



Superfícies Paramétricas Bicúbicas em 3D

Superfícies bicúbicas paramétricas são uma generalização das curvas cúbicas paramétricas. Para induzir a sua notação matemática vamos nos lebrar primeiramente da forma geral de uma curva paramétrica $Q(t) = T \cdot M \cdot G$, onde o vetor de geometria G é uma constante e pode adquirir diferentes formas, dependendo se temos uma curva de Hermite, Bézier ou Spline.

Se nós, por uma questão de conveniência notacional, substituirmos o parâmetro t por s , podemos expressar a fórmula anterior da seguinte forma: $Q(s) = T \cdot M \cdot G$, o que não muda nada.

Porém, já que estamos trabalhando com pontos representados em um espaço tridimensional, se permitirmos que os pontos definindo G variem ao longo de algum caminho que é parametrizado em t , podemos expressar $Q(s)$ como uma família de curvas parametrizadas por t :

$$Q(s, t) = S \cdot M \cdot G(t) = S \cdot M \cdot \begin{bmatrix} G_1(t) \\ G_2(t) \\ G_3(t) \\ G_4(t) \end{bmatrix} \quad (\text{Eq. 7.1})$$

Agora, se tomarmos um valor fixo t_1 qualquer de t , a equação $Q(s, t_1)$ é simplesmente uma curva em 3D porque $G(t_1)$ é uma constante e Q vai variar apenas ao longo de s .

Se permitirmos que t tome um valor levemente diferente t_2 , onde o valor de $t_1 - t_2$ é bastante pequeno, $Q(s, t_2)$ será uma curva levemente diferente.



Se repetirmos este processo para um conjunto arbitrário de valores de t entre 0 e 1, estaremos definindo uma completa família de curvas, cada uma delas arbitrariamente próxima da anterior, dependendo de quanto t variou.

O conjunto de todas estas curvas define uma **superfície curva**. Se as matrizes $\mathbf{G}_i(t)$ definirem por sua vez também curvas cúbicas, estaremos descrevendo uma **superfície paramétrica bicúbica**, que se comporta como uma família de curvas de Hermite, Bézier ou Spline (dependendo de $G(t)$) tanto no sentido de s como no de t .

Se assumimos que as $G_i(t)$ são cúbicas, cada uma delas representada por sua vez através da equação $G_i(t) = T \cdot M \cdot \mathbf{G}_i$

(que nós já conhecemos), onde¹ $\mathbf{G}_i = \begin{bmatrix} \mathbf{g}_{i1} & \mathbf{g}_{i2} & \mathbf{g}_{i3} & \mathbf{g}_{i4} \end{bmatrix}^T$.

Aqui, \mathbf{g}_{i1} é o primeiro elemento do vetor de geometria no sentido t da curva (Hermite, Bézier ou Spline) $G_i(t)$ e assim por diante.

Se transpusermos a equação $G_i(t) = T \cdot M \cdot \mathbf{G}_i$ usando a regra de identidade $(A \cdot B \cdot C)^T = C^T \cdot B^T \cdot A^T$, teremos como resultado:

$$G_i(t) = \mathbf{G}_i^T \cdot M^T \cdot T^T = \begin{bmatrix} \mathbf{g}_{i1} & \mathbf{g}_{i2} & \mathbf{g}_{i3} & \mathbf{g}_{i4} \end{bmatrix} \cdot M^T \cdot T^T \quad (\text{Eq. 7.2})$$

1. Usamos G e g em negrito para diferenciar estes valores representando as matrizes de geometria das funções $G(t)$.



Se substituirmos este resultado para todos os quatro pontos (de controle), teremos:

$$Q(s, t) = S \cdot M \cdot \begin{bmatrix} \mathbf{g}_{11} & \mathbf{g}_{12} & \mathbf{g}_{13} & \mathbf{g}_{14} \\ \mathbf{g}_{21} & \mathbf{g}_{22} & \mathbf{g}_{23} & \mathbf{g}_{24} \\ \mathbf{g}_{31} & \mathbf{g}_{32} & \mathbf{g}_{33} & \mathbf{g}_{34} \\ \mathbf{g}_{41} & \mathbf{g}_{42} & \mathbf{g}_{43} & \mathbf{g}_{44} \end{bmatrix} \cdot M^T \cdot T^T \quad (\text{Eq. 7.3})$$

Ou, representando (Eq. 7.3) de forma mais compacta:

$$Q(s, t) = S \cdot M \cdot \mathbf{G} \cdot M^T \cdot T^T, \quad 0 \leq s, t \leq 1 \quad (\text{Eq. 7.4})$$

Se reescrevermos (Eq. 7.4) de forma separada para cada coordenada x, y e z , teremos a **forma geral** do sistema de equações em forma paramétrica de uma superfície bicúbica:

$$\begin{aligned} x(s, t) &= S \cdot M \cdot \mathbf{G}_x \cdot M^T \cdot T^T \\ y(s, t) &= S \cdot M \cdot \mathbf{G}_y \cdot M^T \cdot T^T, \quad 0 \leq s, t \leq 1 \\ z(s, t) &= S \cdot M \cdot \mathbf{G}_z \cdot M^T \cdot T^T \end{aligned} \quad (\text{Eq. 7.5})$$



Superfícies Bicúbicas de Hermite em 3D

Superfícies de Hermite podem ser completamente definidas por uma matriz de geometria 4x4 \mathbf{G}_H .

Relembrando a definição das Curvas de Hermite em 3D:

$$\begin{aligned}x(t) &= \mathbf{T} \cdot \mathbf{M}_H \cdot \mathbf{G}_{H_x} \\y(t) &= \mathbf{T} \cdot \mathbf{M}_H \cdot \mathbf{G}_{H_y} \\z(t) &= \mathbf{T} \cdot \mathbf{M}_H \cdot \mathbf{G}_{H_z}\end{aligned}\tag{Eq. 7.6}$$

Onde a Matriz dos Parâmetros (da curva cúbica) é dada por:

$$\mathbf{T} = [t^3 \ t^2 \ t \ 1]\tag{Eq. 7.7}$$

e a Matriz de Hermite é dada por:

$$\mathbf{M}_H = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix}\tag{Eq. 7.8}$$

e, por fim, a Matriz de Geometria dada pelos coeficientes:

$$\mathbf{G}_H = \begin{bmatrix} P_1 \\ P_4 \\ R_1 \\ R_4 \end{bmatrix}\tag{Eq. 7.9}$$



Se tomarmos cada uma das equações (Eq. 7.6) e substituirmos t por s , para obter $x(s) = S \cdot M_H \cdot G_{H_x}$, teremos expresso uma curva de Hermite em termos de s .

Se agora supusermos que G_{H_x} não é constante, mas sim que varia em função de t , podemos reescrever:

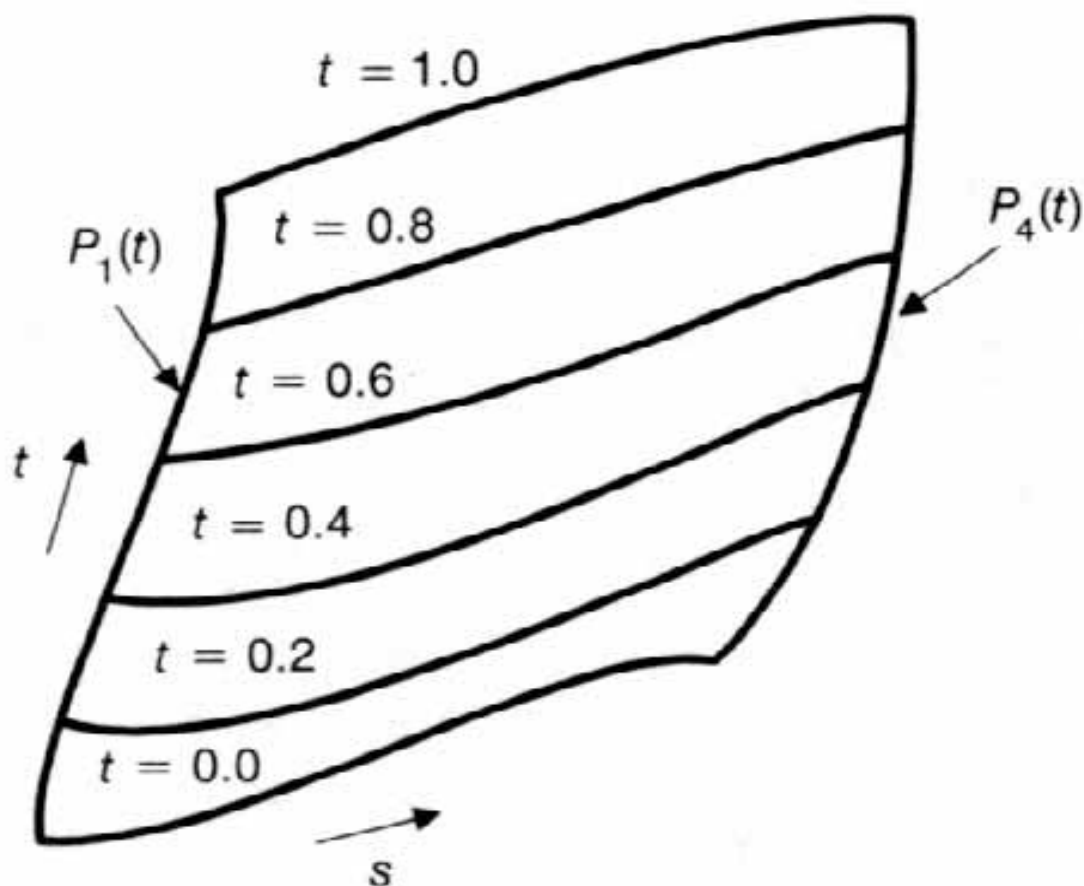
$$x(s, t) = S \cdot M_H \cdot G_{H_x}(t) = S \cdot M_H \cdot \begin{bmatrix} P_1(t) \\ P_4(t) \\ R_1(t) \\ R_4(t) \end{bmatrix}_x \quad (\text{Eq. 7.10})$$

As funções $P_{1x}(t)$ e $P_{4x}(t)$ definem as componentes em x dos pontos iniciais e finais para a curva no parâmetro s . Similarmente, $R_{1x}(t)$ e $R_{4x}(t)$ são os vetores tangentes nestes pontos.

Na Figura 7.1. pode-se observar uma superfície de Hermite definida por duas "bordas" dadas por $P_1(t)$ e $P_4(t)$ e um conjunto de cúbicas s que são definidas nos pontos $t = 0.0, 0.2, 0.4, 0.6, 0.8$ e 1.0 . O "retalho" de superfície é basicamente uma interpolação entre $Q(s, 0)$ e $Q(s, 1)$ com passo 0.2 definido em t .



Figura 7.1. Superfície Bicúbica representada como conjunto de curvas (s variável e t fixo) “armadas” entre duas curvas ao longo de t.





Para podermos definir melhor como funciona uma superfície bicúbica, representemos $P_1(t)$, $P_4(t)$, $R_1(t)$ e $R_4(t)$ na forma de Hermite como:

$$\begin{aligned}
 P_{1_x}(t) &= T \cdot M_H \cdot \begin{bmatrix} \mathbf{g}_{11} \\ \mathbf{g}_{12} \\ \mathbf{g}_{13} \\ \mathbf{g}_{14} \end{bmatrix}_x & P_{4_x}(t) &= T \cdot M_H \cdot \begin{bmatrix} \mathbf{g}_{21} \\ \mathbf{g}_{22} \\ \mathbf{g}_{23} \\ \mathbf{g}_{24} \end{bmatrix}_x \\
 R_{1_x}(t) &= T \cdot M_H \cdot \begin{bmatrix} \mathbf{g}_{31} \\ \mathbf{g}_{32} \\ \mathbf{g}_{33} \\ \mathbf{g}_{34} \end{bmatrix}_x & R_{4_x}(t) &= T \cdot M_H \cdot \begin{bmatrix} \mathbf{g}_{41} \\ \mathbf{g}_{42} \\ \mathbf{g}_{43} \\ \mathbf{g}_{44} \end{bmatrix}_x
 \end{aligned} \tag{Eq. 7.11}$$

E estas quatro cúbicas podem por sua vez ser reescritas como uma única equação (para x , y e z):

$$\begin{bmatrix} P_{1_x}(t) & P_{4_x}(t) & R_{1_x}(t) & R_{4_x}(t) \end{bmatrix} = T \cdot M_H \cdot \mathbf{G}_{H_x}^T \tag{Eq. 7.12}$$

onde:

$$\mathbf{G}_{H_x} = \begin{bmatrix} \mathbf{g}_{11} & \mathbf{g}_{12} & \mathbf{g}_{13} & \mathbf{g}_{14} \\ \mathbf{g}_{21} & \mathbf{g}_{22} & \mathbf{g}_{23} & \mathbf{g}_{24} \\ \mathbf{g}_{31} & \mathbf{g}_{32} & \mathbf{g}_{33} & \mathbf{g}_{34} \\ \mathbf{g}_{41} & \mathbf{g}_{42} & \mathbf{g}_{43} & \mathbf{g}_{44} \end{bmatrix}_x \tag{Eq. 7.13}$$



A transposição de ambos os lados da (Eq. 7.12) resulta em:

$$\begin{bmatrix} P_{1_x}(t) \\ P_{4_x}(t) \\ R_{1_x}(t) \\ R_{4_x}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{g}_{11} & \mathbf{g}_{12} & \mathbf{g}_{13} & \mathbf{g}_{14} \\ \mathbf{g}_{21} & \mathbf{g}_{22} & \mathbf{g}_{23} & \mathbf{g}_{24} \\ \mathbf{g}_{31} & \mathbf{g}_{32} & \mathbf{g}_{33} & \mathbf{g}_{34} \\ \mathbf{g}_{41} & \mathbf{g}_{42} & \mathbf{g}_{43} & \mathbf{g}_{44} \end{bmatrix} \cdot \mathbf{M}_H^T \cdot \mathbf{T}^T = \mathbf{G}_{H_x} \cdot \mathbf{M}_H^T \cdot \mathbf{T}^T \quad (\text{Eq. 7.14})$$

Substituindo (Eq. 7.14) na (Eq. 7.10) temos:

$$\begin{aligned} x(s, t) &= S \cdot \mathbf{M}_H \cdot \mathbf{G}_{H_x} \cdot \mathbf{M}_H^T \cdot \mathbf{T}^T \\ y(s, t) &= S \cdot \mathbf{M}_H \cdot \mathbf{G}_{H_y} \cdot \mathbf{M}_H^T \cdot \mathbf{T}^T \\ z(s, t) &= S \cdot \mathbf{M}_H \cdot \mathbf{G}_{H_z} \cdot \mathbf{M}_H^T \cdot \mathbf{T}^T \end{aligned} \quad (\text{Eq. 7.15})$$

Assim, vemos que podemos definir um ponto qualquer sobre uma superfície de Hermite em função de duas matrizes de parâmetros S e T , representando t^3, t^2, t, s^3, s^2 e s .

As três matrizes 4x4 $\mathbf{G}_{H_x}, \mathbf{G}_{H_y}, \mathbf{G}_{H_z}$ têm o mesmo papel numa superfície de Hermite que tinha a Matriz \mathbf{G}_H para curvas.

Os significados dos 16 elementos de \mathbf{G}_{H_x} podem ser compreendidos se tomarmos por base (Eq. 7.10) e (Eq. 7.10):

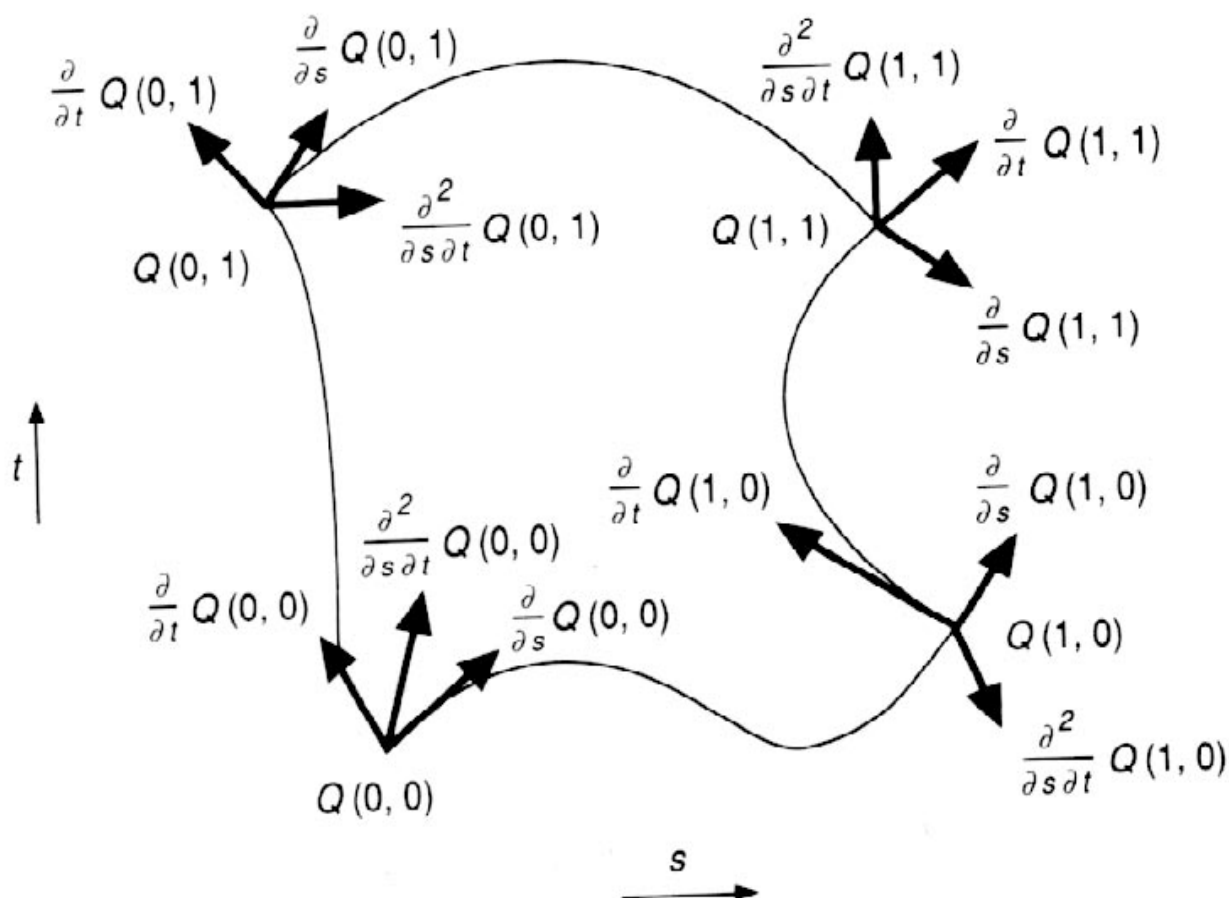
a) O elemento g_{11x} é $x(0,0)$ porque é o ponto inicial $P_{1x}(t)$, que por sua vez é o ponto inicial para $x(s,0)$. Da mesma forma, g_{12x} é $x(0,1)$ pois é o ponto final de $P_{1x}(t)$, que por sua vez é o ponto inicial de $x(s,1)$.



b) Além disso, g_{13x} é $\delta x / \delta t(0,0)$ porque é o vetor tangente inicial para $P_{1x}(t)$ e g_{33x} é $\delta^2 x / \delta s \delta t(0,0)$ porque é o vetor tangente inicial de $R_{1x}(t)$, que por sua vez é a curvatura inicial de $x(s,0)$.

A disposição destes vetores está exemplificada pela figura abaixo.

Figura 7.2: Superfície de Hermite e seus Vetores





Usando estas interpretações podemos reescrever G_{H_x} da seguinte forma:

$$\mathbf{G}_{H_x} = \begin{bmatrix} x(0, 0) & x(0, 1) & \frac{\partial}{\partial t}x(0, 0) & \frac{\partial}{\partial t}x(0, 1) \\ x(1, 0) & x(1, 1) & \frac{\partial}{\partial t}x(1, 0) & \frac{\partial}{\partial t}x(1, 1) \\ \frac{\partial}{\partial s}x(0, 0) & \frac{\partial}{\partial s}x(0, 1) & \frac{\partial^2}{\partial s \partial t}x(0, 0) & \frac{\partial^2}{\partial s \partial t}x(0, 1) \\ \frac{\partial}{\partial s}x(1, 0) & \frac{\partial}{\partial s}x(1, 1) & \frac{\partial^2}{\partial s \partial t}x(1, 0) & \frac{\partial^2}{\partial s \partial t}x(1, 1) \end{bmatrix} \quad (\text{Eq. 7.16})$$

Podemos interpretar G_{H_x} da seguinte forma:

- A porção 2x2 superior esquerda contém as coordenadas x dos quatro cantos da superfície.
- As áreas 2x2 superior direita e inferior esquerda contém os vetores tangentes ao longo de cada direção paramétrica (s e t respectivamente).
- A porção 2x2 no canto inferior direito contém as derivadas parciais simultaneamente a s e t . São chamadas de *twist* (virada) porque quanto maiores, maior será a "orelha de burro" no canto da superfície.

Dados para definir os valores de G_{H_x} ficam por conta do usuário, da mesma forma que ficava em G_H nas curvas. As considerações de continuidade de superfícies adjacentes são as mesmas das curvas.

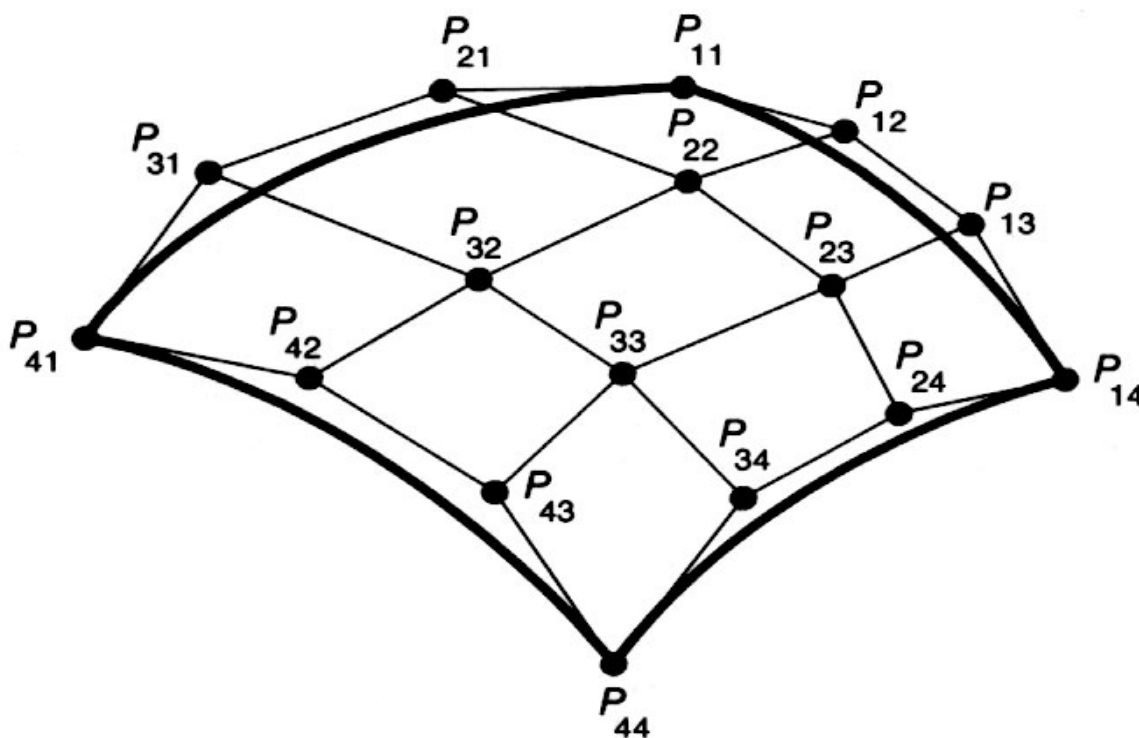
Superfícies Bicúbicas de Bézier

Superfícies Bicúbicas de Bézier podem ser formuladas derivando-se as exatamente da mesma maneira que as superfícies de Hermite. Obtemos:

$$\begin{aligned} x(s, t) &= S \cdot M_B \cdot G_{B_x} \cdot M_B^T \cdot T^T \\ y(s, t) &= S \cdot M_B \cdot G_{B_y} \cdot M_B^T \cdot T^T \\ z(s, t) &= S \cdot M_B \cdot G_{B_z} \cdot M_B^T \cdot T^T \end{aligned} \quad (\text{Eq. 7.17})$$

Uma matriz de geometria de Bézier consiste de 16 pontos de controle.

Figura 7.3: Superfície de Bézier

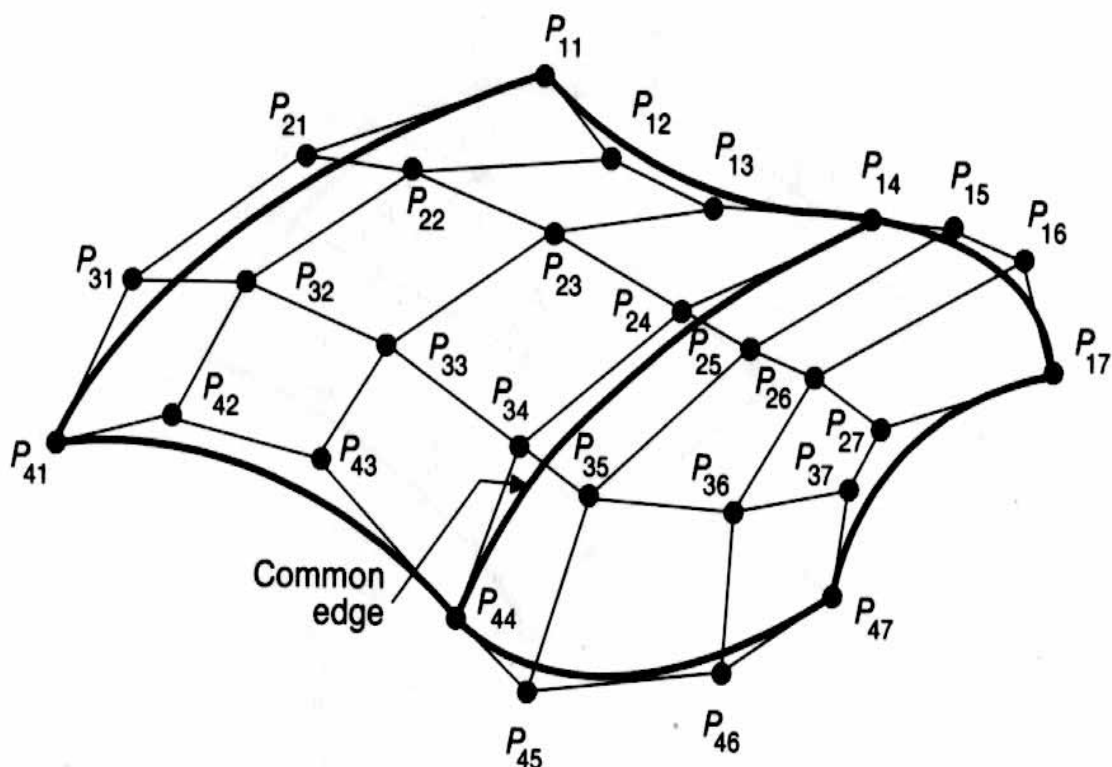




Superfícies de Bézier são interessantes para aplicações de engenharia pelas mesmas razões das curvas: São definidas somente por pontos (descrição intuitiva) e passam através de alguns de seus pontos de controle, permitindo um controle exato de seus limites. Na verdade, uma superfície de Bézier passa pelos quatro pontos de controle extremos: P_{11} , P_{14} , P_{41} e P_{44} .

Continuidade C^0 e G^0 ao longo das bordas das superfícies é obtida fazendo-se os quatro pontos de controle comuns ao longo da borda serem iguais. A continuidade G^1 ocorre quando os dois conjuntos de quatro pontos de controle em cada lado da borda são colineares com os pontos da borda. Isto é exemplificado na onde temos (P_{13}, P_{14}, P_{15}) , (P_{23}, P_{24}, P_{25}) , (P_{33}, P_{34}, P_{35}) e (P_{43}, P_{44}, P_{45}) como conjuntos de pontos, cada qual sobre uma reta.

Figura 7.4: Duas superfícies adjacentes com continuidade G^1 .





Superfícies B-Spline Bicúbicas

De forma similar às superfícies de Bézier, representamos uma superfície B-Spline bicúbica através de um conjunto de pelo menos 16 pontos de controle. Para cada segmento de uma B-Spline bicúbica, representamos a superfície através do sistema:

$$\begin{aligned}x(s, t) &= S \cdot M_{BS} \cdot \mathbf{G}_{BS_x} \cdot M_{BS}^T \cdot T^T \\y(s, t) &= S \cdot M_{BS} \cdot \mathbf{G}_{BS_y} \cdot M_{BS}^T \cdot T^T \\z(s, t) &= S \cdot M_{BS} \cdot \mathbf{G}_{BS_z} \cdot M_{BS}^T \cdot T^T\end{aligned} \quad (\text{Eq. 7.18})$$

As B-Splines são extremamente práticas no que diz respeito à continuidade: Em função de sua definição, a continuidade C^2 é garantida ao longo de qualquer número de B-Splines bicúbicas contíguas.

Nenhum tipo de arranjo especial é necessário, a não ser o cuidado de não duplicar pontos de controle, o que provoca descontinuidades e “dobras”.

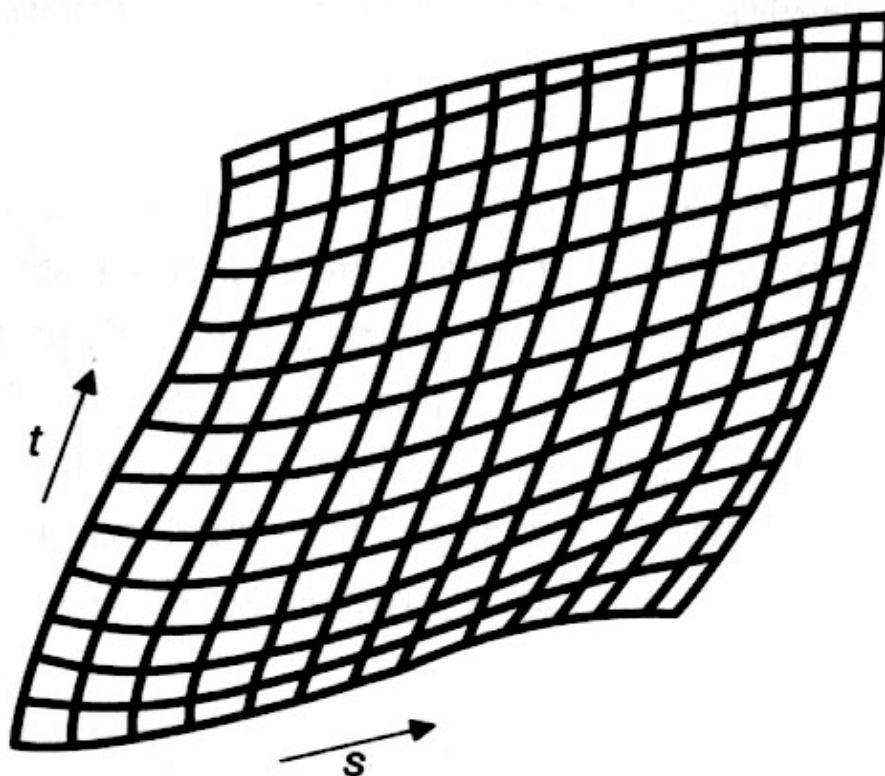


Desenhando Superfícies Bicúbicas

Podemos utilizar as mesmas duas filosofias básicas das curvas para a plotagem de superfícies bicúbicas: a) Cálculo iterativo ou b) subdivisão progressiva. Destes, o cálculo iterativo possui a vantagem de permitir-se começar em qualquer lugar da superfície (o que é útil em caso de superfícies parcialmente clipadas) e de prover um algoritmo incremental extremamente rápido que somente utiliza somas, o algoritmo das *forward differences*, que já foi visto.

Para a plotagem através de um dos dois algoritmos iterativos, basta fixar um dos parâmetros (s ou t) e variar o outro de 0 a 1, de forma que se plota uma curva de cada vez. Pode-se definir um passo fixo em s e depois em t e plotar as várias curvas formando um grid.

Figura 7.5: Uma superfície plotada como conjunto de curvas lado a lado, tanto seguindo s como seguindo t .





Abaixo estão os dois algoritmos já vistos para curvas (plotagem direta e plotagem iterativa através de forward differences) adaptados para curvas e expressos em syntax da linguagem "C".

Figura 7.6: Algoritmo para cálculo de um ponto através do cálculo direto das blending functions

```
typedef double Coeffs[4][4][3];
void DrawSurface (
    Coeffs coefficients,      /* Coefficients for  $Q(s,t)$  */
    int  $n_s$ ,                 /* No. of curves of constant  $s$  to draw, typically 5–10 */
    int  $n_t$ ,                 /* No. of curves of constant  $t$  to draw, typically 5–10 */
    int  $n$ )                  /* No. of steps used to draw each curve, typically 20–100 */
{
    double  $\delta = 1.0 / n$ ;      /* Step size to use in drawing each curve */
    double  $\delta_s = 1.0 / (n_s - 1)$ ; /* Step size in  $s$  to increment to next curve of constant  $t$  */
    double  $\delta_t = 1.0 / (n_t - 1)$ ; /* Step size in  $t$  to increment to next curve of constant  $s$  */
    int  $i, j$ ; double  $s, t$ ;
    /* Draw  $n_s$  curves of constant  $s$ , for  $s = 0, \delta_s, 2\delta_s, \dots, 1$  */
    for ( $i = 0, s = 0.0; i < n_s; i++, s += \delta_s$ ) {
        /* Draw a curve of constant  $s$ , varying  $t$  from 0 to 1. */
        /* X, Y, and Z are functions to evaluate the bicubics. */
        MoveAbs3 (X ( $s, 0.0$ ), Y ( $s, 0.0$ ), Z ( $s, 0.0$ ));
        for ( $j = 1, t = \delta; j < n; j++, t += \delta$ ) {
            /*  $n-1$  steps are used as  $t$  varies from  $\delta$  to 1 for each curve. */
            LineAbs3 (X ( $s, t$ ), Y ( $s, t$ ), Z ( $s, t$ ));
        }
    }
    /* Draw  $n_t$  curves of constant  $t$ , for  $t = 0, \delta_t, 2\delta_t, \dots, 1$  */
    for ( $i = 0, t = 0.0; i < n_t; i++, t += \delta_t$ ) {
        /* Draw a curve of constant  $t$ , varying  $s$  from 0 to 1. */
        MoveAbs3 (X ( $0.0, t$ ), Y ( $0.0, t$ ), Z ( $0.0, t$ ));
        for ( $j = 1, s = \delta; j < n; j++, s += \delta$ ) {
            /*  $n-1$  steps are used as  $s$  varies from  $\delta$  to 1 for each curve. */
            LineAbs3 (X ( $s, t$ ), Y ( $s, t$ ), Z ( $s, t$ ));
        }
    }
} /* DrawSurface */
```



Figura 7.7: Algoritmo para plotagem de superfícies utilizando forward differences

```
typedef double Coeffs[4][4][3];

void DrawSurfaceFwdDif (
    Coeffs A, /* Coefficients for  $Q(s, t)$  */
    int  $n_s$ , /* Number of curves of constant  $s$  to be drawn, typically 5 to 10. */
    int  $n_t$ , /* Number of curves of constant  $t$  to be drawn, typically 5 to 10. */
    int  $n$ ) /* Number of steps to use in drawing each curve, typically 20 to 100. */
{
    /* Initialize */
    double  $\delta_s = 1.0 / (n_s - 1.0)$ ; double  $\delta_t = 1.0 / (n_t - 1.0)$ ;
    /* "*" indicates matrix multiplication */
     $DD_x = E(\delta_s) * A_x * E(\delta_t)^T$ ;
     $DD_y = E(\delta_s) * A_y * E(\delta_t)^T$ ;
     $DD_z = E(\delta_s) * A_z * E(\delta_t)^T$ ;

    /* Draw  $n_s$  curves of constant  $s$ , for  $s = 0, \delta_s, 2\delta_s, \dots, 1$  */
    for ( $i = 0; i < n_s; i++$ ) {
        /* Procedure from Section 11.2.9 to draw a curve */
        DrawCurveFwdDif ( $n$ , First row of  $DD_x$ , First row of  $DD_y$ , First row of  $DD_z$ );
        /* Prepare for next iteration */
        Apply equation 11.92 to  $DD_x, DD_y$ , and  $DD_z$ ;
    }
    /* Transpose  $DD_x, DD_y, DD_z$  so can continue working with rows */
     $DD_x = DD_x^T$ ;  $DD_y = DD_y^T$ ;  $DD_z = DD_z^T$ ;
    /* Draw  $n_t$  curves of constant  $t$ , for  $t = 0, \delta_t, 2\delta_t, 3\delta_t, \dots, 1$  */
    for ( $i = 0; i < n_t; i++$ ) {
        DrawCurveFwdDif ( $n$ , First row of  $DD_x$ , First row of  $DD_y$ , First row of  $DD_z$ );
        /* Prepare for next iteration */
        Apply equation 11.92 to  $DD_x, DD_y$ , and  $DD_z$ ;
    }
} /* DrawSurfaceFwdDif */
```