pow(t,3) +3*pow(t,2) +3*t +1)*Y[i+2] +
( 1*pow(t,3) +0*pow(t,2) +0*t +0)*Y[i+3])/6);
}

```

Os ajustadores de B-Spline abrigam também uma outra característica interessante. Para gerar-se uma curva convexa, com o mesmo grau de continuidade do restante dos segmentos, basta repetir-se os 3 primeiros pontos ao final da sequência de pontos de controle. A Figura 16 ilustra a geração de uma curva convexa e contínua utilizando-se da ferramenta provida pelo autor juntamente com este texto.



◀ ▶ ↻ 🔍