

TEMA 6 - REPRESENTACIÓN DE CURVAS Y SUPERFICIES

La mayor parte de los objetos existentes en el mundo real presentan formas continuas y suaves, y para nada pueden asemejarse a formas poligonales. La mayoría de las aplicaciones gráficas necesitan modelar objetos presentes en el mundo real, y por lo tanto se hace necesario poder generar curvas y superficies de una forma más exacta que una simple sucesión de segmentos rectos cortos. El diseño asistido por ordenador (CAD), los tipos de letra de alta calidad, los trazos de un artista, el path que debe seguir una cámara a lo largo de una secuencia; todos ellos contienen superficies y curvas suaves.

La necesidad de poder representar curvas y superficies viene por dos sentidos. En primer lugar para modelar objetos existentes (coches, caras, montañas, etc.), en donde una descripción matemática del objeto puede no estar disponible. Por supuesto, podemos usar para modelar las coordenadas de los infinitos puntos del objeto, lo cual no es nada viable. También podemos representarlo aproximadamente usando planos, esferas u otras primitivas sencillas, fáciles de describir matemáticamente, pero seguiríamos sin lograr una representación adecuada.

En segundo lugar para modelar algo que aún no exista físicamente, por ejemplo cuando se está diseñando un nuevo prototipo de coche o de avión. Para crearlo, el usuario puede ir esculpiéndolo interactivamente, o bien especificar su descripción matemática, o dar una descripción aproximada para luego ir retocándola.

En este capítulo vamos a ver el modelado de curvas y superficies paramétricas, con lo cual obtendremos una visión general del área de modelado geométrico, el cual es un amplísimo campo dentro de la informática gráfica.

6.1 CURVAS CÚBICAS PARAMÉTRICAS

Cualquier línea curva puede representarse mediante un conjunto de segmentos rectos. Cuanto mayor sea el número de ellos mejor será la aproximación. De igual manera, cualquier tipo de superficie puede aproximarse por un conjunto de polígonos conectados (mesh), acercándonos más a la superficie original cuanto más polígonos usemos. El principal problema de este tipo de aproximaciones es la enorme cantidad de coordenadas de cada punto (extremos de cada segmento recto en las curvas, o vértices de cada polígono en las superficies) que debemos almacenar para lograr una aproximación razonable. Además, otro problema adicional es que la manipulación interactiva de los datos para aproximar una forma dada es tediosa, debido a los numerosos

puntos que hay que posicionar correctamente.

Para solucionar este hándicap, existe una forma de representación mas manipulable y compacta para representar curvas y superficies. La aproximación general consiste en usar funciones de mayor grado que las funciones lineales (que es lo que son los segmentos rectos y los polígonos). Por ejemplo, si tenemos tres puntos no podemos obtener una recta que pase por los tres, si éstos no son colineales. Esto es debido a que la expresión para una recta es $y = ax + b$, por lo que sólo disponemos de dos coeficientes (a y b) para intentar cumplir tres condiciones (una para obligar que pase por cada punto); necesitaríamos guardar dos rectas. Sin embargo, con un polinomio de grado dos tendríamos tres coeficientes, con lo que sí conseguiríamos obtener una única función que pasara por los tres puntos. De ahí que el usar funciones de mayor grado nos permiten aproximar mejor y con menos información un trozo de curva que mediante segmentos rectos.

Estas funciones seguirán siendo todavía aproximaciones a la forma deseada, pero requerirán mucho menos espacio de almacenamiento y ofrecerán una manipulación interactiva mucho más sencilla que con las funciones lineales. Para este tipo de aproximaciones de mayor grado existen tres métodos posibles:

a) Podemos expresar las componentes y y z como **funciones explícitas** de x

$$\left. \begin{array}{l} y = f(x) \\ z = g(x) \end{array} \right\}$$

Un primer problema de esta representación es que es imposible obtener múltiples valores de y para un valor de x , por lo que las curvas tales como circunferencias y elipses deben representarse por varios segmentos curvos. Un segundo problema es que esta definición no es invariante frente a rotaciones; describir una versión rotada de la curva requeriría un gran coste e incluso se necesitaría dividir la curva en varias partes. Un problema más es que describir curvas que posean tangente vertical es muy difícil, ya que es complicado representar una pendiente infinita.

b) Podemos elegir como modelo de la curva la solución de una **ecuación implícita** de la forma

$$f(x, y, z) = 0$$

Esta solución también supone nuevos problemas. En primer lugar, la ecuación dada puede tener más soluciones de las que queremos. Por ejemplo, para modelar una circunferencia podemos usar la ecuación $x^2 + y^2 = 1$, pero ¿cómo podríamos modelar una semicircunferencia? Deberíamos añadir requisitos tales como $x \geq 0$ que no se hayan incluidos dentro de la propia ecuación. En segundo lugar, si deseamos unir dos segmentos curvos definidos en forma implícita, puede ser muy difícil determinar si coinciden sus tangentes en el

punto de unión. La continuidad de las tangentes es un detalle crítico en la mayoría de las aplicaciones.

c) La **representación paramétrica** para las curvas,

$$\left. \begin{aligned} x &= x(t) \\ y &= y(t) \\ z &= z(t) \end{aligned} \right\}$$

soluciona todos los problemas de los dos métodos anteriores, y ofrece además una serie de ventajas adicionales que la hacen ideal para lo que queríamos en un principio. Las curvas paramétricas sustituyen el uso de las pendientes geométricas (las cuales pueden ser infinitas) por vectores tangentes paramétricos (los cuales nunca serán infinitos). De esta manera, una curva va a aproximarse por segmentos curvos polinómicos en lugar de segmentos rectos como antes. Cada segmento curvo Q de la curva completa vendrá representado por tres funciones, x , y , y z , las cuales serán polinomios cúbicos en t .

Se suelen usar los polinomios cúbicos porque los de grado dos ofrecen poca flexibilidad a la hora de controlar la forma de la curva, y los de grado mayor que tres pueden introducir rizos innecesarios y además requieren más computación. Ninguna representación de grado inferior a tres permite que un segmento curvo pase a través de dos puntos dados con dos derivadas específicas en cada uno. Dado un polinomio cúbico con cuatro coeficientes, los usaremos como incógnitas para poder cumplir el sistema de ecuaciones formado por las cuatro condiciones que hemos dicho. Esto es similar a lo que hacemos con una recta: dos coeficientes y dos condiciones por cumplir (los dos extremos). Las derivadas de cada extremos vienen ya definidas por la propia recta, y no pueden ser controladas independientemente. Con polinomios cuadráticos (de grado dos) tenemos tres coeficientes, lo cual nos sirve para los dos extremos y alguna condición más.

Otro detalle más es que las curvas polinómicas cúbicas son las curvas de menor grado que pueden ser no planares en 3D. Es decir, un polinomio de grado dos con tres coeficientes viene dado por tres puntos, los cuales siempre forman un plano en el espacio. La curva resultante por lo tanto permanecerá sobre ese plano.

Los polinomios cúbicos que definen un segmento curvo $Q(t) = [x(t), y(t), z(t)]$ son de la forma

$$\left. \begin{aligned} x(t) &= a_x t^3 + b_x t^2 + c_x t + d_x \\ y(t) &= a_y t^3 + b_y t^2 + c_y t + d_y \\ z(t) &= a_z t^3 + b_z t^2 + c_z t + d_z \end{aligned} \right\} \forall t \in [0, 1]$$

Para tratar con segmentos finitos de la curva, y sin pérdida de generalidad, hemos restringido el parámetro t al intervalo $[0, 1]$. Definiendo el vector

$T = [t^3, t^2, t, 1]$ y definiendo también la matriz C de coeficientes de los tres polinomios como

$$C = \begin{pmatrix} a_x & a_y & a_z \\ b_x & b_y & b_z \\ c_x & c_y & c_z \\ d_x & d_y & d_z \end{pmatrix}$$

podemos reescribir la ecuación para la curva como

$$Q(t) = \begin{pmatrix} x(t) & y(t) & z(t) \end{pmatrix} = T \times C$$

En la figura 3 pueden verse dos segmentos curvos cúbicos paramétricos unidos y sus respectivos polinomios. Además, puede verse la habilidad de la representación paramétrica para representar de forma sencilla múltiples valores de y para un solo valor de x mediante polinomios explícitos en t .

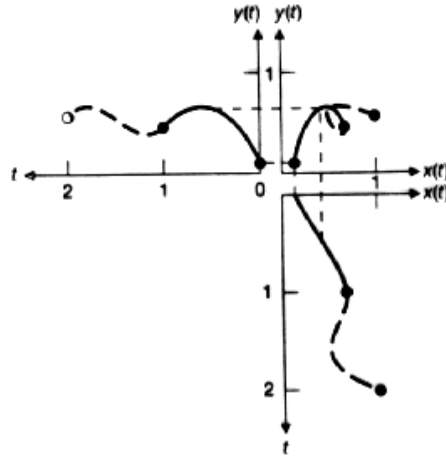


Figure 1

La derivada del vector $Q(t)$ es el vector tangente paramétrico de la curva. Aplicando esta definición obtenemos la siguiente ecuación:

$$\begin{aligned} \frac{d}{dt}Q(t) &= Q'(t) = \begin{pmatrix} x'(t) & y'(t) & z'(t) \end{pmatrix} = \frac{d}{dt}T \times C = \\ &= \begin{pmatrix} 3t^2 & 2t & 1 & 0 \end{pmatrix} \times C = \\ &= \begin{pmatrix} 3a_xt^2 + 2b_xt + c_x & 3a_yt^2 + 2b_yt + c_y & 3a_zt^2 + 2b_zt + c_z \end{pmatrix} \end{aligned}$$

Si dos segmentos curvos se unen en un extremo, diremos que la curva posee **continuidad geométrica** G^0 . Si las direcciones (pero no necesariamente las magnitudes) de los vectores tangentes de ambos vectores son iguales en el punto común, diremos que la curva tiene continuidad geométrica G^1 . En los

objetos hechos en CAD (Diseño Asistido por Ordenador) frecuentemente se requiere la continuidad G^1 entre segmentos consecutivos. Esta continuidad significa que las pendientes de los segmentos son iguales en el punto de unión. Para que dos vectores tangentes V_1 y V_2 tengan la misma dirección debe cumplirse que $V_1 = k \cdot V_2$, con $k > 0$.

Si los vectores tangentes de dos segmentos curvos son iguales (tanto en dirección como en magnitud) en su punto de unión, diremos que la curva tiene continuidad paramétrica C^1 . En general, si la dirección y magnitud del vector $d^n Q(t)/dt^n$ es igual para ambos segmentos, diremos que la curva tiene continuidad C^n . En la figura 4 pueden verse varias curvas con grados diferentes de continuidad. No se olviden que cualquier segmento curvo es siempre continuo en su interior; aquí nos estamos refiriendo a la continuidad de los extremos.

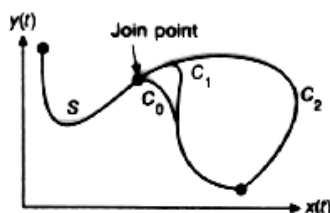


Figure 2

El vector tangente $Q'(t)$ mide la velocidad de un punto de la curva con respecto al parámetro t . Similarmente, la segunda derivada de $Q(t)$ representa la aceleración. Si una cámara se mueve a lo largo de una curva paramétrica a incrementos constantes de tiempo y va grabando un fotograma en cada paso, el vector tangente mide la velocidad de la cámara a lo largo de la curva. Tanto la velocidad como la aceleración de la cámara en los puntos extremos entre segmentos debe ser continua, para evitar movimientos bruscos en la secuencia resultante. Es precisamente esta aceleración en el punto de unión de la figura previa la que provoca que el segmento C^2 se alargue más a la derecha que el segmento C^1 , antes de doblarse hacia el extremo final.

En general, la continuidad C^1 implica G^1 , pero no al revés, ya que la continuidad G^1 es menos restrictiva que C^1 . Sin embargo, visualmente no se aprecia diferente, como puede verse en la figura 5.

El dibujo de una curva paramétrica es completamente diferente al de una función ordinaria, en la cual la variable independiente se va trazando sobre el eje x y la dependiente sobre el eje y . En un plot paramétrico, la variable independiente t nunca se dibuja. Esto significa que no puede determinarse a simple vista el vector tangente a la curva, ya que aunque la dirección es

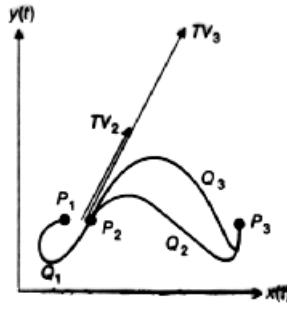


Figure 3

posible de determinar, no ocurre lo mismo con su magnitud. Esto puede verse en el siguiente ejemplo: sea el segmento $\gamma(t)$, $0 \leq t \leq 1$, cuyo vector tangente en $t = 0$ será $\gamma'(0)$. Si creamos un nuevo segmento, $\eta(t) = \gamma(2t)$, $0 \leq t \leq 1/2$, entonces los dibujos de ambas curvas son idénticos, y sin embargo, $\eta'(0) = 2\gamma'(0)$ ¹. Esto demuestra que dos curvas con idéntico dibujo tienen diferentes vectores tangente. Este es el motivo para la definición de la continuidad geométrica: para que dos segmentos se unan suavemente sólo es preciso que las direcciones de las tangentes coincidan, independientemente de sus magnitudes.

Un segmento de curva $Q(t)$ viene definido, como ya hemos dicho, por dos puntos extremos, dos vectores tangentes y una continuidad entre segmentos consecutivos. Cada polinomio cúbico tiene cuatro coeficientes, por lo que se necesitan cuatro condiciones que nos permitan formular cuatro ecuaciones cuyas incógnitas a calcular sean precisamente estos cuatro coeficientes. Existen diferentes tipos de curvas cúbicas, que veremos posteriormente en este capítulo. Para ver como dichos coeficientes dependen de las cuatro condiciones, consideremos que nuestro segmento de curva viene definido como

$$Q(t) = T \cdot C$$

siendo $C = M \cdot G$, donde a su vez M es una **matriz base** de 4x4, y G un vector columna de cuatro elementos de requisitos geométricos, llamado **vector de geometría**. Los requisitos geométricos van a ser precisamente las condiciones establecidas, tales como los puntos extremos o los vectores tangentes, que definen la curva en concreto. Definimos G_x para referirnos al vector columna formado exclusivamente por las componentes x del vector de geometría. G_y y G_z se definen de forma similar. Tanto M como G diferirán para cada tipo de curva cúbica.

Los elementos de M y G son constantes, por lo que el producto $T \cdot M \cdot G$

¹ $\eta(t) = \gamma(2t) = \gamma(u)$, siendo $u = 2t$
 $\eta'(t) = \frac{d\eta(t)}{dt} = \frac{d\gamma(u)}{dt} = \frac{d\gamma(u)}{du} \frac{du}{dt} = 2\gamma'(u)$

nos da tres polinomios cúbicos en t . Expandiendo el producto $Q(t) = T \cdot M \cdot G$ obtenemos²

$$Q(t) = \begin{pmatrix} x(t) & y(t) & z(t) \end{pmatrix} = \begin{pmatrix} t^3 & t^2 & t & 1 \end{pmatrix} \begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{pmatrix} \begin{pmatrix} G_1 \\ G_2 \\ G_3 \\ G_4 \end{pmatrix}$$

El resultado para el polinomio $x(t)$ queda

$$x(t) = (t^3 m_{11} + t^2 m_{21} + t m_{31} + m_{41}) g_{1x} + (t^3 m_{12} + t^2 m_{22} + t m_{32} + m_{42}) g_{2x} \\ + (t^3 m_{13} + t^2 m_{23} + t m_{33} + m_{43}) g_{3x} + (t^3 m_{14} + t^2 m_{24} + t m_{34} + m_{44}) g_{4x}$$

Los polinomios $y(t)$ y $z(t)$ quedan de forma similar.

Si observamos atentamente este polinomio, puede verse como la curva es una suma ponderada de los elementos de la matriz de geometría. Es decir, para cada valor de t

$$x(t) = b_1(t) g_{1x} + b_2(t) g_{2x} + b_3(t) g_{3x} + b_4(t) g_{4x}$$

donde los pesos que multiplican a cada elemento de G son los polinomios cúbicos en t . A estas funciones se les denomina **funciones blending** (to blend=doblar) ya que dependiendo de la forma de estas funciones se van interpolando los valores geométricos de la curva para cada valor de t . Estas funciones se obtienen haciendo $B = T \cdot M$.

Esta forma de representación puede parecer compleja, pero en realidad no es más que una generalización de la aproximación lineal usando segmentos rectos (polinomios de grado 1), para la cual sólo se necesitan dos condiciones geométricas, que son los dos puntos extremos. Así, cada segmento viene definido por la recta que pasa entre los puntos G_1 y G_2 :

$$\begin{cases} x(t) = g_{1x}(1-t) + g_{2x}(t) \\ y(t) = g_{1y}(1-t) + g_{2y}(t) \\ z(t) = g_{1z}(1-t) + g_{2z}(t) \end{cases}$$

¿Cómo se vería esta aproximación lineal usando la misma sintaxis que empleamos para la curva cúbica? Sea el segmento recto (en un caso bidimensional se verá más fácil) de la figura 6.

En coordenadas paramétricas parece claro que la ecuación será de la forma:

$$\left. \begin{aligned} x(t) &= x_1 + t(x_2 - x_1) \\ y(t) &= y_1 + t(y_2 - y_1) \end{aligned} \right\} \text{ siendo } 0 \leq t \leq 1$$

²Nótese que cada G_i es en realidad una terna de valores xyz . Es decir, aunque lo llamemos vector de geometría, a efectos de números reales, G sería una matriz 4×3 , donde G_x sería la primera columna, G_y la segunda y G_z la tercera.

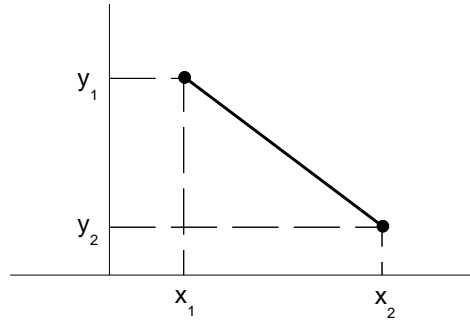


Figure 4

Esto puede ponerse también como

$$\begin{aligned}x(t) &= (1-t)x_1 + tx_2 \\ y(t) &= (1-t)y_1 + ty_2\end{aligned}$$

Por lo tanto el segmento quedará como sigue:

$$Q(t) = \begin{pmatrix} x(t) & y(t) \end{pmatrix} = B \cdot G = B \cdot \begin{pmatrix} x_1 & y_1 \\ x_2 & y_2 \end{pmatrix}$$

De aquí deducimos que el tamaño de la matriz B ha de ser 1×2 . El elemento b_{11} será igual al polinomio que multiplica tanto a x_1 como a y_1 , es decir, $b_{11}(t) = (1-t)$. De igual manera deducimos $b_{12}(t) = t$. Finalmente, ya podemos calcular la matriz base:

$$\begin{aligned}B &= T \cdot M = \begin{pmatrix} t & 1 \end{pmatrix} \begin{pmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{pmatrix} \\ M &= \begin{pmatrix} -1 & 1 \\ 1 & 0 \end{pmatrix}\end{aligned}$$

En resumen, un segmento recto vendrá dado por la ecuación

$$Q(t) = T \cdot M \cdot G = \begin{pmatrix} t & 1 \end{pmatrix} \begin{pmatrix} -1 & 1 \\ 1 & 0 \end{pmatrix} G$$

siendo G el vector de geometría formado por los dos puntos extremos.

Dejando a un lado este ejemplo 2D, pasemos a continuación a ver las matrices base para dos tipos distintos de curvas cúbicas: las curvas de Hermite, y las curvas de Bezier. Existen más tipos de curvas, pero aquí sólo veremos estos dos.

6.1.1 Curvas de Hermite

Un segmento curvo de Hermite viene determinado por sus dos puntos extremos, P_1 y P_4 , y dos vectores tangentes en dichos puntos, R_1 y R_4 . Así, el vector de geometría de Hermite viene definido por

$$G_H = \begin{pmatrix} P_1 \\ P_4 \\ R_1 \\ R_4 \end{pmatrix}$$

y las expresiones para el polinomio cúbico de cada componente (x por ejemplo) será de la forma:

$$x(t) = \begin{pmatrix} t^3 & t^2 & t & 1 \end{pmatrix} \cdot M_H \cdot G_{Hx}$$

Las condiciones para que la curva comience y finalice en cada extremo son:

$$\begin{aligned} x(0) &= P_{1x} = \begin{pmatrix} 0 & 0 & 0 & 1 \end{pmatrix} \cdot M_H \cdot G_{Hx} \\ x(1) &= P_{4x} = \begin{pmatrix} 1 & 1 & 1 & 1 \end{pmatrix} \cdot M_H \cdot G_{Hx} \end{aligned}$$

Para obtener las otras dos ecuaciones referentes a las tangentes habra que encontrar la derivada de x respecto de t :

$$\begin{aligned} x'(t) &= \begin{pmatrix} 3t^2 & 2t & 1 & 0 \end{pmatrix} \cdot M_H \cdot G_{Hx} \\ x'(0) &= R_{1x} = \begin{pmatrix} 0 & 0 & 1 & 0 \end{pmatrix} \cdot M_H \cdot G_{Hx} \\ x'(1) &= R_{4x} = \begin{pmatrix} 3 & 2 & 1 & 0 \end{pmatrix} \cdot M_H \cdot G_{Hx} \end{aligned}$$

Coloquemos ahora las cuatro ecuaciones en forma matricial:

$$\begin{pmatrix} P_1 \\ P_4 \\ R_1 \\ R_4 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{pmatrix} \cdot M_H \cdot G_{Hx}$$

Ahora fijémonos en una cosa curiosa. En realidad, el vector columna que aparece a la izquierda de la igualdad coincide con G_{Hx} ! Por lo tanto, el producto de la matriz 4x4 por M_H debe ser igual a la matriz identidad para que la igualdad se cumpla. De aquí deducimos M_H

$$M_H = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{pmatrix}^{-1} = \begin{pmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

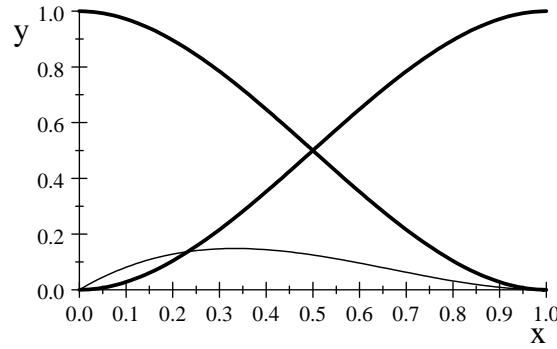
M_H es por supuesto única, y puede usarse en la expresión $x(t) = T \cdot M \cdot G_{Hx}$ para encontrar el polinomio $x(t)$ basado en el vector de geometría G_{Hx} . De igual forma, $y(t) = T \cdot M \cdot G_{Hy}$ y $z(t) = T \cdot M \cdot G_{Hz}$. Así, la expresión para el segmento completo puede escribirse como

$$Q(t) = \begin{pmatrix} x(t) & y(t) & z(t) \end{pmatrix} = T \cdot M_H \cdot G_H$$

Expandiendo el producto $T \cdot M_H$ obtenemos las **funciones blending de Hermite** B_H como los polinomios que pesan a cada elemento del vector de geometría:

$$\begin{aligned} Q(t) &= T \cdot M_H \cdot G_H = B_H \cdot G_H = \\ &= (2t^3 - 3t^2 + 1)P_1 + (-2t^3 + 3t^2)P_4 + (t^3 - 2t^2 + t)R_1 + (t^3 - t^2)R_4 \end{aligned}$$

En la figura siguiente pueden verse las cuatro funciones blending



Las líneas gruesas representan a los polinomios que multiplican a P_1 y a P_4 (el de P_1 empieza en 1) y las finas a R_1 y a R_4

Nótese que en $t = 0$, solamente la función que multiplica a P_1 es distinta de cero, por lo que solamente P_1 afecta a la curva en $t = 0$. A medida que t crece, los otros tres polinomios comienzan a tener influencia.

En la figura 7 se ven las cuatro funciones blending ya pesadas por las componentes y de un vector de geometría, su suma $y(t)$, y la curva paramétrica 2D final $Q(t)$. Nótese que el eje de representación de ésta última figura es xy y no ty .

En la figura 8 se ilustra la influencia que sobre la curva ejerce la variación de la magnitud de la tangente R_1 . Como comentábamos antes, la tangente R_1 viene a medir la derivada en t del inicio de la curva, es decir, como si fuese su velocidad inicial. A mayor magnitud, mayor velocidad tendrá la curva. Ya que el segmento se mantiene constante en el intervalo $(0, 1)$, es decir, t varía entre 0 y 1 (el tiempo es el mismo), al incrementar la velocidad se incrementa

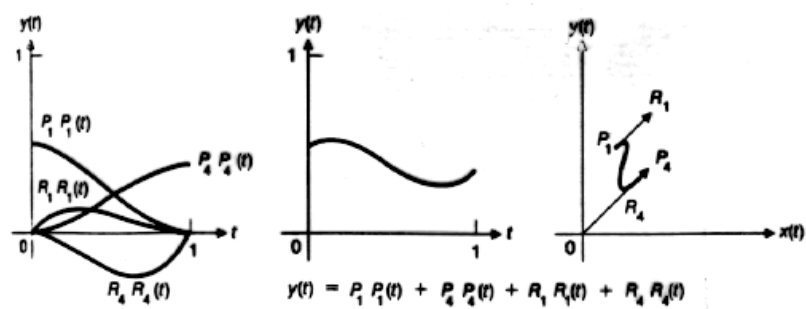


Figure 5

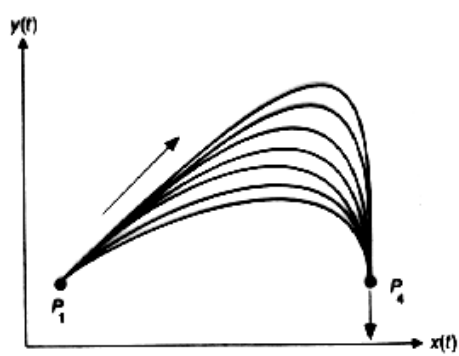


Figure 6

la longitud de la curva. El efecto visual es que la curva se estira en el sentido de la tangente.

En la figura 9 se aprecia el efecto de variar la dirección de la tangente. Lógicamente, la curva partirá en la misma dirección que tenga R_1 , e irá variando suavemente hasta llegar a la posición final llegando siempre con una dirección R_4 .

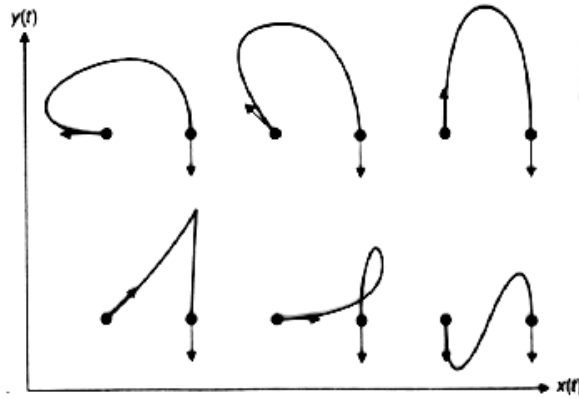


Figure 7

En los sistemas gráficos actuales, el software que permite al diseñador modelar las curvas es capaz de mostrarnos las tangentes de cada segmento para que el usuario interactivamente vaya variando dichos vectores, y en tiempo real ir viendo como va quedando la curva completa. Esta situación se muestra en la figura 10, donde puede verse una curva de Hermite compuesta por dos segmentos. Los puntos extremos pueden reposicionarse arrastrándolos con el ratón, y los vectores tangentes son alargados o encogidos también con el ratón arrastrando las puntas de los vectores.



Figure 8

También puede apreciarse como, para que la curva aparezca suave y sin esquinas, debe mantenerse la continuidad G_1 entre cada dos segmentos consecutivos. De esta manera, si forzamos al software para que nos mantenga

siempre esta restricción, las tangentes final e inicial de dos segmentos conectados siempre serán colineales, como puede verse. Matemáticamente, esta restricción implica que los dos vectores de geometría deben ser de la forma

$$\begin{pmatrix} P_1 \\ P_4 \\ R_1 \\ R_4 \end{pmatrix} \text{ y } \begin{pmatrix} P_4 \\ P_7 \\ kR_4 \\ R_7 \end{pmatrix}, \text{ siendo } k > 0$$

Es decir, debe haber un punto común entre ambos segmentos, y los vectores tangentes deben tener igual dirección. La condición más restrictiva de C_1 se consigue haciendo $k = 1$, coincidiendo entonces dirección y magnitud. En la figura 11 pueden verse dos segmentos con continuidad G_1 .

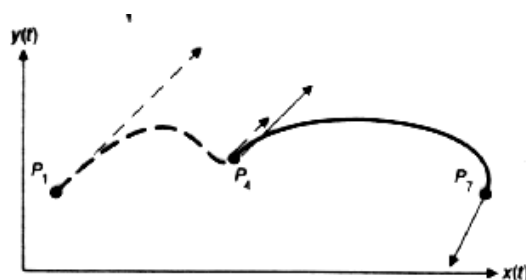


Figure 9

La forma de dibujar una curva de Hermite es relativamente sencillo. Sólo hay que ir evaluando $x(t)$, $y(t)$ y $z(t)$ para n sucesivos valores de t separados por un incremento constante δ . Es decir

$$\left. \begin{aligned} x(t) &= a_x t^3 + b_x t^2 + c_x t + d_x \\ y(t) &= a_y t^3 + b_y t^2 + c_y t + d_y \\ z(t) &= a_z t^3 + b_z t^2 + c_z t + d_z \end{aligned} \right\} \text{ donde } t = 0, \delta, 2\delta, \dots, 1$$

Otra forma más eficiente es usar la regla de Horne para polinomios

$$f(t) = at^3 + bt^2 + ct + d = ((at + b)t + c)t + d$$

lo cual reduce el cálculo a 9 multiplicaciones y 10 sumas para cada punto 3D. Una vez calculados todos los puntos, éstos pueden unirse mediante segmentos rectos en la pantalla.

Otro detalle muy importante a tener en cuenta sobre las curvas cúbicas es que, al ser combinaciones lineales de los cuatro elementos del vector de geometría, para transformar una curva simplemente transformaremos su vector de geometría, y usaremos este nuevo vector obtenido para mostrar la curva transformada. Esto significa que **las curvas son invariantes frente a rotaciones, traslaciones y escalados**. Esta estrategia es por tanto muchísimo más eficiente que generar la curva como una serie de segmentos rectos cortos y transformar cada segmento individual.

6.1.2 Curvas de Bezier

Un segmento curvo de Bezier viene determinado por sus dos puntos extremos, P_1 y P_4 , y por otros dos puntos, P_2 y P_3 , que indirectamente especifican los vectores tangentes en los extremos. Estos dos puntos no pertenecen a la curva, como puede verse en la figura 12. Los vectores tangentes vienen determinados

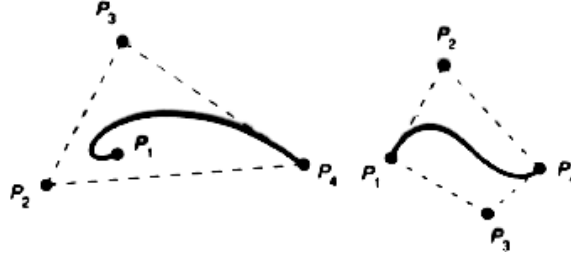


Figure 10

por los vectores P_1P_2 y P_3P_4 , y su relación con los vectores tangente R_1 y R_4 de Hermite es

$$\begin{aligned} R_1 &= Q'(0) = 3(P_2 - P_1) \\ R_4 &= Q'(1) = 3(P_4 - P_3) \end{aligned}$$

Por lo tanto, vemos que en realidad una curva Bezier es como una curva de Hermite, salvo que el vector de geometría es diferente, ya que en vez de consistir en dos puntos y dos vectores, consiste en cuatro puntos, esto es

$$G_B = \begin{pmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{pmatrix}$$

Para encontrar la relación existente entre el vector de geometría de Hermite, G_H , y el de Bezier, G_B , vamos a calcular la matriz M_{HB} que cumple $G_H = M_{HB} \cdot G_B$:

$$G_H = \begin{pmatrix} P_1 \\ P_4 \\ R_1 \\ R_4 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -3 & 3 & 0 & 0 \\ 0 & 0 & -3 & 3 \end{pmatrix} \begin{pmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{pmatrix} = M_{HB} \cdot G_B$$

Para encontrar la matriz base de Bezier usaremos la expresión de la curva de Hermite:

$$Q(t) = T \cdot M_H \cdot G_H = T \cdot M_H \cdot (M_{HB} \cdot G_B) = T \cdot (M_H \cdot M_{HB}) \cdot G_B = T \cdot M_B \cdot G_B$$

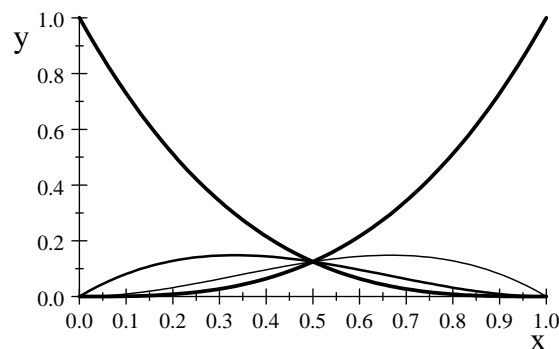
De aquí obtenemos la matriz de Bezier:

$$M_B = M_H \cdot M_{HB} = \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

La expresión para la curva es entonces

$$Q(t) = (1-t)^3 P_1 + 3t(1-t)^2 P_2 + 3t^2(1-t) P_3 + t^3 P_4$$

Los cuatro polinomios $B_B = T \cdot M_B$, que aparecen multiplicando a los elementos del vector de geometría en la ecuación anterior, son conocidos como **polinomios de Bernstein**, y aparecen en la figura siguiente:



Las líneas gruesas representan a los polinomios que multiplican a P_1 y a P_4 y las finas a P_2 y a P_3

Puede verse como en $t = 0$ solamente el polinomio que multiplica a P_1 es distinto de cero, por lo que la curva es igual a P_1 . Similarmente, en $t = 1$ sólo el polinomio de P_4 es diferente de cero, estando la curva en P_4 .

Si examinamos los cuatro polinomios de Bernstein nos damos cuenta de que su suma vale siempre 1 para todo t entre 0 y 1, y además, ninguno de los cuatro polinomios se hace nunca inferior a cero. De esta manera $Q(t)$ es en realidad una media ponderada de los cuatro puntos de control. Esto significa que cada segmento curvo está completamente contenido en el **convex hull**³ formado por los cuatro puntos de control.

Esta propiedad garantiza que cualquier punto del segmento curvo estará en el interior de dicho volumen, como resultado de la media ponderada. Esto

³Me niego a traducir esta expresión. Si la curva es en 2D significa "la envolvente convexa del polígono formado por los cuatro puntos de control". ¿Queda claro?. Si la curva es en 3D, entonces sustituir "polígono" por "poliedro".

puede verse claramente en un caso de dos puntos (un segmento recto) o tres puntos (un triángulo). Cualquier suma ponderada de los puntos (siendo uno la suma de los pesos y todos ellos positivos) cae siempre en el interior de la recta o del triángulo respectivamente (véase la figura anterior).

Esta propiedad es muy útil en la fase de recorte: en lugar de pasarle al algoritmo de recorte todos los segmentos de la curva para determinar su visibilidad, es mucho más eficiente aplicar primero el algoritmo a los convex hull de cada uno. Si el convex hull está completamente dentro del volumen de recorte, el segmento de la curva puede dibujarse enteramente. Si el convex hull está completamente fuera del volumen, el segmento será totalmente invisible. Sólo en aquellos casos en los que el convex hull intersekte al volumen de recorte será cuando pasemos la expresión del segmento al algoritmo del recorte. Obviamente es mucho más rápido calcular el recorte de un polígono o poliedro que el de una función cúbica paramétrica.

Al igual que ocurría con las curvas de Hermite, entre dos segmentos consecutivos debe garantizarse al menos la continuidad G_1 . Para ello debe cumplirse que

$$P_3 - P_4 = k(P_4 - P_5), \text{ siendo } k > 0$$

Es decir, los tres puntos P_3 , P_4 y P_5 deben ser distintos y colineales. Para cumplir C_1 habrá que hacer $k = 1$. En la figura 13 pueden verse dos segmentos de Bezier consecutivos con continuidad G_1 . Obsérvese como cada segmento

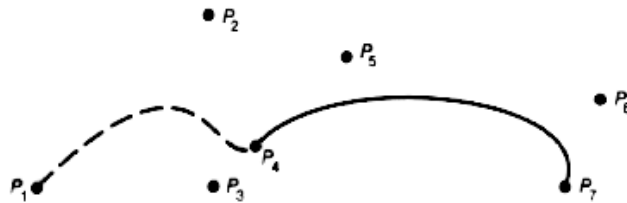


Figure 11

queda siempre en el interior de su convex hull.

6.1.3 Subdivisión de curvas

Supongamos que tenemos creada una serie conectada de segmentos curvos para intentar aproximar una forma geométrica que estamos diseñando. Una vez hechos podemos manipular los puntos de control para variar la forma de la curva, pero probablemente no podamos obtener toda la exactitud que quisiéramos. Probablemente, esto será debido a que no hemos creado los suficientes puntos de control para lograr el efecto deseado. Para solucionar este problema existen dos soluciones.

La primera se conoce como **elevación del grado**, es decir, incrementamos el grado del polinomio de 3 a 4 o más. Esta técnica se hace necesaria a veces, sobre todo si necesitamos garantizar continuidades de grado alto (C^2 ó C^3), pero no es muy conveniente, debido a que se necesitan más condiciones para el segmento (al tener más coeficientes) y al coste computacional adicional para evaluar la curva.

La segunda solución es mucho más práctica, y consiste en subdividir uno o más segmentos de la curva en dos nuevos segmentos. Por ejemplo, un segmento Bezier compuesto por cuatro puntos de control puede subdividirse en dos nuevos segmentos con un total de siete puntos de control (los nuevos segmentos tienen un punto en común). Estos dos nuevos segmentos coinciden perfectamente con el segmento original, hasta que movamos alguno de los puntos de control.

Veamos cómo sería el algoritmo para un segmento Bezier. Sea $Q(t)$ el segmento definido por los puntos P_1, P_2, P_3 y P_4 , y queremos encontrar una curva definida por los puntos L_1, L_2, L_3 y L_4 que coincida con la mitad izquierda de la curva original (left) y otra curva definida por los puntos R_1, R_2, R_3 y R_4 que coincida con la mitad derecha (right). Es decir, la curva izquierda ha de coincidir con $Q(t)$ en el intervalo $0 \leq t < 1/2$, y la curva derecha ha de coincidir en el intervalo $1/2 \leq t < 1$.

Para realizar la subdivisión vamos a usar una técnica de construcción geométrica desarrollada por **Casteljau** para evaluar una curva de Bezier para cualquier valor de t . En nuestro caso nos interesa usar $t = 1/2$, aunque más adelante generalizaremos. El algoritmo es el siguiente. Los segmentos rectos P_1P_2 , P_2P_3 y P_3P_4 los dividimos por la mitad y nos quedamos con el punto medio de cada uno, los cuales llamaremos L_2, H y R_3 respectivamente. A su vez, volvemos a tomar los nuevos segmentos L_2H y HR_3 y de nuevo nos quedamos con el punto medio para obtener dos nuevos puntos: L_3 y R_2 . Por último, este nuevo segmento L_3R_2 lo volvemos a dividir y nos quedamos con su punto medio $L_4 (= R_1)$. El resultado puede verse en la figura 14.

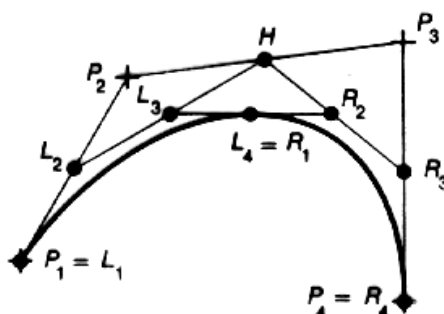


Figure 12

Haciendo $L_1 = P_1$, el segmento Bezier formado por los puntos de control L_1, L_2, L_3 y L_4 coincide perfectamente con la mitad izquierda de $Q(t)$. De forma similar, haciendo $R_4 = P_4$, el otro segmento formado por los puntos R_1, R_2, R_3 y R_4 coincide con la mitad derecha. De esta forma, y retornando al ejemplo inicial, si estamos modelando una forma y nos interesa tener más precisión en un segmento determinado, subdividiríamos la curva con esta técnica y obtendríamos el mismo dibujo inicialmente pero con dos segmentos en lugar de uno, es decir, con más puntos de control. Además esta técnica se presta a la recursividad. Si aún con la subdivisión no logramos la precisión requerida, los segmentos obtenidos pueden ser subdivididos nuevamente aplicando de nuevo la técnica.

Como siempre, lo interesante es reorganizar los cálculos en forma matricial. Nos interesaría calcular por un lado la matriz D_B^L que al mutiplicarla por el vector de geometría del segmento original nos diera el vector de geometría del nuevo segmento izquierdo, y otra matriz D_B^R que nos diera el vector de geometría del derecho. Revisemos la expresiones para los puntos:

$$\begin{aligned} L_1 &= P_1; \quad R_4 = P_4 \\ L_2 &= \frac{P_1 + P_2}{2}; \quad H = \frac{P_2 + P_3}{2}; \quad R_3 = \frac{P_3 + P_4}{2} \\ L_3 &= \frac{L_2 + H}{2} = \frac{P_1 + 2P_2 + P_3}{4}; \quad R_2 = \frac{H + R_3}{2} = \frac{P_2 + 2P_3 + P_4}{4} \\ L_4 &= R_1 = \frac{L_3 + R_2}{2} = \frac{P_1 + 3P_2 + 3P_3 + P_4}{8} \end{aligned}$$

con lo que ya podemos obtener la expresión matricial:

$$\begin{aligned} G_B^L &= D_B^L \cdot G_B = \frac{1}{8} \begin{pmatrix} 8 & 0 & 0 & 0 \\ 4 & 4 & 0 & 0 \\ 2 & 4 & 2 & 0 \\ 1 & 3 & 3 & 1 \end{pmatrix} \begin{pmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{pmatrix} \\ G_B^R &= D_B^R \cdot G_B = \frac{1}{8} \begin{pmatrix} 1 & 3 & 3 & 1 \\ 0 & 2 & 4 & 2 \\ 0 & 0 & 4 & 4 \\ 0 & 0 & 0 & 8 \end{pmatrix} \begin{pmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{pmatrix} \end{aligned}$$

Fíjense que cada uno de los nuevos puntos de control L_i y R_i son una suma ponderada de los puntos P_i , con los pesos todos positivos y cuya suma es siempre 1. Por lo tanto, todos estos nuevos puntos se encuentran en el interior del convex hull. Esto significa que dichos puntos se hallan más cerca de la curva $Q(t)$ que los puntos originales. Esta propiedad nos será muy útil en la siguiente sección.

Por otra parte, el hecho de dividir la curva en $t = 1/2$ suele ser la opción más conveniente en la mayoría de los casos, pero lo ideal sería que el propio usuario de forma interactiva indicara en qué punto de la curva hacer la

subdivisión, aprovechando la técnica de Casteljau. Esta técnica ya habíamos comentado que servía para evaluar la curva en un punto t cualquiera, no necesariamente $t = 1/2$. Si quisiéramos calcular cuál es el punto de la curva para un cierto $0 < t_1 < 1$, la técnica trabajaría de forma similar, salvo que en lugar de dividir siempre por la mitad, dividiríamos usando el ratio $t_1 : (1-t_1)$, es decir, si por ejemplo $t_1 = 1/3$ dividiríamos cada segmento dejando $1/3$ a la izquierda y $2/3$ a la derecha⁴. El punto final obtenido, L_4 , es el punto perteneciente a la curva, que es igual a $Q(t_1)$, y además, se sigue cumpliendo que el segmento formado por los puntos L_i coincide con el primer tercio de $Q(t)$, y el formado por los puntos R_i coincide con los dos últimos tercios de $Q(t)$.

6.1.4 Dibujo de curvas

Hay dos formas básicas para dibujar una curva paramétrica. La primera es por **evaluación iterativa** de $x(t)$, $y(t)$ y $z(t)$ para pequeños incrementos de la variable t , y dibujando líneas rectas entre cada par de valores. Este método ya lo contamos anteriormente, y vimos que el coste computacional necesario era de 9 multiplicaciones y 10 sumas por cada punto 3D que queramos calcular. Una vez calculados, estos puntos se unen mediante segmentos rectos.

Una observación importante es la siguiente: si el número de puntos evaluados, n , es relativamente pequeño, la curva puede verse en pantalla como una sucesión de segmentos rectos cortos, es decir, de forma similar a cuando usamos listas de aristas. Sin embargo, hay una diferencia muy grande con respecto a este otro formato, y es que seguimos manteniendo la información de la curva completa, y en cualquier momento, variando n , podemos tener la curva de forma más exacta, cosa que se hace imposible si lo que tenemos almacenado es un conjunto finito de aristas y puntos.

La segunda manera se conoce como **subdivisión recursiva**, ya que usa la técnica de Casteljau para ir subdividiendo cada segmento hasta que los nuevos segmentos sean lo suficientemente planos que puedan ser aproximados por una recta. El algoritmo general sería el siguiente:

```

Procedure DibujaCurvaRecursivo (curva,  $\varepsilon$ )
  if TestPlanaridad (curva,  $\varepsilon$ ) then
    DibujaRecta (curva)
  else
    SubdivideCurva (curva, Lcurva, Rcurva)
    DibujaCurvaRecursivo (Lcurva,  $\varepsilon$ )

```

⁴Atención: éste es un error típico. Para encontrar el punto C que divide un tramo AB con el ratio $1/3 : 2/3$ se calcula haciendo $C = A + \frac{B-A}{3}$, pero nunca de la forma $C = \frac{A+B}{3}$. ¿Está claro?

DibujaCurvaRecursivo (Rcurva, ε)
end

La representación Bezier es muy apropiada para ser dibujada con este método. La subdivisión es rápida, pues ya vimos en la sección anterior que se necesitaban 6 sumas y 6 divisiones. El test de planaridad para un segmento Bezier es bastante sencillo. Una posible medida de cuan plano es el segmento podría ser considerar la distancia de P_2 y P_3 al segmento P_1P_4 , como se ve en la figura 15. Si esta distancia es inferior al umbral ε podemos considerar que el segmento curvo es lo suficientemente plano. Si no es así, seguiríamos subdividiendo, y como comentábamos antes, con cada nueva subdivisión, los nuevos puntos de control obtenidos se encontrarán más cerca de la curva, por lo que los nuevos convex hulls formados van siendo cada vez más pequeños y estrechos.

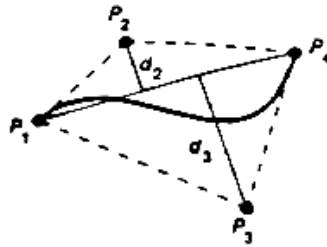


Figure 13

La gran ventaja de la subdivisión recursiva es que evita los cálculos innecesarios en aquellas zonas donde la curva sea más o menos plana. En estos lugares, apenas habrá que subdividir, mientras que en las zonas donde la curvatura sea más elevada se calcularán más puntos para poder visualizarla. En el método iterativo, al tener que elegir un incremento de t constante, deberíamos coger un incremento pequeño para poder mostrar las zonas de alta curvatura, pero este incremento es usado también en las zonas planas perdiendo mucho tiempo de cálculo innecesario. Por otro lado, la desventaja es que el cálculo para el test de planaridad es cómputo extra que no teníamos en el primer método.

6.2 SUPERFICIES BICÚBICAS PARAMÉTRICAS

Las superficies bicúbicas paramétricas son una generalización de las curvas cúbicas paramétricas. Recordemos que la expresión general de una curva era $Q(t) = T \cdot M \cdot G$, donde G era el vector de geometría. Este vector representaba los valores geométricos de la curva (por ejemplo, las coordenadas de los puntos extremos y de los vectores tangente) y por lo tanto era un vector constante independiente del parámetro t .

Para realizar el paso a superficies renombramos la variable t por s , es decir, $Q(s) = S \cdot M \cdot G$. Una vez hecho esto, el truco consiste en considerar que ahora el vector G ya no es constante, sino que varía en 3D a lo largo de una cierta trayectoria parametrizada por t . Es decir, si antes tenía un punto P_1 que indicaba el comienzo de la curva, ahora tendré $P_1(t)$, que me va a indicar el comienzo de la superficie. La expresión resultante es:

$$Q(s, t) = S \cdot M \cdot G(t) = S \cdot M \cdot \begin{pmatrix} G_1(t) \\ G_2(t) \\ G_3(t) \\ G_4(t) \end{pmatrix}$$

Ahora, para un cierto t_1 fijo, $Q(s, t_1)$ es una curva, ya que $G(t_1)$ es constante. Haciendo a continuación $t = t_2$, siendo $t_2 - t_1$ un valor muy pequeño, $Q(s, t_2)$ será una nueva curva ligeramente distinta de la anterior. Repitiendo este proceso para muchos valores distintos de t entre 0 y 1 obtendremos una familia completa de curvas muy pegadas entre sí. El conjunto de todas esas curvas definirá una superficie. Si $G(t)$ son curvas cúbicas, diremos que la superficie es bicúbica (dos variables, tercer grado) paramétrica.

La forma general de la curva vendrá separada en sus tres componentes:

$$\begin{cases} x(s, t) = S \cdot M \cdot G_x(t) \\ y(s, t) = S \cdot M \cdot G_y(t) \\ z(s, t) = S \cdot M \cdot G_z(t) \end{cases}$$

Al igual que ocurría con las curvas, en donde existían varios tipos dependiendo de la matriz base, aquí también existirán otros tantos tipos de superficies distintas. Veamos las de Hermite y las de Bezier.

6.2.1 Superficies de Hermite

Para derivar la expresión para las superficies de Hermite partiremos de la expresión de su curva. Voy a desarrollarlo sólo para la componente x . Las otras dos componentes se realizarían de la misma manera. Como hacíamos antes, renombramos la variable t por s y hagamos que el vector de geometría de Hermite G_H deje de ser constante y pase a depender del parámetro t :

$$x(s, t) = S \cdot M_H \cdot G_{Hx}(t) = S \cdot M_H \cdot \begin{pmatrix} P_{1x}(t) \\ P_{4x}(t) \\ R_{1x}(t) \\ R_{4x}(t) \end{pmatrix}$$

Las funciones $P_{1x}(t)$ y $P_{4x}(t)$ definen las componentes x de los puntos extremos de la curva de parámetro s . Similarmente, $R_{1x}(t)$ y $R_{4x}(t)$ son los

vectores tangentes a esos puntos. Para cualquier valor de t , existen dos puntos extremos fijos y dos vectores tangentes concretos. En la figura 16 se muestran las funciones $P_1(t)$ y $P_4(t)$, y las curvas cúbicas en s definidas para los valores de $t = 0.0, 0.2, 0.4, 0.6, 0.8$ y 1.0 .

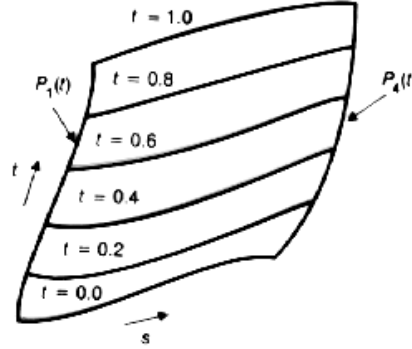


Figure 14

Desarrollemos ahora las expresiones para cada uno de los elementos de $G_{Hx}(t)$:

$$P_{1x}(t) = T \cdot M_H \cdot \begin{pmatrix} g_{11x} \\ g_{12x} \\ g_{13x} \\ g_{14x} \end{pmatrix}; \quad P_{4x}(t) = T \cdot M_H \cdot \begin{pmatrix} g_{21x} \\ g_{22x} \\ g_{23x} \\ g_{24x} \end{pmatrix}$$

$$R_{1x}(t) = T \cdot M_H \cdot \begin{pmatrix} g_{31x} \\ g_{32x} \\ g_{33x} \\ g_{34x} \end{pmatrix}; \quad R_{4x}(t) = T \cdot M_H \cdot \begin{pmatrix} g_{41x} \\ g_{42x} \\ g_{43x} \\ g_{44x} \end{pmatrix}$$

donde g_{1ix} y g_{2ix} son las componentes x de los elementos i -ésimo del vector de geometría para la curvas $P_1(t)$ y $P_4(t)$ respectivamente, y g_{3ix} y g_{4ix} son las componentes x de los elementos i -ésimo del vector de geometría para la curvas $R_1(t)$ y $R_4(t)$. Es decir, g_{23} por ejemplo será la componente x del vector tangente inicial para la curva $P_4(t)$.

Fíjense bien además en un detalle muy importante: para un cierto t_1 fijo, $P_i(t_1)$ representan puntos concretos, mientras que $R_i(t_1)$ representan vectores. Por lo tanto, evaluando para muchos valores de t , $P_i(t)$ puede verse como una sucesión de puntos pegados entre sí formando una curva, mientras que $R_i(t)$ sería como una familia de vectores que van variando suavemente partiendo de $R_i(0)$ hasta llegar a $R_i(1)$.

Reescribiendo las cuatro expresiones anteriores en una única ecuación

obtenemos:

$$\begin{pmatrix} P_{1x}(t) & P_{4x}(t) & R_{1x}(t) & R_{4x}(t) \end{pmatrix} = T \cdot M_H \cdot \mathbf{G}_{Hx}^T$$

siendo

$$\mathbf{G}_{Hx} = \begin{pmatrix} g_{11x} & g_{12x} & g_{13x} & g_{14x} \\ g_{21x} & g_{22x} & g_{23x} & g_{24x} \\ g_{31x} & g_{32x} & g_{33x} & g_{34x} \\ g_{41x} & g_{42x} & g_{43x} & g_{44x} \end{pmatrix}$$

Ahora, realizamos la traspuesta en ambos lados y nos queda:

$$\begin{pmatrix} P_{1x}(t) \\ P_{4x}(t) \\ R_{1x}(t) \\ R_{4x}(t) \end{pmatrix} = \begin{pmatrix} g_{11x} & g_{12x} & g_{13x} & g_{14x} \\ g_{21x} & g_{22x} & g_{23x} & g_{24x} \\ g_{31x} & g_{32x} & g_{33x} & g_{34x} \\ g_{41x} & g_{42x} & g_{43x} & g_{44x} \end{pmatrix} \cdot M_H^T \cdot T^T = \mathbf{G}_{Hx} \cdot M_H^T \cdot T^T$$

Por último sustituimos esta expresión en la que teníamos al comienzo para $x(s, t)$ ($y(s, t)$ y $z(s, t)$ se obtienen de forma similar):

$$\begin{cases} x(s, t) = S \cdot M_H \cdot \mathbf{G}_{Hx} \cdot M_H^T \cdot T^T \\ y(s, t) = S \cdot M_H \cdot \mathbf{G}_{Hy} \cdot M_H^T \cdot T^T \\ z(s, t) = S \cdot M_H \cdot \mathbf{G}_{Hz} \cdot M_H^T \cdot T^T \end{cases}$$

Las tres matrices 4x4 \mathbf{G}_{Hx} , \mathbf{G}_{Hy} y \mathbf{G}_{Hz} juegan el mismo papel en las superficies de Hermite que la matriz G_H para las curvas⁵. El significado de los 16 elementos de \mathbf{G}_H puede entenderse en función del desarrollo que hemos hecho. El elemento g_{11x} es en realidad $x(0, 0)$, ya que es el punto inicial para $P_{1x}(t)$, que a su vez es el punto inicial para $x(s, 0)$. Similarmente, g_{12x} es $x(0, 1)$ ya que es el punto final de $P_{1x}(t)$, que a su vez es el punto inicial para $x(s, 1)$. Por otro lado, g_{13x} es $\partial x / \partial t(0, 0)$ ya que representa el vector inicial para $P_{1x}(t)$, y g_{33x} es $\partial^2 x / \partial s \partial t(0, 0)$ ya que es el vector inicial de $R_{1x}(t)$, que a su vez representa la pendiente inicial de $x(s, 0)$.

Usando estas interpretaciones, podemos reescribir la matriz \mathbf{G}_{Hx} de esta manera:

$$\mathbf{G}_{Hx} = \begin{pmatrix} x(0, 0) & x(0, 1) & \frac{\partial}{\partial t}x(0, 0) & \frac{\partial}{\partial t}x(0, 1) \\ x(1, 0) & x(1, 1) & \frac{\partial}{\partial t}x(1, 0) & \frac{\partial}{\partial t}x(1, 1) \\ \frac{\partial}{\partial s}x(0, 0) & \frac{\partial}{\partial s}x(0, 1) & \frac{\partial^2}{\partial s \partial t}x(0, 0) & \frac{\partial^2}{\partial s \partial t}x(0, 1) \\ \frac{\partial}{\partial s}x(1, 0) & \frac{\partial}{\partial s}x(1, 1) & \frac{\partial^2}{\partial s \partial t}x(1, 0) & \frac{\partial^2}{\partial s \partial t}x(1, 1) \end{pmatrix}$$

⁵Cuando digo matrices de 4x4 me refiero a que cada elemento de la matriz es un punto 3D. En términos de números reales, la resolución de las matrices sería de 4x4x3 (48 números).

Dividamos la matriz en cuatro submatrices de 2×2 . La submatriz superior izquierda de \mathbf{G}_{Hx} contiene las coordenadas x de las cuatro esquinas de la superficie. Las submatrices superior derecha e inferior izquierda indican las componentes x de los vectores tangentes a lo largo de cada dirección paramétrica de la superficie. La submatriz inferior derecha muestra las derivadas parciales cruzadas en las cuatro esquinas. Estas cuatro derivadas son llamadas *twists*, ya que su valor mide, por así decirlo, el grado de ondulación de la esquina. En la figura 17 se indican los 16 elementos de la matriz en relación con la superficie que definen.

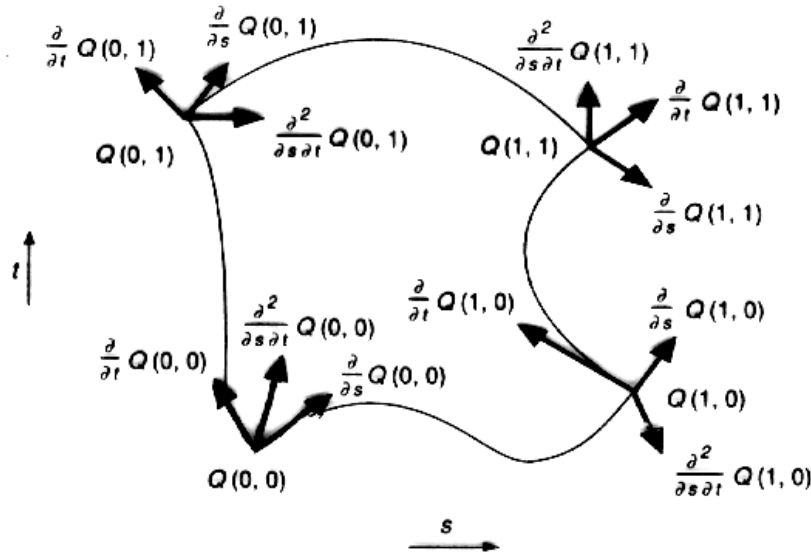


Figure 15

Dejemos a un lado la matriz \mathbf{G}_H y recordemos de nuevo la expresión de un segmento curvo de Hermite. Una curva de Hermite era en realidad una sucesión de segmentos conectados entre sí para formar la curva completa. La expresión $Q(t) = T \cdot M \cdot G$ para todo $t \in [0, 1]$ era en realidad la expresión de cada segmento, y para cada uno existía un vector G específico. Ahora retornemos a la superficie. A la expresión que acabamos de obtener $Q(s, t) = S \cdot M \cdot \mathbf{G} \cdot M^T \cdot T^T$ para todo $s, t \in [0, 1]$ la llamaremos **patch**, y llamaremos superficie de Hermite a una colección de patches conectados entre sí. Cada uno de estos patches tendrá su matriz \mathbf{G} específica.

Además, al igual que en las curvas había que garantizar la continuidad G^1 o C^1 entre segmentos consecutivos, de la misma forma hay que garantizarla entre los patches conectados. Aunque hay algunas diferencias. Antes, cada segmento mantenía la continuidad en los dos puntos extremos que compartía con sus dos segmentos vecinos. Ahora, cada patch debe mantener la continuidad en sus cuatro bordes que a su vez son compartidos por sus cuatro

patches vecinos.

Así, mantener la continuidad C^0 entre dos patches significa que ambos están pegados perfectamente por uno de sus bordes, es decir, el borde final de un patch y el borde inicial del vecino son idénticos. Esto además implica que los puntos de control para los dos patches que identifican el borde deben coincidir. Para mantener la continuidad C^1 es preciso que los puntos de control tanto de la curva que identifica el borde en ambos patches, como los que identifican los vectores tangentes perpendiculares a lo largo de ese borde sean idénticos.

Por ejemplo, supongamos que el patch A está conectado en $s = 1$ con el patch B en $s = 0$, como muestra la figura 18. Para garantizar la continuidad

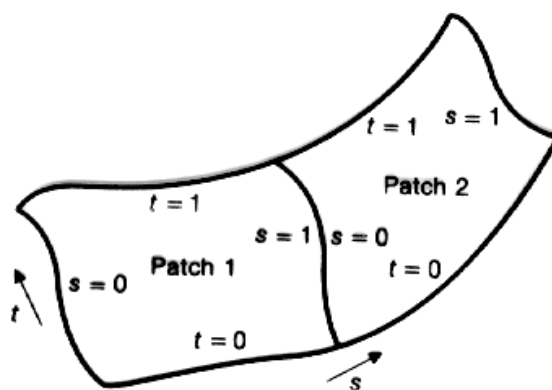


Figure 16

G^1 entre ambos sus matrices deben ser de esta forma:

$$A : \begin{pmatrix} - & - & - & - \\ g_{21} & g_{22} & g_{23} & g_{24} \\ - & - & - & - \\ g_{41} & g_{42} & g_{43} & g_{44} \end{pmatrix}; \quad B : \begin{pmatrix} g_{21} & g_{22} & g_{23} & g_{24} \\ - & - & - & - \\ kg_{41} & kg_{42} & kg_{43} & kg_{44} \\ - & - & - & - \end{pmatrix}$$

siendo $k > 0$. Por supuesto, con $k = 1$ garantizaríamos C^1 . En una superficie completa, cada patch deberá cumplir esta restricción con cada uno de sus cuatro vecinos.

6.2.2 Superficies de Bezier

La expresión para las superficies de Bezier se derivan exactamente de la misma manera que hicimos para las de Hermite. El resultado es:

$$\begin{cases} x(s, t) = S \cdot M_B \cdot \mathbf{G}_{Bx} \cdot M_B^T \cdot T^T \\ y(s, t) = S \cdot M_B \cdot \mathbf{G}_{By} \cdot M_B^T \cdot T^T \\ z(s, t) = S \cdot M_B \cdot \mathbf{G}_{Bz} \cdot M_B^T \cdot T^T \end{cases}$$

La matriz de geometría de Bezier, \mathbf{G}_B , a diferencia de la de Hermite, consiste de 16 puntos de control, como puede verse en la figura 19.

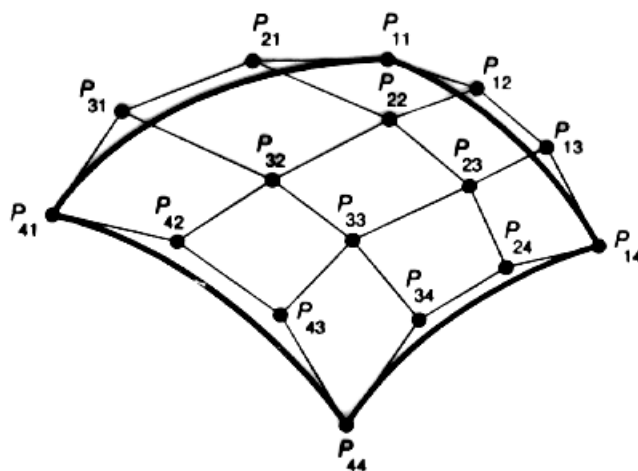


Figure 17

Las superficies de Bezier son muy usadas para el diseño interactivo por las mismas que eran utilizadas las curvas: tenemos un control preciso del patch moviendo los puntos de control, se sigue cumpliendo la propiedad de estar incluido dentro de su convex-hull, y la subdivisión en patches más pequeños se hace de forma sencilla. En la figura 20 puede apreciarse el efecto que obtenemos en el patch cuando estiramos uno de sus puntos de control.

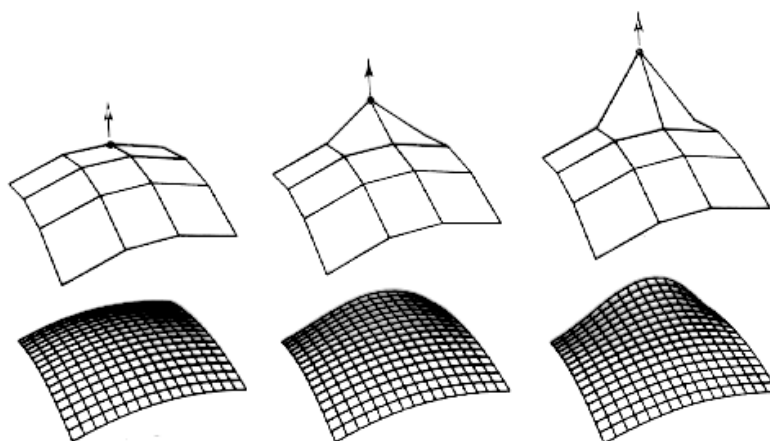


Figure 18

La continuidad C^0 entre dos patches vecinos se consigue haciendo coincidir los cuatro puntos de control que tienen en común. La continuidad G^1 ocurre

cuando los dos conjuntos de puntos de control a cada lado del borde común entre ambos son colineales con los puntos del borde. En el ejemplo mostrado en la figura 21 se ven dos patches con continuidad G^1 , y se aprecia cómo los segmentos $P_{13}P_{14}P_{15}$, $P_{23}P_{24}P_{25}$, $P_{33}P_{34}P_{35}$ y $P_{43}P_{44}P_{45}$ son todos rectos, y además tienen el mismo ratio k entre sus dos partes, es decir, la distancia entre P_{43} y P_{44} es k veces la que existe entre P_{44} y P_{45} . Obviamente, la continuidad C^1 ocurrirá para $k = 1$.

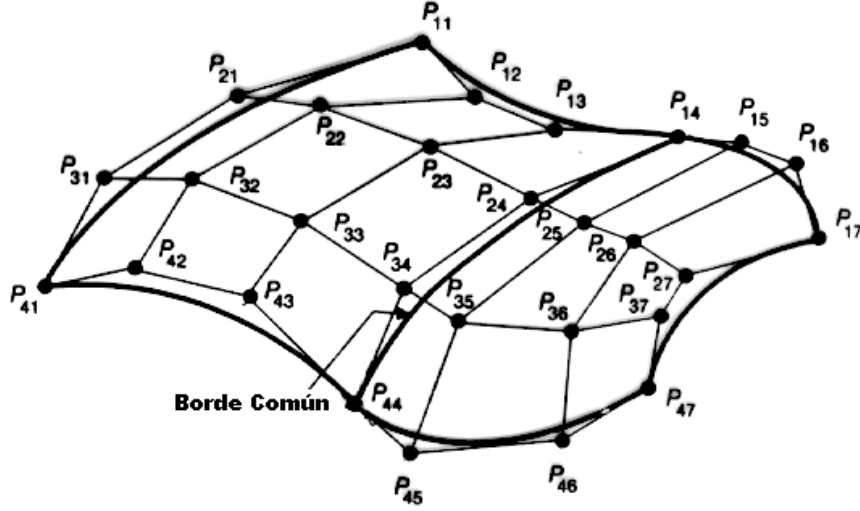


Figure 19

6.2.3 Cálculo de normales a una superficie

El cálculo de normales a una superficie, sea ésta del tipo que sea, es siempre necesario para calcular la iluminación del objeto, para encontrar los objetos más cercanos, para calcular reflexiones, etc. En el caso de las superficies bicúbicas paramétricas es bien sencillo calcularlas.

Comenzemos calculando el vector tangente en la dirección s de la superficie $Q(s, t)$:

$$\begin{aligned} \frac{\partial}{\partial s} Q(s, t) &= \frac{\partial}{\partial s} (S \cdot M \cdot \mathbf{G} \cdot M^T \cdot T^T) = \frac{\partial}{\partial s} (S) \cdot M \cdot \mathbf{G} \cdot M^T \cdot T^T = \\ &= \begin{pmatrix} 3s^2 & 2s & 1 & 0 \end{pmatrix} \cdot M \cdot \mathbf{G} \cdot M^T \cdot T^T \end{aligned}$$

De igual manera calculamos el vector tangente en la dirección t :

$$\begin{aligned} \frac{\partial}{\partial t} Q(s, t) &= \frac{\partial}{\partial t} (S \cdot M \cdot \mathbf{G} \cdot M^T \cdot T^T) = S \cdot M \cdot \mathbf{G} \cdot M^T \cdot \frac{\partial}{\partial t} T^T = \\ &= S \cdot M \cdot \mathbf{G} \cdot M^T \cdot \begin{pmatrix} 3t^2 & 2t & 1 & 0 \end{pmatrix}^T \end{aligned}$$

Recordemos que $Q(s, t)$ representa al punto (s, t) de la superficie Q , y que es por lo tanto una terna de valores reales (sus coordenadas x, y, z). Por lo tanto, estos vectores también nos salen como ternas de valores (las componentes x, y, z del vector). Llamemos entonces x_s, y_s, z_s a las tres componentes del vector tangente en s , y x_t, y_t, z_t a las del vector tangente en t . Ambos vectores tangente son paralelos a la superficie en el punto (s, t) , y por lo tanto, su producto vectorial es perpendicular a la superficie. La expresión para la normal en dicho punto es por tanto (figura 22):

$$N(s, t) = \frac{\partial}{\partial s}Q(s, t) \times \frac{\partial}{\partial t}Q(s, t) = \begin{pmatrix} y_s z_t - y_t z_s & z_s x_t - z_t x_s & x_s y_t - x_t y_s \end{pmatrix}$$

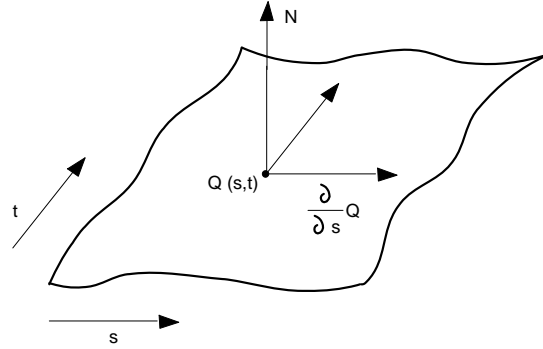


Figure 20

6.2.4 Dibujo de superficies

Al igual que ocurría para dibujar curvas, las superficies pueden dibujarse por evaluación iterativa de los polinomios bicúbicos o por subdivisión, la cual es esencialmente una evaluación adaptativa (dependiendo de la planaridad del patch) de los polinomios.

La evaluación iterativa consiste en dibujar varias curvas a lo largo de t para valores de s concretos, y luego dibujar varias curvas a lo largo s para valores de t concretos. Cada una de estas curvas es una cúbica por lo que aprovecha el algoritmo de la sección anterior. El resultado puede verse en la figura 23.

El algoritmo sería el siguiente:

Procedure DibujaPatch ()

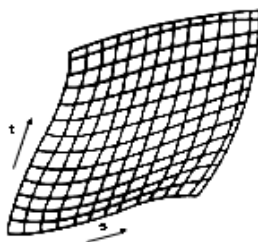


Figure 21

$\delta=1/n$ "incremento para dibujar cada curva"
 $\delta_s=1/(n_s-1)$ "incremento en s entre cada curva en t "
 $\delta_t=1/(n_t-1)$ "incremento en t entre cada curva en s "
 For $s=0$ to 1 by δ_s do "dibujamos una curva para cada s "
 For $t=0$ to $n-\delta$ by δ do "pintamos la curva $0 < t < 1$ "
 DibujaRecta ($t, t+\delta$)
 For $t=0$ to 1 by δ_t do "dibujamos una curva para cada t "
 For $s=0$ to $n-\delta$ by δ do "pintamos la curva $0 < s < 1$ "
 DibujaRecta ($s, s+\delta$)

Este algoritmo es aún más costoso que para las curvas, ya que la ecuación de la superficie debe ser evaluada $2n^2$ veces.

El segundo método es una extensión del procedimiento para las curvas que vimos en la sección anterior. Los patches son subdivididos en nuevos patches hasta que éstos sean similares a cuadriláteros cuasi planos, los cuales se dibujan como polígonos planos. Para evaluar el test de planaridad, una posible solución es hallar el plano que pasa por tres de las cuatro esquinas del patch y encontrar la distancia de cada uno de los 13 puntos de control restantes a dicho plano. Si la mayor de esas distancias está por debajo de un cierto umbral ε entonces el patch es lo suficientemente plano. El algoritmo se muestra a continuación:

Procedure DibujaPatchRecurso (patch, ε)
 If TestPlanaridad (patch, ε) then
 DibujaCuadrilátero (patch)
 else
 SubdividePatch (patch, patch11, patch12, patch21, patch22)
 DibujaPatchRecurso (patch11, ε)
 DibujaPatchRecurso (patch12, ε)
 DibujaPatchRecurso (patch21, ε)
 DibujaPatchRecurso (patch22, ε)
 end

La subdivisión del patch se realiza partiendo por la mitad a lo largo de un

parámetro, por ejemplo s , y entonces subdividir cada uno de los dos patches resultantes por la mitad a lo largo del otro parámetro t . Para ello partimos de los 16 puntos de control iniciales del patch, y aplicamos el método de subdivisión de curvas para cada conjunto de cuatro puntos de control en la dirección del parámetro s , con lo que obtenemos un conjunto de 7×4 puntos (ya que el método de subdivisión de curvas me pasaba de un segmento de 4 puntos de control a 2 nuevos segmentos de 4 puntos con uno en común = 7 puntos). A continuación paso a subdividir los conjuntos de cuatro puntos de control en la dirección t , obteniendo finalmente un conjunto de 7×7 puntos. Estos 49 puntos, cogidos en cuatro grupos de 16 (los puntos del medio son compartidos) forman los cuatro nuevos patches. Este proceso puede verse en la figura 24.

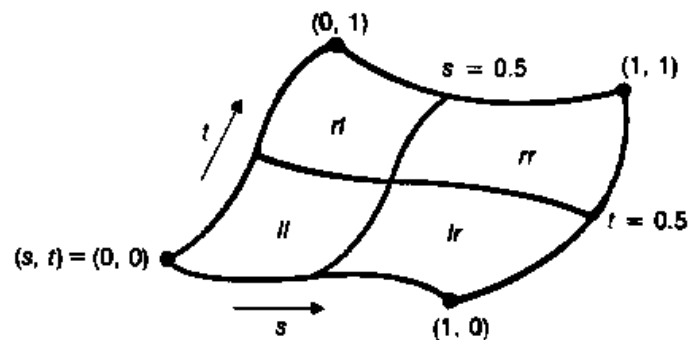


Figure 22

El procedimiento para dibujar el cuadrilátero debe pintarlo como un polígono sombreado. El problema es si el valor ε no es lo suficientemente pequeño, porque en ese caso los cuatro esquinas del patch no serían coplanares. La mejor forma de hacerlo para evitar problemas es calcular el punto promedio de las esquinas y usarlo para pintar cuatro triángulos, tal y como se muestra en la figura 25. La ventaja de los triángulos es que éstos sí que son siempre planos.

6.3 SUPERFICIES CUÁDRICAS

Las superficies cuádricas naturales (esferas, conos, cilindros, paraboloides, etc.) juegan un importante papel en la fabricación de partes mecánicas, así como en la descripción de superficies manufacturadas. Por ejemplo, los balones de fútbol y baloncesto son esféricos, los embudos son cónicos, las latas de cerveza son cilíndricas, y las antenas parabólicas no lo son. Las cuádricas son también importantes en la descripción de superficies más complejas. En general, siempre que dispongamos de la expresión analítica de la superficie, lo mejor es usar una superficie cuádrica.

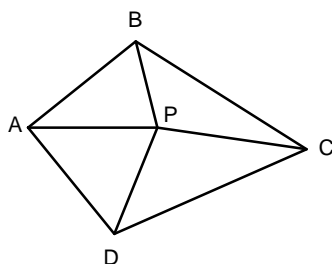


Figure 23

La familia de superficies cuádricas viene definida por las funciones de la forma

$$f(x, y, z) = ax^2 + by^2 + cz^2 + 2dxy + 2eyz + 2fzx + 2gx + 2hy + 2jz + k = 0$$

Por ejemplo, si $a = b = c = -k = 1$ y los restantes coeficientes son cero, tendríamos la esfera unidad definida en el origen; si $a = \dots = f = 0$, la ecuación representa un plano; y así muchos ejemplos más. Todos ellos diremos que pertenecen a la familia de las superficies cuádricas.

Aparte de ser usadas en la mayoría de los procesos de fabricación de piezas se usan también en aplicaciones especializadas como el modelado molecular. Entre sus ventajas cabe destacar lo fácil que es calcular la normal, testear de si un punto pertenece o no a la superficie, calcular la componente z para una x e y fija (muy importante para el cálculo de las superficies ocultas) y calcular intersecciones entre varias superficies.

Como siempre, vamos a intentar expresar la ecuación anterior en forma matricial:

$$P^T \cdot Q \cdot P = 0$$

$$\text{siendo } Q = \begin{pmatrix} a & d & f & g \\ d & b & e & h \\ f & e & c & j \\ g & h & j & k \end{pmatrix} \text{ y } P = \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

La ventaja de tenerla así expresada es que podemos aplicarle a Q cualquier transformación afín y obtener la expresión de la cuádrica transformada. Sea M la matriz de dicha transformación, la nueva cuádrica transformada nos quedaría:

$$\boxed{Q' = (M^{-1})^T \cdot Q \cdot M^{-1}}$$

La expresión para el vector normal a la superficie cuádrica $f(x, y, z) = 0$

en cualquier punto será

$$N(x, y, z) = \left(\frac{df}{dx} \quad \frac{df}{dy} \quad \frac{df}{dz} \right)$$

Esto es mucho más fácil que en el caso de las superficies bicúbicas, ya que aquí podemos calcular la expresión general para la normal, cosa que no podíamos hacer con éstas, sino sólomente calcular para puntos fijos.