Resumen de la asignatura Geometría Aplicada.

Ismael Ruiz Donaire
Mercedes Prado Rodríguez
Jesús Rebollo Bueno

Capítulo 1

Aplicaciones entre Superficies

Vamos a introducir las definciones, enunciados de teoremas y ejemplos, con el fin de poder usar este documento en el examen de la asignatura.

Definición 1 (función diferenciable)**.** Sea M una superficie regular (sup.reg.) en $\mathbb{R}^3$. Una **función** continua $f : M \to \mathbb{R}$ se dice **diferenciable** en un punto $p \in M$ si existe una superficie simple $(U, \mathbf{x})$ en M tal que $p \in U$ y $f \circ \mathbf{x}$ es diferenciable en $p' = \mathbf{x}^{-1}(p)$. f será diferenciable en M si lo es en todo punto de M.

Proposición 2. *La* **Definición 1** *no depende de la supeficie simple tomada.*

Proposición 3. *Sean $f : \mathbb{R}^3 \to \mathbb{R}$ una función diferenciable y M una sup.reg. en $\mathbb{R}^3$. Entonces $f|_M : M \to \mathbb{R}$ es diferenciable.*

Nota 4. La **Proposición 3** vale también si tomamos $f : G \subseteq \mathbb{R}^3 \to \mathbb{R}$ con G abierto y $M \subseteq G$.

Ejemplos 5.

1. **Distancia a un punto (al cuadrado).**

   Sean $p_0 = (x_0, y_0, z_0)$ un punto de $\mathbb{R}^3$ y $f : \mathbb{R}^3 \to \mathbb{R}$ tal que $f(x, y, z) = (x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2$. Entonces $f|_M$ es diferenciable con M sup.reg. de $\mathbb{R}^3$.

2. **Distancia a un punto.**

   Sean $p_0 = (x_0, y_0, z_0)$ un punto de $\mathbb{R}^3$ y $f : \mathbb{R}^3 \setminus \{p_0\} \to \mathbb{R}$ tal que $f(x, y, z) = \sqrt{(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2}$. Entonces $f|_M$ es diferenciable con M sup.reg. de $\mathbb{R}^3$ tal que $p_0 \notin M$.

3. **Función altura sobre un plano π_0.**

   Sean π_0 un plano de $\mathbb{R}^3$ que pasa por $p_0 = (x_0, y_0, z_0)$, con vector característico unitario $\overrightarrow{v} = (a, b, c)$, y $f : \mathbb{R}^3 \to \mathbb{R}$ tal que $f(q) = (q - p_0)\overrightarrow{v}$ (distancia signada a π_0). Así, $f(x, y, z) = a(x - x_0) + b(y - y_0) + c(z - z_0)$. Entonces, si M es sup.reg. de $\mathbb{R}^3$, $f|_M$ es diferenciable.

Definición 6. Una aplicación $f : M \to \mathbb{R}^n$ es **diferenciable** si $\pi_i \circ f : M \to \mathbb{R}$ es diferenciable $\forall i = 1, ..., n$, siendo $\pi_i : \mathbb{R}^n \to \mathbb{R}$ la proyección i-ésima.

Definición 7. Sean M,N sup.reg. y $f : M \to N$ una aplicación continua. Se dice que f es **diferenciable** en $p \in M$ si existen $(U, \mathbf{x})$ en M con $p \in \mathbf{x}(U)$ y $(V, \mathbf{y})$ en N con $f(p) \in \mathbf{y}(V)$ tales que $\mathbf{y}^{-1} \circ f \circ \mathbf{x}$ es diferenciable en $p' = \mathbf{x}^{-1}(p)$.

Nota 8. f es diferenciable en M si f es diferenciable en p, $\forall p \in M$.

Proposición 9. *La* **Definición 7** *no depende de las superficies simples.*

Teorema 10. *Sean $f : \mathbb{R}^3 \to \mathbb{R}^3$ diferenciable, M,N sup.reg. tales que $f(M) \subseteq N$. Entonces $f|_M : M \to N$ es también diferenciable.(Se puede cambiar $f : G \subseteq \mathbb{R}^3 \to \mathbb{R}^3$ con G abierto y $M \subseteq G$).*

Ejemplos 11.

1. Si M es **superficie simétrica respecto del plano** XY, entonces $f : M \to M$ tal que $f(x, y, z) = (x, y, -z)$ es diferenciable, por ser restricción de $\widehat{f} : \mathbb{R}^3 \to \mathbb{R}^3$ (M = catenoide, S^2, cilindro,...).

2. Si M es de **revolución respecto del eje** OZ, la rotación de ángulo θ respecto a Z, $f_\theta : M \to M$, es diferenciable, por ser restricción de $f_\theta : \mathbb{R}^3 \to \mathbb{R}^3$, dada por $f(x, y, z) = (x\cos\theta - y\sin\theta, x\sin\theta + y\cos\theta, z)$.

3. Como $f : \mathbb{R}^3 \to \mathbb{R}^3 : (x, y, z) \longmapsto (ax, by, cz)$ con a,b,$c > 0$ ctes, es diferenciable, entonces su restricción a S^2, $f|_{S^2} : S^2 \to E$, también es diferenciable, siendo E el elipsoide de semiejes a, b, c.

4. Como $f : \mathbb{R}^3 \to \mathbb{R}^3$, $f(p) = Ap+T$, con A matriz ortogonal y T traslación, es diferenciable, entonces $f|_M : M \to f(M)$ también lo es.

5. **Aplicación antipodal.** $f : \mathbb{R}^3 \to \mathbb{R}^3 : (x, y, z) \longmapsto (-x, -y, -z)$ es diferenciable. Entonces $f|_{S^2} : S^2 \to S^2$ también lo es. (Es un caso particular del ejemplo 4, tomando $A = -\mathrm{Id}$, $T = 0$, es decir, la **simetría central respecto del origen**).

Definición 12. Una aplicación $f : M \to N$ se dice **regular** en $p \in M$ si existen $(U, \mathbf{x})$ en M con $p \in \mathbf{x}(U)$ y $(V, \mathbf{y})$ en N con $f(p) \in \mathbf{y}(V)$ tales que $\mathbf{y}^{-1} \circ f \circ \mathbf{x}$ es regular en $p' = \mathbf{x}^{-1}(p)$, i.e., $J_{p'}(\mathbf{y}^{-1} \circ f \circ \mathbf{x})$ es una matriz regular.

Nota 13. La **Definición 12** no depende de las superficies simples.

Definición 14. Sean M, N superficies regulares. Una aplicación $f : M \to N$ se llama **difeomorfismo** si es biyectiva, diferenciable y de inversa diferenciable.

Nota 15. f biyectiva y regular $\Leftrightarrow$ f difeomorfismo.

1.1. La Diferencial o Aplicación Tangente

Definición 16. Dados $f : M \to N$ diferenciable y $p \in M$, se define la **diferencial de f en p** como la aplicación $f_{*p} : T_p(M) \to T_{f(p)}(N)$ dada por $f_{*p}\,\overrightarrow{v} = (f \circ \gamma)'(0)$, $\forall \overrightarrow{v} \in T_p(M)$, siendo $\gamma : (-\varepsilon, \varepsilon) \to M$ una curva diferenciable tal que $\gamma(0) = p$ y $\gamma'(0) = \overrightarrow{v}$.

Teorema 17. *Sean $f : M \to N$ diferenciable y $p \in M$.*

a) $f_{*p} : T_p(M) \to T_{f(p)}(N)$ *no depende de la curva tomada, es lineal y "su"matriz es* $J_{p'}(\boldsymbol{y}^{-1} \circ f \circ \boldsymbol{x})$*, con* $p' = \boldsymbol{x}^{-1}(p)$*.*

b) Si f *es regular en* p*, entonces* f_{*p} *es un isomorfismo (aplicación lineal y biyectiva).*

1.2. Isometrías

Definición 18. Dadas M,N sup.reg., una aplicación diferenciable $f : M \longrightarrow N$ se dice **isometría** si f es biyectiva, regular y f_{*p} es isometía $\forall p \in M$.

Nota 19.

1. **Toda isometría es difeomorfismo**.
2. Que $f_{*p} : T_p(M) \to T_{f(p)}(N)$ sea isometría quiere decir que
$$I_p(\overrightarrow{v_1}, \overrightarrow{v_2}) = I_{f(p)}(f_{*p}\overrightarrow{v_1}, f_{*p}\overrightarrow{v_2}),$$
$\forall p \in M$, $\forall \overrightarrow{v_1}, \overrightarrow{v_2} \in T_p(M)$, siendo I_p (resp. $I_{f(p)}$) la Primera Forma Fundamental en p (resp. $f(p)$).
3. **Las isometrías conservan las longitudes de las curvas**.

Definición 20. Se dice que M es **localmente isométrica** a N si $\forall p \in M$, existen U' entorno abierto de p en M, V' abierto en N y $f : U' \to V'$ isometría. Si M es localmente isométrica a N y N es localmente isométrica a M, se dice que son localmente isométricas.

Ejemplos 21. El catenoide y el helicoide **completos** son localmente isométricos, pero **NO** isométricos, pues no son homeomorfos (el catenoide no es simplemente conexo y el helicoide sí lo es).

Teorema 22 (**Caracterización de superficies localmente isométricas**)**.** *Sean* M*,* N *sup.reg. Entonces* M *es localmente isométrica a* $N \Leftrightarrow \forall p \in M$*, existen* U *abierto en* $\mathbb{R}^2$*,* $\boldsymbol{x} : U \to \mathbb{R}^3$ *sup. simple en* M *con* $p \in \boldsymbol{x}(U)$ *e* $\boldsymbol{y} : U \to \mathbb{R}^3$ *sup. simple en* N *tales que los coeficientes métricos* g_{ij} *de* $\boldsymbol{x}$ *y los* h_{ij} *de* $\boldsymbol{y}$ *coinciden:*
$$g_{ij}(u^1, u^2) = h_{ij}(u^1, u^2), \qquad \forall (u^1, u^2) \in U.$$

Corolario 23. *Si* M*,* N *son localmente isométricas, entonces tienen la misma geometría intrínseca (depende de la* $1^a FF$*) en puntos correspondientes.*

Nota 24. Por el **Corolario 23** se deduce que los mapas están "mal hechos", pues $K(S^2) = \dfrac{1}{r^2} > 0$ y $K(\mathbb{R}^2) = 0$ (no tienen la misma curvatura de Gauss).

Ejemplos 25. El **cilindro** es localmente isométrico al plano, pero no es isométrico (pues no es homeomorfo).

Nota 26. Si dos superficies M,N son isométricas (local. isométricas), entonces $K_M = K_N$. El recíproco no es cierto en general.

Teorema 27 (**Teorema de Minding**)**.** *Dos superficies cualesquiera con la misma curvatura de Gauss constante son localmente isométricas.*

Corolario 28. *Sea M una sup. reg. con $K_M = c = cte$:*

1. *Si $c > 0 \Rightarrow M$ es localmente isométrica a una esfera de radio $\frac{1}{\sqrt{c}}$.*
2. *si $c = 0 \Rightarrow M$ es localmente isométrica al plano euclídeo.*
3. *si $c < 0 \Rightarrow M$ es localmente isométrica a una pseudoesfera.*

Definición 29. Una aplicación diferenciable $f : M \longrightarrow N$ se dice **conforme** si es biyectiva, regular y

$$I_p(\overrightarrow{v_1}, \overrightarrow{v_2}) = \lambda^2(p) I_{f(p)}(f_{*p}\overrightarrow{v_1}, f_{*p}\overrightarrow{v_2}),$$

$\forall p \in M$, $\forall \overrightarrow{v_1}, \overrightarrow{v_2} \in T_p(M)$, siendo λ^2 una función diferenciable en M, no nula en cada punto.

Definición 30. Se dice que M es **localmente conforme** a N si $\forall p \in M$, existen U' entorno abierto de p en M, V' abierto en N y $f : U' \to V'$ conforme.

Nota 31.

1. Las aplicaciones conformes son una generalización obvia de las isometrías ($\lambda^2 = 1$).
2. Una aplicación conforme preserva los ángulos.

Teorema 32 (Caracterización de superficies localmente conformes). *Sean M,N sup.reg. Entonces M es localmente conforme a $N \Leftrightarrow \forall p \in M$, existen U abierto en $\mathbb{R}^2$, $\boldsymbol{x} : U \to \mathbb{R}^3$ sup. simple en M con $p \in \boldsymbol{x}(U)$ e $\boldsymbol{y} : U \to \mathbb{R}^3$ sup. simple en N tales que los coeficientes métricos g_{ij} de $\boldsymbol{x}$ y los h_{ij} de $\boldsymbol{y}$ verifican*

$$g_{ij}(u^1, u^2) = \lambda^2(u^1, u^2) h_{ij}(u^1, u^2) \quad \forall (u^1, u^2) \in U,$$

siendo $\lambda > 0$ una función diferenciable en U.

Teorema 33 (Existencia de cartas isotermales). *Dado $p \in M$ cualquiera, existe una superficie simple $\boldsymbol{x} : U \longrightarrow \mathbb{R}$ con $p \in \boldsymbol{x}(U) \subseteq M$ tal que:*

$$g_{11} = g_{22} = \lambda^2, \quad g_{12} = g_{21} = 0.$$

Corolario 34. *Toda sup. reg. M es localmente conforme al plano.*

Definición 35. Una aplicación diferenciable $f : M \longrightarrow N$ se dice **isoárea** si es biyectiva, regular y verifica que tomada cualquier región R sobre M, R y $f(R)$ tienen la misma área.

Definición 36. Se dice que M es **localmente isoárea** a N si $\forall p \in M$, existen U' entorno abierto de p en M, V' abierto en N y $f : U' \to V'$ isoárea .

Recordatorio 37. Sean $\mathbf{x} : U \longrightarrow \mathbb{R}^3$ sup. simple y $R \subseteq \mathbf{x}(U)$. Entonces:

$$\text{Área}(R) = \iint_{\mathbf{x}^{-1}(R)} \sqrt{g_{11} g_{22} - g_{12}^2}\, du^1 du^2.$$

Teorema 38 (Caracterización de superficies localmente isoáreas). *Sean M,N sup. reg. Entonces M es localmente isoárea a $N \Leftrightarrow \forall p \in M$, existen U abierto en $\mathbb{R}^2$, $\boldsymbol{x} : U \to \mathbb{R}^3$ sup. simple en M con $p \in \boldsymbol{x}(U)$ e $\boldsymbol{y} : U \to \mathbb{R}^3$ sup. simple en N tales que los coeficientes métricos g_{ij} de $\boldsymbol{x}$ y los h_{ij} de $\boldsymbol{y}$ verifican que:*

$$g_{11}g_{22} - g_{12}^2 = h_{11}h_{22} - h_{12}^2.$$

Nota 39.

$$\text{localmente isométrica} \Leftrightarrow \left\{ \begin{array}{c} \text{localmente conforme} \\ \text{y} \\ \text{localmente isoárea} \end{array} \right.$$

pero

$$\left. \begin{array}{c} \text{localmente conforme} \\ \text{ó} \\ \text{localmente isoárea} \end{array} \right\} \not\Rightarrow \text{localmente isométrica}$$

1.3. Rigidez de la Esfera

Teorema 40 (Rigidez de la esfera). *Sea $f : \sum \longrightarrow \mathcal{S}$ una isometría de una esfera $\sum$ en una superficie regular $\mathcal{S}$. Entonces $\mathcal{S}$ es una esfera (del mismo radio).*

Teorema 41. *Sea $\mathcal{S}$ una sup. reg., conexa y compacta, con curvatura de Gauss $K = cte$. Entonces $\mathcal{S}$ es una esfera.*

Lema 42. *Sean $\mathcal{S}$ una sup. reg. y $p \in \mathcal{S}$. Si denotamos por $k_1 \geq k_2$ las curvaturas principales y se verifica que*

$$\begin{array}{ll} 1) & K(p) > 0 \ (p \text{ es un punto elíptico}), \\ 2) & \begin{array}{l} k_1 \text{ tiene máx. local en } p, \\ k_2 \text{ tiene mín. local en } p, \end{array} \end{array}$$

entonces p es un punto umbílico.

Lema 43. *Toda sup. compacta tiene al menos un punto elíptico (existe p con $K(p) > 0$).*

Lema 44. *Sean $\mathcal{S}$ sup. reg. y $A \subseteq \mathcal{S}$. Entonces A es sup. reg. $\Leftrightarrow$ A es abierto en $\mathcal{S}$.*

Nota 45. $K \equiv$ cte se utiliza para deducir que k_2 es función decreciente de k_1.

Teorema 46. *Sea $\underbrace{\mathcal{S} \text{ sup. reg., conexa y compacta con } K > 0}_{ovaloide}$ y H cte. Entonces $\mathcal{S}$ es una esfera (i.e., todo ovaloide con H cte es una esfera).*

Teorema 47 (De Cohn-Vossen). *Dos ovaloides isométricos difieren en un movimiento rígido.*

Capítulo 2

Geometría Computacional

2.1. Herramientas Algorítmicas

2.1.1. Definiciones

Definición 48. Se dice que un **algoritmo** es un procedimiento constructivo para la resolución de un problema que consta de: unos *datos de entrada* de naturaleza precisamente definida, una cantidad finita de *instrucciones* ordenadas que aplicadas a los datos de entrada ofrecen, tras una cantidad finita de operaciones, una solución o *datos de salida*.

Definición 49. Un algoritmo se dice que es **correcto** si produce el resultado deseado y lo hace en un número finito de pasos.

El lenguaje de los algoritmos

El lenguaje de los algoritmos en nuestro modelo computacional RAM está formado por:

- Los **datos de entrada**: son variables que se pueden sustituir por números para que el algoritmo actúe sobre ellas. Normalmente se representan por letras.
- Las **operaciones básicas**:
  - *Asignación*: asignar a una variable x un valor a

    $$x := a.$$

  - Las operaciones de *suma, resta multiplicación y división* para números enteros, esto es, $a + b$, $a - b$, $a * b$, a/b.
  - *Comparación* de números enteros representadas por $a < b$, $a = b$, $a > b$, $a \neq b$...
- Las **instrucciones**:
  1. **Instrucción condicional**: "si $\boldsymbol{B}$ entonces $\boldsymbol{C_1}$ y en otro caso $\boldsymbol{C_2}$". Su significado es: si $\boldsymbol{B}$ es *verdadero* se realiza la instrucción $\boldsymbol{C_1}$ y si $\boldsymbol{B}$ es *falso* se efectúa $\boldsymbol{C_2}$.

2. **Instrucción iterativa**: "mientras $\boldsymbol{B}$ haz $\boldsymbol{C}$", que indica al ordenador que efectúe la instrucción $\boldsymbol{C}$ repetidamente mientras la expresión $\boldsymbol{B}$ sea *verdadera*. La repetición cesa cuando $\boldsymbol{B}$ sea *falsa*.
3. "Desde $\boldsymbol{y = r}$ hasta $\boldsymbol{t}$ haz $\boldsymbol{C}$", que implica la ejecución de la instrucción $\boldsymbol{C}$ para cada valor de $\boldsymbol{y}$ entre $\boldsymbol{r}$ y $\boldsymbol{t}$. Es una abreviatura de una instrucción iterativa.

Complejidad de un algoritmo

Sea un algoritmo cuyos datos de entrada tienen tamaño n:

Definición 50. La **complejidad** (en tiempo) de un algoritmo es el número de operaciones $\boldsymbol{T(n)}$ que realiza el algoritmo en el peor de los casos.

Es decir, el valor máximo del número de operaciones realizadas para todas las aplicaciones del algoritmo a entradas de tamaño n.

Sean $\boldsymbol{f, g : \mathbb{N} \to \mathbb{N}}$ dos funciones:

- $\boldsymbol{f(n)}$ es $\boldsymbol{O(g(n))}$ ("$\boldsymbol{f}$ es *O grande* de $\boldsymbol{g}$ ") $\equiv$ existen una constante $\boldsymbol{C > 0}$ y un número natural $\boldsymbol{n_0 > 0}$ tales que
$$\boldsymbol{f(n) < Cg(n)\,,\ \forall n \geq n_0}$$
- $\boldsymbol{f(n)}$ es $\boldsymbol{\Omega(g(n))}$ ("$\boldsymbol{f}$ es *omega* de $\boldsymbol{g}$") $\equiv$ $\boldsymbol{g(n)}$ es $\boldsymbol{O(f(n))}$.
- $\boldsymbol{f(n)}$ es $\boldsymbol{\theta(g(n))}$ ("$\boldsymbol{f}$ es *theta* de $\boldsymbol{g}$") $\equiv \begin{cases} \boldsymbol{f(n)} \text{ es } \boldsymbol{O(g(n))} \\ \text{y} \\ \boldsymbol{f(n)} \text{ es } \boldsymbol{\Omega(g(n))} \end{cases}$

A continuación tenemos la siguiente comparación entre las distintas O grandes:

$\boldsymbol{O(1)}$	orden constante
$\boldsymbol{O(\log n)}$	orden logarítmico
$\boldsymbol{O(n)}$	orden lineal
$\boldsymbol{O(n \log n)}$	
$\boldsymbol{O(n^2)}$	orden cuadrático
$\boldsymbol{O(n^a)}$	orden polinomial ($a > 2$)
$\boldsymbol{O(a^n)}$	orden exponencial ($a > 1$)
$\boldsymbol{O(n!)}$	orden factorial

2.1.2. Algoritmos de ordenación

El problema de *ordenación* es el siguiente:

> "*Dada una lista de números enteros distintos* $\{x_1, x_2, \ldots, x_n\}$, *queremos ordenarla en orden creciente*".

1. **Burbuja**
2. **Selección**
3. **Inserción**

1. Algoritmo de ordenación de Burbuja

El Algoritmo de Ordenación de burbuja consiste en lo siguiente:

- En el primer paso, se compara x_1 con x_2, y se intercambia si $x_1 > x_2$. Después x_2 con x_3, y así sucesivamente hasta x_{n-1} con x_n.
- Al final de la primera pasada, en x_n está el mayor elemento.
- En la segunda pasada, se comparan igual x_1 con x_2, hasta x_{n-2} con x_{n-1}, intercambiando cuando sea necesario. Al final de esta pasada, $\{x_{n-1}, x_n\}$ está ordenado.
- En el último paso, se realiza la comparación de x_1 con x_2, y la lista resultante ya está ordenada.

Desde $i = 1$ hasta $n - 1$ haz
  Desde $j = 1$ hasta $n - i$ haz
    Si $x_j > x_{j+1}$, entonces intercambia x_j y x_{j+1}

2. Algoritmo de ordenación por Selección

El Algoritmo de Ordenación por Selección consiste en lo siguiente:

1. Seleccionar el elemento más pequeño de la lista, e intercambiar con x_1.
2. Seleccionar el elemento más pequeño de la lista $\{x_2, \ldots, x_n\}$, e intercambiar con x_2.
3. En cada paso k del proceso, la lista de los primeros k elementos está ya ordenada.
4. En el último paso, la lista resultante está ordenada.

3. Algoritmo por Inserción

Hay dos tipos de inserción: la secuencial y la bisección

Inserción secuencial

El Algoritmo de Intersección secuencial consiste en lo siguiente:

1. En el primer paso, se comienza con la lista $\{x_1\}$ y se inserta x_2 en el lugar adecuado respecto de $\{x_1\}$.
2. En cada paso k, se inserta x_{k+1} en la lista parcialmente ordenada $\{x_1, \ldots, x_k\}$, colocándolo en su lugar correspondiente.
3. En el último paso $(n-1)$, la lista queda ordenada.

Inserción por bisección

Ordenemos la lista $\{7, 3, 5, 1, 4, 6\}$

$\{7\}$

$3 \hookrightarrow \{7\}:$
$\quad 3 < 7 \Rightarrow \{3\ ,\ 7\}$

$\{3\ ,\ 7\}$

$5 \hookrightarrow \{3 \mid 7\}:$
$\quad 5 < 7 \Rightarrow 5 \hookrightarrow \{3\}:$
$\qquad 5 > 3 \Rightarrow \{3\ ,\ 5\}$

$\{3\ ,\ 5\ ,\ 7\}$

$1 \hookrightarrow \{3\ ,\ 5 \mid 7\}:$
$\quad 1 < 7 \Rightarrow 1 \hookrightarrow \{3 \mid 5\}:$
$\qquad 1 < 5 \Rightarrow 1 \hookrightarrow \{3\}:$
$\qquad\quad 1 < 3 \Rightarrow \{1\ ,\ 3\}$

$\{1\ ,3\ ,\ 5\ ,\ 7\}$

$4 \hookrightarrow \{1\ ,3 \mid 5\ ,\ 7\}:$
$\quad 4 < 5 \Rightarrow 4 \hookrightarrow \{1 \mid 3\}:$
$\qquad 4 > 3 \Rightarrow 4 \hookrightarrow \{3\}:$
$\qquad\quad 4 > 3 \Rightarrow \{3\ ,\ 4\}$

$\{1\ ,3\ ,\ 4\ ,\ 5\ ,\ 7\}$

$6 \hookrightarrow \{1\ ,3\ ,\ 4 \mid 5\ ,\ 7\}:$
$\quad 6 > 5 \Rightarrow 6 \hookrightarrow \{5 \mid 7\}:$
$\qquad 6 < 7 \Rightarrow 6 \hookrightarrow \{5\}:$
$\qquad\quad 6 > 5 \Rightarrow \{5\ ,\ 6\}$

$\{1\ ,3\ ,\ 4\ ,\ 5\ ,\ 6\ ,\ 7\}$

Complejidad de los algoritmos de ordenación

Algoritmo de ordenación de Burbuja:

- *Comparaciones*: Para cada $i \in \{1, \ldots, n-1\}$ se realizan $n - i$ comparaciones

$$\boldsymbol{(n-1) + (n-2) + \cdots + 2 + 1 = \frac{n(n-1)}{2}} \text{ es } \boldsymbol{O(n^2)}$$

- *Intercambios*: $\boldsymbol{O(n^2)}$ pues, en el peor de los casos, después de cada comparación habrá un intercambio.

Algoritmo de ordenación por selección:

Para cada $i \in \{1, \ldots, n-1\}$ se ejecuta el algoritmo de cálculo del mínimo (máximo) cuya complejidad es lineal. Por tanto el orden de este algoritmo es cuadrático $\boldsymbol{O(n^2)}$.

Algoritmo de ordenación por inserción (secuencial):

- *Comparaciones*: Para cada $i \in \{1, \ldots, n-1\}$ se compara x_i con cada elemento de la lista parcial que tiene $n - i$ elementos

$$\boldsymbol{1 + 2 + \cdots + (n-2) + (n-1)} \text{ es } \boldsymbol{O(n^2)}$$

- *Intercambios*: $\boldsymbol{O(n^2)}$ pues, en el peor de los casos, para insertar x_i hay que desplazar los $i - 1$ números de la lista parcial.

Algoritmo de ordenación por inserción (bisección):

- *Comparaciones*:

$$\boldsymbol{\log_2 2 + \log_2 3 + \cdots + \log_2 n \leq (n-1) \log_2 n} \text{ es } \boldsymbol{O(n \log n)}$$

- *Intercambios*: $\boldsymbol{O(n^2)}$ igual que en la inserción secuencial

2.1.3. Algoritmo Voraz

El esquema algorítmico conocido como *algoritmos voraces* (también *ávidos*; *greedy* en inglés) es el que menos dificultades plantea a la hora de diseñar y comprobar su funcionamiento. Normalmente, este esquema se aplica a los problemas de optimización.

Lo aplicaremos a dos conocidos problemas en Geometría Computacional:

1. Coloración de Vértices.
2. Árbol Spanning Minimal

Funcionamiento:

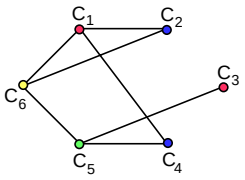
Si llamamos $\boldsymbol{C}$ al conjunto de candidatos posibles, y $\boldsymbol{S}$ al conjunto solución deseado, el algoritmo escoge en cada paso al mejor elemento de $\boldsymbol{C}$ posible, llamémoslo $\boldsymbol{a}$. Acto seguido comprueba si al añadirle a $\boldsymbol{S}$ dicho elemento $\boldsymbol{a}$, el conjunto resultante verifica la condición deseada; en caso contrario lo elimina. Continúa el algoritmo hasta que $\boldsymbol{S}$ es la solución buscada.

```
AlgoritmoVoraz(C: ConjuntoCandidatos)
S := ∅;
Mientras (C ≠ ∅ y S ≠SolucionDeseada) haz
    a:=SeleccionaCandidato(C);
    Si EsFactible(a ∪ S) entonces
        S := S ∪ a;
        Si EsSolucion(S)
            S:=SolucionDeseada;
```

Coloración de vértices

Supongamos que organizamos un congreso con **6** conferencias. Cada uno de los participantes ha seleccionado aquellas conferencias a las que le gustaría asistir. El problema consiste en hacer un horario de forma que todos los participantes puedan asistir a las conferencias que han elegido.

Este problema podemos representarlo mediante un grafo, donde los vértices son las conferencias y las aristas unen aquellas conferencias a las que quieren ir las mismas personas. Para resolver el problema tenemos que asignar sesiones distintas a vértices adyacentes.



Sesión 1	Sesión 2	Sesión 3	Sesión 4
$\boldsymbol{C_1}$ y $\boldsymbol{C_3}$	$\boldsymbol{C_2}$ y $\boldsymbol{C_4}$	$\boldsymbol{C_5}$	$\boldsymbol{C_6}$

Los problemas que pueden resolverse mediante coloración de vértices tienen la siguiente estructura:

"*Dado el grafo* $\boldsymbol{G = (V, A)}$, *se busca una partición de* $\boldsymbol{V}$ *en subconjuntos* $\boldsymbol{V_i}$; *es decir,*

$$\boldsymbol{V = \bigsqcup_{i=1}^{n} V_i},$$

de modo que los vértices de cada $\boldsymbol{V_i}$ *no sean adyacentes y minimizando el tamaño de la partición* (n)."

Definición 51. Una Coloración de Vértices de un grafo $G = (V, A)$ es una función $c : V \to \mathbb{N}$ tal que $xy \in A \Leftrightarrow c(x) \neq c(y)$

Definición 52. Si $k = \text{máx}\{c(v)/v \in V\}$, decimos que G puede colorearse con k colores.

Definición 53. El Número Cromático de G es:

$$\chi(G) = \mathbf{min}\{k \mid G \text{ puede colorearse con } k \text{ colores}\}$$

Sabemos el número cromático de algunos grafos elementales:

- Grafo completo K_n: $\chi(K_n) = n$
- Ciclos C_n: $\chi(C_{2p}) = 2$, $\chi(C_{2p+1}) = 3$
- Árboles T con, al menos, dos vértices: $\chi(T) = 2$

Algoritmo voraz aplicado a la coloración de un grafo:

Este algoritmo, aplicado a la coloración de un grafo, consiste en asignar a cada vértice el primer color que no ha sido asignado a sus vecinos. Si ordenamos los vértices del grafo como $v_1, \ldots, v_n$:

1. Asignamos el primer color a v_1.
2. Para los siguientes vértices v_i, $(i = 2 \cdots n)$ formamos el conjunto S de colores asignados a los vértices anteriores que son adyacentes a v_i
3. Damos a v_i el primer color que no está en S

Veamos dicho algoritmo:

$c(v_1) := 1$;
Desde $i = 2$ hasta n haz
    $S := \emptyset$
    Desde $j = 1$ hasta $i - 1$ haz
        Si v_j es adyacente a v_i entonces $S := S \cup \{c(v_j)\}$
    $k := 1$
    Mientras $k \in S$ haz $k := k + 1$
    $c(v_i) := k$

Complejidad del algoritmo voraz

En el paso i tenemos las operaciones:

- Adyacencias: $i - 1$
- Pertenencias: $i - 1$
- Sumas: $i - 1$

- Asignaciones: $\mathbf{1 + (i - 1) + 1 + (i - 1) + 1}$

En total tenemos:

- Adyacencias: $\mathbf{1 + 2 + \cdots + (n - 2) + (n - 1)} = \frac{\mathbf{n(n-1)}}{\mathbf{2}}$ es $\boldsymbol{O(n^2)}$
- Pertenencias: $\mathbf{1 + 2 + \cdots + (n - 2) + (n - 1)}$ es $\boldsymbol{O(n^2)}$
- Sumas: $\mathbf{1 + 2 + \cdots + (n - 2) + (n - 1)}$ es $\boldsymbol{O(n^2)}$
- Asignaciones: $\mathbf{1 + 3(n - 1) + 2\,(1 + 2 + \cdots + (n - 2) + (n - 1)) = 1 + 3n - 3 + n(n - 1) = n^2 + 2n - 2}$ es $\boldsymbol{O(n^2)}$

Árbol Spanning Minimal (A.S.M.)

Supongamos que queremos construir carreteras entre varias ciudades. Formalmente tenemos un grafo $\boldsymbol{G = (V, A)}$ cuyos vértices son las ciudades y las aristas las carreteras, y una función $\boldsymbol{w : A \to \mathbb{N}}$ de forma que $\boldsymbol{w(a)}$ representa el coste de la carretera $\boldsymbol{a}$. Decimos que $\boldsymbol{G}$ y $\boldsymbol{w}$ constituyen un grafo ponderado y que $\boldsymbol{w}$ es una función de peso.

El objetivo práctico es obtener una red de carreteras de coste mínimo, que se corresponde con el problema de obtener un árbol spanning $\boldsymbol{T = (V, A')}$ de $\boldsymbol{G}$ cuyo peso total

$$\boldsymbol{w(T) = \sum_{a \in A'} w(a)}$$

sea el menor posible. Este problema tiene solución, sin embargo, pueden existir varios árboles spanning distintos cuyo peso total sea mínimo.

Algoritmo voraz aplicado a un Árbol Spanning Minimal:

1. Primero tomamos la arista de menor peso.
2. Después, en cada paso, se añade la arista de menor peso que añada un nuevo vértice al árbol parcial (si hay varias aristas posibles con el mismo peso, se elige cualquiera de ellas).
3. Así continuamos el proceso hasta completar los vértices del grafo de partida y teniendo cuidado de que no se formen ciclos para que el resultado final sea un árbol.

Teorema 54. *Sea $\boldsymbol{G = (V, A)}$ un grafo ponderado conexo con función de peso $\boldsymbol{w : A \to \mathbb{N}}$. Supongamos que $\boldsymbol{T}$ es un árbol spanning de $\boldsymbol{G}$ que resulta de aplicarle el algoritmo voraz. Entonces se verifica:*

$$\boldsymbol{w(T) \leq w(U)}$$

para cualquier otro árbol $\boldsymbol{U}$ spanning de $\boldsymbol{G}$.

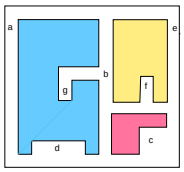
2.1.4. Algoritmos de búsqueda

Supongamos que hemos de llevar a cabo la búsqueda de los vértices de un grafo empezando en un vértice concreto. Podemos emplear dos estrategias distintas:

- podríamos "profundizar", moviéndonos a un nuevo vértice siempre que fuera posible, o
- podríamos "desplegarnos" comprobando todos los vértices en un "nivel" antes de pasar al siguiente.

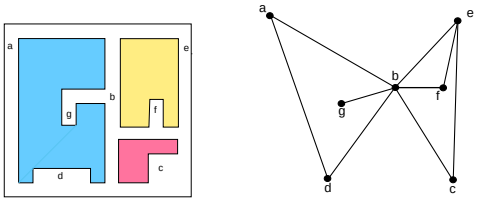
La primera estrategia se conoce con el nombre de **búsqueda en profundidad** (B.E.P.) y la segunda con el nombre de **búsqueda en anchura** (B.E.A.).

La siguiente figura representa una habitación con siete juguetes escondidos, el problema consiste en encontrar los juguetes representados por las letras ***a***, ***b***, ***c***, ***d***, ***e***, ***f*** y ***g***.



Podemos representar esta situación mediante un grafo, donde los vértices serán los juguetes y las aristas representan las rutas disponibles entre los juguetes.

Para el ejemplo dado tendríamos el siguiente grafo:



Algoritmo Búsqueda en Profundidad (B.E.P.):

$\boldsymbol{Q} := \boldsymbol{v}$
Mientras $\boldsymbol{Q} \neq \emptyset$ haz
    $\boldsymbol{x} :=$ Ultimo($\boldsymbol{Q}$)
    Si $\boldsymbol{x}$ es adyacente a un nuevo vértice $\boldsymbol{y}$
        entonces $\boldsymbol{Q} := (\boldsymbol{Q}$ seguida de $\boldsymbol{y})$
        y en otro caso $\boldsymbol{Q} := (\boldsymbol{Q}$ con $\boldsymbol{x}$ eliminado)

B.E.P. $\rightarrow$ D.F.S. = Depth First Search

Algoritmo Búsqueda en Anchura (B.E.A.):

$\boldsymbol{Q} := \boldsymbol{v}$
Mientras $\boldsymbol{Q} \neq \emptyset$ haz
    $\boldsymbol{x} :=$ Primero($\boldsymbol{Q}$)
    Si $\boldsymbol{x}$ es adyacente a un nuevo vértice $\boldsymbol{y}$
        entonces $\boldsymbol{Q} := (\boldsymbol{Q}$ seguida de $\boldsymbol{y})$
        y en otro caso $\boldsymbol{Q} := (\boldsymbol{Q}$ con $\boldsymbol{x}$ eliminado)

B.E.A. $\rightarrow$ B.F.S. = Breadth First Search En la B.E.P. la lista de vértices se conoce con el nombre de pila, ya que se comporta como una pila de platos, sólo se pueden añadir o quitar vértices (platos) en la cima de la pila.

La B.E.A. se comporta como una cola, ya que recuerda a una cola de personas esperando en una tienda a ser atendidos. Los nuevos vértices se añaden al final de la cola y los antiguos se eliminan del principio.

- B.E.P. $\rightarrow$ LIFO=Last in, first out (la última que llega es la primera que sale)
- B.E.A. $\rightarrow$ FIFO=First in, first out (la primera que llega es la primera que sale)

Los procedimientos de búsqueda pueden utilizarse para ver si un grafo $\boldsymbol{G}$ es conexo o no. Ya que si al final hemos pasado por todos los vértices de $\boldsymbol{G}$, es decir, hemos obtenido un árbol spanning, entonces $\boldsymbol{G}$ es conexo.

2.2. Cierre Convexo de una nube de puntos. Algoritmos.

2.2.1. Cierre Convexo (Convex Hull).

Definición 55. Un conjunto $A \subseteq \mathbb{R}^d$ se dice **convexo** si $\overline{xy} \subseteq A$, $\forall x, y \in A$.

Proposición 56. *La intersección de una familia de conjuntos convexos es un conjunto convexo.*

Definición 57. El **cierre convexo** de un conjunto S es el menor convexo que contiene a S. Lo notaremos: **CH(S)**.

Proposición 58. *$CH(S)$ es la intersección de todos los conjuntos convexos que contienen a S.*

Nota 59. A continuación, veamos el cierre convexo para:

$$S \ = \ \text{nube de } n \text{ puntos en } \mathbb{R}^2$$

Proposición 60. *Si P es un polígono convexo cuyos vértices son puntos de S, entonces $P \subseteq CH(S)$.*

Proposición 61. *Sea P un polígono convexo que contiene a S y cuyos vértices son puntos de S. Entonces $P = CH(S)$. En particular, P es único.*

Proposición 62. *La **envolvente convexa** de S es la frontera de $CH(S)$.*

Definición 63. Dados un conjunto $A \subseteq \mathbb{R}^2$ y un punto $p \in A$, se dice que una recta que pasa por p es **soporte** de A si todos los puntos de A quedan en uno de los semiplanos cerrados determinados por dicha recta.

Teorema 64. *Sea P un polígono cuyos vértices están ordenados con orientación positiva. Son equivalentes:*

1. *P es convexo.*
2. *Toda recta que pase por dos vértices consecutivos deja a P en el semiplano de la izquierda.*
3. *Toda recta que pase por dos vértices consecutivos deja al siguiente vértice de P en el semiplano de la izquierda.*
4. *Todo ángulo interior en cada vértice de P es menor o igual que π.*
5. *Todo vértice admite una recta soporte de P que pasa por él.*

Área signada de un triángulo

Dados tres puntos

$$p_1 = (x_1, y_1), \quad p_2 = (x_2, y_2), \quad p_3 = (x_3, y_3)$$

en el plano, el área del triángulo que determinan viene dada por

$$\text{Área} \ = \ \frac{1}{2}\text{abs}\left(\Delta(p_1, p_2, p_3)\right),$$

siendo

$$\Delta(p_1, p_2, p_3) = \begin{vmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{vmatrix}$$

el **área signada**. Su signo coincide con el del vector $\overrightarrow{p_1p_2} \wedge \overrightarrow{p_1p_3}$ y sirve para orientar el triángulo de vértices $p_1p_2p_3$:

- Si $\Delta(p_1, p_2, p_3) > 0$, entonces (p_1, p_2, p_3) tiene orientación positiva.
- Si $\Delta(p_1, p_2, p_3) < 0$, entonces (p_1, p_2, p_3) tiene orientación negativa.

2.2.2. Algoritmos.

Algoritmo de *fuerza bruta*

Descripción:

- **Paso 1:** Para cada par de puntos de S se comprueba si todos los demás están en un mismo semiplano de los dos que determina la recta que pasa por ellos. Si es así, determinan un lado de la envolvente y si no, no.
- **Paso 2:** Se ordenan los lados para que aparezcan tal y como lo hacen al recorrer la frontera.

Complejidad:

- **Paso 1:** $O(n)$ para cada par de puntos. Como hay $\binom{n}{2}$ pares, $O(n^3)$ para encontrar el conjunto de vértices de la envolvente convexa.
- **Paso 2:** $O(n \log n)$ en el peor de los casos.
- **Total:** $O(n^3)$.

Marcha de Jarvis

Idea: Partiendo de un vértice de la envolvente, se busca otro vértice que con él determine una recta soporte de S.

Descripción:

- **Entrada:** $S = (p_1, \ldots, p_n)$.
- **Salida:** Lista cíclica con los vértices de la envolvente convexa de S: $C = (v_1, \ldots, v_k)$.
- **Paso 0:** Encontrar el punto de S con abscisa mínima (si hubiera más de uno, el de menor ordenada): será v_1 que insertaremos en C, inicialmente vacía.
- **Paso i:** Localizar en S el punto de menor ángulo positivo respecto a v_i y al rayo r_i que emana de v_i con la dirección $\overrightarrow{v_{i-1}v_i}$, si $i \neq 1$, y si $i = 1$, al rayo vertical r_1 que emana de v_1 en la dirección de $-\infty$ (si hubiera más de uno, el que esté más alejado de v_i). Ese punto será v_{i+1}. Si $v_{i+1} \neq v_1$, lo insertamos en C y continuamos. En caso contrario, hemos acabado.

Corrección:

- C es la frontera de un polígono convexo con vértices en S y que contiene a S.

Complejidad:

- **Paso 0:** Puede hacerse en tiempo $O(n)$.
- **Paso i:** Para cada i, puede hacerse en tiempo $O(n)$.
- **Total**: $O(n \cdot k)$, donde k es el número de vértices en la envolvente convexa. $O(n^2)$ en el peor de los casos.

Observaciones:

- El cálculo para encontrar el mínimo angular puede hacerse usando áreas signadas.

$$\angle(\overrightarrow{v_i p_j}, r_i) < \angle(\overrightarrow{v_i p_k}, r_i) \Leftrightarrow \Delta(v_i, p_j, p_k) > 0$$

Scan de Graham

- Un polígono convexo es estrellado. Por tanto, los vértices aparecen ordenados angularmente respecto a cualquier punto del núcleo.
- Un polígono convexo es monótono respecto a cualquier dirección. Por tanto, su frontera puede descomponerse en dos cadenas monótonas respecto de OX. En cada una, los vértices están ordenados por abscisas.
- Todos los vértices de un polígono convexo son convexos.

Idea:

1. **Ordenación:** Obtener una lista cíclica L con todos los puntos de S de manera que describa un polígono estrellado o monótono.
2. **Scan:** Quitar de L los puntos que no formen con sus dos adyacentes un ángulo menor que π.

Descripción:
Suponemos que en S no hay más de dos puntos alineados.

- **Entrada:** $S = (p_1, \ldots, p_n)$.
- **Salida:** Lista cíclica con los vértices de la envolvente convexa de S: $C = (v_1, \ldots, v_k)$.
- **Paso 1:** Encontrar el punto de S con abscisa mínima (si hubiera más de uno, el de menor ordenada): será v_1 que insertaremos en C, inicialmente vacía.

- **Paso 2:** Ordenar angularmente el resto de puntos de S respecto a v_1 y en sentido positivo respecto al rayo vertical que emana de él en dirección a $-\infty$. Sea $L = (v_1, \ldots, v_n)$ el resultado.
- **Paso 3:** Insertar v_2 en C. Hacer $q = v_2$.
- **Paso 4:** (Scan) Desde $i = 3$ hasta n, hacer
  1. $p = ant(q)$ (anterior en C).
  2. Mientras $\Delta(p, q, v_i) \leq 0$,
     a) Borrar q de C.
     b) Actualizar $q = p$ y $p = ant(q)$.
  3. Insertar v_i en C y actualizar $q = v_i$.
- Obsérvese que con la ordenación del Paso 2, tanto v_2 como v_n son vértices de la envolvente.

Nota:
Si en S hay más de dos puntos alineados, basta modificar el algoritmo de modo que en el paso de ordenación angular, cuando haya varios puntos con el mismo ángulo, nos quedemos sólo con el más alejado de v_1.

Corrección:

- Los puntos que elimina el algoritmo no son vértices de la envolvente convexa.
- Al finalizar el algoritmo, C es un polígono convexo, pues todos sus vértices son convexos. Además, sus vértices son puntos de S. Para probar que $C = CH(S)$, basta ver que $S \subseteq C$.

Complejidad:

- **Paso 1:** Puede hacerse en tiempo $O(n)$.
- **Paso 2:** Puede hacerse en tiempo $O(n \log n)$.
- **Scan:** El coste es $O(n)$ (el número máximo de test de áreas signadas es $2n - 4$).
- **Total:** $O(n \log n)$.

Algoritmos: Scan de Graham - Variante

- **Entrada:** $S = (p_1, \ldots, p_n)$.
- **Salida:** Lista cíclica con los vértices de la envolvente convexa de S: $C = (v_1, \ldots, v_k)$.
- **Paso 1:** Ordenar lexicográficamente los puntos de S obteniendo la lista L.

- **Paso 2:** Aplicar el scan a L y obtener C_1, que es cadena inferior de la envolvente convexa.
- **Paso 3:** Invertir el orden en L obteniendo L'. Aplicar el scan a L' y obtener C_2, que es cadena superior de la envolvente convexa.
- **Paso 4:** Concatenar C_1 y C_2 eliminando las repeticiones de los extremos para obtener la lista C.

Algoritmos: Scan de Graham - Variante de Akl-Toussaint

- Antes de ordenar, localizar los puntos extremales (abscisas y ordenadas mínimas y máximas). Estos puntos determinan un convexo contenido en $CH(S)$.
- Eliminar los puntos de S que están en el interior de dicho convexo.
- Con los que quedan, se continúa como en el algoritmo anterior, actuando sobre listas parciales.
- El coste del paso adicional es $O(n)$, por lo que la complejidad sigue siendo $O(n \log n)$.

2.3. Cálculo del diámetro de una nube de puntos.

2.3.1. Diámetro de una nube de puntos

Definición 65. El **diámetro** de una nube de n puntos es la mayor distancia entre dos de ellos. Es decir, si $S = \{p_1, \ldots, p_n\}$,

$$diam(S) \ = \ \max_{1 \leq i,j \leq n} d(p_i, p_j)$$

Algoritmo de "fuerza bruta":

1. Para cada par de puntos p_i, p_j, hallar $d(p_i, p_j)$.
2. Tomar el mayor de los valores obtenidos.

Complejidad: $O(n^2)$.

Proposición 66. *El diámetro de S se alcanza en vértices de $CH(S)$.*

2.3.2. Diámetro de un polígono convexo

Sea P un polígono convexo de n vértices.

$$diam(P) \ = \ \max\{d(p,q) \ / \ p, q \in P\}$$

Proposición 67. $diam(P) \ = \ diam(\{\textit{vértices de } P\})$.

Definición 68. Se dice que dos vértices p, q de P son **antípodas** si existen dos rectas paralelas distintas que pasan por esos puntos y son soporte de P.

Proposición 69. *Si $diam(P) = d(p,q)$, entonces p y q son antípodas.*

Algoritmo:

1. Hallar todos los pares de antípodas de P.
2. Hallar la distancia entre los puntos de cada par.
3. Tomar el mayor de los valores obtenidos.

2.3.3. Algoritmo de girar calibres

Nota 70. La comparación entre los ángulos α y β se puede hacer usando áreas signadas

$$\alpha < \beta \Leftrightarrow \Delta(p, SIG(p), x) < 0$$

$$x = p + \overrightarrow{qSIG(q)}$$

- **Entrada:** Lista cíclica P con los n vértices de un polígono convexo en orientación positiva.
- **Salida:** Lista Q con las parejas de antípodas de P.
- **Paso 1:** Hacer $Q := \emptyset$.
- **Paso 2:** Localizar en P los puntos mínimo y máximo con orden lexicográfico: x_{min} y x_{max}.
- **Paso 3:** Hacer $p := x_{min}$ y $q := x_{max}$.
- **Paso 4:** Mientras $q \neq x_{min}$ o $p \neq x_{max}$ repetir:
  - Introducir en Q el par (p,q).
  - Hacer $x := p + \overrightarrow{qSIG(q)} = p + SIG(q) - q$.
  - Si $\Delta(p, SIG(p), x) < 0$, hacer $p := SIG(p)$
  - Si $\Delta(p, SIG(p), x) > 0$, hacer $q := SIG(q)$
  - Si $\Delta(p, SIG(p), x) = 0$, añadir a Q los pares $(p, SIG(q))$ y $(SIG(p), q)$ y hacer $p := SIG(p)$ y $q := SIG(q)$.
- **Paso 5:** Acabar devolviendo Q.

Corrección: El algoritmo acaba.
Complejidad: $O(n)$.

2.4. Problemas de proximidad

Dada una nube S con n puntos, se plantean los siguientes problemas:

1. Fijado un punto de S, encontrar la región del plano formada por los puntos que están más próximos a él que a cualquier otro de S.
2. **Todos los vecinos más cercanos:** Para cada punto de S encontrar cuál es su punto más cercano.
3. **El par más cercano:** Encontrar el par de puntos de S cuya distancia entre sí sea mínima.
4. **El vecino más cercano:** Dado un nuevo punto p, encontrar el elemento de S más cercano a p.
5. **Mayor círculo vacío:** Encontrar el mayor círculo que no contenga ningún punto de S y cuyo centro esté dentro de $CH(S)$.
6. **Triangulación:** Encontrar una triangulación que contenga a los puntos de S como vértices.
7. **Árbol spanning mínimo:** Construir un árbol de longitud mínima que tenga a los puntos de S como vértices.

2.4.1. Diagrama de Voronoi

$$S = \{p_1, \ldots, p_n\} \subset \mathbb{R}^2$$

Definición 71. Se llama **región**, **célula** o **polígono de Voronoi** de p_i al conjunto

$$Vor(p_i) = \{q \in \mathbb{R}^2 | d(q, p_i) \leq d(q, p_j), \forall j \neq i\}$$

.

Denotamos por $h(p_i, p_j)$ al semiplano que contiene a p_i y tiene por borde la mediatriz m_{ij} del segmento $\overline{p_i p_j}$.

Lema 72. *Se verifica:*

$$Vor(p_i) = \bigcap_{j \neg i} h(p_i, p_j)$$

Definición 73. Se llama **diagrama de Voronoi** de S, $Vor(S)$, a la teselación del plano determinada por la familia $\{Vor(p_i)\}_{i=1,\ldots,n}$. Los puntos p_i se llaman **generadores** del diagrama, y los vértices y segmentos o semirrectas de $Vor(S)$ se llaman **vértices** y **aristas de Voronoi** de S, respectivamente.

Nota 74.

- El diagrama de Voronoi constituye una partición del plano en polígonos convexos (los polígonos de Voronoi), tal que cada punto de S pertenece a un único polígono.
- Cada arista de $Vor(S)$ es común a dos polígonos y es perpendicular al segmento que une los puntos de S que están en estos polígonos.

- $Vor(S)$ contiene toda la información de proximidad relativa a S.
- El diagrama de Voronoi se puede definir en dimensiones superiores.

Además, suponiendo que los puntos de S están en posición general:

- Los puntos de S no están todos alineados.
- No hay cuatro puntos de S que sean cocirculares.

Teorema 75. *Cada vértice del diagrama de Voronoi es la intersección de tres aristas del diagrama.*

Nota 76. Los vértices de Voronoi son los centros de circunferencias definidas por tres puntos de S.

Teorema 77. *Para cada vértice v de $Vor(S)$, la circunferencia correspondiente no contiene a ningún otro punto de S.*

Teorema 78. *Para cada $p_i \in S$, cada vecino más próximo a p_i define una arista en el polígono $Vor(p_i)$.*

Teorema 79. *El polígono $Vor(p_i)$ es no acotado si y sólo si p_i es un punto de la envolvente convexa de S.*

Corolario 80. *Las semirrectas de $Vor(S)$ corresponden a pares de puntos adyacentes de la envolvente convexa de S.*

2.4.2. Triangulación de Delaunay

Definición 81. El grafo dual de $Vor(S)$ recibe el nombre de **grafo de Delaunay** de S.

Teorema 82. *Si los puntos de S están en posición general, el grafo dual de $Vor(S)$ es una triangulación de S, llamada **triangulación de Delaunay** de S.*

Solución de los problemas de proximidad

- Fijado un punto de S, encontrar la región del plano formada por los puntos que están más próximos a él que a cualquier otro de S.

Sol.: Dado $p_i \in S$, dicha región es $Vor(p_i)$.

- **Todos los vecinos más cercanos:** Para cada punto de S encontrar cuál es su punto más cercano.

Sol.: Los vecinos más próximos a p_i definen las aristas de $Vor(p_i)$, por lo que bastará conocer dichas aristas.

- **El par más cercano:** Encontrar el par de puntos de S cuya distancia entre sí sea mínima.

Sol.: Se resuelve en tiempo lineal una vez conocida la solución del problema anterior.

- **El vecino más cercano:** Dado un nuevo punto p, encontrar el elemento de S más cercano a p.

Sol.: Basta determinar en qué región de $Vor(S)$ se encuentra p.

- **Mayor círculo vacío:** Encontrar el mayor círculo que no contenga ningún punto de S y cuyo centro esté dentro de $CH(S)$.

Sol.: Se define la función $f(x, y)$ como la distancia del punto (x, y) a su punto más cercano de S y se prueba que f (restringida a $CH(S)$) alcanza su máximo en un vértice de uno de los polígonos que se obtienen al hallar $Vor(S) \cap CH(S)$.

- **Triangulación:** Encontrar una triangulación que contenga a los puntos de S como vértices.

Sol.: Considerar la Triangulación de Delaunay.

- **Árbol spanning mínimo:** Construir un árbol de longitud mínima que tenga a los puntos de S como vértices.

Sol.: Puede obtenerse a partir de la Triangulación de Delaunay, con el siguiente procedimiento: elegido un árbol T, localizar la arista (u', v') tal que $d(u', v') = \min\{d(u, v) \ / \ u \in T, \ v \in S \setminus T\}$.

2.4.3. Algoritmos

1. Divide y vencerás:

- **Paso 1:** Dividir el conjunto S en dos subconjuntos S_1 y S_2 (con igual número de puntos, aproximadamente) separados por una recta vertical, de manera que S_1 quede a la izquierda de dicha recta.
- **Paso 2:** Construir $Vor(S_1)$ y $Vor(S_2)$.
- **Paso 3:** Construir la cadena poligonal σ que separa S_1 y S_2.
- **Paso 4:** Obtener $Vor(S)$ como

$$(Vor(S_1) \cap \pi_L) \cup (Vor(S_2) \cap \pi_R)$$

donde π_L (resp. π_R) es la región del plano que queda a la izquierda (resp. derecha) de σ.

Para detalles sobre el Paso 3, la corrección y la complejidad de este algoritmo, véase: F. P. Preparata, M. I. Shamos. *Computational Geometry. An Introduction.* Springer-Verlag, New York, 1985. Páginas 205-214.

2. Fortune:

Véase: M. Berg et al. *Computational Geometry: Algorithms and Applications.* Springer-Verlag, Berlin, 2000.

Capítulo 3

Modelado Geométrico

3.1. Introducción al Modelado

3.1.1. Curvas Paramétricas

Definición 83. Una curva parametrizada regular de clase C^k es una aplicación

$$\begin{array}{rccc} \alpha: & I & \longrightarrow & \mathbb{R}^3 \\ & t & \mapsto & \alpha(t) = (x(t), y(t), z(t)) \end{array}$$

donde $I \subseteq \mathbb{R}$ es un intervalo abierto ($I = \mathbb{R}$ es posible), que verifica las siguientes condiciones:

1. α es diferenciable de clase C^k para algún $k \geq 1$;
2. $\alpha'(t) = (x'(t), y'(t), z'(t)) \neq 0,\ \forall t \in I$.

El vector $\alpha'(t)$ se denomina vector velocidad o vector tangente de α en el instante t.

Triedro De Frenet:

$\alpha(t)$	$\alpha(s)$ s=p.n.
$\mathbf{t}(t) = \dfrac{\alpha'(t)}{\| \alpha'(t) \|}$	$\mathbf{t}(s) = \dot{\alpha}(s)$
$\mathbf{b}(t) = \dfrac{\alpha'(t) \times \alpha''(t)}{\| \alpha'(t) \times \alpha''(t) \|}$	$\mathbf{n}(s) = \dfrac{\ddot{\alpha}(s)}{\| \ddot{\alpha}(s) \|}$
$\mathbf{n}(t) = \mathbf{b}(t) \times \mathbf{t}(t)$	$\mathbf{b}(s) = \mathbf{t}(s) \times \mathbf{n}(s)$

Ecuaciones de Frenet-Serret:

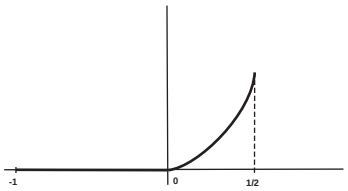
$$\left\{ \begin{array}{lllll} \dot{\mathbf{t}} = & & \kappa\mathbf{n} & \\ \dot{\mathbf{n}} = & -\kappa\mathbf{t} & & +\tau\mathbf{b} \\ \dot{\mathbf{b}} = & & -\tau\mathbf{n} & \end{array} \right\} \sim \begin{pmatrix} \dot{\mathbf{t}} \\ \dot{\mathbf{n}} \\ \dot{\mathbf{b}} \end{pmatrix} = \begin{pmatrix} 0 & \kappa & 0 \\ -\kappa & 0 & \tau \\ 0 & -\tau & 0 \end{pmatrix} \begin{pmatrix} \mathbf{t} \\ \mathbf{n} \\ \mathbf{b} \end{pmatrix}$$

Curvatura y Torsión:

$\alpha(t)$	$\alpha(s)$ $\quad s$=p.n.
$\kappa(t) = \dfrac{\parallel \alpha'(t) \times \alpha''(t) \parallel}{\parallel \alpha'(t) \parallel^3}$	$\kappa(s) = \parallel \ddot{\alpha}(s) \parallel$
$\tau(t) = \dfrac{(\alpha'(t), \alpha''(t), \alpha'''(t))}{\parallel \alpha'(t) \times \alpha''(t) \parallel^2}$	$\tau(s) = \dfrac{(\dot{\alpha}(s), \ddot{\alpha}(s), \dddot{\alpha}\ (s))}{\parallel \ddot{\alpha}(s) \parallel^2}$

Continuidad Geométrica:

$$\alpha(t) = \begin{cases} (t,0,0) & si \quad t \in [-1,0] \\ (2t, 4t^2, 0) & si \quad t \in (0, \frac{1}{2}] \end{cases}$$



$$\alpha'(t) = \begin{cases} (1,0,0) & si \quad t \in [-1,0] \\ (2,8t,0) & si \quad t \in (0, \frac{1}{2}] \end{cases} \qquad \alpha'(0^-) \neq \alpha'(0^+)$$

Por tanto

$$\alpha \in \mathcal{C}^0([-1, \frac{1}{2}]), \quad \alpha \notin \mathcal{C}^1([-1, \frac{1}{2}])$$

Sin embargo la dirección del vector tangente si se desplaza de forma continua a lo largo de la curva.

$$\mathbf{t}(t) = \tfrac{\alpha'(t)}{\|\alpha'(t)\|} = \begin{cases} (1,0,0) & si \quad t \in [-1,0] \\ \dfrac{1}{\sqrt{1+16t^2}}(1,4t,0) & si \quad t \in (0, \frac{1}{2}] \end{cases}$$

Entonces, $\mathbf{t}(t)$ es continua: $\mathbf{t}(0^-) = \mathbf{t}(0^+)$

Definición 84. Una curva continua $\alpha(t)$ para la que la dirección del vector tangente, es decir $\mathbf{t}(t)$, es continua se dice que posee continuidad visual o geométrica de clase $\mathcal{G}^1$.

Consecuencias:

- $\mathcal{C}^1 \Rightarrow \mathcal{G}^1, \qquad \mathcal{G}^1 \not\Rightarrow \mathcal{C}^1$. (Aunque es posible reparametrizar naturalmente la curva para obtener una parametrización $\mathcal{C}^1$)
- $\mathcal{G}^1$ es una propiedad intrínseca de la geometría de la curva.

Definición 85. Una curva $\alpha(t)$ se dice que posee continuidad geométrica de clase $\mathcal{G}^2$ si el vector curvatura, $V(t) = \kappa(t)\,\mathbf{n}(t)$, es continuo.

Consecuencias:

- $\mathcal{C}^2 \Rightarrow \mathcal{G}^2$
- $\alpha(s)$ parametrizada naturalmente, entonces $\alpha(s)$ es $\mathcal{G}^2 \Leftrightarrow \ddot{\alpha}(s)$ es continua.
- $\alpha(t)$ es una curva con $\kappa(t)$ y $\mathbf{n}(t)$ continuas $\Rightarrow \alpha$ es $\mathcal{G}^2$.

3.1.2. Espacios Vectoriales de Funciones

Estudiaremos: $y = f(x)$

$f(x) \in \mathcal{F}(x)$ siendo $\mathcal{F}(x)$ un $\mathbb{R}$-espacio vectorial de dimensión finita.

$$f(x) = a_1 B_1(x) + a_2 B_2(x) + \cdots + a_n B_n(x)$$

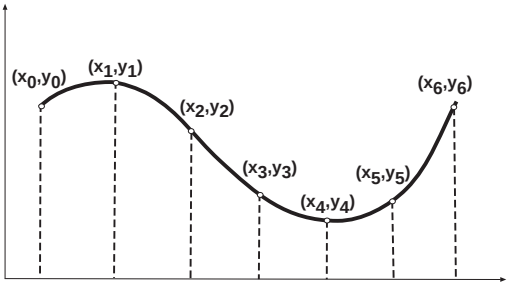
3.1.3. Interpolación Polinomial

Sean $(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)$, $n+1$ puntos de $\mathbb{R}^2$. O bien $y = f(x)$ de cálculo costoso o complejo, entonces elegimos $n+1$ puntos $\{(x_i, f(x_i))\}$, con $0 \leq i \leq n$.

Podemos considerar $x_0 < x_1 < \cdots < x_n$.

Objetivo: Dibujar una curva, lo más simple o sencilla posible, que intuitivamente siga la forma de estos puntos.

Posible solución: Polinomio de Interpolación



El polinomio de interpolación $P(x)$ que pasa por $\{(x_i, y_i)\}$ con $0 \leq i \leq n$, es de grado menor o igual que n, por tanto $P(x) \in \mathcal{P}(x)$.

Si consideramos la base $\mathcal{B} = \{1, x, \ldots, x^n\}$:

$$P(x) = a_0 + a_1\, x + a_2\, x^2 + \cdots + a_n\, x^n$$

$$\begin{bmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix}$$

$det(B) = \prod_{i>j}(x_i - x_j) \neq 0 \Rightarrow$ El polinomio de interpolación existe y es único.

Lagrange:

Del sistema anterior $Ba = y$, la solución $a = B^{-1}y = Ly$

$$P(x) = [1\ x\ x^2\ \cdots\ x^n] \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = [1\ x\ x^2\ \cdots\ x^n] \quad L \quad \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix} =$$

$$= [L_0(x)\ L_1(x)\ L_2(x)\ \cdots\ L_n(x)] \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix} =$$

$$= y_0\ L_0(x) + y_1\ L_1(x) + \cdots + y_n\ L_n(x).$$

Se deduce: $L_i(x_i) = 1, \quad L_i(x_j) = 0,\ i \neq j$. Por tanto $P(x)$ es combinación de la base de Lagrange con coeficientes las ordenadas de los puntos.

Newton:

No es necesario que las abscisas de los puntos a interpolar estén en orden creciente.

- $P_0(x) = y_0$
- $P_1(x)$ polinomio de interpolación de $(x_0, y_0), (x_1, y_1)$. Y así hasta
- $P(x) := P_n(x)$ polinomio de interpolación de $(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)$

Definición 86. $P_i(x) = P_{i-1}(x) + r_i(x) \quad 1 \leq i \leq n$

$y_j = P_{i-1}(x_j) = P_i(x_j),\ 0 \leq j \leq i-1 \Rightarrow r_i(x_j) = 0,\ 0 \leq j \leq i-1.$
Usando la Base de Newton, se tiene que:

$$r_i(x) = c_i(x - x_0)(x - x_1)\cdots(x - x_{i-1}) = c_i N_i(x)$$

Además $y_i = P_i(x_i) = P_{i-1}(x_i) + c_i N_i(x_i)$. Por tanto

$$c_i = \frac{y_i - P_{i-1}(x_i)}{N_i(x_i)} = \frac{y_i - P_{i-1}(x_i)}{\prod_{j=0}^{i-1}(x_i - x_j)}$$

Problemas:

- La obtención de $P(x)$ requiere coste computacional elevado.
- Manejar y operar con polinomios de grado elevado.
- $P(x)$ no sigue la forma dictada por los puntos de interpolación.
- Oscilación del polinomio a ambos lados de la curva teórica.

- Carácter global del polinomio de interpolación: Es inviable retocar localmente el polinomio.

  Si $P(x) = a_0B_0(x) + \cdots + a_kB_k(x) + \cdots + a_nB_n(x)$ y $\overline{P}(x) = a_0B_0(x) + \cdots + \overline{a}_kB_k(x) + \cdots + a_nB_n(x)$ son los polinomios de interpolación. Entonces la variación entre ellos es:

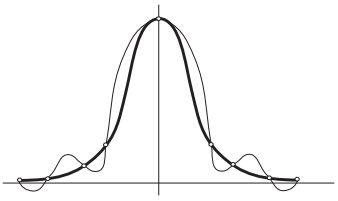
  $$P(x) - \overline{P}(x) = (a_k - \overline{a}_k)B_k(x)$$

  Para que $P(x)$ y $\overline{P}(x)$ se diferencien sólo en un pequeño intervalo (siendo idénticos en el resto), nos interesan bases que sean nulas en gran parte de $[x_0, x_n]$.

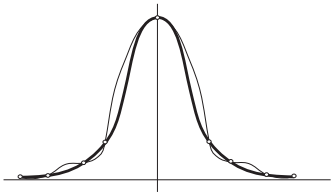
3.1.4. Spline Cúbico

Minimizando Oscilación:

Definición 87. Spline cúbico es una función de clase C^2 constituida por polinomios de 3^{er} grado definidos en cada subintervalo $[x_i, x_{i+1}]$ y con segunda derivada nula en sus extremos.



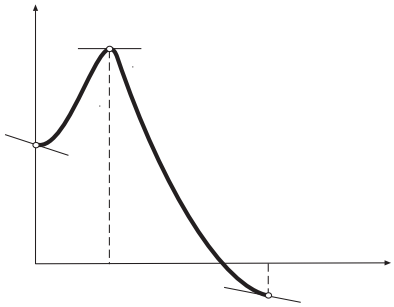
Polinomio de Interpolación



Spline Cúbico

3.1.5. Interpolación De Hermite

Dados los puntos $(x_0, y_0), \ldots, (x_n, y_n)$ buscamos un polinomio $P(x)$ que además de interpolar, lo haga con pendientes especificadas $y'_0, \ldots, y'_n$.



Tal polinomio existe y es único (ver Cordero y Cortés, *Curvas y Superficies para el modelado geométrico*, Ed. Ra-Ma)

Construcción:

$P(x)$ ha de verificar: $\begin{cases} P(x_i) = y_i \\ \\ P'(x_i) = y_i' \end{cases} \quad 0 \leq i \leq n$

Tenemos $2n+2$ condiciones $\Rightarrow$ Grado de $P(x) \leq 2n+1$.

$$P(x) = a_{2n+1}x^{2n+1} + a_{2n}x^{2n} + \cdots + a_1 x + a_0$$
$$P'(x) = (2n+1)a_{2n+1}x^{2n} + 2na_{2n}x^{2n-1} + \cdots + 2a_2 x + a_1$$

$$\begin{bmatrix} 1 & x_0 & x_0^2 & x_0^3 & \cdots & x_0^{2n+1} \\ 1 & x_1 & x_1^2 & x_1^3 & \cdots & x_1^{2n+1} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & x_n & x_n^2 & x_n^3 & \cdots & x_n^{2n+1} \\ 0 & 1 & 2x_0 & 3x_0^2 & \cdots & (2n+1)x_0^{2n} \\ 0 & 1 & 2x_1 & 3x_1^2 & \cdots & (2n+1)x_1^{2n} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 1 & 2x_n & 3x_n^2 & \cdots & (2n+1)x_n^{2n} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ \\ \vdots \\ \\ \vdots \\ a_{2n+1} \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \\ y_0' \\ y_1' \\ \vdots \\ y_n' \end{bmatrix}$$

3.2. Curvas de Bézier

3.2.1. Polinomios de Bernstein:

Fijado $n \in \mathbb{N}$ existen $n+1$ polinomios de Bernstein de grado n:

$$B_k^n(x) = \binom{n}{k} x^k (1-x)^{n-k}, \qquad 0 \leq k \leq n$$

definidos generalmente para $x \in [0,1]$.

Propiedades:

- $B_k^n(x) \geq 0, \qquad \forall x, n, k$
- $B_k^n(x)$ alcanza exactamente un máximo en $x = \frac{k}{n}$
- $\sum_{k=0}^{n} B_k^n(x) = 1, \qquad \forall x$

  Partición de la Unidad
- Recurrencia: $B_k^n(x) = (1-x)B_k^{n-1}(x) + xB_{k-1}^{n-1}(x), \quad \forall n \geq 2, \quad 1 \leq k \leq n.$
- Derivada: $\dfrac{d}{dx}B_k^n(x) = n(B_{k-1}^{n-1}(x) - B_k^{n-1}(x))$

3.2.2. Curvas de Bézier

Curvas Simples:

Definición 88. Lamamos curva de Bézier simple o de un tramo a la curva en forma paramétrica donde cada coordenada es expresada como combinación lineal de la base de Bernstein:

$$C(t) = \sum_{k=0}^{n} P_k \, B_k^n(t), \qquad t \in [0,1]$$

con $P_k = (x_k, y_k)$ si es una curva en $\mathbb{R}^2$ o $P_k = (x_k, y_k, z_k)$ si es una curva en $\mathbb{R}^3$.

Los coeficientes P_k se denominan puntos de control de la curva de Bézier.

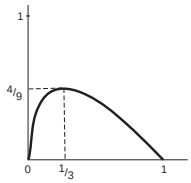
Variación de un punto de control

$$C(t) = (1-t)^3 P_0 + 3t(1-t)^2 P_1 + 3t^2(1-t)P_2 + t^3 P_3$$

$$\overline{C}(t) = (1-t)^3 P_0 + 3t(1-t)^2 \overline{P}_1 + 3t^2(1-t)P_2 + t^3 P_3$$

Entonces:

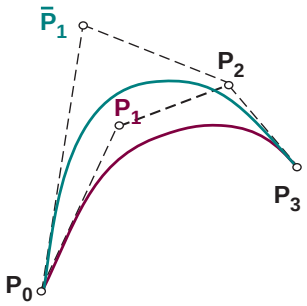
$$C(t) - \overline{C}(t) = 3t(1-t)^2(P_1 - \overline{P}_1) = B_1^3(t)(P_1 - \overline{P}_1)$$



$max_{t\in[0,1]} B_1^3(t) = \frac{4}{9}$

Por tanto

$$max_{t\in[0,1]} \parallel C(t) - \overline{C}(t) \parallel = \frac{4}{9} \parallel P_1 - \overline{P}_1 \parallel$$



Las curvas difieren a lo sumo en algo menos de la mitad de la norma de la diferencia de los puntos de control y es nula al aproximarnos a los extremos.

Propiedades:

- Interpola sólo los puntos extremos: $C(0) = P_0, \quad C(1) = P_n$
- Tangencia al polígono de control en los extremos: $C'(0) = n(P_1 - P_0), \quad C'(1) = n(P_n - P_{n-1})$
- Derivadas: $C'(t) = n\sum_{k=0}^{n-1}(P_{k+1} - P_k)B_k^{n-1}(t)$. Denotamos $\Delta P_j = P_{j+1} - P_j$,

$$C'(t) = n\sum_{k=0}^{n-1}\Delta P_k B_k^{n-1}(t)$$

$C'(t)$ es a su vez una curva de Bézier de grado $n-1$ con puntos de control $n\Delta P_k$.

Es fácil calcular derivadas sucesivas:

$$C''(t) = n(n-1)\sum_{k=0}^{n-2}\Delta(\Delta P_k)B_k^{n-2}(t)$$

Denotamos: $\Delta^2 P_k = \Delta(\Delta P_k) = \Delta P_{k+1} - \Delta P_k = (P_{k+2} - P_{k+1}) - (P_{k+1} - P_k)$ y en general $\Delta^r P_k = \Delta^{r-1}P_{k+1} - \Delta^{r-1}P_k$ se tiene:

$$C^{r)}(t) = n(n-1)\cdots(n-r+1)\sum_{k=0}^{n-r}\Delta^r P_k B_k^{n-r}(t)$$

- Control seudolocal: $C(t) - \overline{C}(t) = B_k^n(t)(P_k - \overline{P}_k)$
- Restricción a la envolvente convexa.
- Invariancia afín: Si f es una afinidad, la curva imagen por la afinidad $f(C(t))$ coincide con la curva de Bézier generada por los puntos $f(P_k)$, imagen por la afinidad de los puntos de control de $C(t)$.
- Recurrencia: consideremos una curva de Bézier de 2^o grado

$$C(t) = (1-t)^2 P_0 + 2t(1-t)P_1 + t^2 P_2$$

También es posible expresarla:

$$C(t) = (1-t)[(1-t)P_0 + tP_1] + t[(1-t)P_1 + tP_2]$$

si denotamos $P_0^{(1)} = (1-t)P_0 + tP_1$ y $P_1^{(1)} = (1-t)P_1 + tP_2$

$$C(t) = (1-t)P_0^{(1)} + tP_1^{(1)}$$

Los puntos $C(t)$ están también sobre una curva de Bézier de 1^{er} grado con puntos de control móviles.

Veamos este proceso recursivo para curvas de Bézier de 3^{er} grado:

$$C(t) = (1-t)^3 P_0 + 3(1-t)^2 tP_1 + 3(1-t)t^2 P_2 + t^3 P_3$$

Podemos expresarla por:

$$C(t) = (1-t)[(1-t)^2P_0 + 2t(1-t)P_1 + t^2P_2] + t[(1-t)^2P_1 + 2t(1-t)P_2 + t^2P_3]$$

también

$$C(t) = (1-t)[(1-t)[(1-t)P_0 + tP_1] + t[(1-t)P_1 + tP_2]] +$$

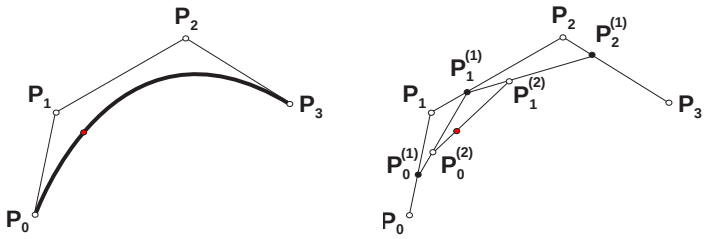
$$+t[(1-t)[(1-t)P_1 + tP_2] + t[(1-t)P_2 + tP_3]]$$

Denotamos $P_0^{(1)} = (1-t)P_0 + tP_1$, $P_1^{(1)} = (1-t)P_1 + tP_2$ y $P_2^{(1)} = (1-t)P_2 + tP_3$,

$$C(t) = (1-t)[(1-t)P_0^{(1)} + tP_1^{(1)}] + t[(1-t)P_1^{(1)} + tP_2^{(1)}]$$

y si denotamos $P_0^{(2)} = (1-t)P_0^{(1)} + tP_1^{(1)}$ y $P_1^{(2)} = (1-t)P_1^{(1)} + tP_2^{(1)}$,

$$C(t) = (1-t)P_0^{(2)} + tP_1^{(2)}$$



Observación: En este dibujo los puntos $P_0^{(1)}, P_1^{(1)}, P_2^{(1)}, P_0^{(2)}, P_1^{(2)}$ están fijados para $t = \frac{1}{3}$.

Algoritmo de De Casteljau:

$P_0, P_1, \ldots, P_n$, $n+1$ puntos en $\mathbb{R}^2$ ó $\mathbb{R}^3$

$$\begin{aligned}
&P_k^{(0)} = P_k && k = 0, \ldots, n \\
&P_k^{(1)} = (1-t)P_k^{(0)} + tP_{k+1}^{(0)} && k = 0, \ldots, n-1 \\
&P_k^{(2)} = (1-t)P_k^{(1)} + tP_{k+1}^{(1)} && k = 0, \ldots, n-2 \\
&\cdots\cdots && \cdots\cdots \\
&P_k^{(n-1)} = (1-t)P_k^{(n-2)} + tP_{k+1}^{(n-2)} && k = 0, 1 \\
&P_k^{(n)} = (1-t)P_k^{(n-1)} + tP_{k+1}^{(n-1)} && k = 0
\end{aligned}$$

Se tiene que

$$C(t) = P_0^{(n)}$$

Reparametrización:

Aunque las curvas de Bézier simples $C(t)$ están definidas para $t \in [0,1]$, es posible reparametrizarlas para que su intervalo de definición sea cualquier $[a,b]$ mediante un cambio de variable afín $t = \frac{s-a}{b-a}$:

$$\overline{C}(s) = \sum_{k=0}^{n} P_k B_k^n \left(\frac{s-a}{b-a} \right), \qquad s \in [a,b]$$

El vector tangente para cada parametrización de la curva es: $C'(t) = \sum_{k=0}^{n} P_k B_k^{n\prime}(t)$, $\overline{C}'(s) = \frac{1}{b-a} \sum_{k=0}^{n} P_k B_k^{n\prime}(\frac{s-a}{b-a})$

Por tanto: $\frac{1}{b-a} C'(t_0) = \frac{1}{b-a} \sum_{k=0}^{n} P_k B_k^{n\prime}(t_0) = \frac{1}{b-a} \sum_{k=0}^{n} P_k B_k^{n\prime}(\frac{s_0-a}{b-a}) = \overline{C}'(s_0)$

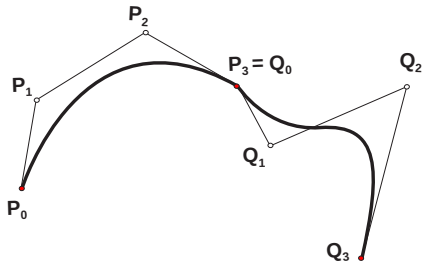
Luego, en puntos correspondientes los dos vectores tangentes son el mismo salvo contante.

Definición 89. Una curva de Bézier compuesta es la unión sucesiva de varias curvas de Bézier simples de bajo grado (2 ó 3).

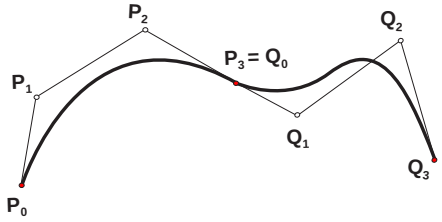
Composición de dos curvas de Bézier simples:

Sean $P_0, P_1, \ldots, P_n$ y $Q_0, Q_1, \ldots, Q_m$ los puntos de control de dos curvas de Bézier simples. Simplificaremos la notación si consideramos las dos curvas del mismo grado $m = n$.

Si queremos que la curva compuesta sea continua $P_n = Q_0$



Además, podemos exigir que las dos curvas se unan manteniendo una cierta suavidad. Para ello, los vectores tangentes a las dos curvas han de tener la misma dirección en el punto de unión $P_n = Q_0$.



Es decir, los vectores tangentes $P_n - P_{n-1}$ y $Q_1 - Q_0 = Q_1 - P_n$ han de ser proporcionales. Cuando se verifique esta condición, diremos que la curva compuesta tiene continuidad geométrica o visual de clase $\mathcal{G}^1$.

Para que la curva compuesta tenga continuidad de clase $\mathcal{C}^1$, es decir que el vector tangente sea continuo, en primer lugar hay parametrizar la curva compuesta con un parámetro global.

Tenemos $C_1(t) = \sum_{k=0}^{n} P_k B_k^n(t)$ y $C_2(s) = \sum_{k=0}^{n} Q_k B_k^n(s)$. Hemos de encontrar dos intervalos $[u_0, u_1]$ y $[u_1, u_2]$ tales que reparametrizando $C_1(t)$ y $C_2(s)$

$$C_1(u) = \sum_{k=0}^{n} P_k B_k^n \left(\frac{u - u_0}{u_1 - u_0} \right) \qquad u \in [u_0, u_1]$$

$$C_2(u) = \sum_{k=0}^{n} Q_k B_k^n \left(\frac{u - u_1}{u_2 - u_1} \right) \qquad u \in [u_1, u_2]$$

La curva de Bézier compuesta $C(u)$ parametrizada por $u \in [u_0, u_2]$ queda definida por

$$C(u) = \begin{cases} C_1(u), & u \in [u_0, u_1] \\ C_2(u), & u \in [u_1, u_2] \end{cases}$$

$C(u)$ es continua de clase $\mathcal{C}^0$ ya que $C_1(u_1) = C_2(u_1)$ pues $P_n = Q_0$.

Para que $C(u)$ sea de clase $\mathcal{C}^1$ ha de verificar $C'(u_1^-) = C'(u_1^+)$. Como sabemos:

$$C_1'(u) = \frac{1}{u_1 - u_0} \sum_{k=0}^{n} P_k B_k^{n\prime} \left(\frac{u - u_0}{u_1 - u_0} \right)$$

$$C_2'(u) = \frac{1}{u_2 - u_1} \sum_{k=0}^{n} Q_k B_k^{n\prime} \left(\frac{u - u_1}{u_2 - u_1} \right)$$

Entonces ha de verificarse $C'(u_1^-) = C_1'(u_1) = C'(u_1^+) = C_2'(u_1)$:

$$C_1'(u_1) = \frac{1}{u_1 - u_0} \sum_{k=0}^{n} P_k B_k^{n\prime}(1) = \frac{1}{u_1 - u_0} n(P_n - P_{n-1})$$

$$C_2'(u_1) = \frac{1}{u_2 - u_1} \sum_{k=0}^{n} Q_k B_k^{n\prime}(0) = \frac{1}{u_2 - u_1} n(Q_1 - Q_0)$$

$$C(u) \text{ es de clase } \mathcal{C}^1 \Leftrightarrow \frac{1}{u_1 - u_0} n(P_n - P_{n-1}) = \frac{1}{u_2 - u_1} n(Q_1 - Q_0)$$

$$\Leftrightarrow Q_1 - Q_0 = \lambda(P_n - P_{n-1}) \text{ con } \lambda = \frac{u_2 - u_1}{u_1 - u_0} = \frac{\| Q_1 - Q_0 \|}{\| P_n - P_{n-1} \|}$$

Si $P_{n-1}, P_n = Q_0, Q_1$ están alineados, la curva compuesta será $\mathcal{G}^1$. Si además podemos encontrar $\{u_0, u_1, u_2\}$ verificando la relación anterior, entonces la curva compuesta será $\mathcal{C}^1$.

Continuamos buscando condiciones sobre los puntos de control para que la curva compuesta sea $\mathcal{C}^2$. Para ello recordemos que si $C(t)$ con $t \in [0, 1]$ es curva simple:

$$C''(0) = n(n-1)\Delta^2 P_0 = n(n-1)(P_2 - 2P_1 + P_0)$$

$$C''(1) = n(n-1)\Delta^2 P_{n-2} = n(n-1)(P_n - 2P_{n-1} + P_{n-2})$$

Entonces tenemos que encontrar $[u_0, u_1]$ y $[u_1, u_2]$ en los que reparametrizar las dos curvas simples, $C_1(t)$ y $C_2(s)$, y definir la curva compuesta. En el punto unión de las dos curvas simples:

$$C_1''(u_1) = \frac{n(n-1)}{(u_1-u_0)^2}\Delta^2 P_{n-2} = \frac{n(n-1)}{(u_1-u_0)^2}(P_n - 2P_{n-1} + P_{n-2})$$

$$C_2''(u_1) = \frac{n(n-1)}{(u_2-u_1)^2}\Delta^2 Q_0 = \frac{n(n-1)}{(u_2-u_1)^2}(Q_2 - 2Q_1 + Q_0)$$

Por tanto la derivada segunda será continua en $u = u_1$ si y sólo si:

$$\frac{1}{(u_1-u_0)^2}\Delta^2 P_{n-2} = \frac{1}{(u_2-u_1)^2}\Delta^2 Q_0$$

o bien,

$$\frac{1}{(u_1-u_0)^2}(P_n - 2P_{n-1} + P_{n-2}) = \frac{1}{(u_2-u_1)^2}(Q_2 - 2Q_1 + Q_0)$$

Pero $P_n = Q_0$, P_{n-1} y Q_1 están alineados y a cierta distancia.

Así, de la expresión anterior obtenemos la siguiente restricción sobre Q_2 en función de P_{n-2}, P_{n-1}, $P_n = Q_0$ y $Q_1 = \frac{u_2-u_1}{u_1-u_0}(P_n - P_{n-1}) + Q_0$:

$$Q_2 = \frac{(u_2-u_1)^2}{(u_1-u_0)^2}(P_n - 2P_{n-1} + P_{n-2}) + 2\frac{u_2-u_1}{u_1-u_0}(P_n - P_{n-1}) + P_n$$

Podemos generalizar las condiciones para que la curva compuesta tenga continuidad de clase $\mathcal{C}^n$:

$$\frac{1}{(u_1-u_0)^r}\Delta^r P_{n-r} = \frac{1}{(u_2-u_1)^r}\Delta^r Q_0, \qquad \text{para } 1 \le r \le n$$

En caso de que las dos curvas simples sean de grado n, estas condiciones nos determinan totalmente los puntos de control de la segunda curva $Q_i, \quad 0 \le i \le n$ fijados los puntos de control de la primera curva. Por tanto, en este caso afirmar que de la curva compuesta es $\mathcal{C}^n$ es equivalente a afirmar que es $\mathcal{C}^\infty$.

Como vemos la composición de curvas simples con continuidad de clase mayor que 1 nos proporcionará cada vez mayor restricción sobre los puntos de control y por tanto pérdida de flexibilidad. Entonces, en adelante consideraremos curvas compuestas de case $\mathcal{C}^1$.

Curvas de Bézier compuestas $\mathcal{C}^1$ de varios tramos:

Sean m curvas de Bézier simples de grado n,

$$\begin{array}{l} C_1(t) = \sum_{k=0}^n P_{k,(1)} B_k^n(t) \\ C_2(t) = \sum_{k=0}^n P_{k,(2)} B_k^n(t) \qquad t \in [0,1] \\ \cdots\cdots\cdots \\ C_m(t) = \sum_{k=0}^n P_{k,(m)} B_k^n(t) \end{array}$$

Para que la curva compuesta por estas sea de clase $\mathcal{C}^1$ es necesario, en primer lugar, definirla con un parámetro global $u \in [u_0, u_m]$ y reparametrizar cada curva simple $C_i(u)$ con $u \in [u_{i-1}, u_i]$.

Debe verificarse la continuidad: $P_{n,(i)} = P_{0,(i+1)}$, para $i = 1, \ldots, m-1$.

Además, la continuidad del vector tangente: $P_{n-1,(i)}, P_{n,(i)} = P_{0,(i+1)}, P_{1,(i+1)}$ alineados (continuidad geométrica $\mathcal{G}^1$) y a cierta distancia, es decir,

$$\frac{u_{i+1} - u_i}{u_i - u_{i-1}} = \frac{\parallel P_{1,(i+1)} - P_{0,(i+1)} \parallel}{\parallel P_{n,(i)} - P_{n-1,(i)} \parallel}, \qquad i = 1, \ldots, m-1$$

La curva compuesta será $\mathcal{C}^1([u_0, u_m])$ sii podemos elegir una partición $u_0 < u_1 < \ldots < u_{m-1} < u_m$ verificando:

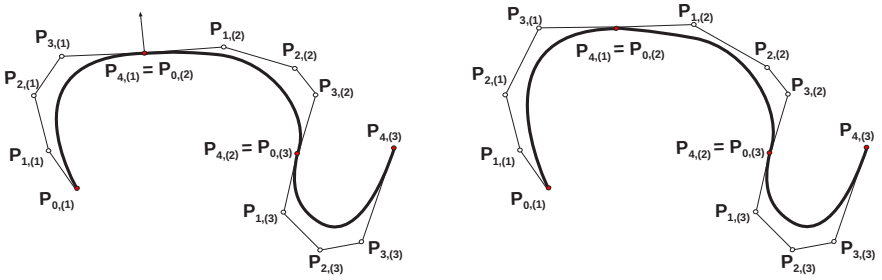
$$u_{i+1} = u_i + (u_i - u_{i-1}) \frac{\parallel P_{1,(i+1)} - P_{0,(i+1)} \parallel}{\parallel P_{n,(i)} - P_{n-1,(i)} \parallel}, \qquad i = 1, \ldots, m-1$$

Curvas de Bezier compuestas $\mathcal{C}^1$ de varios tramos: control de forma

Si modificamos un punto de una de las curvas simples $P_{k,(i)}$ por $\overline{P}_{k,(i)} = P_{k,(i)} + h$, siendo h un vector ¿qué puntos tenemos que modificar también para la curva compuesta siga siendo $\mathcal{C}^1$?

- Si el punto modificado es un punto unión de dos curvas simples:

$$\begin{aligned} \overline{P}_{0,(i+1)} &= P_{0,(i+1)} + h \\ \overline{P}_{n,(i)} &= P_{n,(i)} + h \end{aligned}$$



Entonces

$$\begin{aligned} \overline{P}_{n-1,(i)} &= P_{n-1,(i)} + h \\ \overline{P}_{1,(i+1)} &= P_{1,(i+1)} + h \end{aligned}$$

De forma que

$$\overline{P}_{1,(i+1)} - \overline{P}_{0,(i+1)} = P_{1,(i+1)} - P_{0,(i+1)}$$

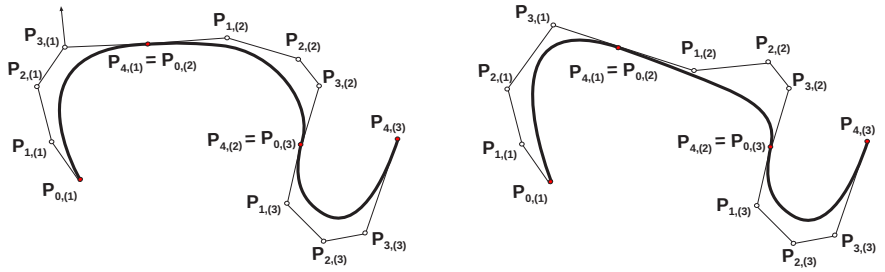
$$\overline{P}_{n,(i)} - \overline{P}_{n-1,(i)} = P_{n,(i)} - P_{n-1,(i)}$$

Y por tanto

$$\frac{\parallel \overline{P}_{1,(i+1)} - \overline{P}_{0,(i+1)} \parallel}{\parallel \overline{P}_{n,(i)} - \overline{P}_{n-1,(i)} \parallel} = \frac{\parallel P_{1,(i+1)} - P_{0,(i+1)} \parallel}{\parallel P_{n,(i)} - P_{n-1,(i)} \parallel}$$

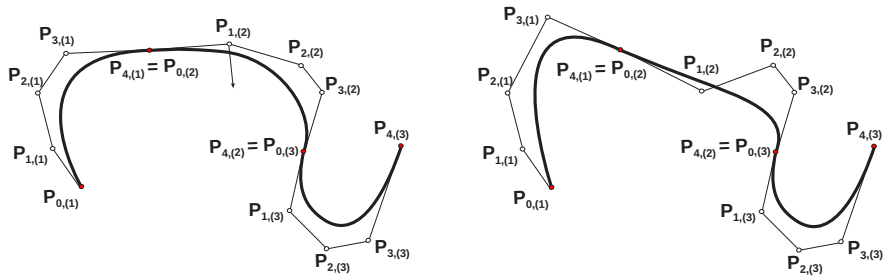
con lo que la partición $u_0 < u_1 < \ldots < u_{m-1} < u_m$ sigue proporcionando que la nueva curva compuesta sea de clase $\mathcal{C}^1([u_0, u_m])$.

- Si el punto modificado es el penúltimo punto de un tramo $\overline{P}_{n-1,(i)} = P_{n-1,(i)} + h$,



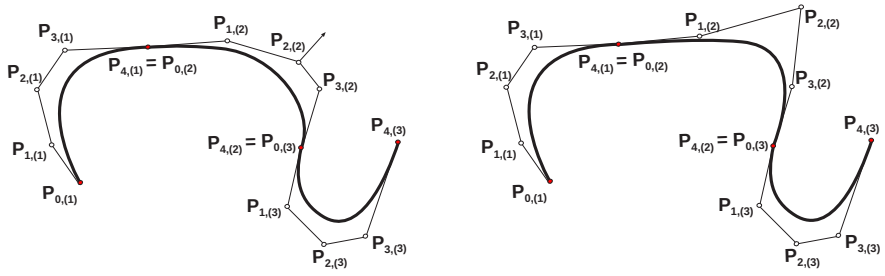
esto obliga (salvo que se trate del último tramo, $i = m$) a modificar el segundo punto del tramo siguiente, $\overline{P}_{1,(i+1)}$, de tal forma que esté alineado con $P_{0,(i+1)}$ y $\overline{P}_{n-1,(i)}$ y se verifique la relación entre los módulos. La partición $u_0 < u_1 < \ldots < u_{m-1} < u_m$ sigue haciendo que la nueva curva compuesta sea de clase $\mathcal{C}^1([u_0, u_m])$.

- Si el punto modificado es el segundo punto de un tramo $\overline{P}_{1,(i+1)} = P_{1,(i+1)} + h$, esto obliga a modificar el penúltimo punto del tramo anterior, $\overline{P}_{n-1,(i)}$, de tal forma que esté alineado con $P_{0,(i+1)}$ y $\overline{P}_{1,(i+1)}$ y se verifique la relación entre los módulos.



La partición $u_0 < u_1 < \ldots < u_{m-1} < u_m$ sigue proporcionando que la nueva curva compuesta sea de clase $\mathcal{C}^1([u_0, u_m])$.

- Si el punto modificado no encuadrado en uno de los casos anteriores, no conlleva la modificación de ningún otro punto.



Resumen:

1. La modificación de un punto de interpolación (o de unión de tramos, excepto $P_{0,(1)}$ y $P_{n,(m)}$) afecta a los dos tramos de los que el punto es unión.

2. La modificación del penúltimo punto de un tramo (excepto el último) afecta a ese y al anterior tramo.

3. La modificación del segundo punto de un tramo (excepto el primero) afecta a ese y al siguiente tramo.

4. La modificación de cualquier otro punto sólo afecta al tramo donde se encuentra dicho punto.

En una curva de Bézier compuesta de curvas de tercer grado, todos los puntos de control excepto los dos primeros y los dos últimos se encuentran en alguna de las categorías 1, 2 ó 3 anteriores.

Curvas de Bézier compuestas: Interpolación cúbica

Fijados unos puntos de interpolación, estudiaremos como obtener la curva de Bézier compuesta de clase $\mathcal{C}^1$ que interpole estos puntos discutiendo distintas alternativas en la elección de los puntos de control intermedios.

Nos restringiremos a curvas de tercer grado, pues nos permiten bastante flexibilidad sin necesidad de manejar gran número de puntos intermedios (sólo dos por tramo).

Cambiaremos la notación:

P_0, P_1, P_2, P_3	tramo 1
P_3, P_4, P_5, P_6	tramo 2
$\cdots\cdots\cdots$	
$P_{3i-3}, P_{3i-2}, P_{3i-1}, P_{3i}$	tramo i
$\cdots\cdots\cdots$	
$P_{3m-3}, P_{3m-2}, P_{3m-1}, P_{3m}$	tramo m

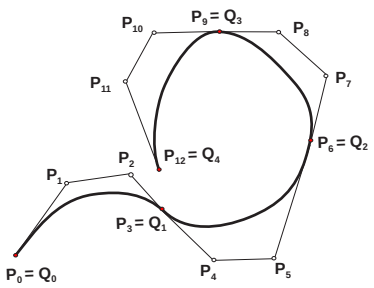
Podemos parametrizar la curva compuesta para $u \in [0,1]$ y denotar $0 = u_0 < u_1 < \cdots < u_m = 1$ una partición del intervalo de definición.

Los puntos de interpolación $Q_0, Q_1, \ldots, Q_m$ serán $Q_i = P_{3i}$ para $i = 0, \ldots, m$

Los puntos de control intermedios, exceptuando P_1 y P_{3m-1}, estarán relacionados de dos en dos mediante la relación de colinealidad de $P_{3i-1}, P_{3i} = Q_i, P_{3i+1}$ y además

$$\frac{u_{i+1} - u_i}{u_i - u_{i-1}} = \frac{\| P_{3i+1} - P_{3i} \|}{\| P_{3i} - P_{3i-1} \|}, \qquad i = 1, \ldots, m-1$$

Consideremos el siguiente ejemplo para $m = 4$:



Los puntos P_1 y P_{11} son libres y los $P_0, P_3, P_6, P_9, P_{12}$ fijos.
Los puntos P_2 y P_4 está en una varilla rígida centrada en P_3 la cual puede girar y extenderse siempre que

$$\frac{u_2 - u_1}{u_1 - u_0} = \frac{\| P_4 - P_3 \|}{\| P_3 - P_2 \|}.$$

Igual pasa con los pares P_5, P_7 y P_8, P_{10}.

Curvas de Bézier compuestas: Interpolación cúbica de Hermite

Construyamos la curva de Bézier compuesta $C(u)$ con continuidad de clase $\mathcal{C}^1$ que interpola los puntos $Q_0, \ldots, Q_m$, conocida la derivada de la curva $D_0, \ldots, D_m$ en esos puntos. Siguiendo la notación anterior, $C(u_i) = Q_i$ y $C'(u_i) = D_i$.

Recordemos que la curva que describe el tramo i-ésimo será

$$C_i(t) = (1-t)^3 P_{3i-3} + t(1-t)^2 P_{3i-2} + t^2(1-t) P_{3i-1} + t^3 P_{3i}$$

Reparametrizando $t = \frac{u-u_i}{u_i - u_{i-1}}$ con $u \in [u_{i-1}, u_i]$, la curva $C(u)$ verifica: $C'(u_i) = C_i'(u_i) = \frac{3}{u_i - u_{i-1}}(P_{3i} - P_{3i-1}) = C_{i+1}'(u_i) = \frac{3}{u_{i+1} - u_i}(P_{3i+1} - P_{3i})$,

$C'(u_0) = \frac{3}{u_1 - u_0}(P_1 - P_0)$,

$C'(u_m) = \frac{3}{u_m - u_{m-1}}(P_{3m} - P_{3m-1})$.

Luego

$$P_{3i-1} = P_{3i} - \frac{u_i - u_{i-1}}{3} C'(u_i) = Q_i - \frac{u_i - u_{i-1}}{3} D_i, \qquad i = 1, \ldots, m-1$$

y también

$$P_{3i+1} = P_{3i} + \frac{u_{i+1} - u_i}{3} C'(u_i) = Q_i + \frac{u_{i+1} - u_i}{3} D_i, \qquad i = 1, \ldots, m-1$$

Los puntos P_1 y P_{3m-1} se obtienen:

$$P_1 = P_0 + \frac{u_1 - u_0}{3} C'(u_0) = Q_0 + \frac{u_1 - u_0}{3} D_0$$

$$P_{3m-1} = P_{3m} - \frac{u_m - u_{m-1}}{3} C'(u_m) = Q_m - \frac{u_m - u_{m-1}}{3} D_m$$

3.3. Superficies de Bézier

3.3.1. Introducción

Sean $(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)$ $n+1$ puntos en $\mathbb{R}^2$ y sean $z_0, z_1, \ldots, z_n$ los valores a interpolar. Buscamos un polinomio $P(x, y)$ tal que

$$z_i = P(x_i, y_i), \qquad i = 0, \ldots, n.$$

¿Qué grado tiene el polinomio $P(x,y)$?
¿Qué base de polinomios elijo para expresar $P(x,y)$?
Ejemplo:
Supongamos que tenemos 4 puntos a interpolar, es decir $n = 3$.

- Podemos tomar $B = \{1, x, y, xy\}$ y $P(x,y) = a_0 + a_1x + a_2y + a_3xy$
- Si $P(x,y)$ es de grado 2, podemos tomar una base más general $B = \{1, x, y, xy, x^2, y^2\}$ y entonces $P(x,y) = a_0+a_1x+a_2y+a_3xy+a_4x^2+a_5y^2$
- Si tenemos 5 puntos, podemos tomar $B = \{1, x, y, x^2, y^2\}$, o bien la anterior base.

3.3.2. Polinomio de Interpolación

Por razones de simetría, para grados fijos n y m respectivamente de x e y, vamos a considerar la base que resulta del producto tensorial de bases de potencias:

$$B = [1x \cdots x^n] \otimes [1y \cdots y^m] = \{1, x, x^2, \ldots, x^n, y, yx, \ldots, yx^n,$$
$$y^2, xy^2, \ldots, y^2x^n, \ldots, y^m, xy^m, \ldots, x^ny^m\}$$

Esta base tiene forma rectangular:

$$\begin{array}{ccccc} 1 & x & x^2 & \cdots & x^n \\ y & yx & yx^2 & \cdots & yx^n \\ y^2 & y^2x & y^2x^2 & \cdots & y^2x^n \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ y^m & y^mx & y^mx^2 & \cdots & y^mx^n \end{array}$$

El espacio que genera tiene dimensión $(n+1)(m+1)$.
Cuando $m = n$ podemos considerar también la base de potencias de polinomios de grado menor o igual que que n:

$$B = \{1, x, x^2 \ldots, x^n, y, xy, x^2y \ldots, x^{n-1}y, \ldots, y^2, xy^2, \ldots, x^{n-2}y,$$
$$\ldots, x^2y^{n-2}, y^{n-1}x, y^n\}$$

Esta base posee forma triangular:

$$\begin{array}{ccccccc} 1 & x & x^2 & \cdots & x^{n-2} & x^{n-1} & x^n \\ y & yx & yx^2 & \cdots & yx^{n-2} & yx^{n-1} & \\ y^2 & y^2x & y^2x^2 & \cdots & y^2x^{n-2} & & \\ \cdots & \cdots & \cdots & \cdots & & & \\ \cdots & \cdots & \cdots & & & & \\ y^{n-1} & xy^{n-1} & & & & & \\ y^n & & & & & & \end{array}$$

Genera un espacio de dimensión $\frac{(n+1)(n+2)}{2}$.

3.3.3. Superficies Producto Tensorial

Considerando la base rectangular de potencias

$$P(x,y) = \sum_{i=0}^{n}\sum_{j=0}^{m} a_{ij}x^i y^j = \sum_{i=0}^{n}\left(\sum_{j=0}^{m} a_{ij}y^j\right)\ x^i = \sum_{i=0}^{n} c_i x^i$$

Esto es, la superficie puede considerarse como una curva polinomial en la variable x con coeficientes una curva polinomial en la variable y.

Podemos cambiar la base de polinomios: $B_x = \{B_0(x), \ldots, B_n(x)\}$, $B_y = \{C_0(y), \ldots, C_m(y)\}$. Entonces considerando la base producto tensorial $[B_0(x), \ldots, B_n(x)]\otimes [C_0(y), \ldots, C_m(y)]$ tenemos

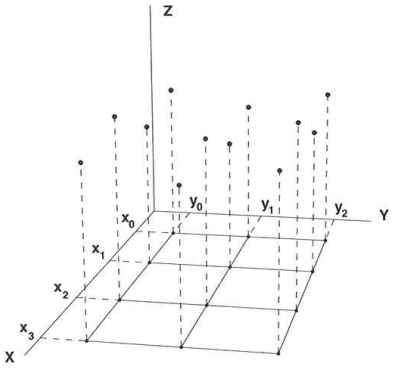
$$P(x,y) = \sum_{i=0}^{n}\sum_{j=0}^{m} b_{ij}B_i(x)C_j(y)$$

En este caso $P(x,y)$ referida a esta nueva base puede no ser un polinomio arbitrario de grado $n+m$.

Dado que la dimensión del espacio de polinomios que estamos considerando es $N = (n+1)(m+1)$, podemos interpolar este número de puntos. Es decir, tenemos $(x_1, y_1), \ldots, (x_N, y_N)$ puntos del plano con sus correspondientes valores funcionales $z_1, \ldots, z_N$ que pueden expresarse como:

$$\begin{array}{ccccc}
(x_0, y_0, z_{00}) & (x_1, y_0, z_{10}) & (x_2, y_0, z_{20}) & \cdots & (x_n, y_0, z_{n0}) \\
(x_0, y_1, z_{01}) & (x_1, y_1, z_{11}) & (x_2, y_1, z_{21}) & \cdots & (x_n, y_1, z_{n1}) \\
\cdots & \cdots & \cdots & \cdots & \cdots \\
(x_0, y_m, z_{0m}) & (x_1, y_m, z_{1m}) & (x_2, y_m, z_{2m}) & \cdots & (x_n, y_m, z_{nm})
\end{array}$$

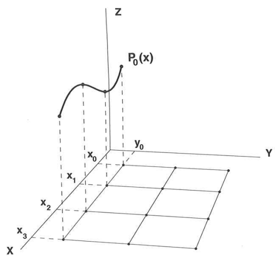
De esta forma los argumentos (x_i, y_j) son nodos de un malla rectangular.
Cada punto de la malla (x_i, y_j) tiene asignado una altura z_{ij}.



Buscamos un polinomio $P(x,y)$ que verifique:

$$P(x_i, y_j) = z_{ij}, \qquad 0 \le i \le n, \quad 0 \le j \le m.$$

Las alturas $z_{00}, z_{10}, \ldots, z_{n0}$, correspondientes a los puntos $(x_0, y_0), (x_1, y_0), \ldots, (x_n, y_0)$, pueden ser interpoladas por un polinomio de grado menor o igual que n :

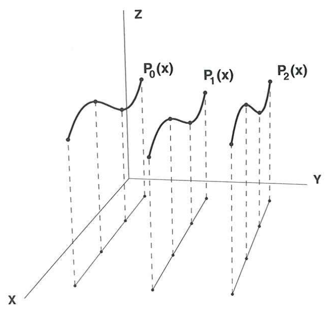


$$\left\{ \begin{array}{rl} z = P_0(x) = & \sum_{i=0}^{n} a_{i0} B_i(x) \\ y = & y_0 \end{array} \right.$$

verificando $P_0(x_k) = z_{k0}, \quad k = 0, \ldots, n$ donde $B_i(x)$ es la base de polinomios a la que está referida el polinomio $P_0(x)$

Reiteramos el proceso para $(x_0, y_1), (x_1, y_1), \ldots, (x_n, y_1)$ obteniendo el polinomio $P_1(x)$ que interpola $z_{01}, z_{11}, \ldots, z_{n1}$. Así calculamos $m+1$ polinomios de interpolación en la variable x:

$$\begin{array}{lll} P_0(x) = \sum_{i=0}^{n} a_{i0} B_i(x) & \text{verifica} & P_0(x_k) = z_{k0}, \quad k = 0, \ldots, n \\ P_1(x) = \sum_{i=0}^{n} a_{i1} B_i(x) & \text{verifica} & P_1(x_k) = z_{k1}, \quad k = 0, \ldots, n \\ \cdots\cdots\cdots & & \\ P_m(x) = \sum_{i=0}^{n} a_{im} B_i(x) & \text{verifica} & P_m(x_k) = z_{km}, \quad k = 0, \ldots, n \end{array}$$



Finalizada la fase de interpolación en la variable x, comencemos con la interpolación con respecto a la variable y. Denotemos $C_j(y)$ la base de polinomios de grado menor o igual que m. Consideramos

$$Q_0(y) = \sum_{j=0}^{m} b_{0j} C_j(y), \quad \text{tal que} \quad Q_0(y_k) = a_{0k}, \quad k = 0, \ldots, m$$

que interpola los coeficientes a_{0k}. Análogamente,

$$Q_1(y) = \sum_{j=0}^{m} b_{1j} C_j(y), \quad \text{tal que} \quad Q_1(y_k) = a_{1k}, \quad k = 0, \ldots, m$$

Y así hasta

$$Q_n(y) = \sum_{j=0}^{m} b_{nj} C_j(y), \quad \text{tal que} \quad Q_n(y_k) = a_{nk}, \quad k = 0, \ldots, m$$

Entonces el polinomio de interpolación buscado será:

$$P(x, y) = \sum_{i=0}^{n} Q_i(y) B_i(x)$$

ya que para cualquier (x_k, y_h), con $0 \leq k \leq n, 0 \leq h \leq m$:

$$P(x_k, y_h) = \sum_{i=0}^{n} Q_i(y_h) B_i(x_k) = \sum_{i=0}^{n} a_{ih} B_i(x_k) = P_h(x_k) = z_{kh}$$

Este procedimiento se puede simplificar si en la segunda fase de la interpolación utilizamos la base de Lagrange. Es decir,

$$Q_i(y) = \sum_{j=0}^{m} a_{ij} L_j(y)$$

entonces

$$P(x,y) = \sum_{i=0}^{n} Q_i(y) B_i(x) = \sum_{i=0}^{n} \left(\sum_{j=0}^{m} a_{ij} L_j(y) \right) B_i(x) =$$

$$\sum_{j=0}^{m} \left(\sum_{i=0}^{n} a_{ij} B_i(x) \right) L_j(y) = \sum_{j=0}^{m} P_j(x) L_j(y)$$

Por tanto, $P(x,y)$ interpola las curvas $P_k(x)$ en la variable y. Es decir, $P(x, y_k) = \sum_{j=0}^{m} P_j(x) L_j(y_k) = P_k(x)$.

Si además elegimos la base de Lagrange para la primera fase de la interpolación, tenemos:

$$P_j(x) = \sum_{i=0}^{n} z_{ij} \overline{L}_i(x)$$

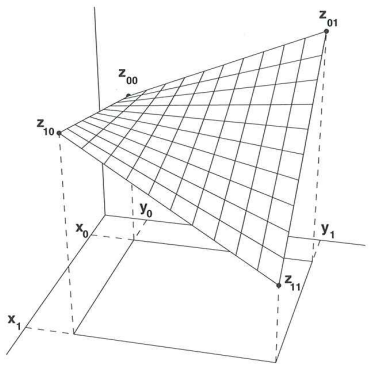
entonces,

$$P(x,y) = \sum_{j=0}^{m} P_j(x) L_j(y) = \sum_{j=0}^{m} \left(\sum_{i=0}^{n} z_{ij} \overline{L}_i(x) \right) L_j(x) = \sum_{i=0}^{n} \sum_{j=0}^{m} z_{ij} \overline{L}_i(x) L_j(y)$$

expresión en la que aparece explícitamente los valores interpolados z_{ij}.

3.3.4. Superficie bilineal

Definición 90. Una superficie bilineal es la superficie que interpola 4 puntos (x_0, y_0, z_{00}), (x_0, y_1, z_{01}), (x_1, y_0, z_{10}), (x_1, y_1, z_{11}); es decir, una superficie producto tensorial para $n = m = 1$.



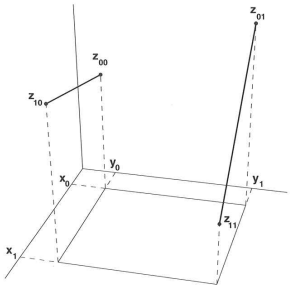
Podemos construirla siguiendo el mismo procedimiento descrito anteriormente. Interpolamos en primer lugar con respecto a la variable x:

- Para $y = y_0$ interpolamos (x_0, z_{00}) y (x_1, z_{10}):

$$P_0(x) = \frac{x_1 z_{00} - x_0 z_{10}}{x_1 - x_0} + \frac{z_{10} - z_{00}}{x_1 - x_0} x = a_{00} + a_{10} x$$

- Para $y = y_1$ interpolamos (x_0, z_{01}) y (x_1, z_{11}):

$$P_1(x) = \frac{x_1 z_{01} - x_0 z_{11}}{x_1 - x_0} + \frac{z_{11} - z_{01}}{x_1 - x_0} x = a_{01} + a_{11} x$$



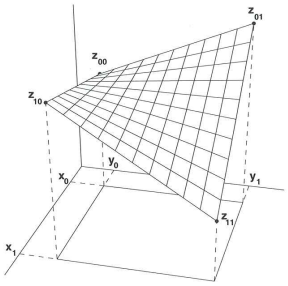
A continuación interpolamos con respecto a y:

- Para $x = x_0$ interpolamos (y_0, a_{00}) y (y_1, a_{01}):

$$Q_0(y) = \frac{y_1 a_{00} - y_0 z_{01}}{y_1 - y_0} + \frac{a_{01} - a_{00}}{y_1 - y_0} y$$

- Para $x = x_1$ interpolamos (y_0, a_{10}) y (y_1, a_{11}):

$$Q_1(y) = \frac{y_1 a_{10} - y_0 a_{11}}{y_1 - y_0} + \frac{a_{11} - a_{10}}{y_1 - y_0} y$$



Entonces

$$P(x, y) = Q_0(y) + Q_1(y)x = \frac{y_1 a_{00} - y_0 z_{01}}{y_1 - y_0} +$$

$$\frac{a_{01} - a_{00}}{y_1 - y_0} y + \frac{y_1 a_{10} - y_0 a_{11}}{y_1 - y_0} x + \frac{a_{11} - a_{10}}{y_1 - y_0} yx.$$

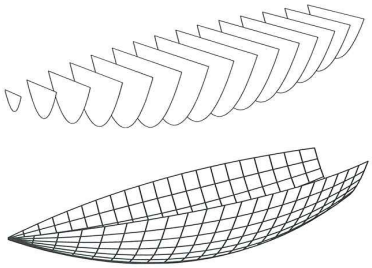
Si sustituimos las expresiones de a_{ij} y agrupamos términos:

$$P(x,y) = z_{00} + \frac{z_{10} - z_{00}}{x_1 - x_0}(x - x_0) + \frac{z_{01} - z_{00}}{y_1 - y_0}(y - y_0) +$$
$$+ \frac{z_{00} - z_{01} - z_{10} + z_{11}}{(x_1 - x_0)(y_1 - y_0)}(x - x_0)(y - y_0).$$

Nota: Los métodos hasta hora descritos construyen superficies $z = P(x, y)$.

3.3.5. Lofting

Técnica basada en la construcción de una superficie recubridora de ciertas curvas planas (secciones de buques) posicionadas en $\mathbb{R}^3$.



Con este procedimiento la superficie obtenida puede ser una superficie paramétrica $S(u, v) = (x(u, v), y(u, v), z(u, v))$.

3.3.6. v-Lofting

Fijadas $n + 1$ curvas, $C_0(u), C_1(u), \ldots, C_n(u)$, para $u \in [u_0, u_1]$

Se pretende construir una superficie $S(u, v)$ con $v \in [v_0, v_n]$ tal que considerando la partición $v_0 < v_1 < \cdots < v_n$ se verifique:

$$S(u, v_i) = C_i(u), \qquad i = 0, \ldots, n.$$

Para la construcción de esta superficie $S(u, v)$ podemos generalizar el procedimiento utilizado en la interpolación de superficies producto tensorial.

Es decir, si consideramos $L_0(v), L_1(v), \cdots, L_n(v)$ polinomios de la base de Lagrange de grado n asociada a la partición fijada del intervalo $[v_0, v_n]$,

$$\begin{cases} L_k(v_i) = 0, & \text{si} \quad i \neq k \\ L_k(v_k) = 1 \end{cases}$$

Tenemos que

$$S(u, v) = \sum_{i=0}^{n} C_i(u) L_i(v)$$

Observemos que las curvas fijadas $C_k(u)$ son curvas paramétricas de la superficie v-Lofting:

$$S(u, v_k) = \sum_{i=0}^{n} C_i(u) L_i(v_k) = C_k(u)$$

También es posible utilizar otra base que no sea la de Lagrange. Sea $B = \{B_0(v), \ldots, B_n(v)\}$ base de polinomios. Entonces

$$S(u,v) = \sum_{i=0}^{n} a_i(u) B_i(v)$$

Para determinar $a_i(u)$ imponemos las condiciones de interpolación:

$$S(u, v_k) = \sum_{i=0}^{n} a_i(u) B_i(v_k) = C_k(u), \qquad k = 0, \ldots, n$$

Podemos escribir estas condiciones matricialmente:

$$\begin{pmatrix} B_0(v_0) & B_1(v_0) & \cdots & B_n(v_0) \\ B_0(v_1) & B_1(v_1) & \cdots & B_n(v_1) \\ \vdots & \vdots & \vdots & \vdots \\ B_0(v_n) & B_1(v_n) & \cdots & B_n(v_n) \end{pmatrix} \begin{pmatrix} a_0(u) \\ a_1(u) \\ \vdots \\ a_n(u) \end{pmatrix} = \begin{pmatrix} C_0(u) \\ C_1(u) \\ \vdots \\ C_n(u) \end{pmatrix}$$

Las curvas $a_i(u)$ quedan determinadas como combinaciones lineales de las curvas $C_k(u)$ de partida.

Esto es, si denotamos L la matriz inversa de B:

$$\begin{pmatrix} a_0(u) \\ a_1(u) \\ \vdots \\ a_n(u) \end{pmatrix} = \begin{pmatrix} l_{00} & l_{01} & \cdots & l_{0n} \\ l_{10} & l_{11} & \cdots & l_{1n} \\ \vdots & \vdots & \vdots & \vdots \\ l_{n0} & l_{1n} & \cdots & l_{nn} \end{pmatrix} \begin{pmatrix} C_0(u) \\ C_1(u) \\ \vdots \\ C_n(u) \end{pmatrix} =$$

$$= \begin{pmatrix} \sum_{j=0}^{n} l_{0j}\, C_j(u) \\ \sum_{j=0}^{n} l_{1j}\, C_j(u) \\ \vdots \\ \sum_{j=0}^{n} l_{nj}\, C_j(u) \end{pmatrix}$$

3.3.7. Superficies de Bézier

Definición 91. Dados un conjunto de puntos de control $\{P_{ij} = (a_{ij}, b_{ij}, c_{ij})\}$ con $0 \le i \le n, \quad 0 \le j \le m$, definimos la superficie de Bézier generada por estos puntos como la siguiente superficie paramétrica:

$$S(u,v) = \sum_{i=0}^{n} \sum_{j=0}^{m} P_{ij} B_i^n(u) B_j^m(v)$$

para $u, v \in [0,1]$, donde

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

$$B_j^m(v) = \binom{m}{j} v^j (1-v)^{m-j}$$

son polinomios de Bernstein de grado n y m respectivamente.

Esto es:

$$S(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m}(a_{ij}, b_{ij}, c_{ij})B_i^n(u)B_j^m(v) =$$

$$\left(\sum_{i=0}^{n}\sum_{j=0}^{m}a_{ij}B_i^n(u)B_j^m(v), \sum_{i=0}^{n}\sum_{j=0}^{m}b_{ij}B_i^n(u)B_j^m(v), \sum_{i=0}^{n}\sum_{j=0}^{m}c_{ij}B_i^n(u)B_j^m(v)\right)$$

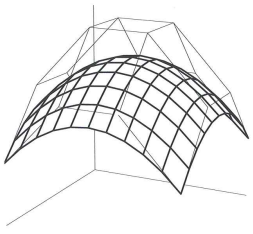
Recordemos algunas de las propiedades de los polinomios de Bernstein:

- $B_k^n(x) \geq 0, \qquad \forall x, n, k.$
- $B_k^n(0) = 0$, para $\quad k \neq 0, \quad B_0^n(0) = 1$ $B_k^n(1) = 0$, para $\quad k \neq n, \quad B_n^n(1) = 1.$
- $\sum_{k=0}^{n} B_k^n(x) = 1, \qquad \forall x.$
- Recurrencia: $B_k^n(x) = (1-x)B_k^{n-1}(x) + xB_{k-1}^{n-1}(x), \quad \forall n \geq 2, \quad 1 \leq k \leq n.$
- Derivada: $\dfrac{d}{dx}B_k^n(x) = n(B_{k-1}^{n-1}(x) - B_k^{n-1}(x))$

Propiedades:

- $S(u,v)$ interpola los puntos vértices de la malla de puntos de control:

$$S(0,0) = P_{00}$$
$$S(0,1) = P_{0m}$$
$$S(1,0) = P_{n0}$$
$$S(1,1) = P_{nm}$$



- Derivadas parciales:

$$\frac{\partial S(u,v)}{\partial u} = \sum_{i=0}^{n}\sum_{j=0}^{m}P_{ij}\frac{dB_i^n(u)}{du}B_j^m(v)$$

Teniendo en cuenta la derivada de los polinomios de Bernstein y reagrupando términos, tenemos:

$$\frac{\partial S(u,v)}{\partial u} = n\sum_{i=0}^{n-1}\sum_{j=0}^{m}(P_{i+1,j} - P_{ij})B_i^{n-1}(u)B_j^m(v)$$

De la misma forma:

$$\frac{\partial S(u,v)}{\partial v} = m\sum_{i=0}^{n}\sum_{j=0}^{m-1}(P_{i,j+1} - P_{ij})B_i^n(u)B_j^{m-1}(v)$$

Si consideremos la curva paramétrica frontera $v = 0$, es decir $\alpha(u) = S(u,0)$ tenemos:

$$\alpha'(u) = \frac{\partial S(u,0)}{\partial u} = n\sum_{i=0}^{n-1}(P_{i+1,0} - P_{i0})B_i^{n-1}(u)$$

Por tanto, $\alpha'(0) = n(P_{10} - P_{00})$ y $\alpha'(1) = n(P_{n0} - P_{n-1,0})$.

- Los puntos $S(u,v)$ están en la envolvente convexa de los puntos P_{ij}.

$$\sum_{i=0}^{n}\sum_{j=0}^{m}B_i^n(u)B_j^m(v) = \sum_{i=0}^{n}\left(\sum_{j=0}^{m}B_j^m(v)\right)B_i^n(u) = \sum_{i=0}^{n}B_i^n(u) = 1$$

- Invariancia afín (igual que en caso de las curvas de Bézier).

- Si consideramos una curva paramétrica $v = v_0$,

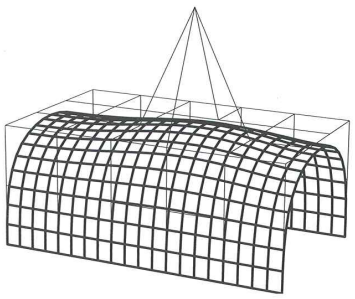
$$\alpha(u) = S(u,v_0) = \sum_{i=0}^{n}\left(\sum_{j=0}^{m}P_{ij}B_j^m(v_0)\right)B_i^n(u)$$

  es una curva de Bézier de grado n con puntos de control $Q_i = \sum_{j=0}^{m}P_{ij}B_j^m(v_0)$.

- Control seudolocal: si un punto de control P_{kh} es modificado a $\overline{P}_{kh}$, entonces las superficies resultantes se diferencian en:

$$S(u,v) - \overline{S}(u,v) = (P_{kh} - \overline{P}_{kh})B_k^n(u)B_h^m(v)$$

  y al igual que en el caso de las curvas es posible cuantificar esta deferencia en módulo.



3.3.8. Superficies de Bézier Compuestas

Propiedades:

Sean $\{P_{ij}\}$ con $0 \le i \le n$, $0 \le j \le m$ los puntos de control de una superficie de Bézier $S_1(u,v)$ definida en $[u_0,u_1]\times[v_0,v_1]$:

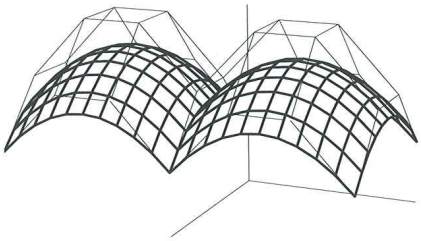
$$S_1(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} P_{ij} B_i^n(t) B_j^m(s)$$

donde $t = \frac{u-u_0}{u_1-u_0}, \quad s = \frac{v-v_0}{v_1-v_0}$.

Y sean $\{Q_{ij}\}$ con $0 \le i \le n, \quad 0 \le j \le m$ los puntos de control de una superficie de Bézier $S_2(u,v)$ definida en $[u_1,u_2]\times[v_0,v_1]$:

$$S_2(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} Q_{ij} B_i^n(r) B_j^m(s)$$

donde $r = \frac{u-u_1}{u_2-u_1}$.



La superficie compuesta, definida para $u \in [u_0,u_2], \quad v \in [v_0,v_1]$, es continua con clase $\mathcal{C}^k$ si lo es en los puntos de la curva paramétrica $u = u_1$, unión de $S_1(u,v)$ y $S_2(u,v)$.

$$S_1(u_1,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} P_{ij} B_i^n(1) B_j^m(s) = \sum_{j=0}^{m} P_{nj} B_j^m(v)$$

$$S_2(u_1,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} Q_{ij} B_i^n(0) B_j^m(s) = \sum_{j=0}^{m} Q_{0j} B_j^m(v)$$

$S(u,v)$ será $\mathcal{C}^0$ sii $S_1(u_1,v) = S_2(u_1,v)$ sii, teniendo en cuenta que $B_j^m(v)$ es base,

$$P_{nj=Q_{0j},} \qquad {\scriptstyle j=0\ldots,m}$$

$S(u,v)$ será $\mathcal{C}^1$ sii $\frac{\partial S_1}{\partial v}(u_1,v) = \frac{\partial S_2}{\partial v}(u_1,v)$ y $\frac{\partial S_1}{\partial u}(u_1,v) = \frac{\partial S_2}{\partial u}(u_1,v)$:

$$\frac{\partial S_1}{\partial v}(u_1,v) = \frac{m}{v_1-v_0}\sum_{j=0}^{m-1}(P_{n,j+1}-P_{nj}) B_j^{m-1}(s)$$

$$\frac{\partial S_2}{\partial v}(u_1,v) = \frac{m}{v_1-v_0}\sum_{j=0}^{m-1}(Q_{0,j+1}-Q_{0j}) B_j^{m-1}(s)$$

Lo que no añade condiciones nuevas.

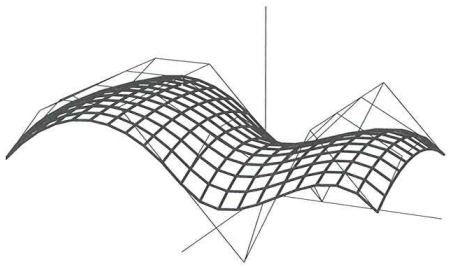
$$\frac{\partial S_1}{\partial u}(u_1,v) = \frac{n}{u_1-u_0}\sum_{j=0}^{m}(P_{nj}-P_{n-1,j}) B_j^{m}(s)$$

$$\frac{\partial S_2}{\partial u}(u_1, v) = \frac{n}{u_2 - u_1} \sum_{j=0}^{m} (Q_{1j} - Q_{0j}) B_j^m(s)$$

Por tanto $S(u, v)$ será $\mathcal{C}^1$ sii

$$\frac{n}{u_1 - u_0}(P_{nj} - P_{n-1,j}) = \frac{n}{u_2 - u_1}(Q_{1j} - Q_{0j}), \qquad j = 0, \ldots, m$$

Es decir, para cada $j = 0, \ldots, m$ los puntos $P_{n-1,j}, P_{nj} = Q_{0j}, Q_{1j}$ tienen que estar alineados y a cierta distancia.



Recordamos que el caso de curvas de Bézier compuestas, fijados los puntos de control, podíamos buscar una partición del intervalo global de definición y reparametrizar las curva simples para que la curva composición sea de clase $\mathcal{C}^1$. En el caso de superficies, fijados los puntos de control, no podemos hacer lo mismo. Los puntos de control y la partición de los intervalos han de ser fijados a priori para que la superficie de Bézier compuesta sea $\mathcal{C}^1$.