DISEÑO ASISTIDO POR ORDENADOR

4º Curso Ingeniería Informática

Dpto. Lenguajes y Sistemas Informáticos

Curso 2005/2006

TEMA 4. Diseño de curvas y superficies

TEMA 4. Diseño de curvas y superficies

Bibliografía

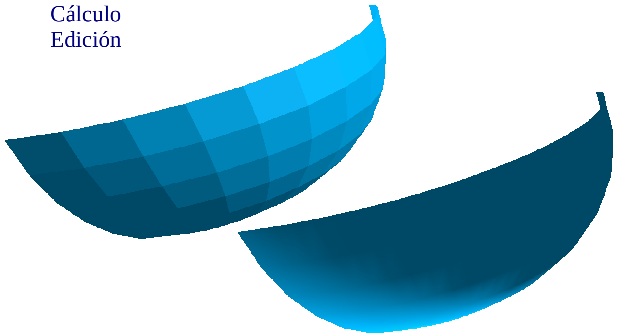
Anand V.B.: "Computer Graphics and Geometric Modelling for Engineers". John Wiley & Sons, 1993.

Foley J.D.; van Dam A.; Feiner S.K.; Hughes J.F.: "Computer Graphics. Theory and Practice". Addison-Wesley 1996. (Hay edición resumida en castellano: "Introduccion a la graficacion por computador", 1996).

Hearn D.D.; Baker M.P.: "Computer graphics. C version". 2nd Ed. 1997. Prentice Hall.

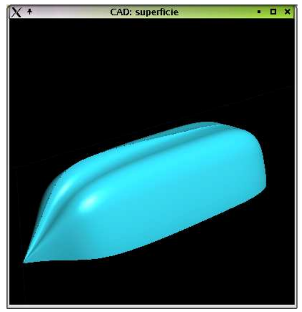
4.1 Representación y visualización.

Una superficie no es una malla de polígonos:
- Visualización
- Cálculo
- Edición



4.1 Representación y visualización.

¿Como podemos describir las curvas y superficies?
- Tipo de ecuaciones.
- Parámetros que va a editar el usuario.



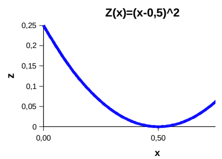
4.1.1 Representación

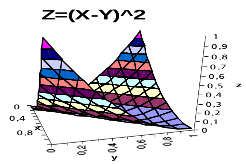
Ecuaciones explícitas

3D $z = f(x,y)$ (superficie)

ó $y = f_1(x)$, $z = f_2(x)$ (curva en el espacio)

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



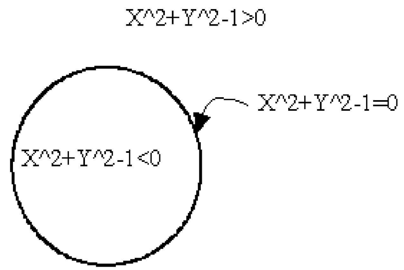


4.1.1 Representación

Ecuaciones implícitas

3D $f(x,y,z) = 0$ (superficie)

2D $f(x,y) = 0$ (curva en el plano)



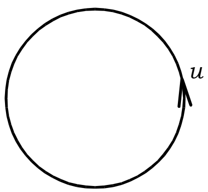
4.1.1 Representación

Ecuaciones paramétricas

3D
$x = f_1(u,v)$,
$y = f_2(u,v)$,
$z = f_3(u,v)$ $u \in [u_1,u_2]$, $v \in [v_1,v_2]$ (superficie)

2D
$x = f_1(u)$
$y = f_2(u)$ $u \in [u_1,u_2]$ (curva en el plano)

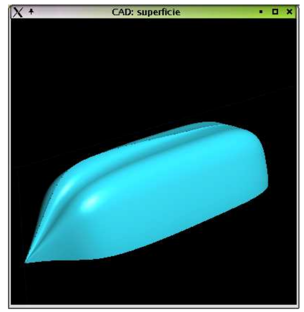
$x = \cos(2\pi u)$
$y = \text{sen}(2\pi u)$ $u \in [0,1]$



4.1.1 Representación

¿Como podemos describir las curvas y superficies?

- Tipo de ecuaciones: **Paramétricas**
- Parámetros que va a editar el usuario.



4.1.1 Representación

$x = f_1(u)$

$y = f_2(u) \quad u \in [u_1, u_2]$ (curva en el plano)

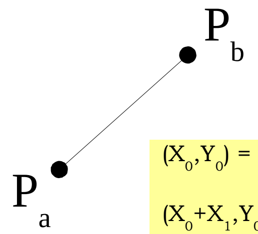
¿Cos(u)? ¿u^2?
¿Cos(u^2)? ¿u^2·Cos(u)?
¿Tan(u^2)? ¿Tan(u^2)·Cos(u)?

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

4.1.1 Representación

$x = f_1(u)$

$y = f_2(u) \quad u \in [u_1, u_2]$ (curva en el plano)



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

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

$(X_0, Y_0) = \mathbf{P_a}$

$(X_0 + X_1, Y_0 + Y_1) = \mathbf{P_b} \quad \Rightarrow \quad (X_1, Y_1) = \mathbf{P_a} - \mathbf{P_b}$

$X(u) = P_a.x + (P_a.x - P_b.x) \cdot u$

$Y(u) = P_a.y + (P_a.y - P_b.y) \cdot u$

$X(u) = (1-u) \cdot P_a.x + u \cdot P_b.x$

$Y(u) = (1-u) \cdot P_a.y + u \cdot P_b.y$

4.1.1 Representación

$X = f_1(u)$

$y = f_2(u) \quad u \in [u_1, u_2]$ (curva en el plano)

$X(u) = (1-u) \cdot P_a.x + u \cdot P_b.x$

$Y(u) = (1-u) \cdot P_a.y + u \cdot P_b.y$

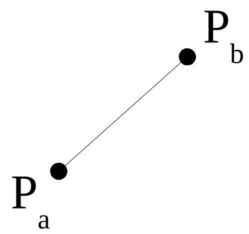
$\mathbf{P(u)} = (1-u) \cdot \mathbf{P_a} + u \cdot \mathbf{P_b}$

Funciones de forma

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

Puntos de control

4.1.1 Representación



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

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

$B_0(u) = 1-u$

$B_1(u) = u$

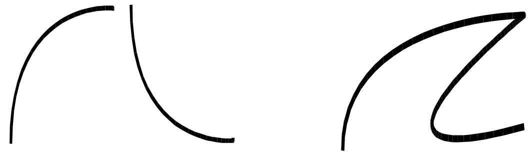
$$\boldsymbol{P}(u) = \begin{pmatrix} B_0(u), & B_1(u) \end{pmatrix} \cdot \begin{pmatrix} \boldsymbol{P_0} \\ \boldsymbol{P_1} \end{pmatrix}$$

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

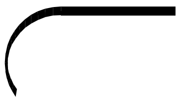
4.1.2 Propiedades

Discontinua

Continua C^0

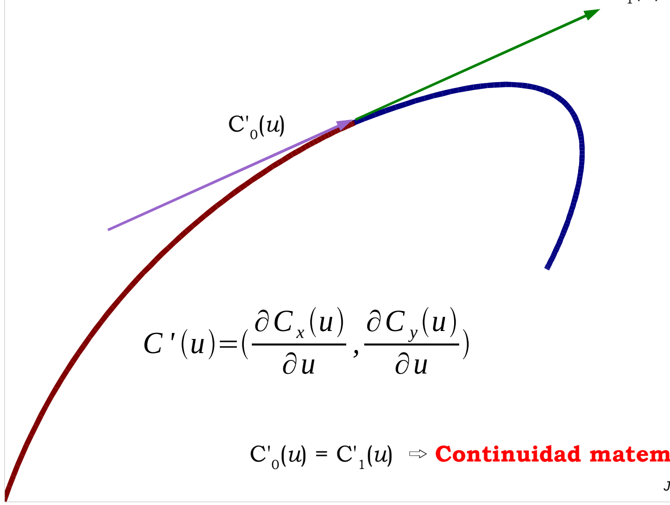


Continuamente diferenciable C^1



C^k

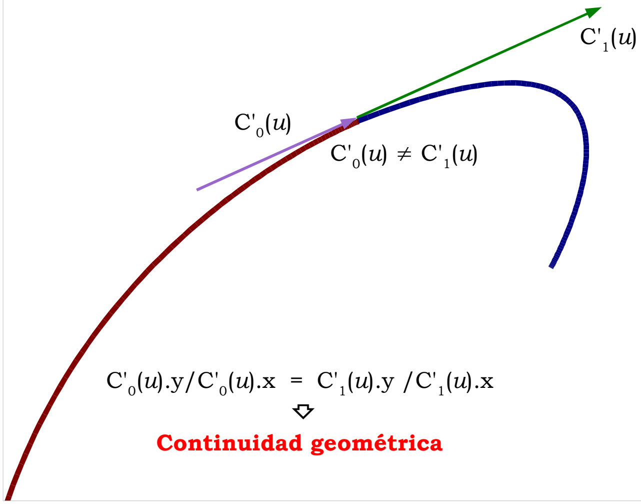
4.1.2 Propiedades: Continuidad



$$C'(u) = \left(\frac{\partial C_x(u)}{\partial u}, \frac{\partial C_y(u)}{\partial u}\right)$$

$C'_0(u) = C'_1(u)$ ⇨ **Continuidad matemática**

4.1.2 Propiedades: Continuidad

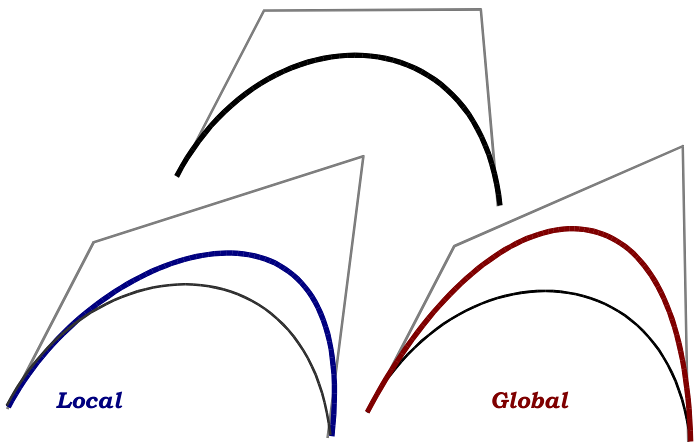


$C'_0(u).y / C'_0(u).x = C'_1(u).y / C'_1(u).x$

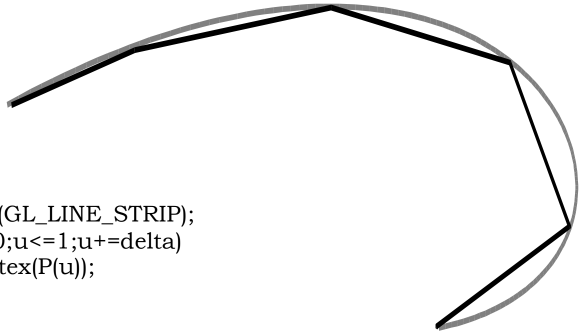
⇩

Continuidad geométrica

4.1.2 Propiedades



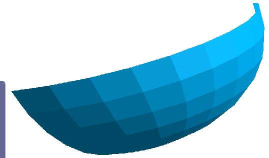
4.1.3 Dibujo de curvas

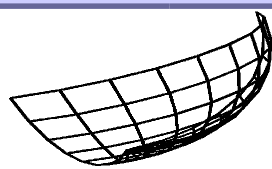


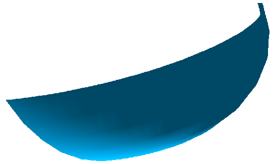
```
glBegin(GL_LINE_STRIP);
for (u=0;u<=1;u+=delta)
    glVertex(P(u));
glEnd();
```


4.1.4 Visualización de superficies

$$S(u,v)=\sum_{i=1}^{n}\sum_{j=1}^{m} P_{i,j}\cdot B_{i,j}(u,v)$$







```
glBegin(GL_TRIANGLES);
 for (u=0;u<=1;u+=delta)
     for (v=0;v<=1;v+=delta)
        glVertex(S(u,v));
 glEnd();
```

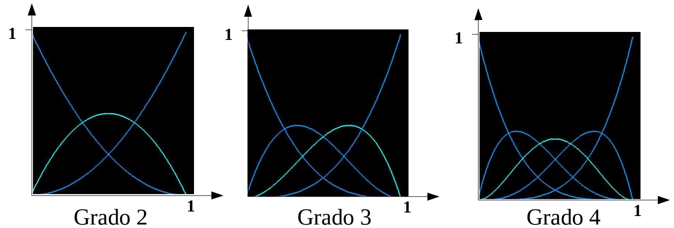

4.2.1 Curvas de Bèzier

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

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

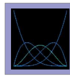
$$\binom{n}{i}=\frac{n!}{i!\cdot(n-i)!}$$

4.2.1 Curvas de Bèzier



$$\sum_{i=0}^{n} B_i^n(u)=1$$

$$B_i^n(u)\geqslant 0$$



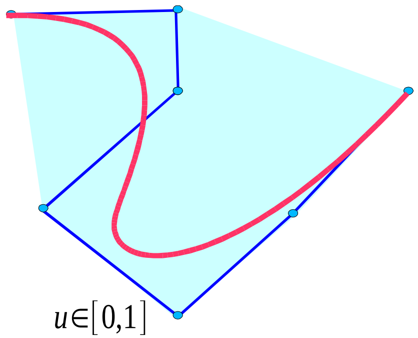
Ejecutar Bezier

4.2.1 Curvas de Bèzier: propiedades

La curva está contenida en la envolvente convexa de los puntos de control

$$\sum_{i=0}^{n} B_i^n(u)=1$$

$$B_i^n(u)\geqslant 0$$



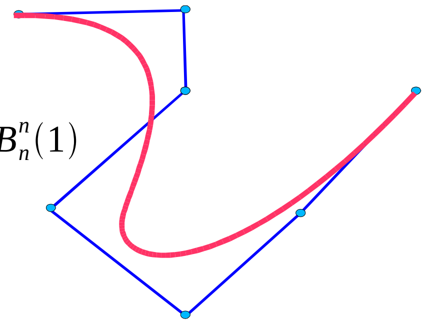
$$\boldsymbol{P}(u)=\sum_{i=0}^{n} \boldsymbol{P}_i B_i^n(u) \qquad u\in[0,1]$$

4.2.1 Curvas de Bèzier: propiedades

La curva no interpola los puntos de control, salvo el primero y último

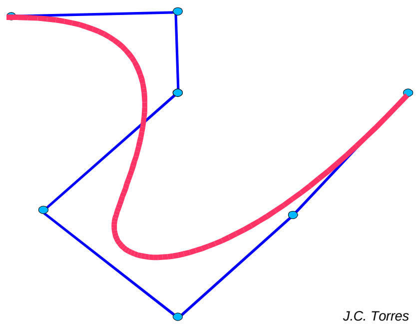
$$B_i^n(u)\neq 1$$

$$salvo\ B_0^n(0)\ y\ B_n^n(1)$$

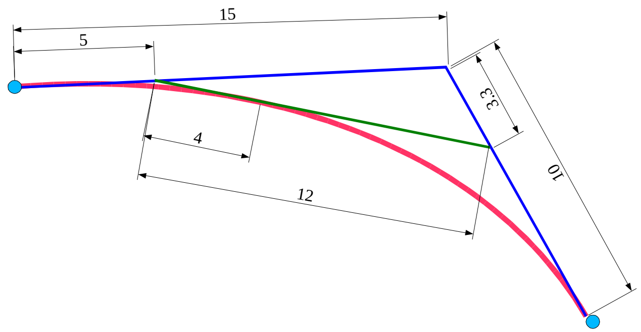


4.2.1 Curvas de Bèzier: propiedades

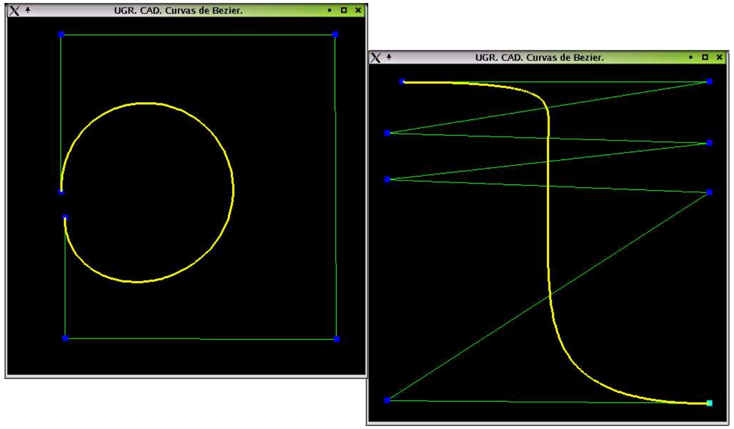
- *La dirección en los extremos está determinada por P_1 y P_{n-1}.*
- *El grado del polinomio es el número de puntos menos uno.*
- *La modificación de un punto de control afecta a toda la curva.*
- *La curva sigue la forma de la poligonal.*
- *La continuidad es C^{∞}.*



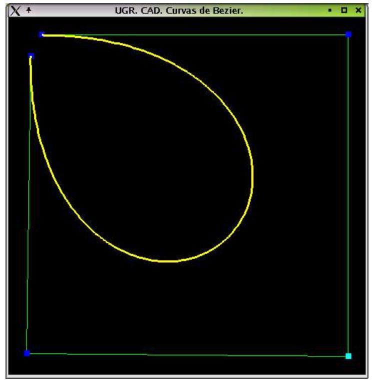
4.2.1 Curvas de Bèzier: método de De Casteljau



4.2.1 Curvas de Bèzier: ejemplos



4.2.1 Curvas de Bèzier: Demo



4.2.2 B-Splines

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

$$N_{i,1}(u)=\begin{cases} 1 & si \quad t_i \leqslant u < t_{i+1} \\ 0 & \end{cases}$$

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

$$t_j \leqslant t_{j+1} \qquad 0 \leqslant j \leqslant n+k$$

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

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

K Orden

Número de puntos de control: n+1

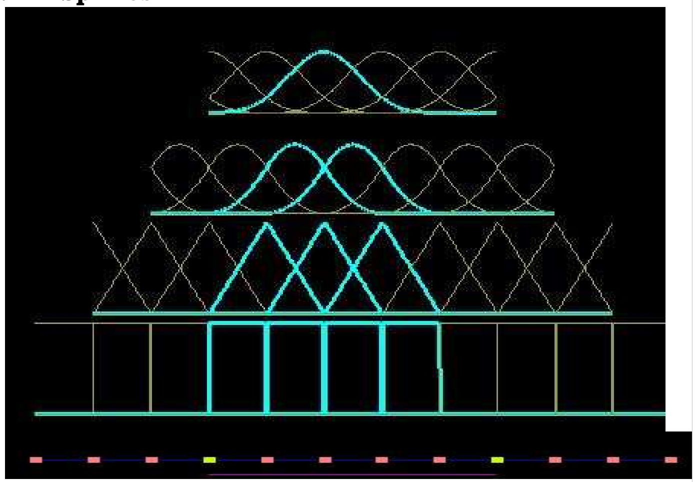
Grado polinomio: k-1

4.2.2 B-Splines

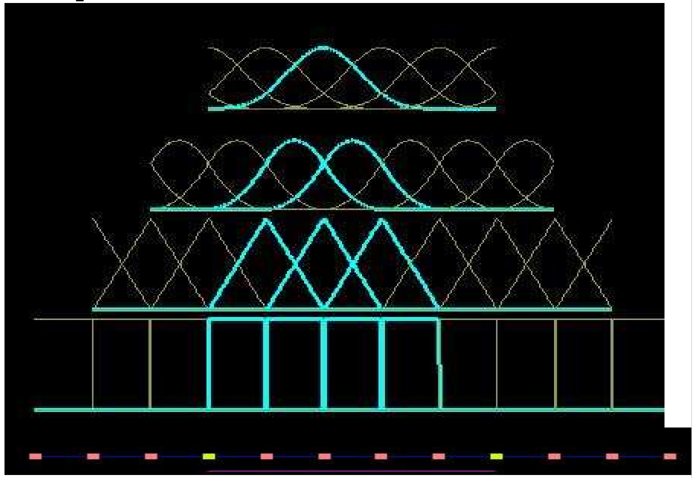
K=4

K=3

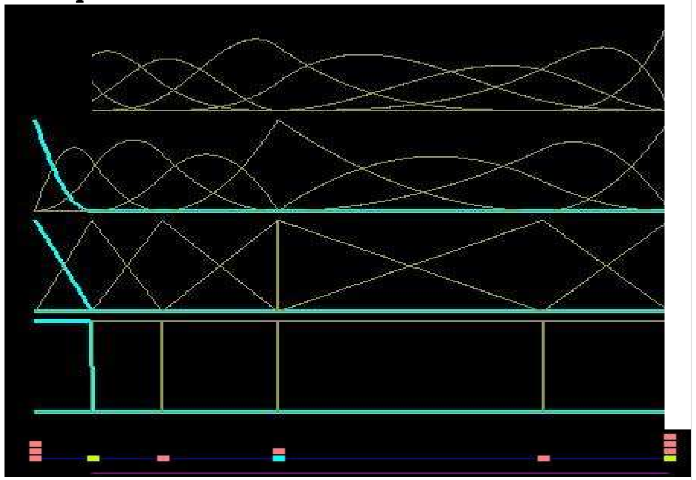
K=2



4.2.2 B-Splines



4.2.2 B-Splines

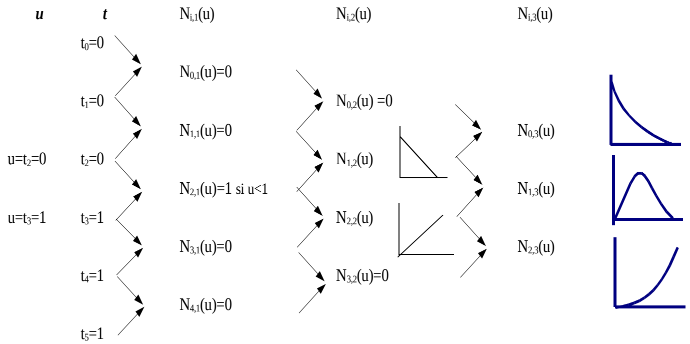


4.2.2 B-Splines: propiedades

- 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 $u \in [t_i, t_{i+k+1}]$
- Continuidad: $N_{i,k}$ es (k-1) veces diferenciable

B-Splines

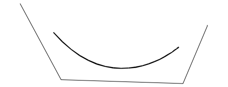
(k=3, n=2)



4.2.2 B-Splines: vector de nudos

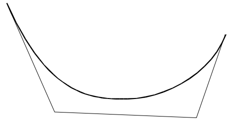
Uniformes y periódicos

$t_i = t_{i-1} + \delta$

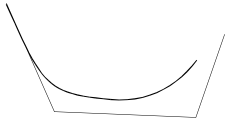


No periódicos

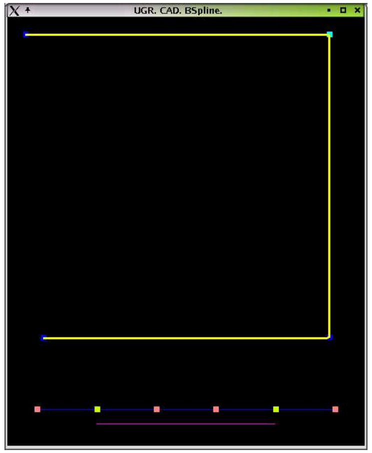
$$t_i = \begin{cases} 0 & \text{si } 0 \in i < k \\ i-k+1 & \text{si } k \leqslant i \leqslant n \\ n-k+2 & \text{si } i > n \end{cases}$$



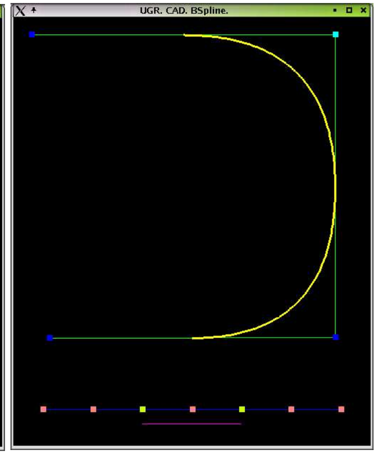
No uniformes



4.2.2 B-Splines: ejemplos

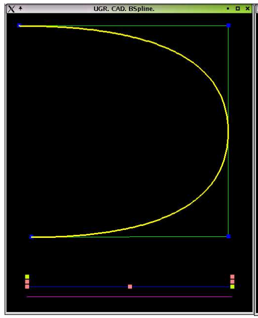


Uniforme. Orden 2

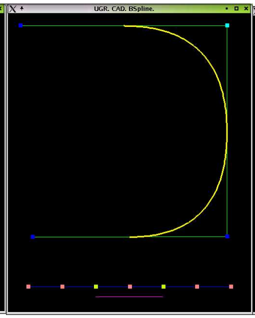


Uniforme. Orden 3

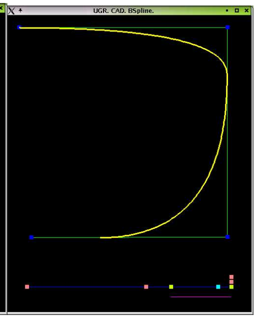
4.2..2 B-Splines: ejemplos



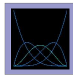
No periódico Orden 3



Uniforme. Orden 3

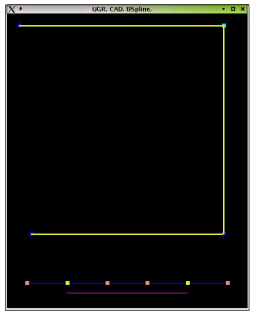


No uniforme. Orden 3

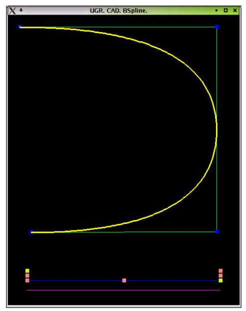


Ejecutar BSpline

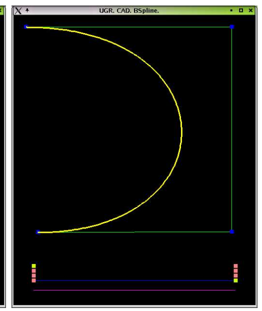
4.2.2 B-Splines: ejemplos



No periódico
Orden 2

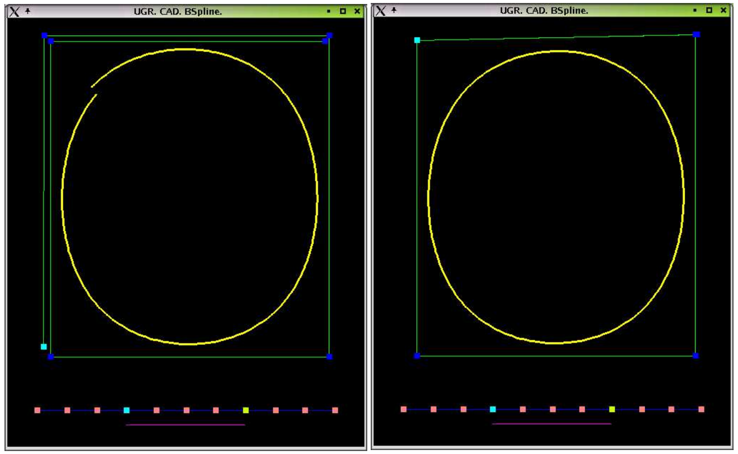


No periódico
Orden 3



No periódico
Orden 4

4.2.2 **B-Splines:** ejemplos



Construccioón de una curva cerrada. (Uniforme. Orden 4)

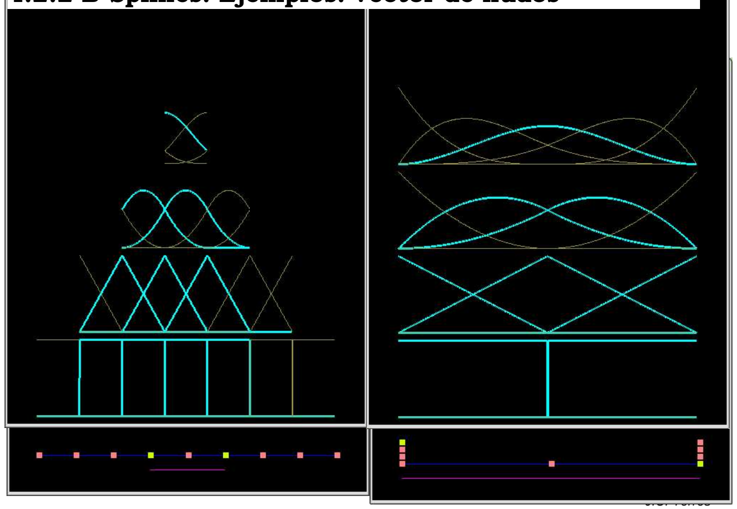
4.2.2 **B-Splines:** OpenGL

```
GLUnurbsObj *NurbId;
float nudos[2*max];
float P[max][3];

void initCurva( )
{
NurbId = gluNewNurbsRenderer();
gluNurbsProperty(NurbId, GLU_SAMPLING_TOLERANCE, 5.0);
}
```


4.2.2 **B-Splines:** OpenGL

```
void draw()
{
....
gluBeginCurve(NurbId);   /* Comienza la definición de la curva */
gluNurbsCurve(NurbId,   /* Genera la superficie                */
    M, nudos,                /* Vector de nodos dirección u      */
    3,               /* Número de datos entre puntos de control  */
    &P[0][0],        /* Array de puntos de control        */
    orden,           /* Orden                       */
    GL_MAP1_VERTEX_3);

//  - El número de puntos de control es el número de elementos
// en vector de nodos menos el orden

    gluEndCurve(NurbId);   /* fin de definición de curva */
....
```


4.2.2 B-Splines. Ejemplos: vector de nudos

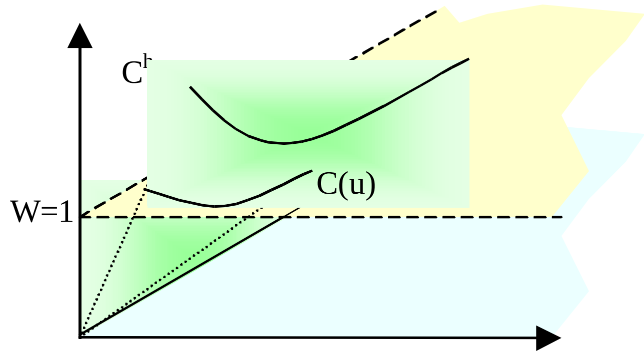


4.2.3 Curvas racionales

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

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

4.2.3 Curvas racionales



4.2.3 Curvas racionales: Circunferencia

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

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

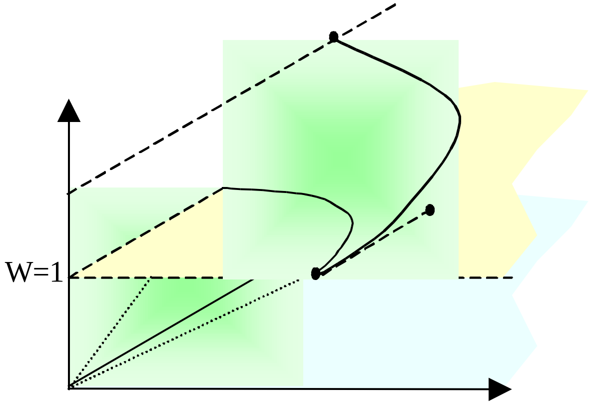
$$B(u)=\sum_{i=0}^{2} P_i\cdot B_i(u)=(1-u^2)\cdot P_0+2u\cdot(1-u)\cdot P1+u^2\cdot P_2=$$
$$=P_0+2u\cdot(P_1-P0)+u^2\cdot(P_2-2P1-P0)$$

4.2.3 Curvas racionales: Circunferencia

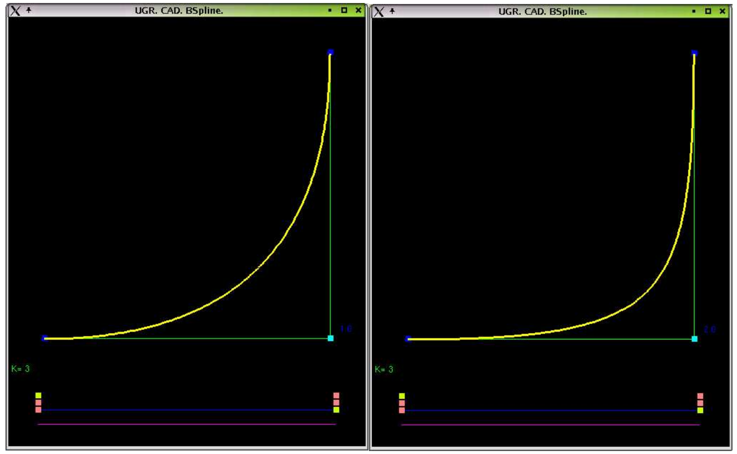
$$B(u)=\sum_{i=0}^{2} P_i\cdot B_i(u)=(1-u^2)\cdot P_0+2u\cdot(1-u)\cdot P1+u^2\cdot P_2=$$
$$=P_0+2u\cdot(P_1-P0)+u^2\cdot(P_2-2P1-P0)$$

$$B(u)=C^h(u)\Rightarrow\begin{cases} P_0.x=1 & P_1.x=P_0.x=1 & P_2.x=2\cdot P_1.x-P_0.x-1=0\\ P_0.y=0 & P_1.y=1 & P_2.y=2\cdot P_1.y=2\\ P_0.w=1 & P_1.w=P_0.w=1 & P_2.w=2\cdot P_1.w-P_0.w+1=2\end{cases}$$

4.2.3 Curvas racionales: Circunferencia

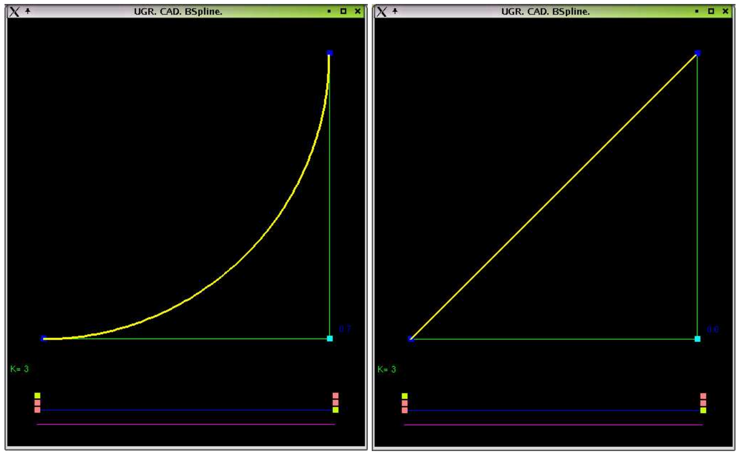


4.2.3 Curvas racionales.



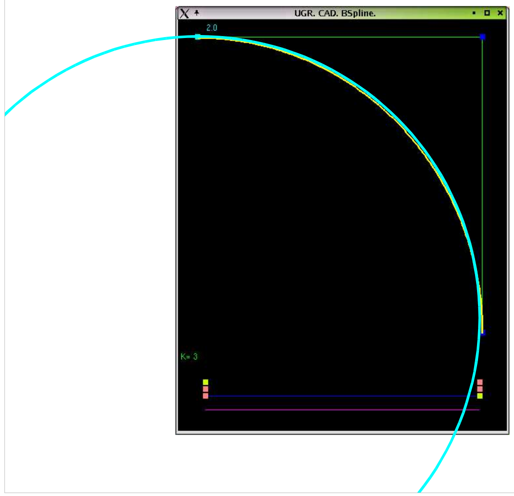
w=1 w=2

4.2.3 Curvas racionales.



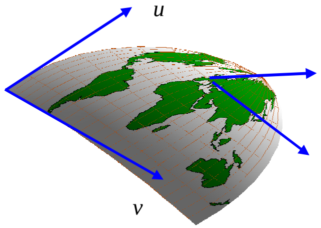
w=0.7 w=0.0

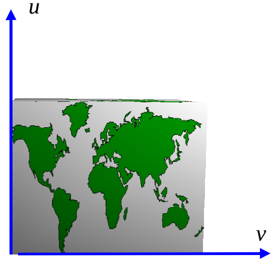
4.2.3 Curvas racionales.



NURBS

4.3 Superficies. *Parametrización de una superficie*

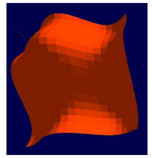




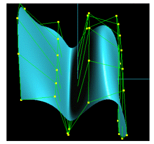
4.3 Superficies: Métodos de construcción

Representación

Malla de triángulos



Malla de puntos de control

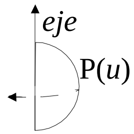


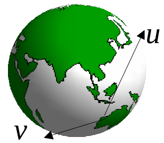
Mediante curvas:

- **Cilíndricas (1)**
- **Revolución (1)**
- **Regladas (2)**
- **Unión (n)**

Generación de perfiles

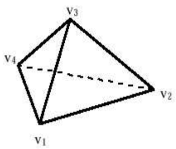
- **Revolución**
- **Dirigido por un eje**





4.3.1 Superficies: Mallas de triángulos

Mallas irregulares



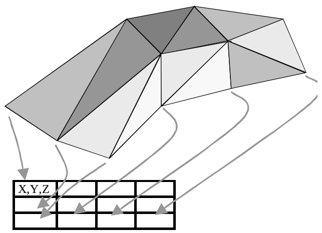
V

x_1	y_1	z_1
x_2	y_2	z_2
x_3	y_3	z_3
x_4	y_4	z_4

T

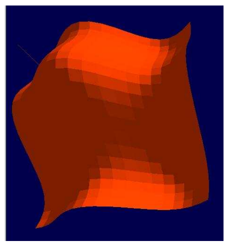
v_1	v_2	v_3
v_1	v_3	v_4
v_3	v_4	v_2
v_2	v_4	v_3

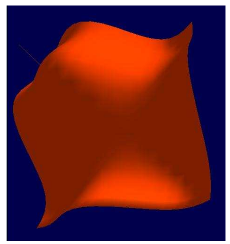
Mallas regulares

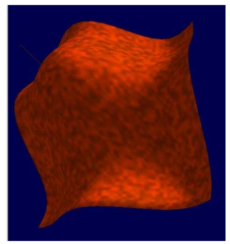


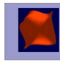
4.3.1 Superficies: Mallas de triángulos

Visualización







 Ejecutar mesh

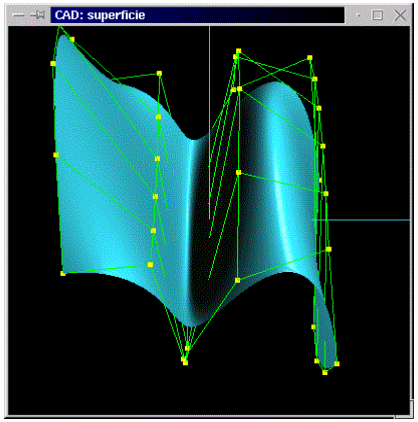
4.3.2 Superficies: Superficies Bspline

$$S(u,v)=\sum_{i=0}^{n}\sum_{j=0}^{m}P_{ij}\cdot N_{i,k}(u)\cdot N_{j,l}(v)$$

$$u_{min}\leqslant u\leqslant u_{max} \qquad 2\leqslant k\leqslant n+1$$

$$v_{min}\leqslant v\leqslant v_{max} \qquad 2\leqslant l\leqslant m+1$$

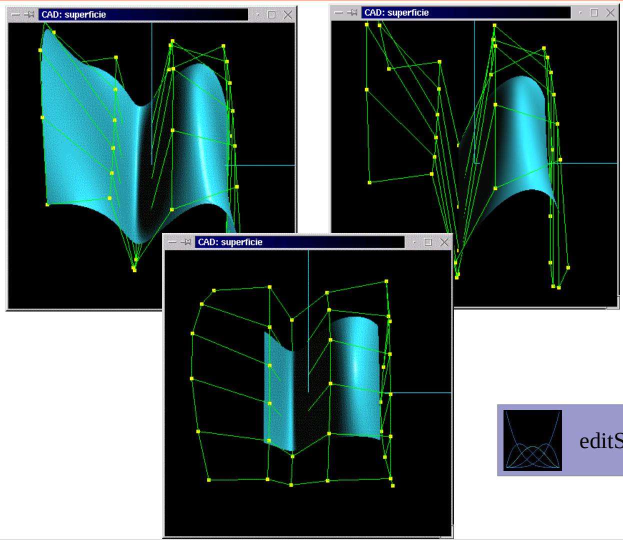
4.3.2 Superficies: Superficies Bspline



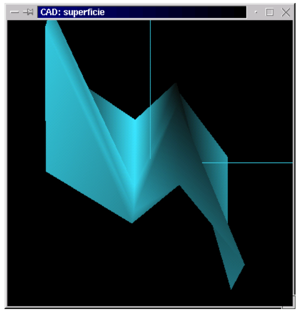
4.3.2 Superficies: Superficies Bspline

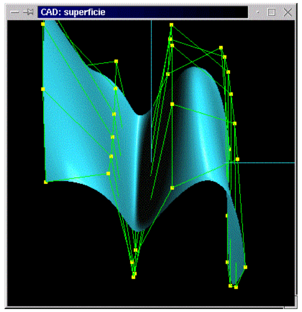
```
GLfloat ctlpoints[4][4][3]; // 4x4 puntos de control 3D
GLfloat knots[8] = {0.0,0.0,0.0,0.0,1.0,1.0,1.0,1.0};
GLUnurbsObj *theNurb;

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

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 def. De superficie
```

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

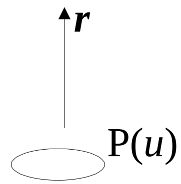



editSuperficie

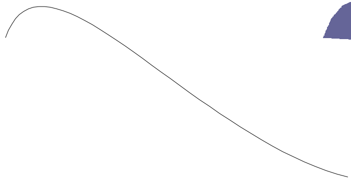


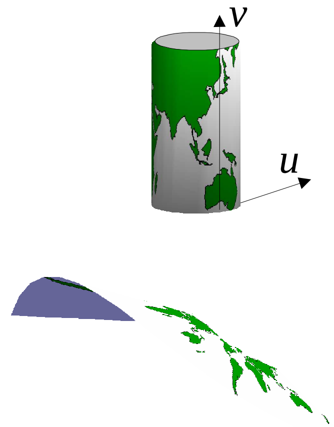


4.3.3 Superficies: Cilíndricas

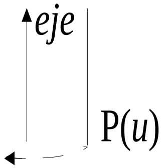


$\mathbf{S}(u,v) = \mathbf{P}(u) + v \cdot \mathbf{r}$
$v \in [0, 1]$

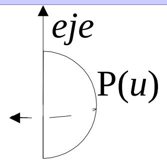


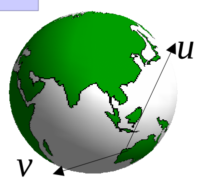


4.3.3 Superficies: Revolución

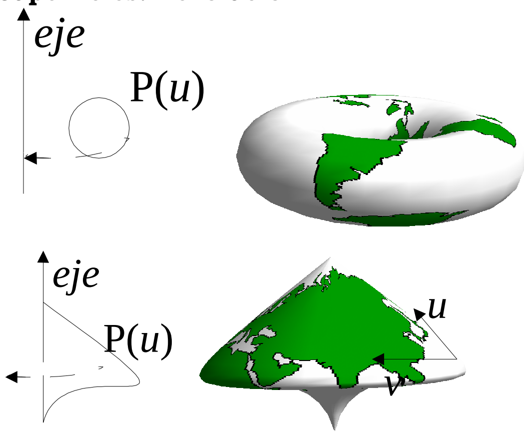


$C(u) = (x(u), 0, z(u))$
$S(u,v) = (x(u) \cdot \cos(v), x(u) \cdot \mathrm{sen}(v), z(u))$
$u \in [0, 2\pi]$

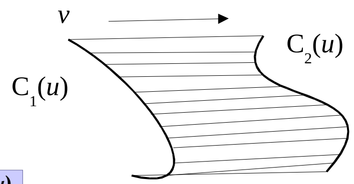




4.3.3 Superficies: Revolución

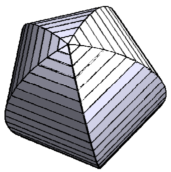


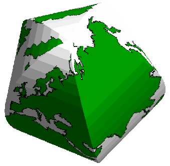
4.3.3 Superficies: Regladas



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

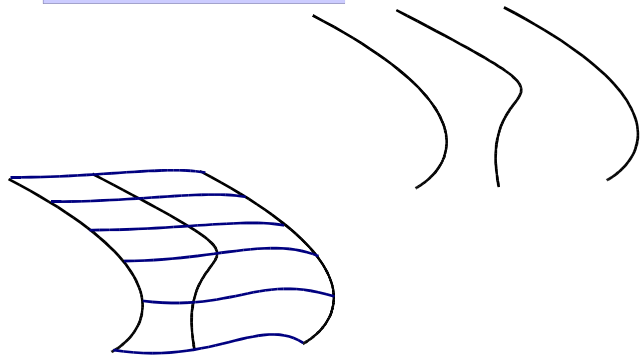
$v \in [0, 1]$





4.3.3 Superficies: Unión

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



4.3.3 Superficies: Generación de perfiles