

# Diseño asistido por ordenador

4ª Curso Ingeniería Informática

J.C. Torres  
Dpt. Lenguajes y Sistemas Informáticos  
ETS. Ingeniería Informática  
Universidad de Granada

## **TEMA 4. Diseño de curvas y superficies**

Este tema aborda un aspecto más concreto del modelado geométrico: la representación de curvas y superficies. Ambas comparten los mismos fundamentos matemáticos y las mismas técnicas.

## INDICE

### 4.1 Representación y visualización.

4.1.1 Representación.

4.1.2 Proceso de diseño.

4.1.3 Dibujo de curvas.

4.1.4 Visualización de superficies.

### 4.2 Métodos de diseño de curvas

4.2.1 Bézier.

4.2.2 B-Splines.

4.2.3 Curvas racionales.

### 4.3 Diseño de superficies.

4.3.1 Mallas de polígonos.

4.3.2 Generación de superficies a partir de curvas.

4.3.3 Superficies B-Splines.

## 4.1 Representación de curvas y superficies.

Las curvas y superficies poseen una representación matemática precisa, bien estudiada y suficientemente flexible. No obstante, y salvo raras excepciones, no es factible, en un sistema CAD, representar una curva, o superficie, mediante una ecuación, debido fundamentalmente a la necesidad de editar la representación. Comenzaremos estudiando su representación matemática, para justificar la representación computacional que se hace en sistemas CAD.

A nivel matemático, las curvas y superficies se pueden representar como ecuaciones de varias formas, atendiendo a como aparezcan las distintas variables involucradas:

- **Ecuaciones explícitas**, en las que aparece de forma explícita una de las variables en función de las otras dos. Estas expresiones tienen alguna de las formas siguientes:

en 3D             $z = f(x,y)$             (superficie)  
 ó                 $y = f_1(x), z = f_2(x)$             (curva en el espacio)

en 2D             $y = f(x)$                                 (curva en el plano)

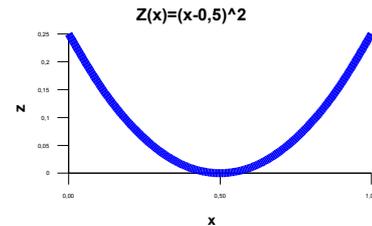


Figura 4.1 Curva explícita

Estas expresiones se suelen utilizar para curvas (fig. 4.1) y superficies univaluadas (fig. 4.2), como las que se suelen obtener como resultado de procesos de medida experimental (p.e. presión en puntos de la superficie de la tierra, o elevación de un camino). La evaluación de la superficie es muy fácil. Sin embargo es difícil utilizarlas para casos generales, dado que las superficies que nos encontramos en la realidad, y que queremos modelar no serán, normalmente, univaluadas..

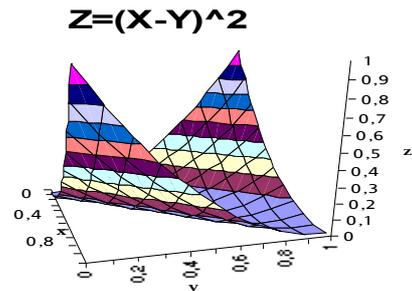


Figura 4.2 Superficie explícita

- **Ecuaciones implícitas**, expresadas como una ecuación de las variables igualada a cero. En ellas no aparece ninguna variable despejada, por lo que su evaluación puede ser compleja. La expresión tendrá una de las formas siguientes:

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

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

según que el elemento se defina en el espacio o en el plano, e independientemente de la dimensión de este. Un caso notable de curvas y superficies que se pueden expresar usando ecuaciones implícitas son las cuádricas. Por ejemplo un círculo de radio unidad se puede expresar como

$$y^2 + x^2 - 1 = 0$$

Las ecuaciones implícitas poseen la ventaja de ser *orientables*, es decir, es factible determinar hacia que lado de la superficie, o curva, se encuentra un punto substituyendo sus coordenadas

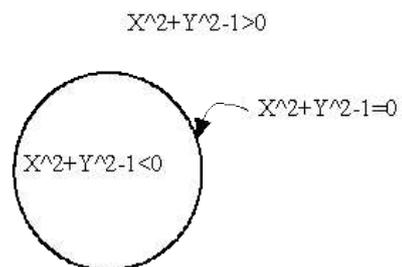


Figura 4.3 Curva implícita

en la ecuación. Por ejemplo, el punto (0,0) está a la izquierda del círculo, el punto (1,0) está sobre el círculo, y el (2,2) está a la derecha.

Por contra, las ecuaciones implícitas pueden ser difíciles de evaluar. A partir de una expresión explícita se puede obtener una implícita. Obviamente, el recíproco no es siempre cierto.

- **Ecuaciones paramétricas.** La curva o superficie se describe en base a un conjunto de parámetros que la recorren, como un conjunto de ecuaciones que permiten obtener cada una de las coordenadas a medida que el parámetro evoluciona sobre el elemento. Una curva se expresa usando un parámetro,  $u$ , que toma valores en un intervalo predeterminado, y dos o tres ecuaciones (según que se defina en el plano o el espacio). Así, una curva en el plano se define mediante el par de ecuaciones:

$$\begin{aligned}x &= f_1(u) \\ y &= f_2(u) \quad u \in [u_1, u_2]\end{aligned}$$

El círculo anterior se puede expresar mediante las ecuaciones paramétricas:

$$\begin{aligned}x &= \cos(2\pi u) \\ y &= \text{sen}(2\pi u) \quad u \in [0, 1]\end{aligned}$$

Una superficie se describe en base a dos parámetros,  $(u, v)$ , que describen dos direcciones ortogonales sobre ella, y a tres ecuaciones que expresan las coordenadas en base a estos parámetros:

$$\begin{aligned}x &= f_1(u, v), \\ y &= f_2(u, v), \\ z &= f_3(u, v) \quad u \in [u_1, u_2], v \in [v_1, v_2]\end{aligned}$$

Modificando los valores de los parámetros se recorre la superficie, obteniéndose las coordenadas de los puntos que están sobre ella.

Las expresiones paramétricas presentan la ventaja de ser flexibles, aunque no son orientables. Por otra parte, cualquier expresión explícita se puede poner, trivialmente, en forma paramétrica.

En aplicaciones de diseño se suelen utilizar expresiones *paramétricas* por su generalidad. No obstante, no es suficiente con la utilización de una expresión paramétrica. La necesidad de *editar* la curva, o superficie, obligaría a modificar, tanto los intervalos de variación de los parámetros como las funciones que describen las coordenadas ( $f_1$ ,  $f_2$  y  $f_3$ ). Esto último no es factible, en un caso general, a menos que se fije la forma de estas funciones, haciendo que su modificación implique el cambio de un número reducido de parámetros cuyo efecto en la forma de la curva sea predecible. Una forma de conseguir esto es utilizar funciones polinómicas:

$$X(u) = X_0 + X_1 \cdot u + X_2 \cdot u^2 + \dots$$

El conjunto de curvas que se pueden generar depende del grado de los polinomios usados. En cualquier caso, la especificación de una curva se hace con una *secuencia* finita de parámetros, tal como pretendíamos. No obstante, tal como decíamos, eso no es suficiente para el diseño de curvas, necesitamos, además, que los parámetros que introduzca el usuario sean relevantes para él, es decir, que el usuario pueda intuir el valor de los parámetros a partir de la forma de la curva. La relación entre la forma de la curva y los coeficientes del polinomio.

Vamos a ver como establecer la relación entre parámetros relevantes para el usuario y los coeficientes del polinomio con un ejemplo. Supongamos que utilizamos polinomios de grado uno, esto es:

$$X(u) = X_0 + X_1 \cdot u$$

$$Y(u) = Y_0 + Y_1 \cdot u \quad u \in [0,1]$$

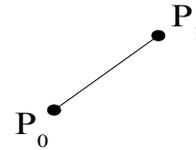


Figura 4. 4 Línea generada por polinomios de grado 1.

La curva generada será la recta que pasa por los puntos (X<sub>0</sub>,Y<sub>0</sub>) y (X<sub>0</sub>+X<sub>1</sub>,Y<sub>0</sub>+Y<sub>1</sub>). Si consideramos como intervalo de variación del parámetro el [0,1], la curva generada será la línea delimitada por estos puntos (ver fig. 4.4). Parece natural que el usuario introduzca los puntos extremos de la línea para especificar está, y que los valores de los parámetros se obtengan de estos. El cálculo de los coeficientes puede realizarse como sigue:

$$(X_0, Y_0) = \mathbf{P}_0$$

$$(X_0+X_1, Y_0+Y_1) = \mathbf{P}_1 \quad \Rightarrow \quad (X_1, Y_1) = \mathbf{P}_1 - \mathbf{P}_0$$

donde **P<sub>1</sub>** y **P<sub>0</sub>** son los puntos extremos (notaremos los puntos y vectores en negrita, esto es: **P(u)** = (P.x(u),P.y(u),P.z(u))). Generalizando, un método satisfactorio para expresar curvas y superficies es definir las a partir de un conjunto finito de puntos de control. Esto equivale a poner las expresiones anteriores en la forma:

$$X(u) = \sum_{i=1}^n x_i \cdot F_i(u)$$

donde x<sub>i</sub> es la coordenada x del i-ésimo punto de control, y las F<sub>i</sub> son funciones de forma, que expresan la dependencia con el parámetro. En el ejemplo anterior,

$$X(u) = X_0 + X_1 \cdot u = \mathbf{P}_0.x + (\mathbf{P}_1.x - \mathbf{P}_0.x) \cdot u = \mathbf{P}_0.x \cdot (1-u) + \mathbf{P}_1.x \cdot u$$

luego, las funciones de forma son:

$$F_0(u) = 1-u$$

$$F_1(u) = u$$

### 4.1.1 Proceso de diseño

En el ciclo de diseño de una curva, o superficie, el usuario solo trabaja con los puntos de control, que son en sí la representación de la curva, o superficie, en el modelo. Por tanto, en el proceso de diseño el usuario seleccionará y modificará los puntos de control, hasta que se obtenga la curva o superficie deseada [Brunet87]. El proceso puede comenzar con valores predeterminados asignados a los puntos de control.

En este proceso, el dibujo de la curva debe realizarse repetidas veces, al menos cada vez que el usuario mueve un punto, e idealmente mientras éste mueve los puntos con el dispositivo de entrada (durante el ciclo de realimentación). Por este motivo, cuando la máquina no es suficientemente potente, la visualización durante la edición se simplifica, aproximando la superficie como una malla de curvas.

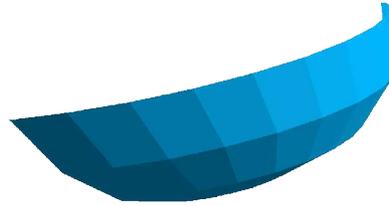


Figura 4.5 Dibujo de una curva

### 4.1.2 Dibujo de curvas.

Es necesario visualizar los elementos generados. El método más simple para visualizar una curva es el basado en la aproximación del segmento por una poligonal. Para ello basta con evaluar la curva en una secuencia de valores del parámetros, dibujando la poligonal que une los puntos así obtenidos (figura 4.5). Si el conjunto de puntos generados es suficientemente grande, no se apreciará la aproximación.

Debe tenerse en cuenta que el que los puntos en coordenadas paramétricas estén equiespaciados no implican que los estén los puntos evaluados en coordenadas reales. Existen métodos adaptativos, basados en la elección de los vértices de la poligonal en función de la curvatura o de la distancia al último punto. Obsérvese que, en cualquier caso, lo que se aproxima es la imagen y no la representación de la curva en sí.



**Figura 4.6** Dibujo de una superficie como una malla de triángulos, con 10x5 parches y sombreado constante por parche.

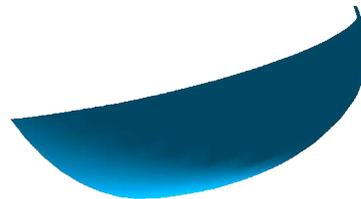
### 4.1.3 Visualización de superficies

El método descrito previamente para dibujar una curva nos permitirá aproximar una superficie paramétrica por una malla de triángulos. Para ello evaluaremos la superficie en un subconjunto de puntos del espacio paramétrico distribuidos en una rejilla. Por cada cuatro puntos vecinos formaremos dos triángulos (ver figura 4.6). Cada trozo de superficie que aproximamos mediante dos triángulos se denomina parche. La calidad de la imagen depende del número de parches, y también de como se dibujen estos.

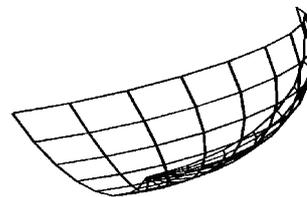
La figura 4.7 muestra la misma superficie dibujada con un mayor número de parches y suavizando el color dentro de cada parches, para dar una sensación de suavidad.

Otra alternativa es basarse en algoritmos para realizar el test de intersección rayo-superficie, lo que permite visualizar la superficie usando trazado de rayos.

Tal como indicamos anteriormente, una superficie se puede también visualizar dibujando una familia de curvas contenidas dentro de ésta. Las curvas se obtienen normalmente fijando el valor de uno de los parámetros (denominándose en este caso isoparamétricas). La figura 4.8 muestra la misma superficie anterior dibujada como un conjunto de curvas isoparamétricas.



**Figura 4.7** Dibujo de una superficie como una malla de triángulos, con 30x12 parches y sombreado de Gouraud



**Figura 4.8** Dibujo de una superficie como un conjunto de curvas isoparamétricas.

## 4.2 Métodos de diseño

La mayor parte de los métodos de diseño de curvas y superficies se basan en la utilización de puntos de control a partir de los cuales se define ésta como un promedio de los puntos de control:

$$P(u) = \sum_{i=1}^n P_i \cdot B_i(u)$$

donde  $P_i$  son los puntos de control y  $B_i(u)$  son funciones de forma, dadas en forma paramétrica.

Existen diversos métodos de diseño de curvas, con diferentes características. Entre estas, cabe destacar el *carácter* del método (que puede ser *local* o *global*) y el *comportamiento* respecto a los puntos de control (*interpolante* o no). Un método tiene carácter local cuando la modificación de un punto de control afecta solamente a la forma de la curva, o superficie, en las proximidades del punto de control. Por el contrario, en un método global, la modificación de un punto de control afecta a toda la curva, o superficie. Es más fácil editar una curva o superficie utilizando un método local, ya que permite ajustar la forma de la curva trozo a trozo.

Decimos que un método interpola a los puntos de control cuando el elemento generado (curva o superficie) pasa por ellos.

Habitualmente es necesario usar polinomios continuos a trozos, para conseguir métodos de diseño local, lo que influye en el *grado de continuidad* de la curva. El grado de continuidad indica el número de veces que se puede derivar su ecuación obteniendo una función continua. Geométricamente esto está relacionado con la continuidad del elemento. Una curva con continuidad  $C^0$  es continua, pero su pendiente no. Si la continuidad es  $C^1$  la curva y su pendiente son continuas.

La continuidad depende de la forma en que está parametrizada la curva. Para que la curva sea continua en un punto, sus vectores tangentes a izquierda y derecha deben coincidir. No obstante, la magnitud del vector tangente no influye en la apreciación que podemos hacer de continuidad en el punto. Por este motivo, se suele hablar de continuidad geométrica, notada con  $G$ , en lugar de continuidad matemática. Una curva tiene continuidad geométrica en un punto si las tangentes a izquierda y derecha tienen la misma dirección, independientemente de la magnitud del vector tangente. Por este motivo, la continuidad matemática implica continuidad geométrica (salvo el caso especial en que el vector tangente es nulo). Normalmente se requiere como mínimo continuidad  $G^1$  y con frecuencia  $G^2$ .

También se debe tener en cuenta la influencia del grado del polinomio de las funciones de forma. Cuando mayor sea el grado del polinomio más restricciones podremos aplicar a la curva (ya que hay más coeficientes), pero también será mayor oscilación de ésta, y más costoso su cálculo.



Figura 4.9 Continuidad

En este tema veremos las curvas de Bézier, los splines y los B-Splines. Las curvas de Bézier destacan por su simplicidad de formulación, tanto en la forma de Bézier usando polinomios de Bernstein, como en el planteamiento de De Casteljaou. Los splines tienen una utilidad muy limitada en diseño, aunque son una buena opción cuando se quiere un método que interpole los puntos de control.

Las curvas B-Splines a diferencia de las curvas anteriores, permiten controlar el grado de continuidad. Por otro lado, es posible obtener el método de Bézier como caso particular de B-Spline.

La siguiente tabla resume las características de estos métodos:

<b>Método</b>	<b>Grado del polinomio</b>	<b>Interpola</b>	<b>Carácter (nº de puntos que afectan un tramo)</b>	<b>Continuidad</b>
Lineal	1	Si	Local (2)	C <sup>0</sup>
Bézier	n-1	No (Solo extremos)	Global (n)	C <sup>∞</sup>
Spline local	3	Si	Local (4)	C <sup>1</sup>
Spline Gobal	3	Si	Global	G <sup>2</sup>
B-Spline	Ajustable	No (se puede forzar)	Local (Ajustable)	Ajustable
B-Spline (cúbico)	3	No	Local (4)	G <sup>2</sup>

**Tabla 4.1** Característica de algunos métodos de diseño de curvas. n es el número de puntos de control. La última fila muestra un ejemplo de ajuste para un B-Spline cúbico.

Independientemente del método usado, la descripción de la curva se puede realizar usando notación matricial, separando la dependencia con el parámetro de la dependencia con los puntos de control. Así, para el caso de interpolación lineal, podemos expresar la ecuación

$$P(u) = \sum_{i=1}^n P_i \cdot B_i(u)$$

con

$$B_0(u) = 1-u$$

$$B_1(u) = u$$

para n=2, en la forma:

$$P(u) = P_0 \cdot B_0(u) + P_1 \cdot B_1(u)$$

y representando los puntos de control en un vector columna:

$$P(u) = \begin{pmatrix} B_0(u) \\ B_1(u) \end{pmatrix} \cdot \begin{pmatrix} P_0 \\ P_1 \end{pmatrix}$$

y separando en la primera matriz la dependencia en u como un vector fila:

$$P(u) = (1, u) \cdot \begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix} \cdot \begin{pmatrix} P_0 \\ P_1 \end{pmatrix}$$

En general, la curva se puede expresar como una ecuación matricial en la que interviene una matriz de potencias del parámetro, una matriz de coeficientes (cuyos valores dependerán del método de diseño) y una matriz de puntos de control.

### 4.2.1 Curvas de Bézier

El método de Bézier utiliza como funciones base, o de forma, los polinomios de Bernstein, cuya expresión es

$$B_i^n(u) = \binom{n}{i} (1-u)^{n-i} \cdot u^i$$

donde  $\binom{n}{i} = \frac{n!}{i! \cdot (n-i)!}$  y  $n$  es el número de puntos de control. El parámetro está definido en  $[0,1]$ , y el número de puntos de control es  $n+1$ . La ecuación de la curva es:

$$P(u) = \sum_{i=0}^n P_i B_i^n(u) \quad u \in [0,1]$$

La figura 4.10 muestra las funciones de forma para  $n=2$ , y la figura 4.11 para  $n=3$ . Podemos observar que todas las funciones de forma son nulas en los extremos, salvo la primera y la última que valen uno para  $u=0$  o  $u=1$ . Esto implica que en los puntos extremos, es decir, al principio y final de la curva solo influya uno de los puntos de control, y que en los extremos la curva pase por el primer o último punto de control.

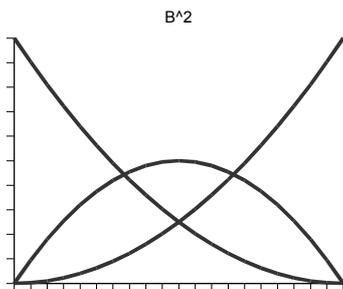


Figura 4.10 Funciones de forma de grado 2

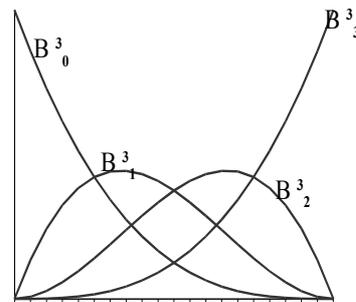


Figura 4.11 Funciones de forma de grado 3

Para ningún otro valor de  $u$ , hay funciones de forma con valor igual a uno. Esto implica que la curva no pase por ningún otro punto de control.

Para cualquier valor de  $u$  la suma de las funciones de forma es igual a uno. Es decir, podemos ver las funciones de forma como pesos en una media ponderada de los puntos de control. Como consecuencia de esto, la curva estará siempre dentro de la envolvente convexa de los puntos de control (mínimo polígono convexo que contiene a los puntos de control).

Podemos observar también que cada una de las funciones de forma presenta un máximo, que se presenta en valores de  $u$  crecientes con el índice de la función. Esto hace que la influencia de cada punto de control sea mayor para un valor de  $u$  que crece con el índice del punto, por lo que la curva sigue la forma de la poligonal, pero de forma más suave.

Es fácil comprobar (calculando las pendientes en los extremos) que la dirección de salida en el primer punto de control coincide con la de la recta que pasa por el segundo punto de control, e igualmente la dirección de llegada al último punto es la de la recta que lo une al penúltimo.

En resumen, la curva tiene las siguientes propiedades:

- La curva es **interior a la frontera convexa**.

- Pasa por  $P_0$  ( $u=0$ ) y por  $P_n$  ( $u=1$ ), pero no por los intermedios.
- La dirección de salida de estos puntos está determinada por  $P_1$  y  $P_{n-1}$ .
- La curva sigue la forma de la poligonal.
- La continuidad es  $C^\infty$ .
- El grado del polinomio es el número de puntos de control menos uno.
- La modificación de un punto de control afecta a toda la curva.

Se pueden conectar curvas con continuidad prefijada, por ejemplo  $C^1$  o  $C^2$ .

### 4.2.2 B-Splines

Son curvas polinómicas que se construyen conectando polinomios de un determinado grado. Los polinomios se obtienen por combinación de  $n$  puntos de control, utilizando una base de funciones de forma, cuyo orden es independiente del número de puntos de control

$$P(u) = \sum_{i=0}^n P_i \cdot N_{i,k}(u) \quad u_{min} \leq u \leq u_{max} \quad 2 \leq k \leq n+1$$

donde  $k$  es el **orden** curva.

Las funciones de forma se definen de forma recursiva, a partir de las funciones de grado inferior.

$$N_{i,1}(u) = \begin{cases} 1 & \text{si } t_i \leq u < t_{i+1} \\ 0 & \text{en otro caso} \end{cases}$$

$$N_{i,k}(u) = \frac{u-t_i}{t_{i+k-1}-t_i} \cdot N_{i,k-1}(u) + \frac{t_{i+k}-u}{t_{i+k}-t_{i+1}} \cdot N_{i+1,k-1}(u)$$

Al aumentar  $k$  aumenta el grado de los polinómios. El **grado** de los polinómios, y por tanto de la curva, de orden  $k$  es  $k-1$ . Para  $k=2$  los polinómios son de grado uno.

La secuencia de valores de  $t$ , que se denominan **nodos**, define una partición del intervalo de variación del parámetro, permitiendo ajustar la zona de influencia de cada punto de control. Al conjunto de nodos se le llama **vector de nodos**, y debe satisfacer las siguientes condiciones:

$$t_j \leq t_{j+1} \quad 0 \leq j \leq n+k$$

$$u_{min} = t_{k-1}$$

$$u_{max} = t_{n+1}$$

El **vector de nodos**  $[t_i]$  relacionan la variable paramétrica  $u$  con los puntos de control  $P_i$ , delimitando la zona de influencia de cada punto de control. Por este motivo está formado por una secuencia de valores reales no decrecientes. Además, el vector de nodos determina el rango de variación del parámetro.

Las funciones base son polinómicas a trozos, que son no nulos tan solo en un número pequeño de intervalos (ver fig. 4.12). Por tanto, cada punto de control afecta a la curva sobre un rango de valores del parámetro donde la función de forma es no nula. Otra consecuencia de esto es que la curva no sea infinitamente diferenciable,

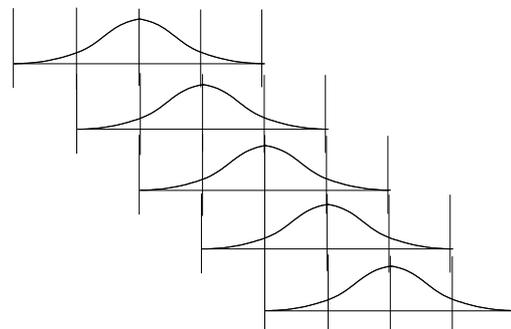


Figura 4.12 Funciones base para B-Splines cúbicos

como las Bezier. En los nodos, se utiliza una ecuación diferente a cada lado. En estos puntos la curva tendrá un nivel de continuidad que dependerá de su orden y de la distribución de los valores de los nodos. Si todos los valores del vector de nodos son distintos, la curva será  $k-1$  veces diferenciable. Es decir tendrá continuidad  $C^{k-2}$ . Por ejemplo, un Bspline de orden 4 tiene continuidad  $C^2$ . Si  $p$  nodos (del vector nodos) coinciden,  $u_j = \dots = u_{j+p-1}$ , la curva B-spline tiene continuidad  $C^{k-2-(p-1)}$  en  $u_j$ , (es decir  $C^{k-1-p}$ ). Siempre se debe cumplir que  $p \leq n+1$ .

Formalmente se pueden definir a partir de un conjunto de propiedades y de una relación de recurrencia sobre el grado. Las funciones de forma de los B-splines  $N_{i,k}$  están definidos por las siguientes propiedades

- Partición de la unidad:  $\sum_i N_{i,k}(t) = 1$
- Positividad:  $N_{i,k}(t) \geq 0$
- Soporte local:  $N_{i,k}(t) = 0$  si  $u \notin [t_i, t_{i+k+1}]$
- Continuidad:  $N_{i,k}$  es  $(k-1)$  veces diferenciable

En Bezier, el número de puntos determina el grado de la curva. En B-splines, el grado se controla por un parámetro especial ( $k$ ), que es independiente del número de puntos de control. [Mort85], aumentando la flexibilidad del método.

La figura 4.13 muestra el proceso de cálculo, siguiendo la definición recursiva de las funciones de forma, de una curva de orden 3 y tres puntos de control.

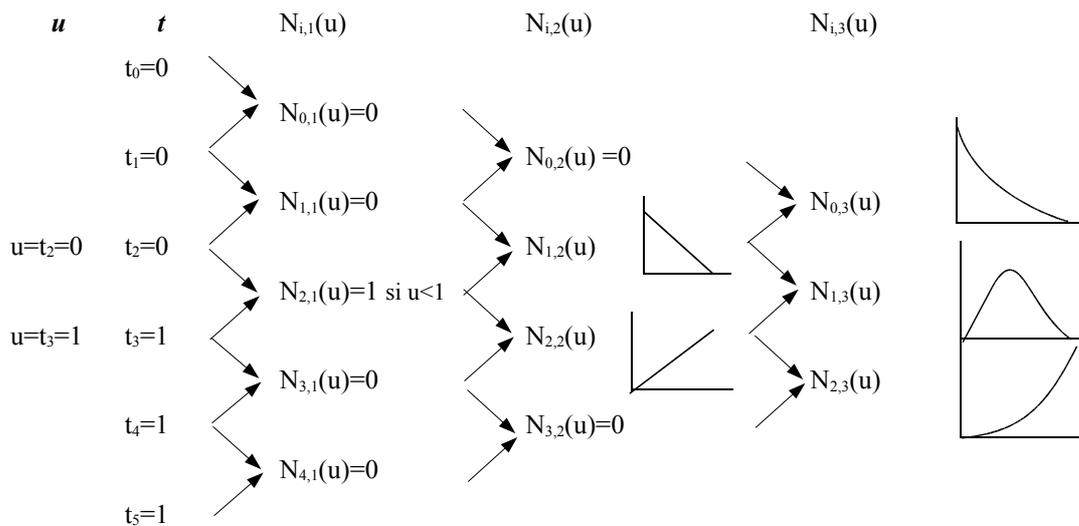


Figura 4.13. Proceso de cálculo para  $k=3$  y  $n=2$

Existen varias configuraciones usuales del vector de nodos, que se pueden utilizar como valores iniciales a los que posteriormente se le pueden realizar ajustes para conseguir determinados comportamientos locales. Un Bspline es **uniforme y periódico** si los elementos del vector de nodos están equiespaciados

$$t_j = t_{j-1} + \delta$$

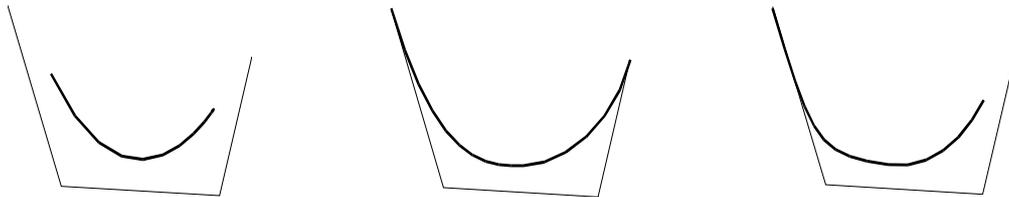
En este caso, todas las funciones de forma son iguales salvo una traslación (figura 4.12). La curva no interpola a ninguno de los puntos de control (ver figura 4.14 izquierda). Esta es la configuración habitual para definir curvas cerradas. Obsérvese que para hacer una curva cerrada se deben repetir los  $k-1$  puntos del principio al final.

Un B-spline es **no periódico** si los elementos del extremo del vector de nodos son iguales, manteniéndose equiespaciados los centrales

$$t_j = \begin{cases} 0 & \text{si } 0 \leq j < k \\ j-k+1 & \text{si } k \leq j \leq n \\ n-k+2 & \text{si } n < j \leq n+k \end{cases}$$

En este caso, la curva interpola a los puntos extremos, pero no a los demás. Ver figura 4.14 centro. Cuando el orden es máximo,  $k=n+1$  la curva, que está descrita por una única ecuación, coincide con una curva Bezier.

Cuando el vector de nodos no satisface ninguna de las condiciones anteriores decimos que el B-spline es **no uniforme** (ver figura 4.14 derecha).



**Figura 4.14.** Efecto de la modificación del vector de nodos. Uniforme periódico (izquierda), no periódico (centro), no uniforme (derecha)

Para los B-Splines cúbicos, uniformes y periódicos la ecuación de los polinomios se puede expresar en forma matricial:

$$P_i(u) = \frac{1}{6} \begin{pmatrix} 1 & u & u^2 & u^3 \end{pmatrix} \cdot \begin{pmatrix} 1 & 4 & 1 & 0 \\ -3 & 0 & 3 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{pmatrix} \cdot \begin{pmatrix} P_i \\ P_{i+1} \\ P_{i+2} \\ P_{i+3} \end{pmatrix}$$

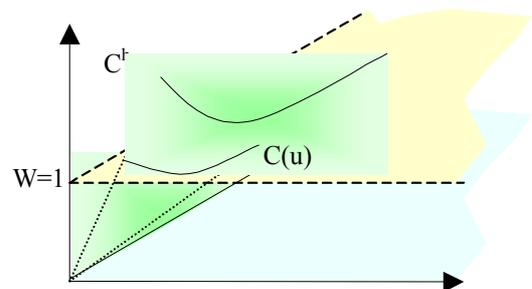
Las curvas B-spline mantienen, independientemente de los valores de los nodos, la propiedad de combinación convexa y variación menor que la poligonal, que tienen las curvas de Bezier.

#### 4.2.4 Curvas racionales

Las curvas racionales pueden interpretarse geoméricamente como la proyección de una curva racional 4D. Generalizan a las curvas no racionales, incluyéndolas como casos particulares, y permiten representar cuádricas y cónicas

$$C^h(u) = (x(u), y(u), z(u), w(u)) = \sum_{i=0}^n P_i^h \cdot B_i(u)$$

$$C(u) = (x(u), y(u), z(u)) = \frac{\sum_{i=0}^n P_i \cdot w_i \cdot B_i(u)}{\sum_{i=0}^n w_i \cdot B_i(u)}$$



Circunferencia:

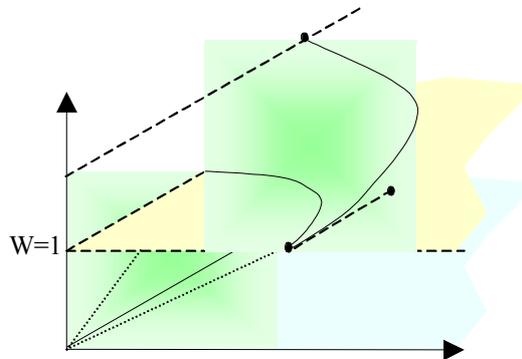
$$C(u) = \left( \frac{1-u^2}{1+u^2}, \frac{2 \cdot u}{1+u^2} \right)$$

$$C^h(u) = (1-u^2, 2 \cdot u, 1+u^2)$$

$$B(u) = \sum_{i=0}^2 P_i \cdot B_i(u) = (1-u^2) \cdot P_0 + 2u \cdot (1-u) \cdot P_1 + u^2 \cdot P_2 =$$

$$= P_0 + 2u \cdot (P_1 - P_0) + u^2 \cdot (P_2 - 2P_1 + P_0)$$

$$B(u) = C^h(u) \Rightarrow \begin{cases} P_{0,x} = 1 & P_{1,x} = P_{0,x} = 1 & P_{2,x} = 2 \cdot P_{1,x} - P_{0,x} - 1 = 0 \\ P_{0,y} = 0 & P_{1,y} = 1 & P_{2,y} = 2 \cdot P_{1,y} - P_{0,y} = 2 \\ P_{0,w} = 1 & P_{1,w} = P_{0,w} = 1 & P_{2,w} = 2 \cdot P_{1,w} - P_{0,w} - 1 = 2 \end{cases}$$



### 4.3 Diseño de superficies.

Una superficie se describe matemáticamente usando dos parámetros, que establecen un sistema de coordenadas sobre ella, permitiéndonos recorrerla.

La superficie se puede definir directamente a partir de una malla de puntos de control, al igual que una curva, con la única diferencia de que los puntos de control forma una distribución bidimensional y que las funciones de forma dependen de dos parámetros,  $B_{ij}(u,v)$ .

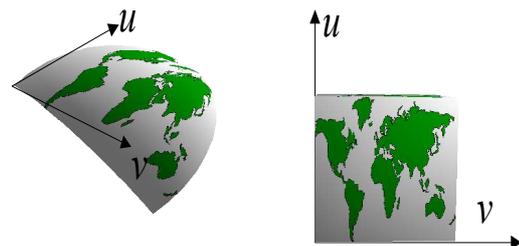


Figura 4.15 Parametrización de una superficie

Alternativamente, es posible definir una superficie a partir de una o varias curvas. Por otra parte, es cada vez más frecuente utilizar mallas de triángulos como una aproximación a la superficie. En esta sección se aborda la representación y diseño de superficies, comenzando con la aproximación por mallas de triángulos, la construcción a partir de curvas y la generación directa usando una malla de puntos de control.

### 4.3.1 Generación a a partir de curvas

Se pueden generar superficies simples a partir de un número reducido de curvas.

#### 4.3.1.1 Cilíndricas.

Formadas por la traslación de un vector sobre una curva cualquiera. Equivale al barrido por traslación de la curva. La ecuación de la superficie es:

$$S(u,v) = P(u) + v \cdot r$$

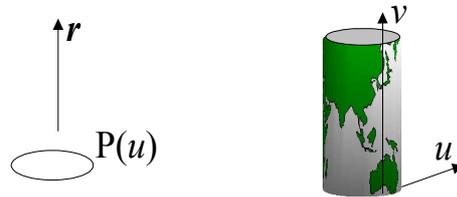


Figura 4.16 Superficie cilíndrica

donde  $P(u)$  es la ecuación de la curva y  $r$  es un vector. Con este método es posible generar superficies que tengan simetría traslacional. Cuando la curva es una circunferencia la superficie generada es la lateral de un cilindro (ver figura 4.16).

#### 4.3.1.2 Superficies de revolución.

Se obtiene al girar una curva plana respecto a un eje contenido en dicho plano (figura 4.17). Si el eje de giro es el eje z, y la curva,  $C(u)$  está contenida en el plano Z-X, esto es

$$C(u) = (x(u), 0, z(u))$$

la ecuación de la superficie es:

$$S(u,v) = (x(u) \cdot \cos(v), x(u) \cdot \text{sen}(v), z(u)),$$

con  $u \in [0, 2\pi]$

Cuando la curva es un segmento de recta paralelo al eje de simetría, la superficie es resultante es la cara exterior de un cilindro (figura 4.18).

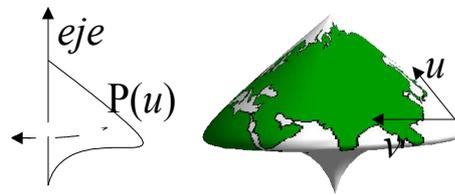


Figura 4.17 Superficie de revolución

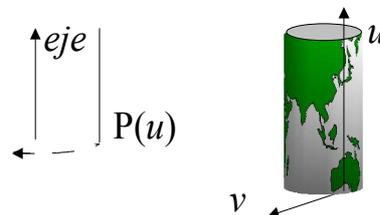


Figura 4.18 Superficie de revolución

Si la curva es una semicircunferencia, la superficie puede ser una esfera (figura 4.19). Si la curva es una circunferencia que no corta al eje, la superficie resultante es un toro (ver figura 4.20).

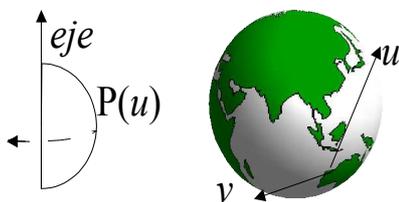


Figura 4.19 Esfera como superficie de revolución

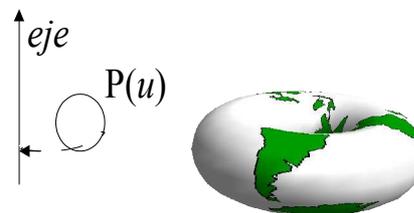


Figura 4.20 Toro como superficie de revolución

#### 4.3.1.3 Superficies regladas.

La superficie está definida por la unión por interpolación lineal de dos curvas del espacio. Las dos curvas deben de tener el mismo rango de variación del parámetro.

$$S(u,v) = (1-v) \cdot C_1(u) + v \cdot C_2(u)$$

Geoméricamente equivale a unir mediante segmentos los puntos de las dos curvas que tienen el mismo valor del parámetro (figura 4.21).

Si una de las curvas es una traslación de la otra, el resultado es el que se obtendría como una superficie cilíndrica construida con una de las curvas y el vector de traslación a hasta la otra. Por tanto, las superficies cilíndricas son un caso particular de superficie reglada.

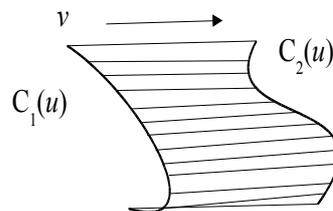


Figura 4.21 Superficie reglada

#### 4.3.1.4 Superficie de unión.

Dada una familia de curvas con la misma parametrización \$S\_i(u)\$, se puede construir una superficie interpolando los puntos con el mismo valor de parámetro en todas las curvas. La interpolación puede realizarse, por ejemplo con Splines. Si \$F\_i\$ son las funciones de forma del método de interpolación usado, la superficie vendrá dada por la expresión:

$$S(u,v) = \sum_{i=1}^n S_i(u) \cdot F_i(v)$$

Las superficies regladas son un caso particular de superficie de unión con interpolación lineal.

#### 4.3.1.5 Generación de perfiles.

Si se crean superficies regladas o de unión se deben de generar previamente las curvas que describen los perfiles que se han de interpolar. Dos formas usuales de crear perfiles son:

- **Por revolución.** Dada una curva \$P(u)\$ y un eje, se obtienen perfiles a partir de la curva girándola respecto al eje. La figura 4.22 muestra el resultado de generar ejes por revolución para posteriormente generar superficies regladas. Obsérvese que el resultado no es una superficie de revolución.

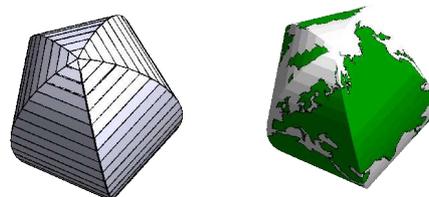


Figura 4.22 Aproximación de una esfera como cinco superficies regladas

- **Perfil dirigido por un eje.** Dada una curva plana  $P(u)$  y una curva cualquiera  $Q(u)$  se pueden obtener perfiles colocando la curva  $P$  sobre la  $Q$  con una orientación fija respecto a ésta (figura 4.23).

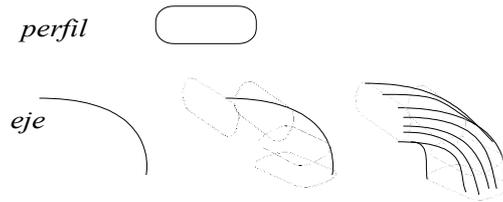


Figura 4.23 Perfil dirigido por un eje

Para ello se sigue el siguiente proceso:

1. trasladar  $P$  hasta un punto de  $Q$ ,
2. calcular tangente de  $Q$  en el punto de contacto
3. girar  $P$  para que el plano que lo contiene sea perpendicular a la tangente

### 4.3.2 Generación a partir de mallas de puntos de control

Se pueden generalizar los métodos de diseño de curvas para superficies. En este caso se usará una malla rectangular de puntos de control,  $P_{ij} = (x_{ij}, y_{ij}, z_{ij})$ . La superficie se calculará a trozos, *parches*, en la forma:

$$S(u,v) = (x(u,v), y(u,v), z(u,v)) = \sum_{j=0}^m \sum_{i=0}^n P_{ij} \cdot B_{ij}(u,v)$$

Las funciones de forma  $B_{ij}(u,v)$  se pueden obtener por producto de las funciones de forma en  $u$  y  $v$ , usadas para curvas

$$B_{ij}(u,v) = F_i(u) \cdot G_j(v)$$

Donde  $F$  y  $G$  son funciones de forma. Obsérvese que las funciones de forma para  $u$  y  $v$  no tienen que ser las mismas. Pueden variar en grado, vectores de nodos, o incluso ser métodos diferentes. Conceptualmente el proceso se puede ver como la generación de la superficie como una superficie de unión con el método  $G$ , usando como perfiles la familia de  $m$  curvas  $C_i(u) = \sum_{j=0}^n P_{ij} \cdot F_i(u)$ .

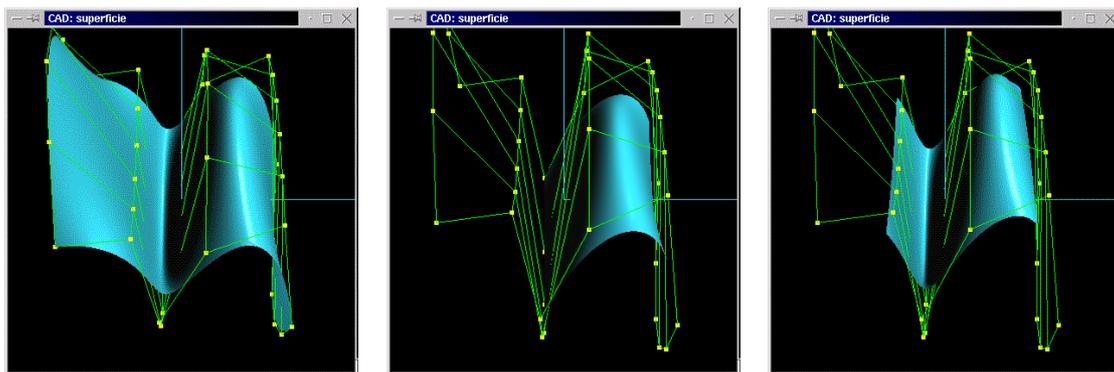


Figura 4.24 Superficies Bspline. En los tres casos la superficie es uniforme y *no-periódica* en  $v$ . A la izquierda *no-periódica* en  $u$ . En el centro *uniforme y periódica* en  $u$ . A la derecha, *no uniforme* en  $u$ .

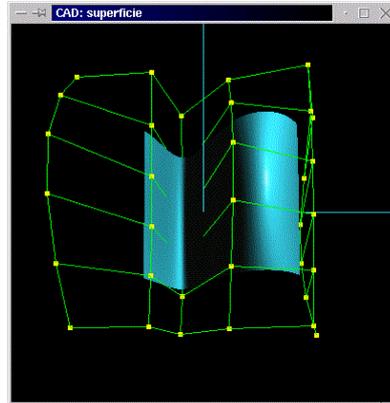
Las figuras 4.24 y 4.25 muestran varios ejemplos de superficies Bspline de orden 4 en  $u$  y  $v$ , definidos con una malla de  $6 \times 6$  puntos de control.

Los vectores de nodos utilizados son:

{0.0, 0.0, 0.0, 0.0, 1.0, 2.0, 3.0, 3.0, 3.0, 3.0} para uniforme **-no periódica**

{0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0} para **uniforme - periódica**

{0.0, 1.0, 2.0, 3.0, 3.0, 3.0, 6.0, 7.0, 8.0, 9.0} para **no uniforme**



**Figura 4.25** Superficie uniforme - periódica en  $u$  y  $v$

## Bibliografía

Existe una amplia bibliografía sobre el tema, aunque la mayor parte de ella posee un enfoque excesivamente matemático o teórico para ser usada como guía. Por otra parte, la mayoría de los textos de Informática Gráfica incluye algún tema sobre diseño de curvas y superficies, aunque en ocasiones es demasiado general.

En la primera sección se puede utilizar el libro de Foley, que hace un buen tratamiento de lo conceptos generales y los algoritmos de visualización, pero que no trata con la suficiente profundidad los métodos de diseño.

Las secciones siguientes se basan en el libro de Anand, que hace un tratamiento conciso, pero claro y completo, de la mayor parte del tema. En algunos aspectos concretos se puede utilizar el survey de Böhm, Farin y Kahman sobre curvas y superficies.

### Complementos 4.1 Superficies en OpenGL

La librería glut de OpenGL contiene funciones para dibujar superficies y curvas Bsplines tanto no racionales como racionales (abreviadamente **nurbs**, de non-uniform rational B-Splines). OpenGL dibuja las superficies discretizandolas en triángulos. Permite asignarles propiedades de materiales, posibilitando su visualización con el mismo nivel de realismo que el resto de los objetos.

Para utilizar superficies en un programa OpenGL es necesario definir las estructuras de datos para almacenar la descripción de ésta, inicializar los parámetros de visualización y almacenamiento y crear la superficie. Veremos cada uno de estas secciones por separado.

Las estructuras de datos que es necesario crear son la matriz de puntos de control, el vector (o vectores) de nodos y un manejador de la superficie. Las matrices de puntos de control y nodos son del tipo GLfloat y el manejador es del tipo GLUnurbs. A continuación se muestra un ejemplo de definición de estos elementos:

```
GLfloat ctlpoints[4][4][3]; // 4x4 puntos de control 3D
GLfloat knots[8] = {0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 1.0};
GLUnurbsObj *theNurb;
```

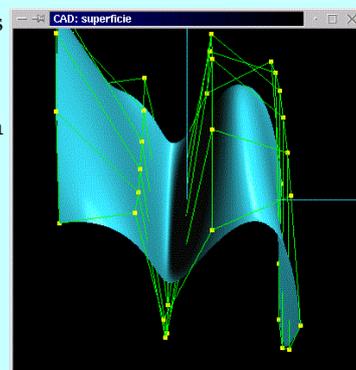
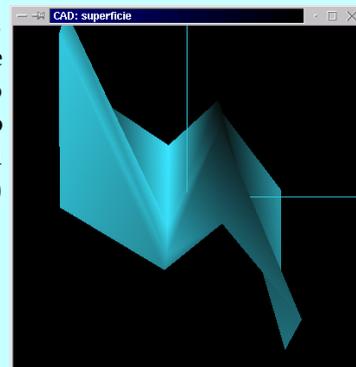
En la inicialización se debe crear el manejador de nurbs, e inicializar sus parámetros de visualización: tolerancia de muestreo, modo de dibujo (isocurvas o relleno) y activar la generación automática de normales. Un ejemplo de llamada a estas funciones es:

```
theNurb = gluNewNurbsRenderer(); // crea un manejador de nurbs
gluNurbsProperty(theNurb, GLU_SAMPLING_TOLERANCE, 5.0);
gluNurbsProperty(theNurb, GLU_DISPLAY_MODE, GLU_FILL);
glEnable(GL_AUTO_NORMAL); // genera automaticamente las normales
```

El valor asignado a `GLU_SAMPLING_TOLERANCE` indica el tamaño de la máxima arista de los triángulos en que se subdividirá la superficie para dibujarla. Por tanto cuando menor sea el valor mayor será la precisión y el tiempo necesario para visualizar la superficie. La figura muestra el efecto de utilizar un valor muy grande (500 a la derecha) frente a uno pequeño (5 a la izquierda).

Por último, es necesario dibujar la superficie, para ello se utilizan las instrucciones `gluBeginSurface`, `gluNurbsSurface` y `gluEndSurface`. A todas ellas se les pasa el manejador de Nurbs como primer argumento. Por ejemplo, las siguientes intrucciones crean una superficie de orden 4x4:

```
gluBeginSurface(theNurb); // Se va a dar una
superficie
    gluNurbsSurface(theNurb, 8, knots, 8,
        knots, 4 * 3, 3,
        &ctlpoints[0][0][0], 4, 4,
        GL_MAP2_VERTEX_3);
gluEndSurface(theNurb); // fin de
definición de superficie
```



A continuación se describe los argumentos de la función `gluNurbsSurface`.

```
gluNurbsSurface(theNurb, nNodosU, MnodosU, nNodosV, MnodosV, deltaUP,
    deltaVP, &P, ordenU, ordenV, TipoVertices);
```

- **theNurb:** Manejador de nurbs.
- **nNodosU:** Tamaño del vector de nodos para  $u$ .
- **MnodosU:** Matriz lineal de reales conteniendo el vector de nodos para  $u$ .
- **nNodosV:** Tamaño del vector de nodos para  $v$ .
- **MnodosV:** Matriz lineal de reales conteniendo el vector de nodos para  $v$ .
- **deltaUP:** Espaciamento entre puntos de control en la matriz de puntos de control con valores de  $u$  consecutivos. La matriz de puntos de control se pasa como una matriz lineal. El sistema necesita saber cuantos valores reales debe avanzar en el array cuando incrementa el índice en la dirección  $u$  en uno. El espaciamento dependerá de la dimensión de los puntos de control (3 o 4 para racionales), y de la forma en que se ha definido la matriz de puntos de control.
- **deltaVP:** Igual para  $v$ .
- **&P:** Puntero al primer elemento de la matriz de puntos de control. Los puntos de controles son valores reales. Su dimensión, y por tanto el número de reales usados para cada punto, depende del tipo de Bspline definido.
- **ordenU:** Orden en  $u$
- **ordenV:** Orden en  $v$
- **TipoVertices:** `GL_MAP2_VERYEX_3` o `GL_MAP2_VERTEX_4`, para puntos 3D (Bsplines normales o 4D para racionales).

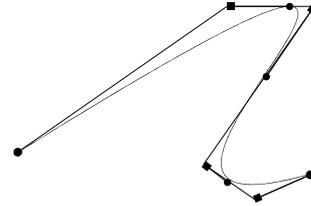
Obsérvese que no se especifica el número de puntos de control de forma directa. El número de puntos de control es el número de elementos en vector de nodos menos el orden.

## Referencias

- Anand V.B.: "Computer Graphics and Geometric Modelling for Engineers". John Wiley & Sons, 1993.
- Böhm<sup>84</sup> Böhm W.; Farin G.; Kahmann J.: "A survey of curve and surface methods in CAGD". Computer Aided Geometric Design". Vol.1, N.1, 1984. pp. 1-60.
- Foley J.D.; van Dam A.; Feiner S.K.; Hughes J.F.: "Computer Graphics. Theory and Practice". Addison-Wesley 1992.
- Hearn D.D.; Baker M.P.: "Gráficas por computadora". Prentice Hall 1994
- Piegl L. (ed.): "Fundamental Developments of Computer-Aided Geometric Design". Academic Press. 1993.
- Barsky B.A.: "Computer Graphics and Geometric Modelling Using Beta-splines". Springer-Verlag 1988
- Farin G.: "Curves and Surfaces for computer aided geometric design. A practical Guide". Academic Press 1988

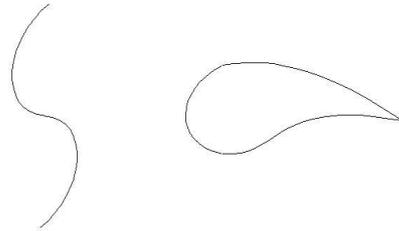
### Ejercicios

1. Obtener una formula de interpolación mediante splines cúbicos locales que utilice solo la mitad de los puntos de control como puntos de paso, de forma que la pendiente en cada punto de paso sea la de la línea que une dicho punto con el siguiente punto de control (que no es punto de paso), tal como se muestra en la figura.

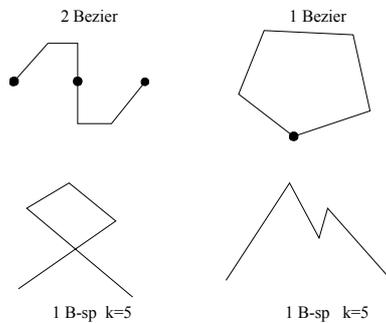


2. Obtener la forma matricial de la expresión de las curvas de Bézier cúbicas.

3. ¿Dónde deben colocarse los puntos de control para generar las curvas de la figura usando curvas de Bezier de grado 3? Marcar los puntos de control numerándolos y dibujar el polígono de control.



4. Valorar las ventajas e inconvenientes de la utilización de polinomios de interpolación de grado alto.
5. Demostrar que una curva de Bézier está dentro de la envolvente convexa.
6. Dados los polígonos de control de la figura dibujar la curva que describen.

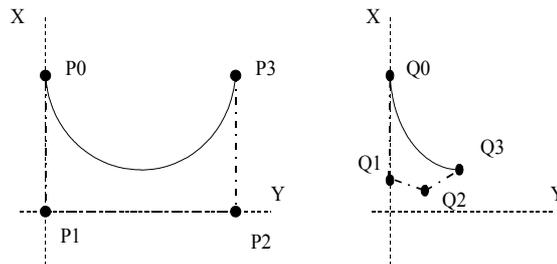


7. Dada la parábola  $y=x^2$  ( $0 \leq x \leq 2$ ). Calcular los puntos de control de la curva de Bezier que representa a dicha parábola.
8. Calcular el valor del punto con coordenadas  $(u,v)=(0.75,0.5)$  de la superficie reglada que se obtiene a partir de las siguientes curvas:

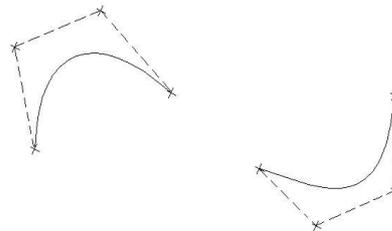
$$C_1: \begin{matrix} y = 1-x^2 \\ z = 0 \end{matrix} \quad -1 \leq x \leq 1$$

$$C_2: \begin{matrix} y = -x \\ z = 1 \end{matrix} \quad -1 \leq x \leq 1$$

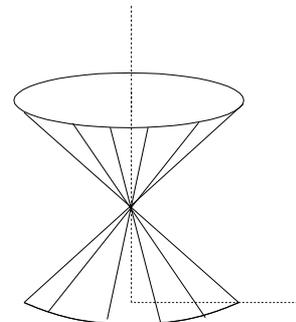
9. Modelar un balón de playa cuya sección ecuatorial sea un hexágono
10. Modelar la superficie de un trampolín de salto de esquí.
11. Construir la superficie de una rampa en forma de escalera de caracol.
12. Aproximar un arco de radio unidad con: a) bezier de 3 puntos y b) bezier 4 puntos. Justificar los puntos seleccionados.
13. Dada una curva de Bezier cualquiera,  $P(t)$ , con cuatro puntos de control, obtener una curva de Bezier,  $Q(u)$ , del mismo grado, que sea igual a  $P(t)$  para el intervalo de variación del parámetro  $t$ ,  $[0,a]$  (ver figura).



14. Demostrar que las curvas de Bézier pasan por los puntos inicial y final, y que su pendiente está determinada por el punto contiguo.
15. Obtener la expresión paramétrica de la superficie de un toroide.
16. Dados los siguientes puntos  $P_0(0,1,0)$ ,  $P_1(1,0,0)$ ,  $P_2(0,0,1)$  y  $P_3(1,1,1)$ , determinar la superficie bilineal  $S(l,m)$ .
17. Dadas dos curvas Bezier de grado tres cualesquiera, encontrar una tercera curva de Bezier que una las dos con continuidad  $C^1$ . Dibujar la curva resultante sobre la figura.



18. Dados dos círculos con centro en el eje Y y radio R, contenidos en los planos  $Y=0$  e  $Y=10$ , construir la superficie de la figura.



19. Encontrar la superficie reglada que se forma con el semicírculo de radio unidad ( $x^2+y^2=1, z=0, y>0$ ) y la parábola ( $y=1-x^2, z=1, -1 \leq x \leq 1$ ).
20. Dadas las curvas de Bézier C1 ( $P_0, P_1, P_2$ ) y C2 ( $P_3, P_4, P_5$ ). Comentar como se alcanzaría continuidad C0 y C1. Suponiendo fijos los puntos extremos y el común, encontrar la configuración que permite obtener continuidad C2.