

# Aproximación normal de curvas y superficies regulares

Alfonso Carriazo

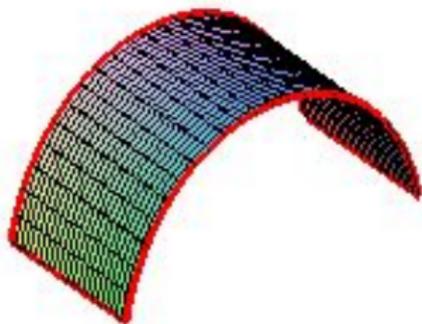
Departamento de Geometría y Topología



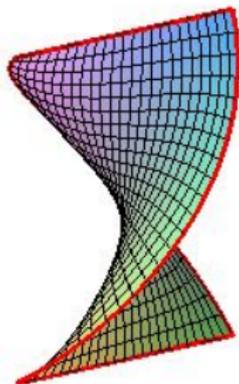
V Jornadas Doctorales

Programa de Doctorado en Matemáticas, Universidad de Cádiz

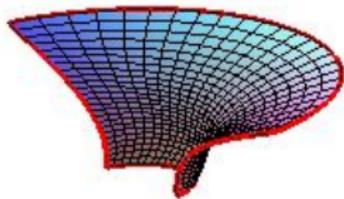
¿Es esto un trozo de un cilindro?



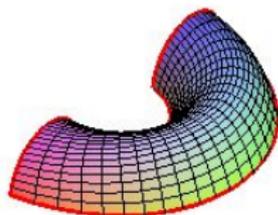
¿Es esto un trozo de un helicoides?



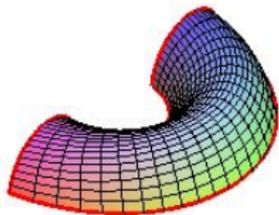
¿Es esto un trozo de un catenoide?



¿Es esto un trozo de un toro?

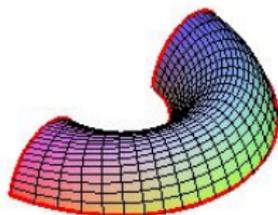


¿Es esto un trozo de un toro?



La respuesta es: NO.

¿Es esto un trozo de un toro?



La respuesta es: NO. De hecho, son Superficies de **Bézier**.



Francia, 01/09/1910 - 25/11/1999



Francia, 01/09/1910 - 25/11/1999

- Ingeniero Mecánico y Eléctrico.



Francia, 01/09/1910 - 25/11/1999

- Ingeniero Mecánico y Eléctrico.
- Trabajó para Renault (1933-1975).



Francia, 01/09/1910 - 25/11/1999

- Ingeniero Mecánico y Eléctrico.
- Trabajó para Renault (1933-1975).
- Se doctoró en Matemáticas en 1977.



Francia, 01/09/1910 - 25/11/1999

- Ingeniero Mecánico y Eléctrico.
- Trabajó para Renault (1933-1975).
- Se doctoró en Matemáticas en 1977.
- Conservatoire National des Arts et Métiers (1968-1979).

Dados  $n + 1$  puntos  $P_0, \dots, P_n$ , determinan la **curva de Bézier**

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

Dados  $n + 1$  puntos  $P_0, \dots, P_n$ , determinan la **curva de Bézier**

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

donde

$$B_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i}, \quad 0 \leq i \leq n$$

es el **polinomio de Bernstein** de grado  $n$ .

# Curvas de Bézier

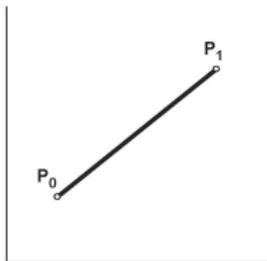
Dados  $n + 1$  puntos  $P_0, \dots, P_n$ , determinan la **curva de Bézier**

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

donde

$$B_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i}, \quad 0 \leq i \leq n$$

es el **polinomio de Bernstein** de grado  $n$ .



# Curvas de Bézier

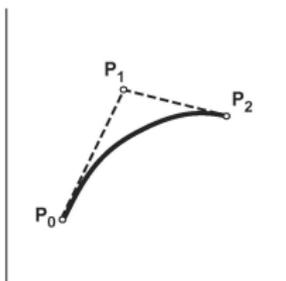
Dados  $n + 1$  puntos  $P_0, \dots, P_n$ , determinan la **curva de Bézier**

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

donde

$$B_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i}, \quad 0 \leq i \leq n$$

es el **polinomio de Bernstein** de grado  $n$ .



# Curvas de Bézier

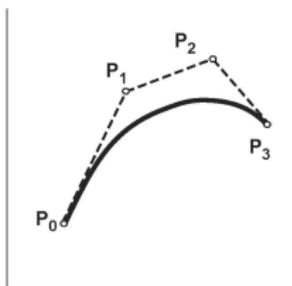
Dados  $n + 1$  puntos  $P_0, \dots, P_n$ , determinan la **curva de Bézier**

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

donde

$$B_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i}, \quad 0 \leq i \leq n$$

es el **polinomio de Bernstein** de grado  $n$ .



Dada una malla de  $m \cdot n$  puntos  $P_{ij}$ ,  $0 \leq i \leq m, 0 \leq j \leq n$ , determina la **superficie de Bézier**

$$B(u, v) = \sum_{i=0}^m \sum_{j=0}^n B_i^m(u) B_j^n(v) P_{ij}, \quad u, v \in [0, 1],$$

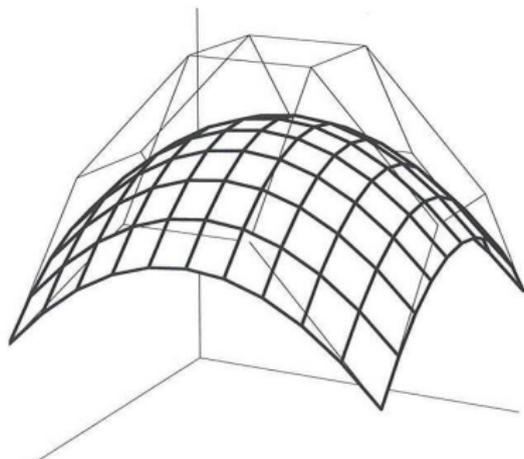
donde  $B_i^m$  y  $B_j^n$  son polinomios de Bernstein de grados  $m$  y  $n$ , respectivamente.

# Superficies de Bézier

Dada una malla de  $m \cdot n$  puntos  $P_{ij}$ ,  $0 \leq i \leq m, 0 \leq j \leq n$ , determina la **superficie de Bézier**

$$B(u, v) = \sum_{i=0}^m \sum_{j=0}^n B_i^m(u) B_j^n(v) P_{ij}, \quad u, v \in [0, 1],$$

donde  $B_i^m$  y  $B_j^n$  son polinomios de Bernstein de grados  $m$  y  $n$ , respectivamente.



**Problema 1:** Dada una curva regular

$$\alpha : [0, 1] \longrightarrow \mathbb{R}^2,$$

**Problema 1:** Dada una curva regular

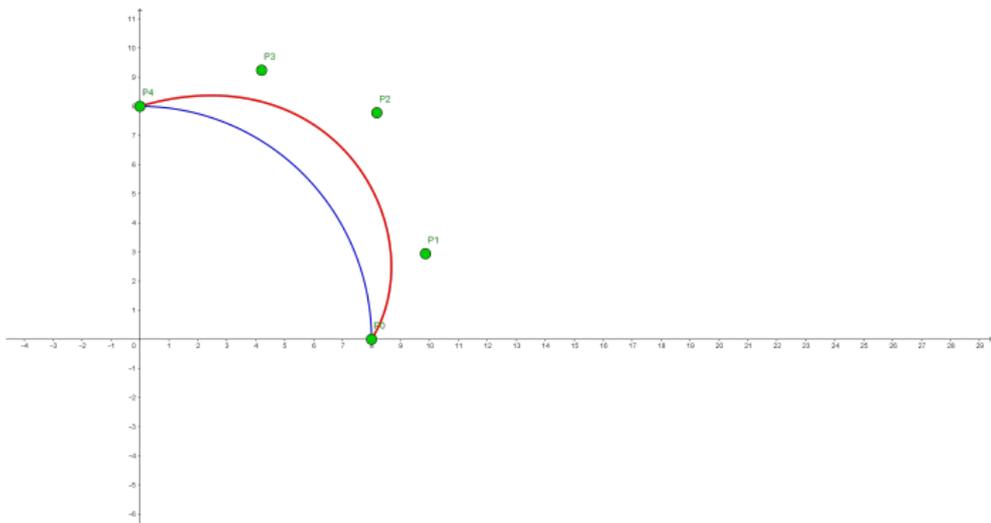
$$\alpha : [0, 1] \longrightarrow \mathbb{R}^2,$$

¿cómo podemos elegir los puntos de control para aproximar  $\alpha$  por una curva de **Bézier**?

**Problema 1:** Dada una curva regular

$$\alpha : [0, 1] \longrightarrow \mathbb{R}^2,$$

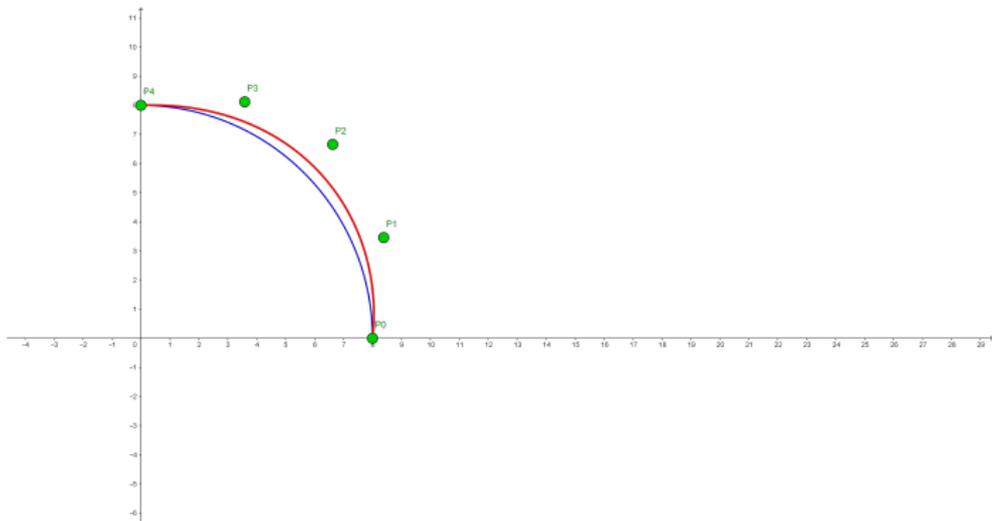
¿cómo podemos elegir los puntos de control para aproximar  $\alpha$  por una curva de **Bézier**?



**Problema 1:** Dada una curva regular

$$\alpha : [0, 1] \longrightarrow \mathbb{R}^2,$$

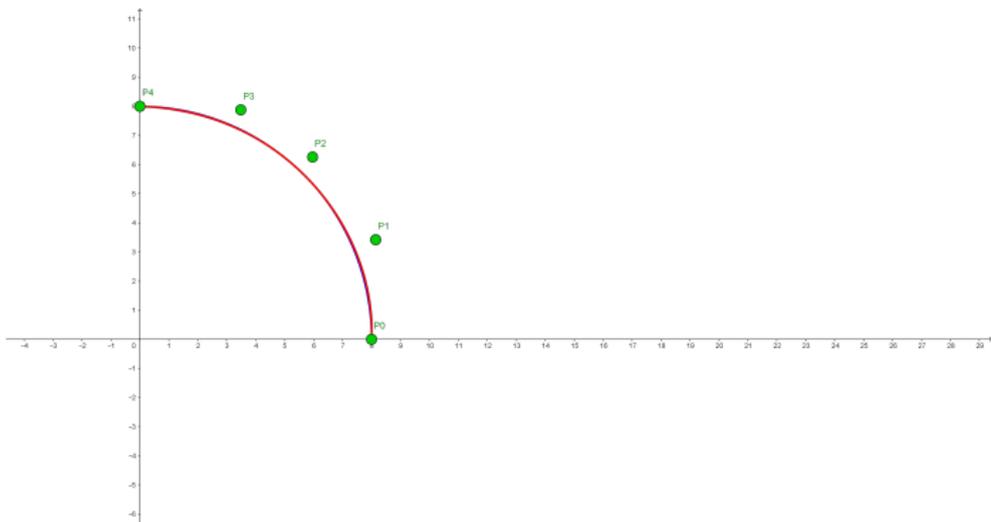
¿cómo podemos elegir los puntos de control para aproximar  $\alpha$  por una curva de **Bézier**?



**Problema 1:** Dada una curva regular

$$\alpha : [0, 1] \longrightarrow \mathbb{R}^2,$$

¿cómo podemos elegir los puntos de control para aproximar  $\alpha$  por una curva de **Bézier**?



**Problema 1:** Dada una curva regular

$$\alpha : [0, 1] \longrightarrow \mathbb{R}^3,$$

¿cómo podemos elegir los puntos de control para aproximar  $\alpha$  por una curva de **Bézier**?

**Problema 1:** Dada una curva regular

$$\alpha : [0, 1] \longrightarrow \mathbb{R}^3,$$

¿cómo podemos elegir los puntos de control para aproximar  $\alpha$  por una curva de **Bézier**?

**Problema 2:** Dada una superficie regular

$$\mathbf{x} : [0, 1] \times [0, 1] \longrightarrow \mathbb{R}^3,$$

¿cómo podemos elegir los puntos de control para aproximar  $\mathbf{x}$  por una superficie de **Bézier**?

# Nuestro método

Sea  $\alpha : [0, 1] \rightarrow \mathbb{R}^2$  una curva regular.

# Nuestro método

Sea  $\alpha : [0, 1] \rightarrow \mathbb{R}^2$  una curva regular.

Elegimos  $t_0 = 0 < t_1 < \cdots < t_{n-1} < t_n = 1$ .

# Nuestro método

Sea  $\alpha : [0, 1] \rightarrow \mathbb{R}^2$  una curva regular.

Elegimos  $t_0 = 0 < t_1 < \dots < t_{n-1} < t_n = 1$ . Sean  $P_0 = \alpha(0)$ ,  
 $P_n = \alpha(1)$ .

# Nuestro método

Sea  $\alpha : [0, 1] \rightarrow \mathbb{R}^2$  una curva regular.

Elegimos  $t_0 = 0 < t_1 < \dots < t_{n-1} < t_n = 1$ . Sean  $P_0 = \alpha(0)$ ,  $P_n = \alpha(1)$ . Para cada  $i = 1, \dots, n-1$ , consideramos el punto

$$P_i(\lambda_i) = \alpha(t_i) + \lambda_i \mathbf{n}(t_i),$$

donde  $\mathbf{n}(t_i)$  es el vector normal a  $\alpha$  en  $\alpha(t_i)$ .

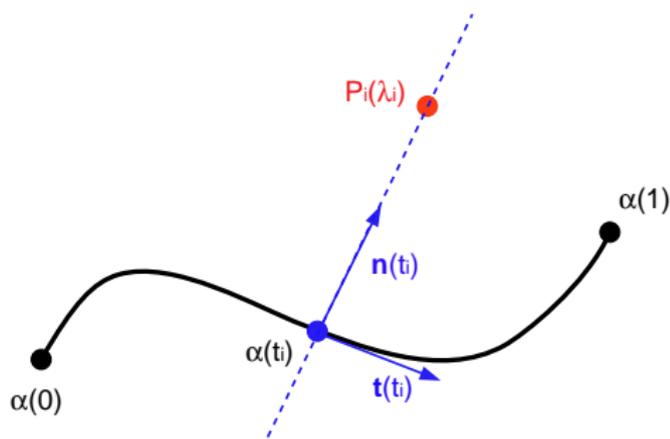
# Nuestro método

Sea  $\alpha : [0, 1] \rightarrow \mathbb{R}^2$  una curva regular.

Elegimos  $t_0 = 0 < t_1 < \dots < t_{n-1} < t_n = 1$ . Sean  $P_0 = \alpha(0)$ ,  $P_n = \alpha(1)$ . Para cada  $i = 1, \dots, n-1$ , consideramos el punto

$$P_i(\lambda_i) = \alpha(t_i) + \lambda_i \mathbf{n}(t_i),$$

donde  $\mathbf{n}(t_i)$  es el vector normal a  $\alpha$  en  $\alpha(t_i)$ .



## Nuestro método (2)

Entonces, definimos la curva de **Bézier**

$$B(\lambda_1, \dots, \lambda_{n-1}, t) = B_0^n(t)P_0 + \sum_{i=1}^{n-1} B_i^n(t)P_i(\lambda_i) + B_n^n(t)P_n,$$

donde los  $B_i^n(t)$  son los polinomios de Bernstein.

## Nuestro método (2)

Entonces, definimos la curva de **Bézier**

$$B(\lambda_1, \dots, \lambda_{n-1}, t) = B_0^n(t)P_0 + \sum_{i=1}^{n-1} B_i^n(t)P_i(\lambda_i) + B_n^n(t)P_n,$$

donde los  $B_i^n(t)$  son los polinomios de Bernstein.

Ahora, definimos la función:

$$F(\lambda_1, \dots, \lambda_{n-1}) = \int_0^1 |\alpha(t) - B(\lambda_1, \dots, \lambda_{n-1}, t)|^2 dt.$$

## Nuestro método (2)

Entonces, definimos la curva de **Bézier**

$$B(\lambda_1, \dots, \lambda_{n-1}, t) = B_0^n(t)P_0 + \sum_{i=1}^{n-1} B_i^n(t)P_i(\lambda_i) + B_n^n(t)P_n,$$

donde los  $B_i^n(t)$  son los polinomios de Bernstein.

Ahora, definimos la función:

$$F(\lambda_1, \dots, \lambda_{n-1}) = \int_0^1 |\alpha(t) - B(\lambda_1, \dots, \lambda_{n-1}, t)|^2 dt.$$

Minimizando esta función, encontramos los valores  $\lambda_1^0, \dots, \lambda_{n-1}^0$  que nos dan la mejor aproximación  $B(\lambda_1^0, \dots, \lambda_{n-1}^0, t)$  para la curva  $\alpha(t)$ .

## Nuestro método (3)

La minimización se puede hacer resolviendo un sistema de ecuaciones lineales, dado por  $\partial F / \partial \lambda_j = 0$ , para cada  $j = 1, \dots, n - 1$ .

## Nuestro método (3)

La minimización se puede hacer resolviendo un sistema de ecuaciones lineales, dado por  $\partial F / \partial \lambda_j = 0$ , para cada  $j = 1, \dots, n - 1$ . De hecho:

$$\frac{\partial F}{\partial \lambda_j} = \int_0^1 \frac{\partial G}{\partial \lambda_j} dt,$$

donde

$$G(\lambda_1, \dots, \lambda_{n-1}, t) = \sum_{k=1}^2 (\alpha^k(t) - B^k(\lambda_1, \dots, \lambda_{n-1}, t))^2.$$

## Nuestro método (3)

La minimización se puede hacer resolviendo un sistema de ecuaciones lineales, dado por  $\partial F / \partial \lambda_j = 0$ , para cada  $j = 1, \dots, n - 1$ . De hecho:

$$\frac{\partial F}{\partial \lambda_j} = \int_0^1 \frac{\partial G}{\partial \lambda_j} dt,$$

donde

$$G(\lambda_1, \dots, \lambda_{n-1}, t) = \sum_{k=1}^2 (\alpha^k(t) - B^k(\lambda_1, \dots, \lambda_{n-1}, t))^2.$$

Así:

$$\frac{\partial G}{\partial \lambda_j} = 2 \sum_{k=1}^2 (\alpha^k(t) - B^k(\lambda_1, \dots, \lambda_{n-1}, t)) \frac{\partial (\alpha^k(t) - B^k(\lambda_1, \dots, \lambda_{n-1}, t))}{\partial \lambda_j}.$$

## Nuestro método (3)

La minimización se puede hacer resolviendo un sistema de ecuaciones lineales, dado por  $\partial F / \partial \lambda_j = 0$ , para cada  $j = 1, \dots, n - 1$ . De hecho:

$$\frac{\partial F}{\partial \lambda_j} = \int_0^1 \frac{\partial G}{\partial \lambda_j} dt,$$

donde

$$G(\lambda_1, \dots, \lambda_{n-1}, t) = \sum_{k=1}^2 (\alpha^k(t) - B^k(\lambda_1, \dots, \lambda_{n-1}, t))^2.$$

Así:

$$\frac{\partial G}{\partial \lambda_j} = 2 \sum_{k=1}^2 (\alpha^k(t) - B^k(\lambda_1, \dots, \lambda_{n-1}, t)) (-B_j^n(t) \mathbf{n}^k(t)).$$

## Nuestro método (3)

La minimización se puede hacer resolviendo un sistema de ecuaciones lineales, dado por  $\partial F / \partial \lambda_j = 0$ , para cada  $j = 1, \dots, n - 1$ . De hecho:

$$\begin{aligned} \frac{\partial F}{\partial \lambda_j} = & -2 \sum_{k=1}^2 \mathbf{n}^k(t_j) \left( \int_0^1 \alpha^k(t) B_j^n(t) dt - P_0^k \int_0^1 B_0^n(t) B_j^n(t) dt \right. \\ & \left. - \sum_{i=1}^{n-1} \alpha^k(t_i) \int_0^1 B_i^n(t) B_j^n(t) dt - P_n^k \int_0^1 B_n^n(t) B_j^n(t) dt \right) \\ & + 2 \sum_{i=1}^{n-1} \left( \sum_{k=1}^2 \mathbf{n}^k(t_i) \mathbf{n}^k(t_j) \int_0^1 B_i^n(t) B_j^n(t) dt \right) \lambda_i, \end{aligned}$$

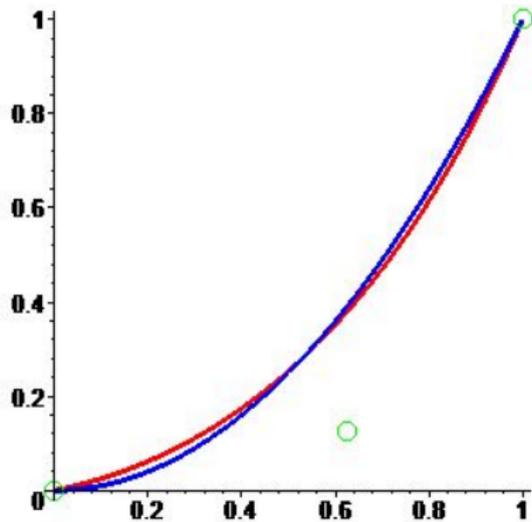
## Nuestro método (3)

La minimización se puede hacer resolviendo un sistema de ecuaciones lineales, dado por  $\partial F/\partial \lambda_j = 0$ , para cada  $j = 1, \dots, n-1$ . De hecho:

$$\begin{aligned} \frac{\partial F}{\partial \lambda_j} = & -2 \sum_{k=1}^2 \mathbf{n}^k(t_j) \left( \int_0^1 \alpha^k(t) B_j^n(t) dt - P_0^k \int_0^1 B_0^n(t) B_j^n(t) dt \right. \\ & \left. - \sum_{i=1}^{n-1} \alpha^k(t_i) \int_0^1 B_i^n(t) B_j^n(t) dt - P_n^k \int_0^1 B_n^n(t) B_j^n(t) dt \right) \\ & + 2 \sum_{i=1}^{n-1} \left( \sum_{k=1}^2 \mathbf{n}^k(t_i) \mathbf{n}^k(t_j) \int_0^1 B_i^n(t) B_j^n(t) dt \right) \lambda_i, \end{aligned}$$

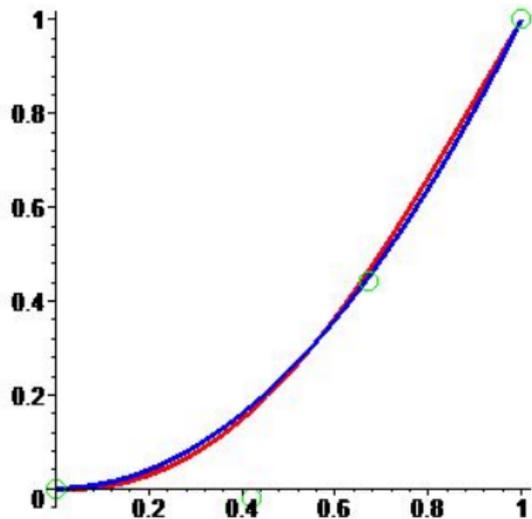
Esta expresión es afín en  $\lambda_1, \dots, \lambda_{n-1}$ .

# Ejemplo: la parábola



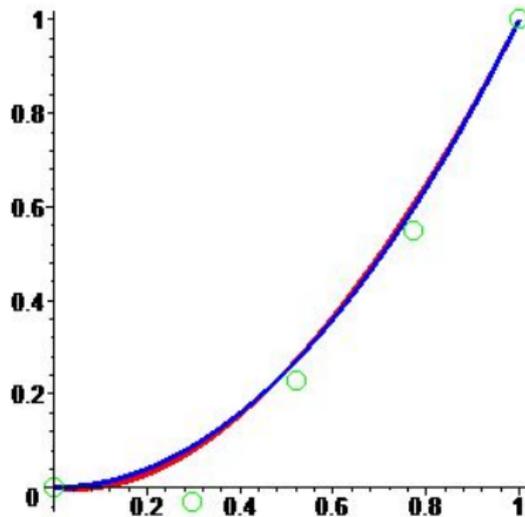
Estimación del error: 0.004166666667

# Ejemplo: la parábola



Estimación del error: 0.001459250709

# Ejemplo: la parábola



Estimación del error: 0.0008372327736

# Nuestro método: una mejora

Sea  $\alpha : [0, 1] \rightarrow \mathbb{R}^2$  una curva regular.

# Nuestro método: una mejora

Sea  $\alpha : [0, 1] \rightarrow \mathbb{R}^2$  una curva regular.

Es bien sabido que, si  $B(t)$  es una curva de **Bézier**, entonces

$$B'(0) = n(P_1 - P_0), \quad B'(1) = n(P_n - P_{n-1}).$$

# Nuestro método: una mejora

Sea  $\alpha : [0, 1] \rightarrow \mathbb{R}^2$  una curva regular.

Es bien sabido que, si  $B(t)$  es una curva de **Bézier**, entonces

$$B'(0) = n(P_1 - P_0), \quad B'(1) = n(P_n - P_{n-1}).$$

En nuestro caso:

$$B'(\lambda_1, \dots, \lambda_{n-1}, 0) = n(\alpha(t_1) + \lambda_1 \mathbf{n}(t_1) - \alpha(0)),$$

$$B'(\lambda_1, \dots, \lambda_{n-1}, 1) = n(\alpha(1) - \alpha(t_{n-1}) - \lambda_{n-1} \mathbf{n}(t_{n-1})).$$

## Nuestro método: una mejora

Sea  $\alpha : [0, 1] \rightarrow \mathbb{R}^2$  una curva regular.

Es bien sabido que, si  $B(t)$  es una curva de **Bézier**, entonces

$$B'(0) = n(P_1 - P_0), \quad B'(1) = n(P_n - P_{n-1}).$$

En nuestro caso:

$$B'(\lambda_1, \dots, \lambda_{n-1}, 0) = n(\alpha(t_1) + \lambda_1 \mathbf{n}(t_1) - \alpha(0)),$$

$$B'(\lambda_1, \dots, \lambda_{n-1}, 1) = n(\alpha(1) - \alpha(t_{n-1}) - \lambda_{n-1} \mathbf{n}(t_{n-1})).$$

Si imponemos

$$B'(\lambda_1, \dots, \lambda_{n-1}, 0) \cdot \mathbf{n}(0) = 0, \quad B'(\lambda_1, \dots, \lambda_{n-1}, 1) \cdot \mathbf{n}(1) = 0,$$

entonces obtenemos dos ecuaciones lineales que nos permiten determinar fácilmente  $\lambda_1, \lambda_{n-1}$ .

# Nuestro método: una mejora

Sea  $\alpha : [0, 1] \rightarrow \mathbb{R}^2$  una curva regular.

Es bien sabido que, si  $B(t)$  es una curva de **Bézier**, entonces

$$B'(0) = n(P_1 - P_0), \quad B'(1) = n(P_n - P_{n-1}).$$

En nuestro caso:

$$B'(\lambda_1, \dots, \lambda_{n-1}, 0) = n(\alpha(t_1) + \lambda_1 \mathbf{n}(t_1) - \alpha(0)),$$

$$B'(\lambda_1, \dots, \lambda_{n-1}, 1) = n(\alpha(1) - \alpha(t_{n-1}) - \lambda_{n-1} \mathbf{n}(t_{n-1})).$$

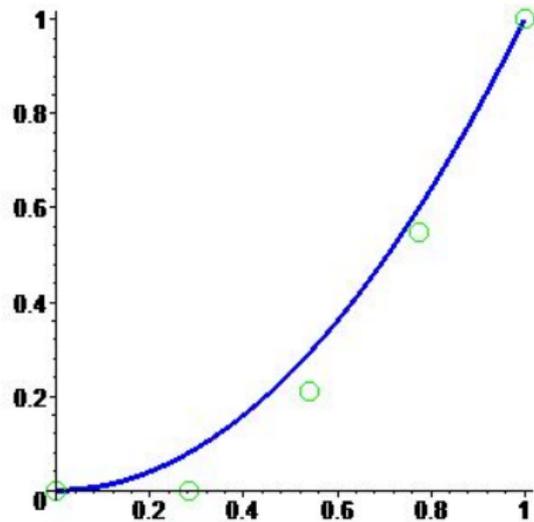
Si imponemos

$$B'(\lambda_1, \dots, \lambda_{n-1}, 0) \cdot \mathbf{n}(0) = 0, \quad B'(\lambda_1, \dots, \lambda_{n-1}, 1) \cdot \mathbf{n}(1) = 0,$$

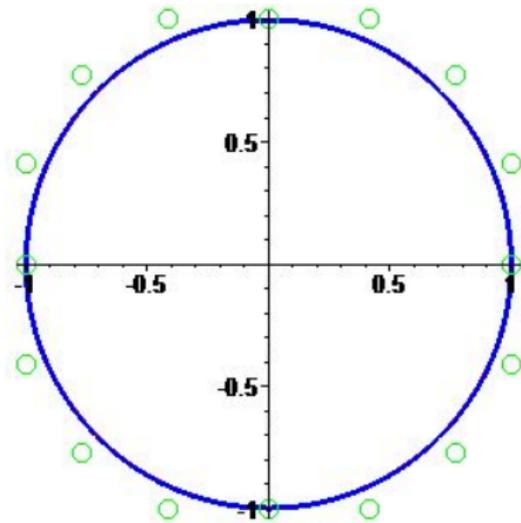
entonces obtenemos dos ecuaciones lineales que nos permiten determinar fácilmente  $\lambda_1, \lambda_{n-1}$ .

Decimos que  $B(\lambda_1, \dots, \lambda_{n-1}, t)$  está “**clavada**” en los puntos  $P_0, P_n$ .

# Ejemplo: la parábola



# Ejemplo: la circunferencia

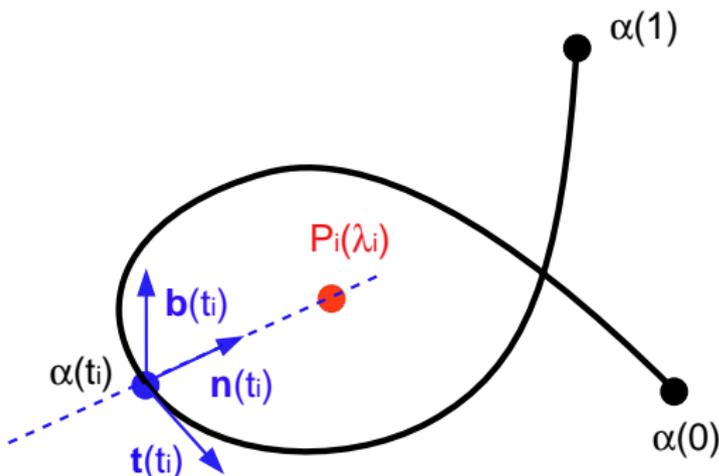


## Curvas espaciales:

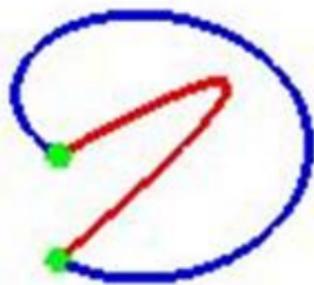
**Curvas espaciales:** El mismo método se puede usar para curvas en  $\mathbb{R}^3$ . En tal caso,  $\mathbf{n}(t_i)$  sería el vector normal principal de  $\alpha$  en  $\alpha(t_i)$  (necesitamos que  $\kappa(t_i) \neq 0$ ).

# Nuestro método: extensiones

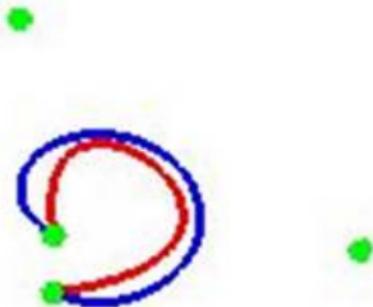
**Curvas espaciales:** El mismo método se puede usar para curvas en  $\mathbb{R}^3$ . En tal caso,  $\mathbf{n}(t_i)$  sería el vector normal principal de  $\alpha$  en  $\alpha(t_i)$  (necesitamos que  $\kappa(t_i) \neq 0$ ).



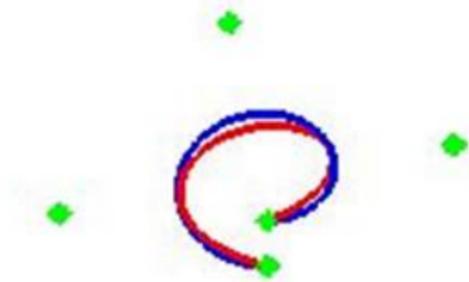
# Ejemplo: hélice circular



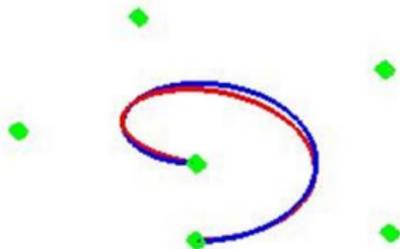
# Ejemplo: hélice circular



# Ejemplo: hélice circular



# Ejemplo: hélice circular



# Nuestro método: extensiones (2)

**Superficies:**

## Nuestro método: extensiones (2)

**Superficies:** Para aproximar una superficie  $\mathbf{x} : [0, 1] \times [0, 1] \rightarrow \mathbb{R}^3$ , seguimos los siguientes pasos:

## Nuestro método: extensiones (2)

**Superficies:** Para aproximar una superficie  $\mathbf{x} : [0, 1] \times [0, 1] \rightarrow \mathbb{R}^3$ , seguimos los siguientes pasos:

**Paso 1:** Aproximamos las curvas

$$\alpha_1(t) = \mathbf{x}(t, 0), \quad \alpha_2(t) = \mathbf{x}(0, t), \quad \alpha_3(t) = \mathbf{x}(t, 1), \quad \alpha_4(t) = \mathbf{x}(1, t),$$

por curvas de **Bézier**  $B_1(t), \dots, B_4(t)$ .

## Nuestro método: extensiones (2)

**Superficies:** Para aproximar una superficie  $\mathbf{x} : [0, 1] \times [0, 1] \rightarrow \mathbb{R}^3$ , seguimos los siguientes pasos:

**Paso 1:** Aproximamos las curvas

$$\alpha_1(t) = \mathbf{x}(t, 0), \quad \alpha_2(t) = \mathbf{x}(0, t), \quad \alpha_3(t) = \mathbf{x}(t, 1), \quad \alpha_4(t) = \mathbf{x}(1, t),$$

por curvas de **Bézier**  $B_1(t), \dots, B_4(t)$ . Así, determinamos los puntos exteriores de la malla de control.

## Nuestro método: extensiones (2)

**Superficies:** Para aproximar una superficie  $\mathbf{x} : [0, 1] \times [0, 1] \rightarrow \mathbb{R}^3$ , seguimos los siguientes pasos:

**Paso 1:** Aproximamos las curvas

$$\alpha_1(t) = \mathbf{x}(t, 0), \quad \alpha_2(t) = \mathbf{x}(0, t), \quad \alpha_3(t) = \mathbf{x}(t, 1), \quad \alpha_4(t) = \mathbf{x}(1, t),$$

por curvas de **Bézier**  $B_1(t), \dots, B_4(t)$ . Así, determinamos los puntos exteriores de la malla de control.

**Paso 2:** Para los puntos interiores de la malla de control, escribimos

$$P_{ij}(\lambda_{ij}) = \mathbf{x}(u_i, v_j) + \lambda_{ij} \mathbf{N}(u_i, v_j),$$

donde  $\mathbf{N}(u_i, v_j)$  es el vector normal de  $\mathbf{x}$  en  $\mathbf{x}(u_i, v_j)$ , y construimos la superficie de **Bézier**  $B(\lambda_{ij}, u, v)$ .

## Nuestro método: extensiones (2)

**Superficies:** Para aproximar una superficie  $\mathbf{x} : [0, 1] \times [0, 1] \rightarrow \mathbb{R}^3$ , seguimos los siguientes pasos:

**Paso 1:** Aproximamos las curvas

$$\alpha_1(t) = \mathbf{x}(t, 0), \quad \alpha_2(t) = \mathbf{x}(0, t), \quad \alpha_3(t) = \mathbf{x}(t, 1), \quad \alpha_4(t) = \mathbf{x}(1, t),$$

por curvas de **Bézier**  $B_1(t), \dots, B_4(t)$ . Así, determinamos los puntos exteriores de la malla de control.

**Paso 2:** Para los puntos interiores de la malla de control, escribimos

$$P_{ij}(\lambda_{ij}) = \mathbf{x}(u_i, v_j) + \lambda_{ij} \mathbf{N}(u_i, v_j),$$

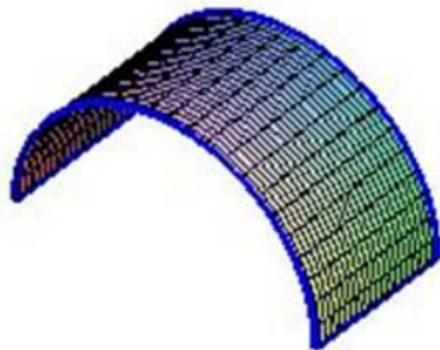
donde  $\mathbf{N}(u_i, v_j)$  es el vector normal de  $\mathbf{x}$  en  $\mathbf{x}(u_i, v_j)$ , y construimos la superficie de **Bézier**  $B(\lambda_{ij}, u, v)$ .

**Paso 3:** Minimizamos la función

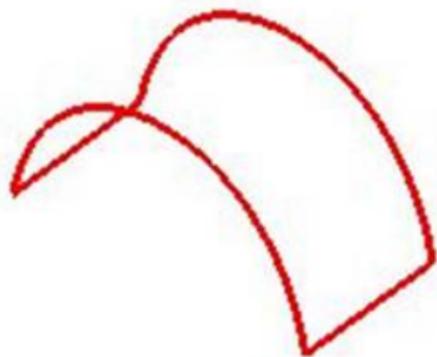
$$F(\lambda_{ij}) = \int_0^1 \int_0^1 |\mathbf{x}(u, v) - B(\lambda_{ij}, u, v)|^2 dudv,$$

para obtener los valores de  $\lambda_{ij}^0$ .

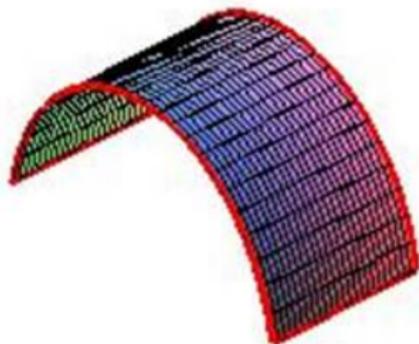
# Ejemplo: cilindro circular



# Ejemplo: cilindro circular

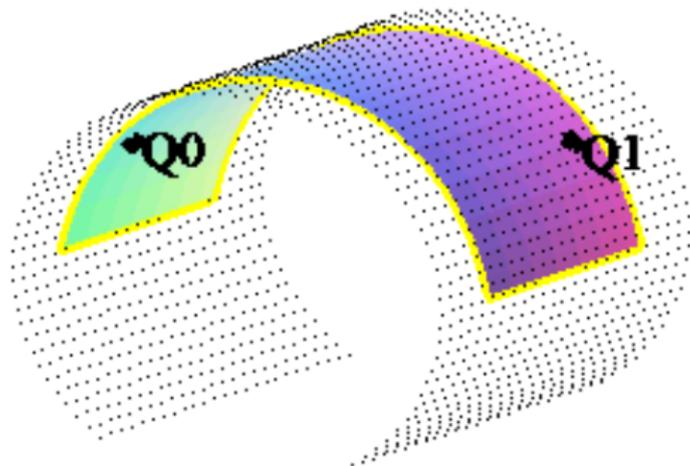


# Ejemplo: cilindro circular

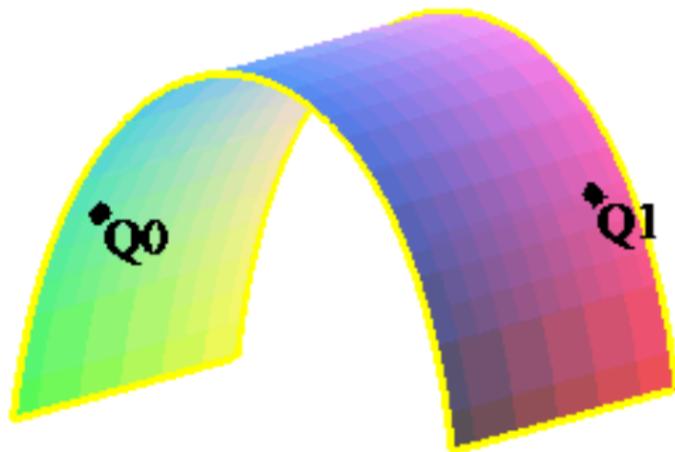


**Minimizar la longitud de curvas en superficies:**

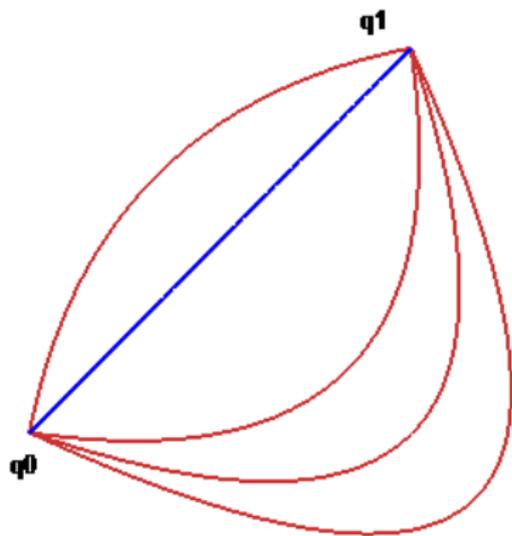
Minimizar la longitud de curvas en superficies:



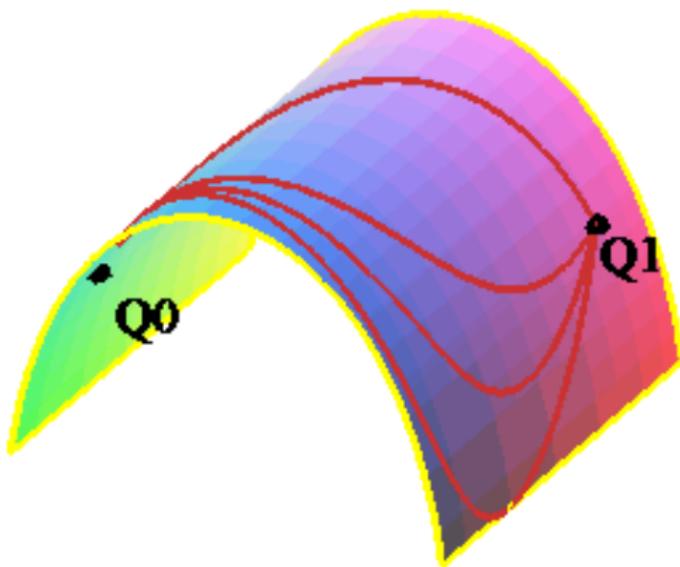
Minimizar la longitud de curvas en superficies:



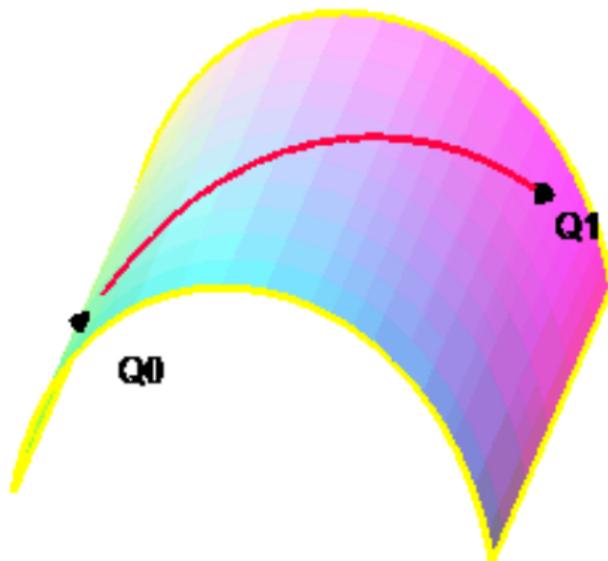
Minimizar la longitud de curvas en superficies:



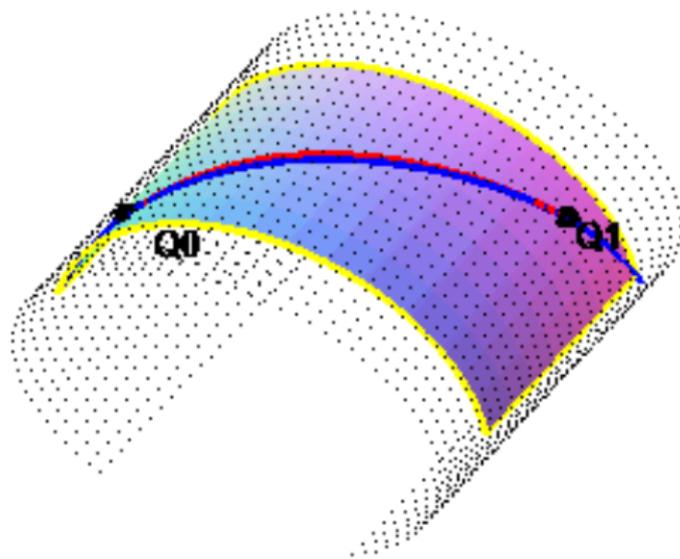
Minimizar la longitud de curvas en superficies:



Minimizar la longitud de curvas en superficies:



Minimizar la longitud de curvas en superficies:



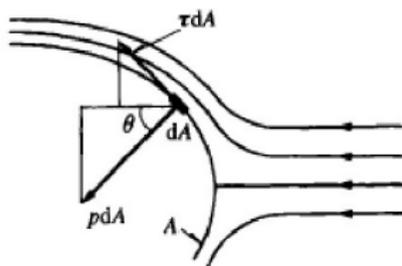
**Mejora aerodinámica de curvas y superficies:**

# Nuestro método: aplicaciones (2)

## Mejora aerodinámica de curvas y superficies:

Nuestro objetivo es minimizar la fuerza de presión que viene determinada por la siguiente ecuación:

$$F_p = \int_A p dA \cos \theta$$

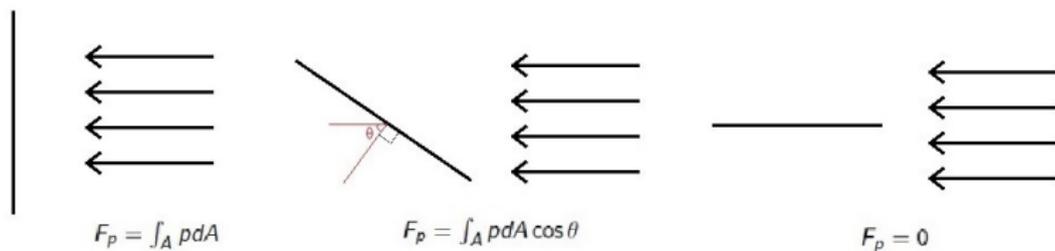


# Nuestro método: aplicaciones (2)

## Mejora aerodinámica de curvas y superficies:

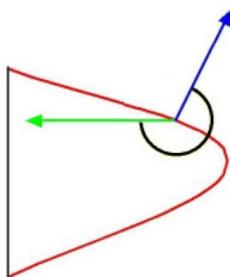
Nuestro objetivo es minimizar la fuerza de presión que viene determinada por la siguiente ecuación:

$$F_p = \int_A p dA \cos \theta$$



## Mejora aerodinámica de curvas y superficies:

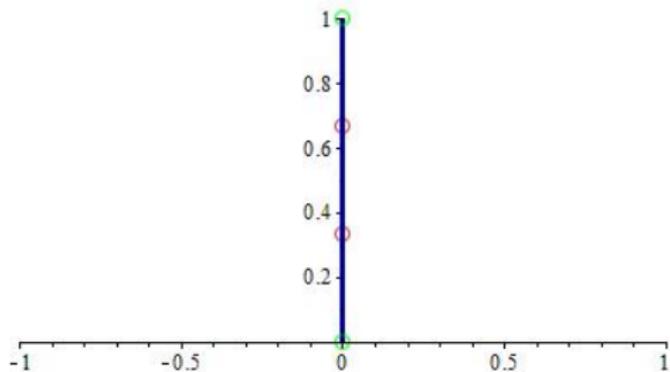
**Objetivo:** Minimizar  $\cos^2 \theta$  (para que no influya el signo), siendo  $\theta$  el ángulo que forman la normal de la curva y la dirección del aire  $\vec{v} = (-1, 0)$ .



## Nuestro método: aplicaciones (2)

$$B(\lambda_1, \lambda_2, t) = (3t(1-t)^2\lambda_1 + 3t^2(1-t)\lambda_2, (1-t)^3 + 2t(1-t)^2 + t^2(1-t))$$

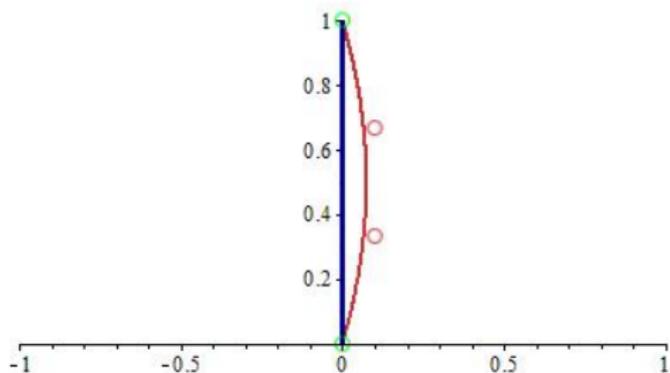
$$\cos^2 \theta(\lambda_1, \lambda_2, 1/3) = \frac{1}{1 + \lambda_2^2}, \quad \cos^2 \theta(\lambda_1, \lambda_2, 2/3) = \frac{1}{1 + \lambda_1^2}$$



## Nuestro método: aplicaciones (2)

$$B(\lambda_1, \lambda_2, t) = (3t(1-t)^2\lambda_1 + 3t^2(1-t)\lambda_2, (1-t)^3 + 2t(1-t)^2 + t^2(1-t))$$

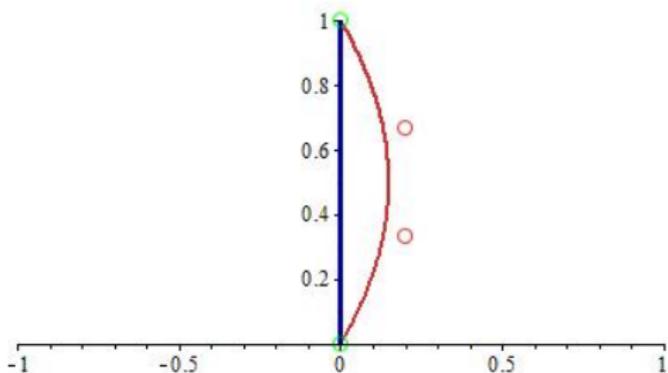
$$\cos^2 \theta(\lambda_1, \lambda_2, 1/3) = \frac{1}{1 + \lambda_2^2}, \quad \cos^2 \theta(\lambda_1, \lambda_2, 2/3) = \frac{1}{1 + \lambda_1^2}$$



## Nuestro método: aplicaciones (2)

$$B(\lambda_1, \lambda_2, t) = (3t(1-t)^2\lambda_1 + 3t^2(1-t)\lambda_2, (1-t)^3 + 2t(1-t)^2 + t^2(1-t))$$

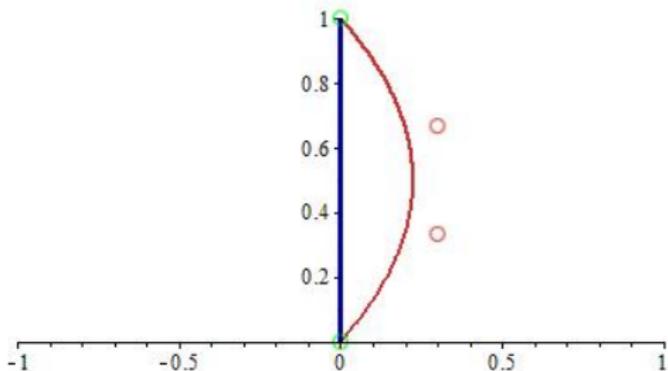
$$\cos^2 \theta(\lambda_1, \lambda_2, 1/3) = \frac{1}{1 + \lambda_2^2}, \quad \cos^2 \theta(\lambda_1, \lambda_2, 2/3) = \frac{1}{1 + \lambda_1^2}$$



## Nuestro método: aplicaciones (2)

$$B(\lambda_1, \lambda_2, t) = (3t(1-t)^2\lambda_1 + 3t^2(1-t)\lambda_2, (1-t)^3 + 2t(1-t)^2 + t^2(1-t))$$

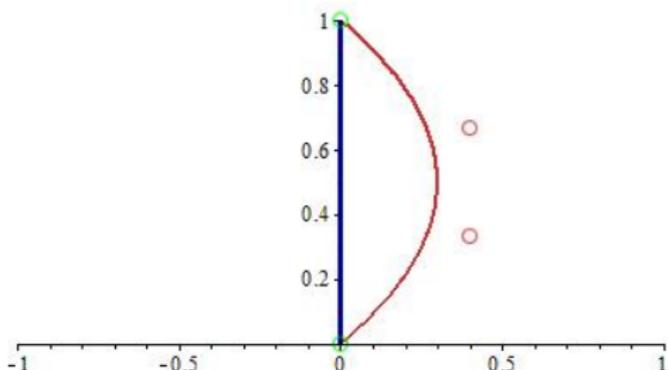
$$\cos^2 \theta(\lambda_1, \lambda_2, 1/3) = \frac{1}{1 + \lambda_2^2}, \quad \cos^2 \theta(\lambda_1, \lambda_2, 2/3) = \frac{1}{1 + \lambda_1^2}$$



## Nuestro método: aplicaciones (2)

$$B(\lambda_1, \lambda_2, t) = (3t(1-t)^2\lambda_1 + 3t^2(1-t)\lambda_2, (1-t)^3 + 2t(1-t)^2 + t^2(1-t))$$

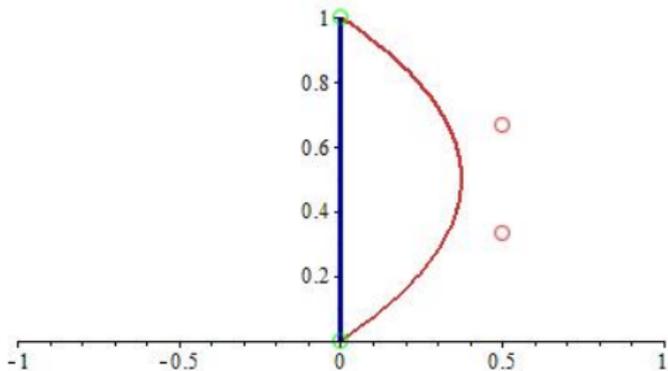
$$\cos^2 \theta(\lambda_1, \lambda_2, 1/3) = \frac{1}{1 + \lambda_2^2}, \quad \cos^2 \theta(\lambda_1, \lambda_2, 2/3) = \frac{1}{1 + \lambda_1^2}$$



## Nuestro método: aplicaciones (2)

$$B(\lambda_1, \lambda_2, t) = (3t(1-t)^2\lambda_1 + 3t^2(1-t)\lambda_2, (1-t)^3 + 2t(1-t)^2 + t^2(1-t))$$

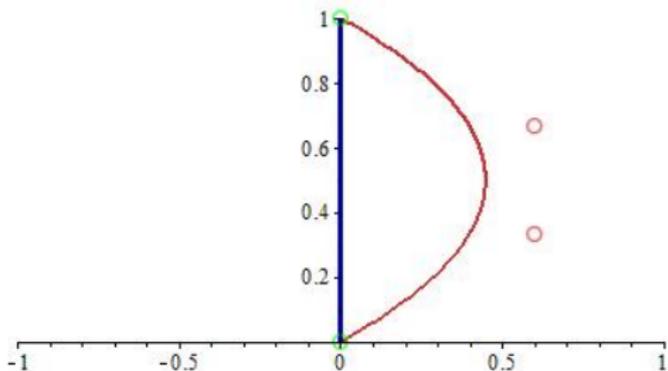
$$\cos^2 \theta(\lambda_1, \lambda_2, 1/3) = \frac{1}{1 + \lambda_2^2}, \quad \cos^2 \theta(\lambda_1, \lambda_2, 2/3) = \frac{1}{1 + \lambda_1^2}$$



## Nuestro método: aplicaciones (2)

$$B(\lambda_1, \lambda_2, t) = (3t(1-t)^2\lambda_1 + 3t^2(1-t)\lambda_2, (1-t)^3 + 2t(1-t)^2 + t^2(1-t))$$

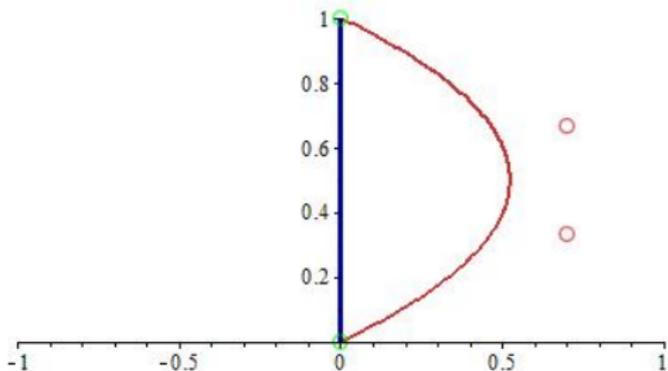
$$\cos^2 \theta(\lambda_1, \lambda_2, 1/3) = \frac{1}{1 + \lambda_2^2}, \quad \cos^2 \theta(\lambda_1, \lambda_2, 2/3) = \frac{1}{1 + \lambda_1^2}$$



## Nuestro método: aplicaciones (2)

$$B(\lambda_1, \lambda_2, t) = (3t(1-t)^2\lambda_1 + 3t^2(1-t)\lambda_2, (1-t)^3 + 2t(1-t)^2 + t^2(1-t))$$

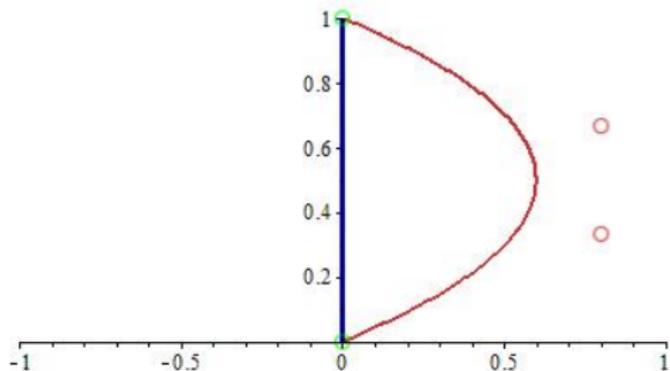
$$\cos^2 \theta(\lambda_1, \lambda_2, 1/3) = \frac{1}{1 + \lambda_2^2}, \quad \cos^2 \theta(\lambda_1, \lambda_2, 2/3) = \frac{1}{1 + \lambda_1^2}$$



## Nuestro método: aplicaciones (2)

$$B(\lambda_1, \lambda_2, t) = (3t(1-t)^2\lambda_1 + 3t^2(1-t)\lambda_2, (1-t)^3 + 2t(1-t)^2 + t^2(1-t))$$

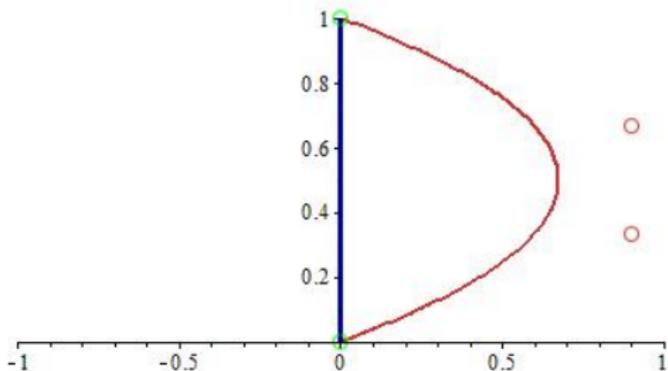
$$\cos^2 \theta(\lambda_1, \lambda_2, 1/3) = \frac{1}{1 + \lambda_2^2}, \quad \cos^2 \theta(\lambda_1, \lambda_2, 2/3) = \frac{1}{1 + \lambda_1^2}$$



## Nuestro método: aplicaciones (2)

$$B(\lambda_1, \lambda_2, t) = (3t(1-t)^2\lambda_1 + 3t^2(1-t)\lambda_2, (1-t)^3 + 2t(1-t)^2 + t^2(1-t))$$

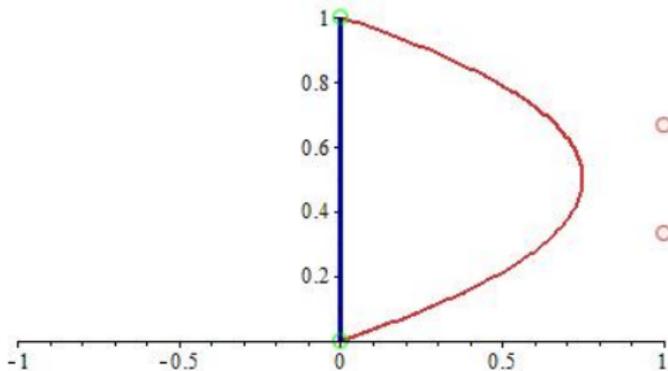
$$\cos^2 \theta(\lambda_1, \lambda_2, 1/3) = \frac{1}{1 + \lambda_2^2}, \quad \cos^2 \theta(\lambda_1, \lambda_2, 2/3) = \frac{1}{1 + \lambda_1^2}$$



## Nuestro método: aplicaciones (2)

$$B(\lambda_1, \lambda_2, t) = (3t(1-t)^2\lambda_1 + 3t^2(1-t)\lambda_2, (1-t)^3 + 2t(1-t)^2 + t^2(1-t))$$

$$\cos^2 \theta(\lambda_1, \lambda_2, 1/3) = \frac{1}{1 + \lambda_2^2}, \quad \cos^2 \theta(\lambda_1, \lambda_2, 2/3) = \frac{1}{1 + \lambda_1^2}$$



A. Carriazo, M. C. Márquez, H. Ugail. Normal approximations of regular curves and surfaces. *FILOMAT* **29** (2015), 457-464.

A. Carriazo, M. C. Márquez, H. Ugail. Normal approximations of regular curves and surfaces. *FILOMAT* **29** (2015), 457-464.

A. Carriazo, M. C. Márquez, H. Ugail. On the curve of minimal length between two points on a Bézier surface. *SKIMA 2019*.

A. Carriazo, M. C. Márquez, H. Ugail. Normal approximations of regular curves and surfaces. *FILOMAT* **29** (2015), 457-464.

A. Carriazo, M. C. Márquez, H. Ugail. On the curve of minimal length between two points on a Bézier surface. *SKIMA 2019*.

A. G. Bedoya, A. Carriazo, M. C. Márquez. Una aproximación geométrica a la aerodinámica de curvas y superficies. En: Proceedings of XI Encuentro Andaluz de Geometría, A. Carriazo, D. de la Fuente, M. C. Márquez, A. Romero (editors). Godel, Granada, 2017.