

Interpolación

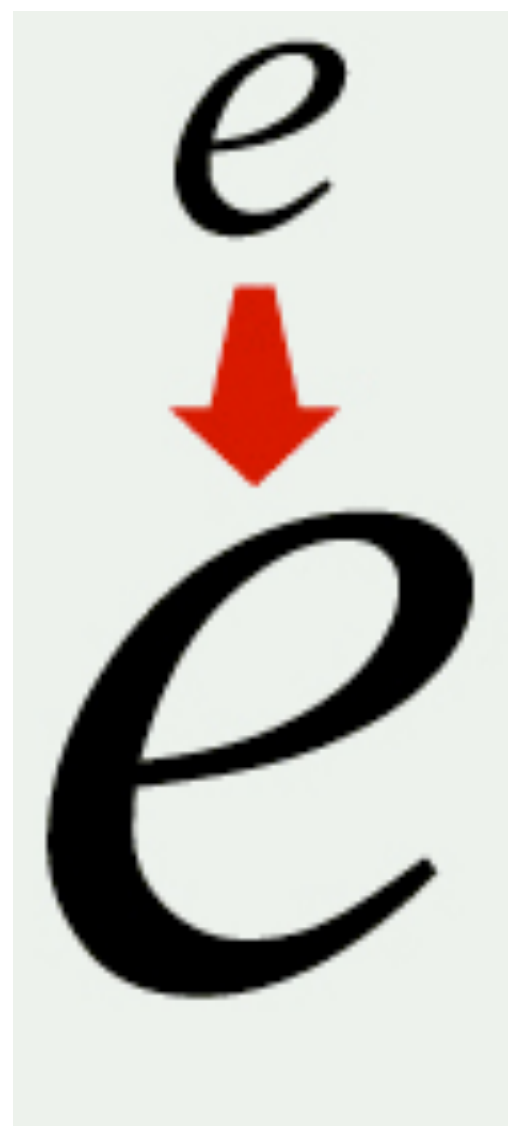
Dan Casas

Motivación

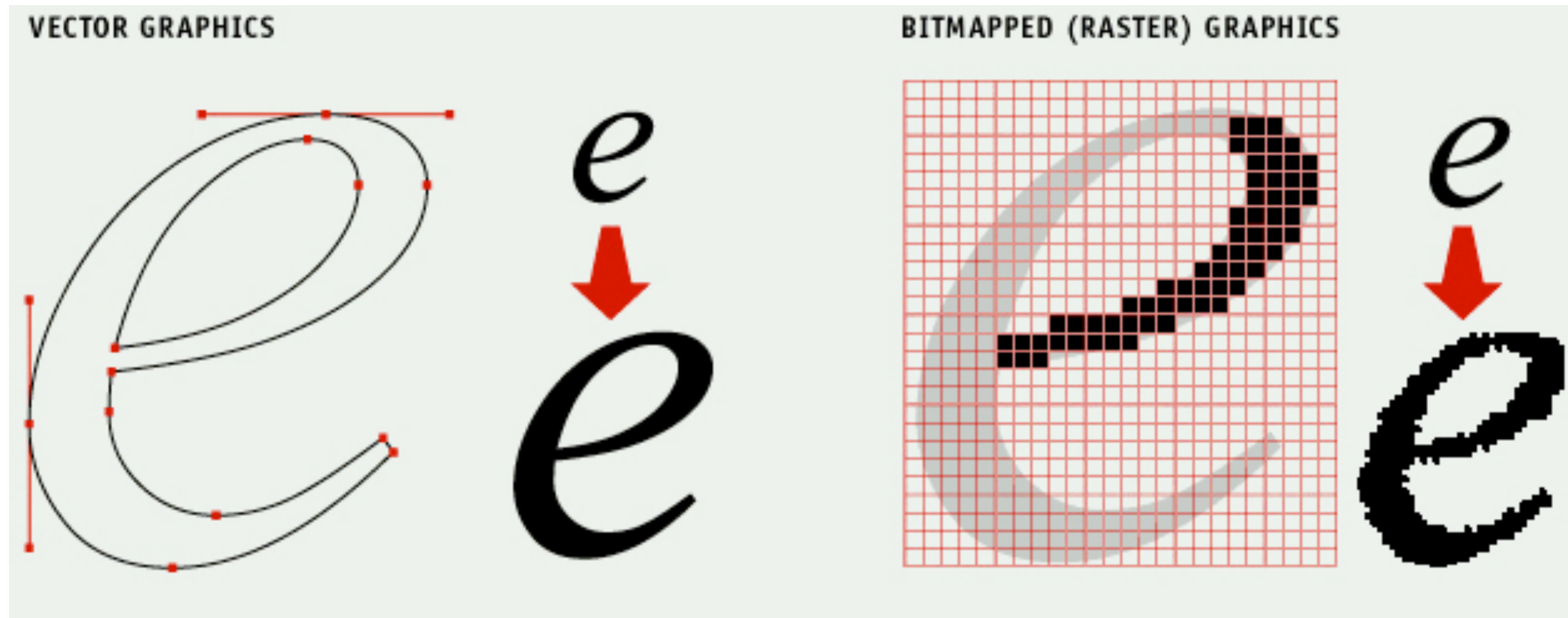
e

e

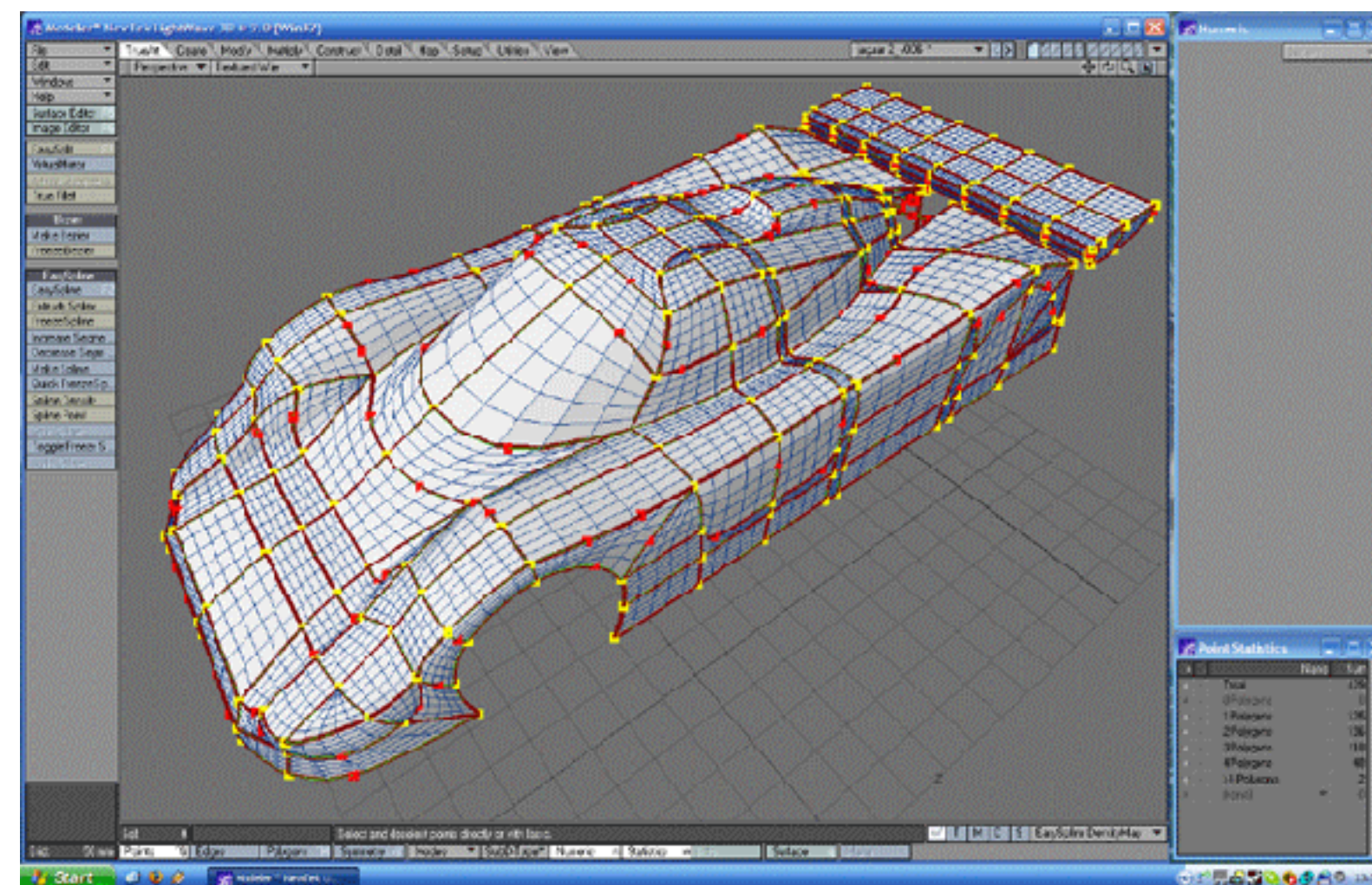
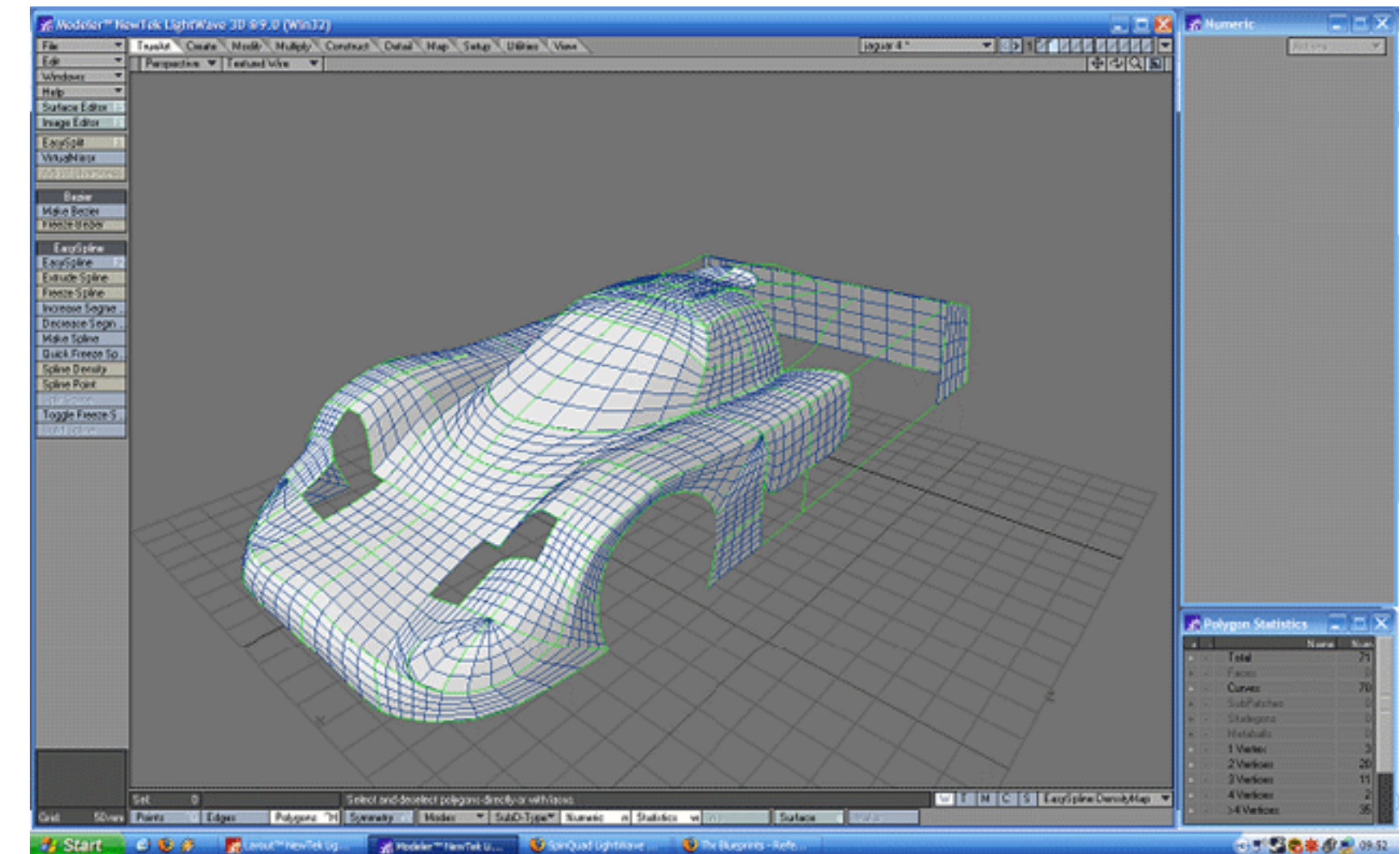
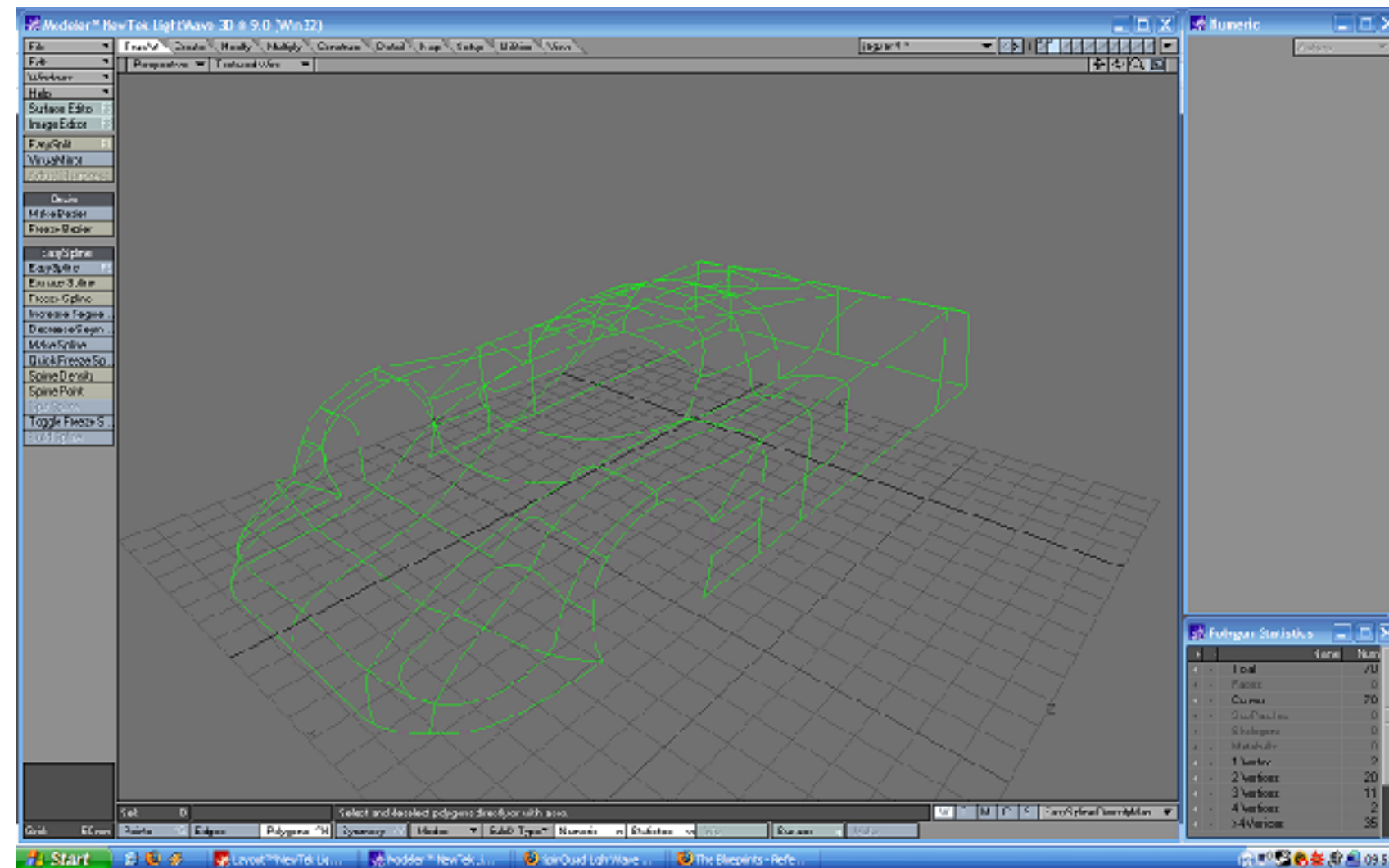
Motivación



Motivación



Motivación



Interpolación

1. Introducción

La mayor parte de los procesos relacionados con la Animación se basan en la Interpolación.

Interpolación

1. Introducción

La mayor parte de los procesos relacionados con la Animación se basan en la Interpolación.



Interpolación

1. Introducción

La mayor parte de los procesos relacionados con la Animación se basan en la Interpolación.



Interpolación

1. Introducción

La mayor parte de los procesos relacionados con la Animación se basan en la Interpolación.



Interpolación

1. Introducción

La mayor parte de los procesos relacionados con la Animación se basan en la Interpolación.



¿Qué necesitamos?

Interpolación

1. Introducción

La mayor parte de los procesos relacionados con la Animación se basan en la Interpolación.



¿Qué necesitamos?

1. Una función de interpolación apropiada.

Interpolación

1. Introducción

La mayor parte de los procesos relacionados con la Animación se basan en la Interpolación.



¿Qué necesitamos?

1. Una **función de interpolación** apropiada.
2. La parametrización de la función en base a la **distancia** recorrida.

Interpolación

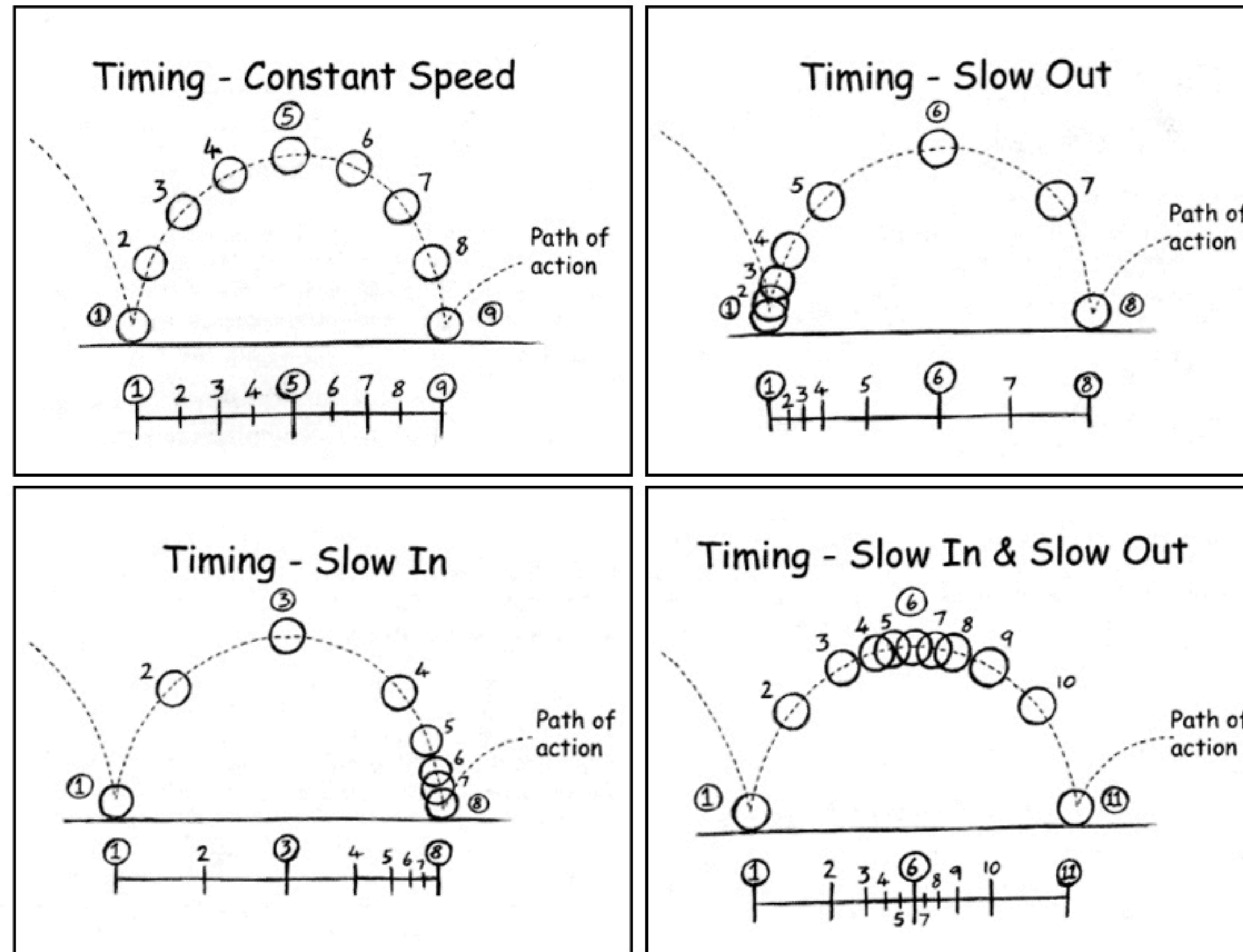
1. Introducción

La mayor parte de los procesos relacionados con la Animación se basan en la Interpolación.



¿Qué necesitamos?

1. Una **función de interpolación** apropiada.
2. La parametrización de la función en base a la **distancia** recorrida.
3. El mantenimiento del control de la posición interpolada en el **tiempo**.



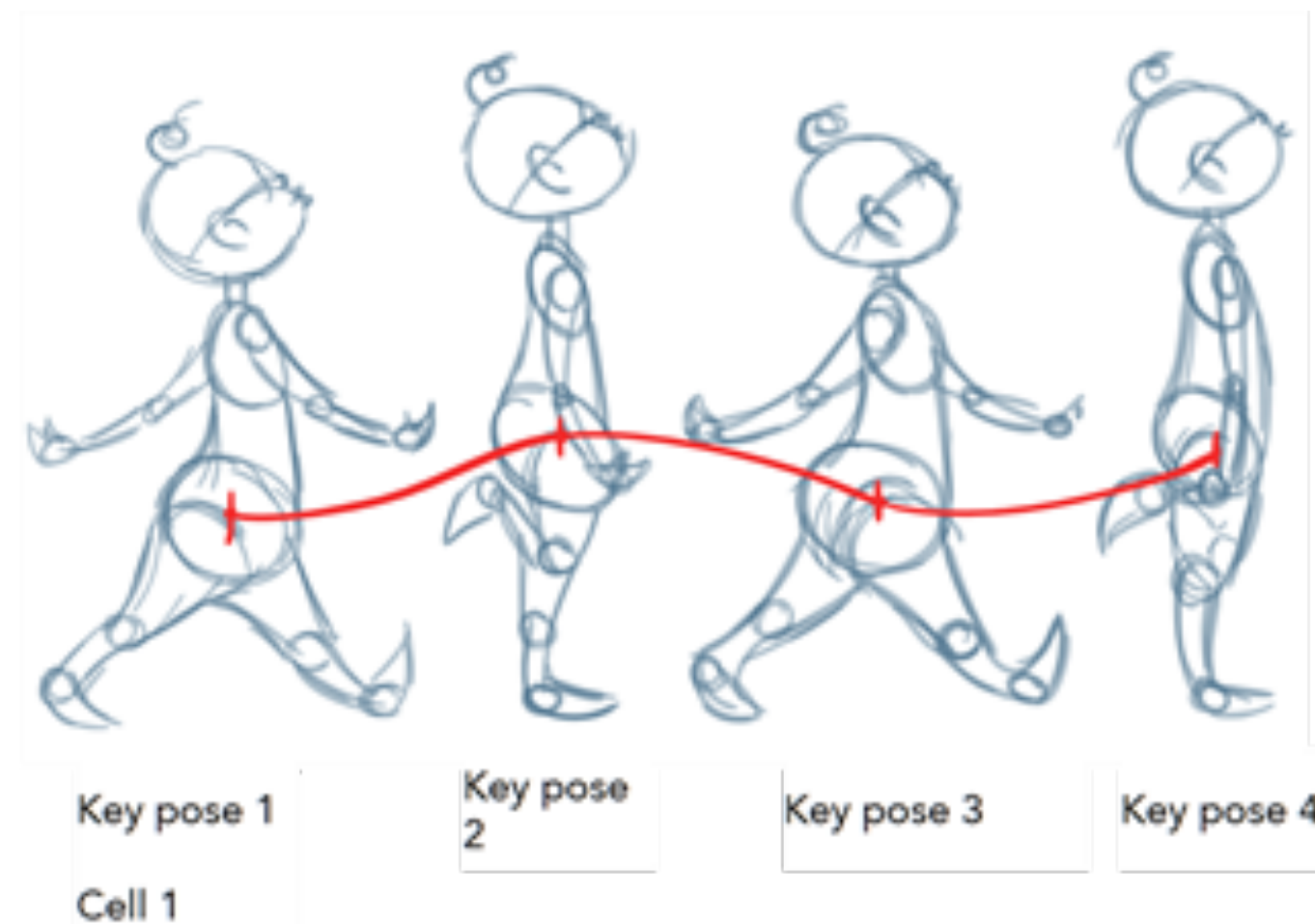
<http://gizma.com/easing/#expo3>

Interpolación

Animador —> valores asociados con un parámetro en unos **keyframes**.

¿Cómo generar el resto de los valores del parámetro entre los keyframes?

Un keyframe en animación es un dibujo que define los puntos de inicio y fin de cualquier **transición**.



Interpolación

Para vídeo, mínimo 24 fps (frames per second)



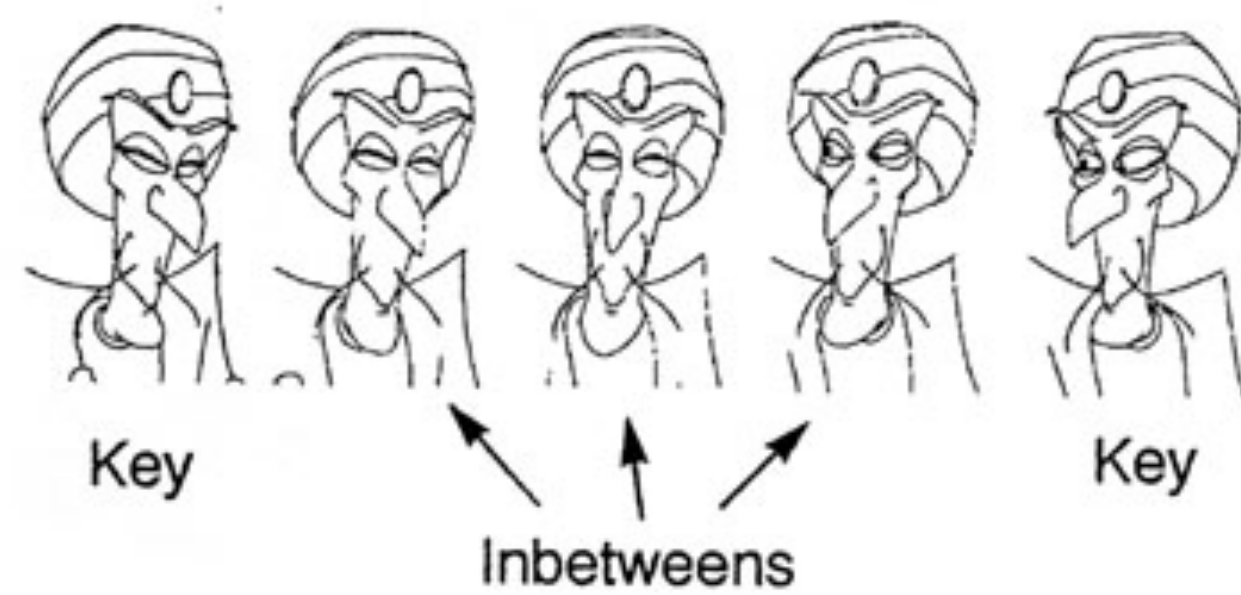
Key



Key

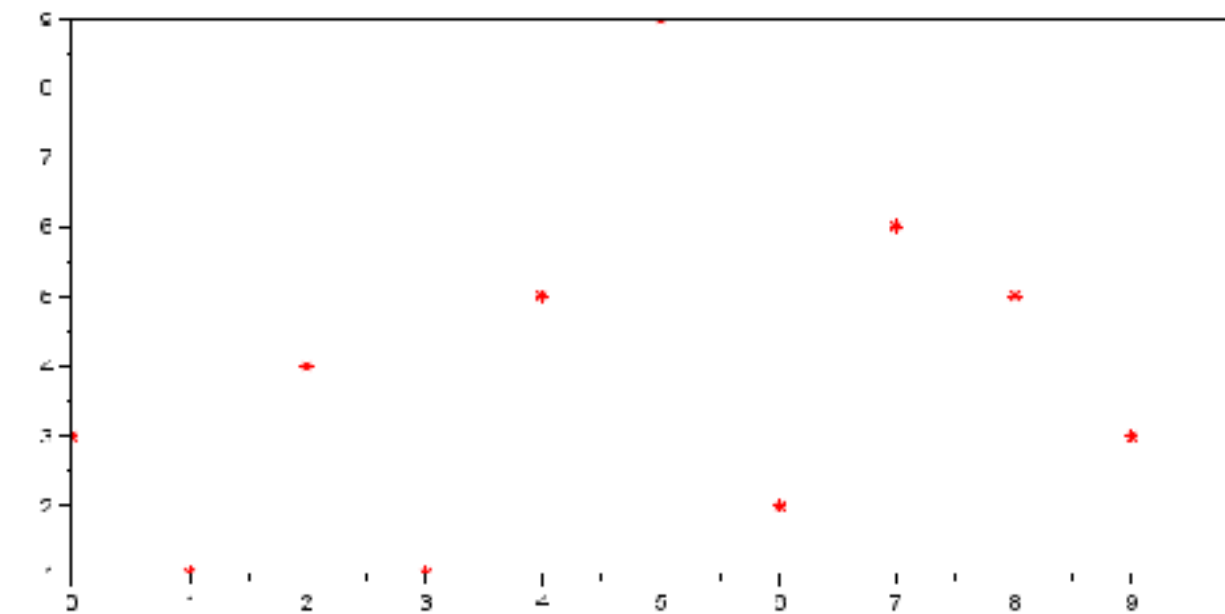
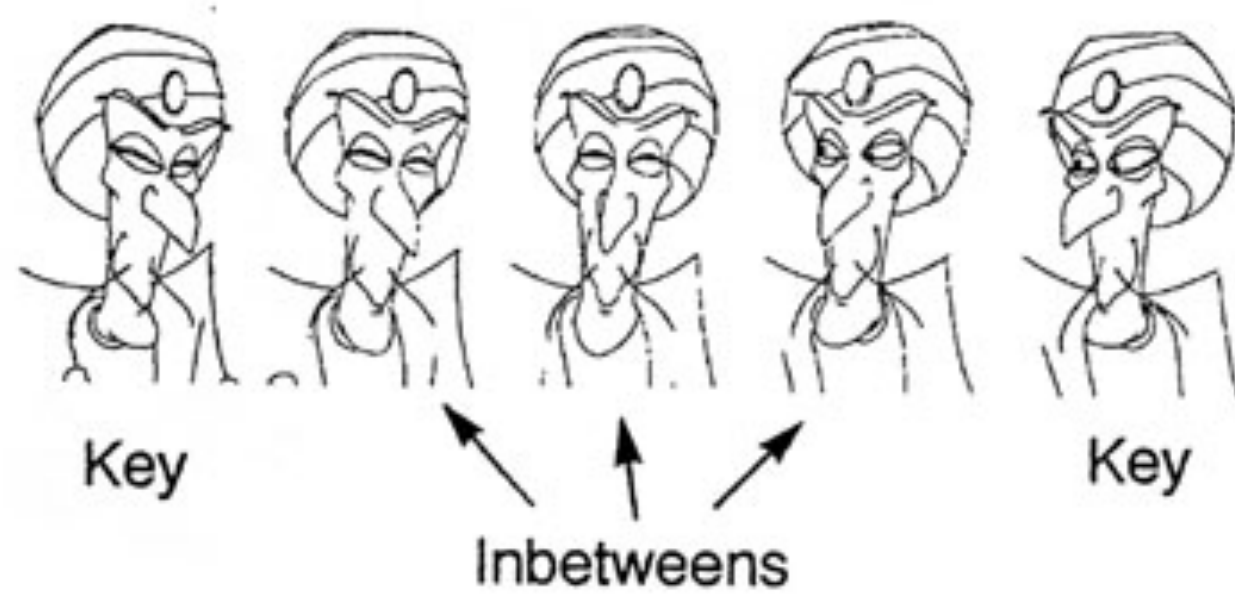
Interpolación

Para vídeo, mínimo 24 fps (frames per second)



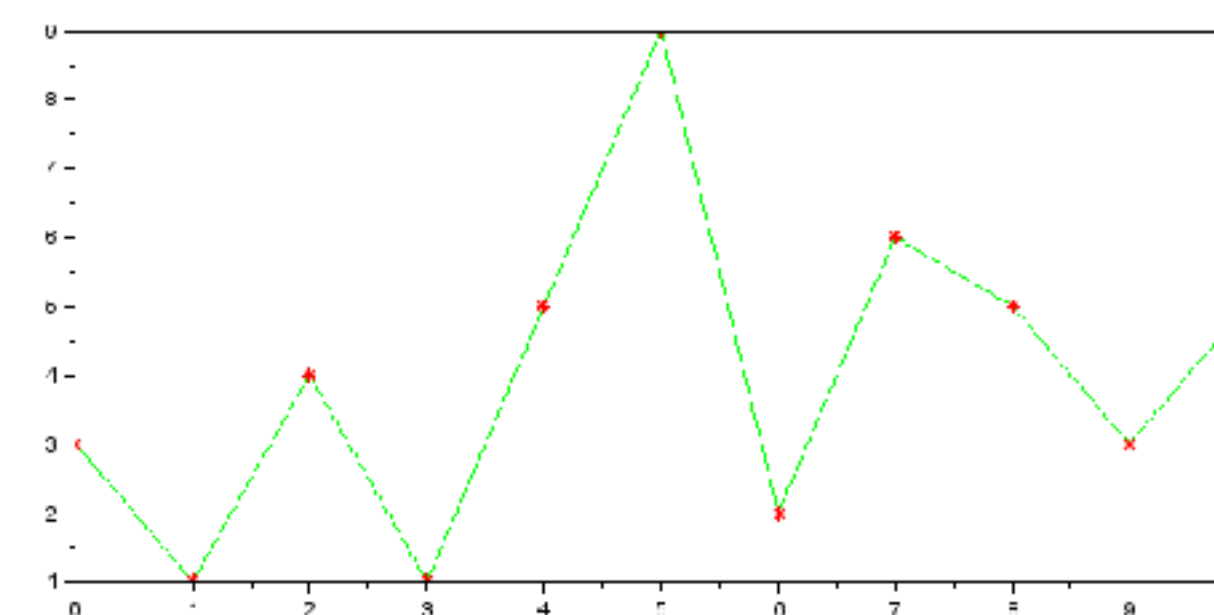
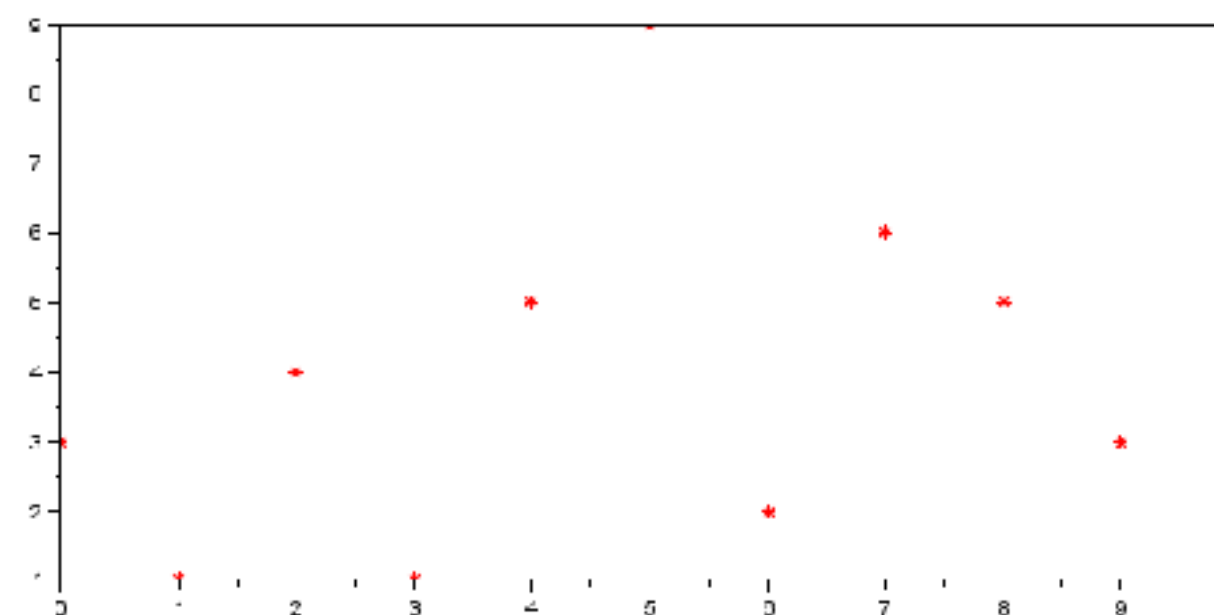
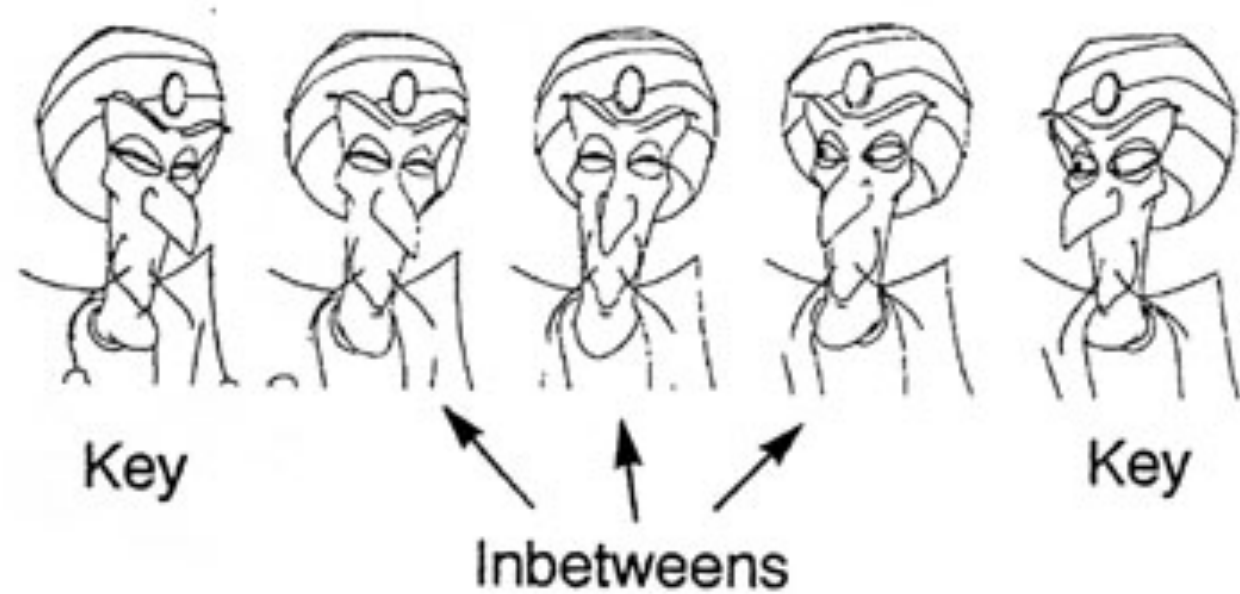
Interpolación

Para vídeo, mínimo 24 fps (frames per second)



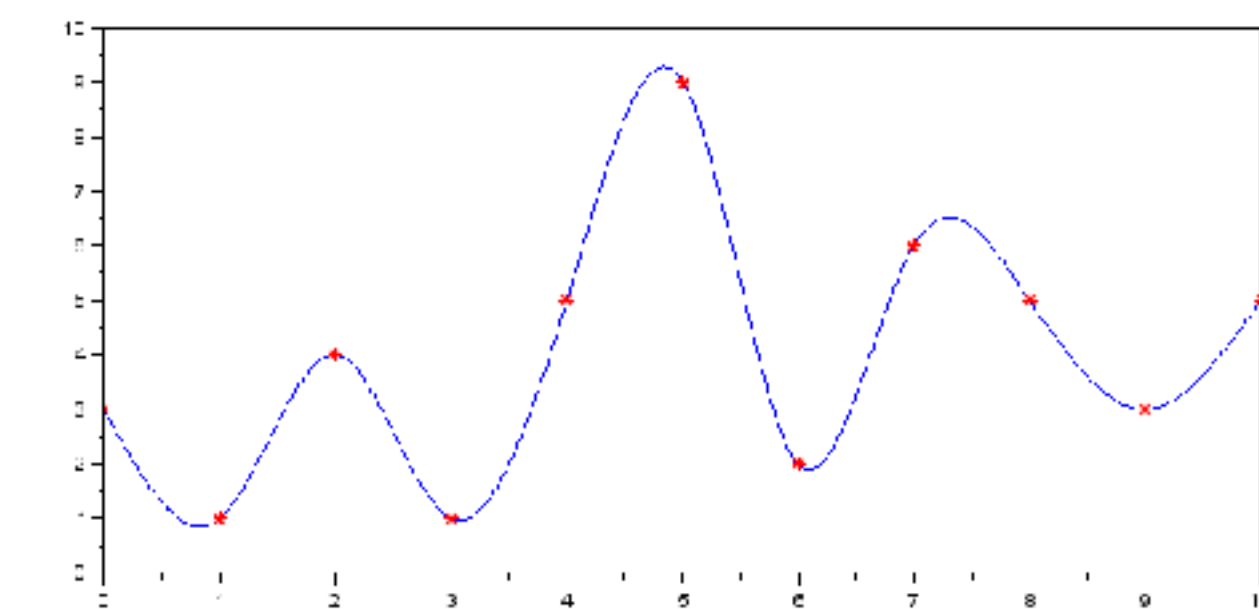
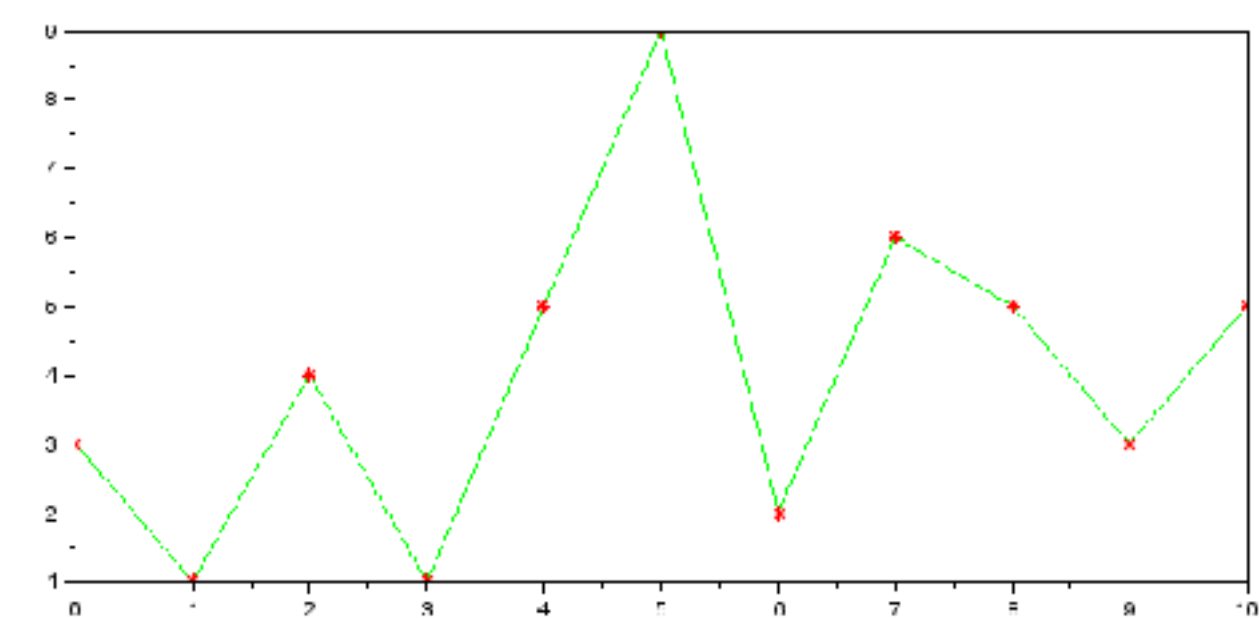
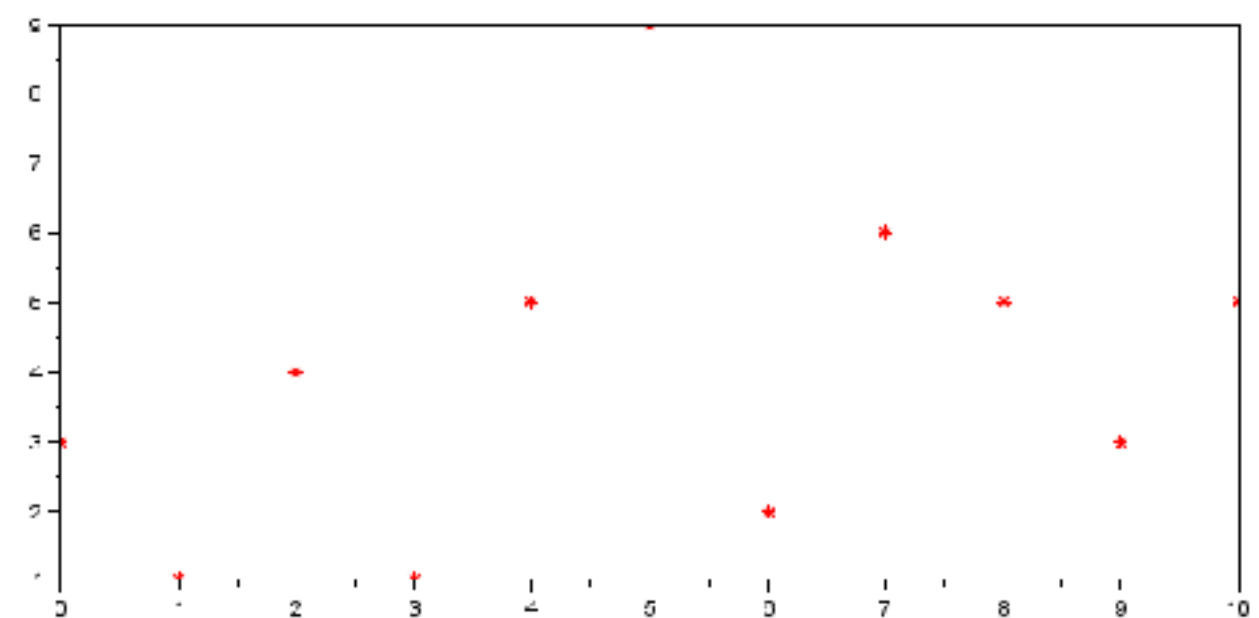
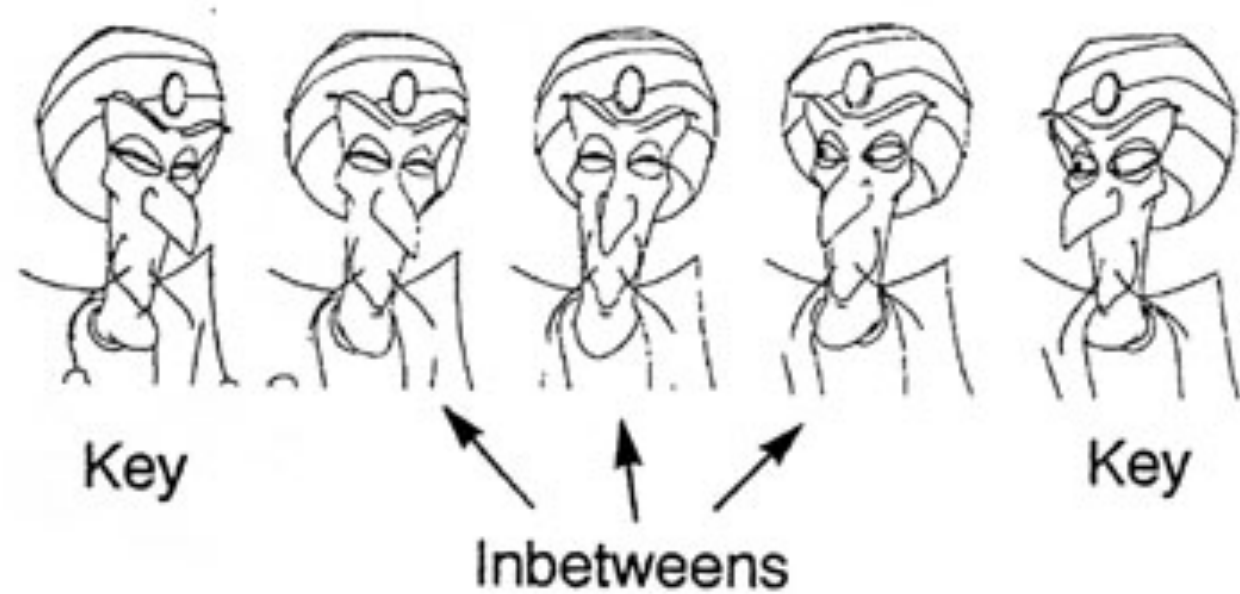
Interpolación

Para vídeo, mínimo 24 fps (frames per second)



Interpolación

Para vídeo, mínimo 24 fps (frames per second)



Interpolación

2.- Búsqueda de la función apropiada.

¿Cómo elegir la técnica de interpolación más adecuada?
¿una vez seleccionada, cómo aplicarla a mi caso de animación?

Interpolación vs Aproximación



An interpolating spline in which the spline passes through the interior control points



An approximating spline in which only the endpoints are interpolated; the interior control points are used only to design the curve

Interpolación

En el caso de ser puntos de la curva, forman parte de un spline de interpolación.

Un spline es una curva diferenciable definida en porciones mediante polinomios. Veremos que se van a usar normalmente polinomios de orden bajo para no realizar cambios bruscos en la forma a aproximar.

En el caso de ser una aproximación, los puntos dados son un ejemplo de cómo diseñar el spline por aproximación, y en cualquier caso, permite más margen en el diseño de la curva.

Interpolación

Interpolación:

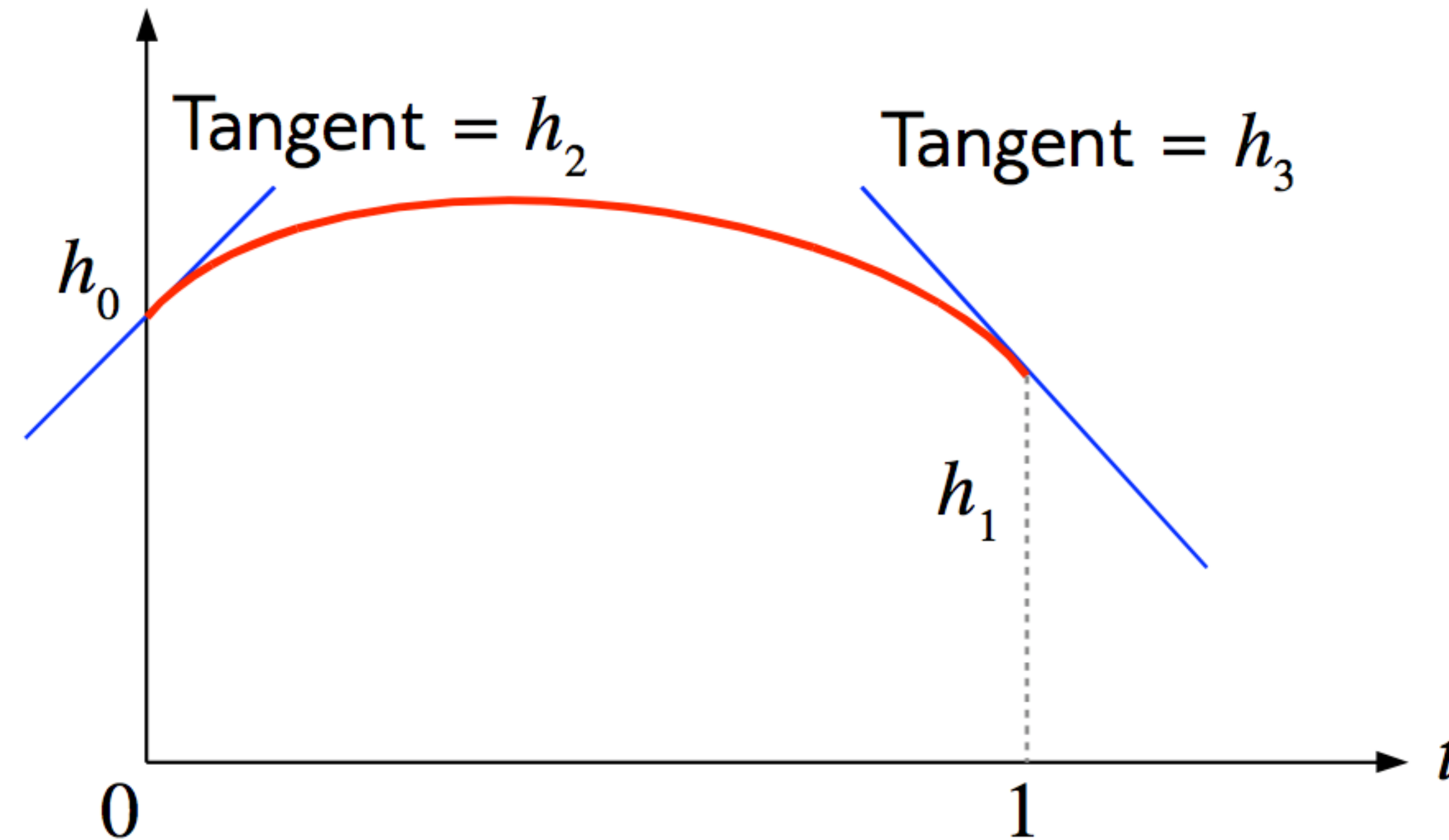
- **Hermite:** Requiere saber tangentes en los puntos finales.
- **Polinomio interpolador de Lagrange**
- **Splines, Catmull-Rom:** Sólo necesita saber los puntos por donde pasaría la curva. Defino por mi mismo las tangentes.

Aproximaciones:

- **Curvas de Bezier.**
- **B-splines.**

Cubic Hermite Interpolation

- Specify positions h_0, h_1 and tangents (slopes, derivatives) h_2, h_3 at two points: $t = 0$ and $t = 1$



Cubic Hermite Interpolation

$$P(t) = at^3 + bt^2 + ct + d$$

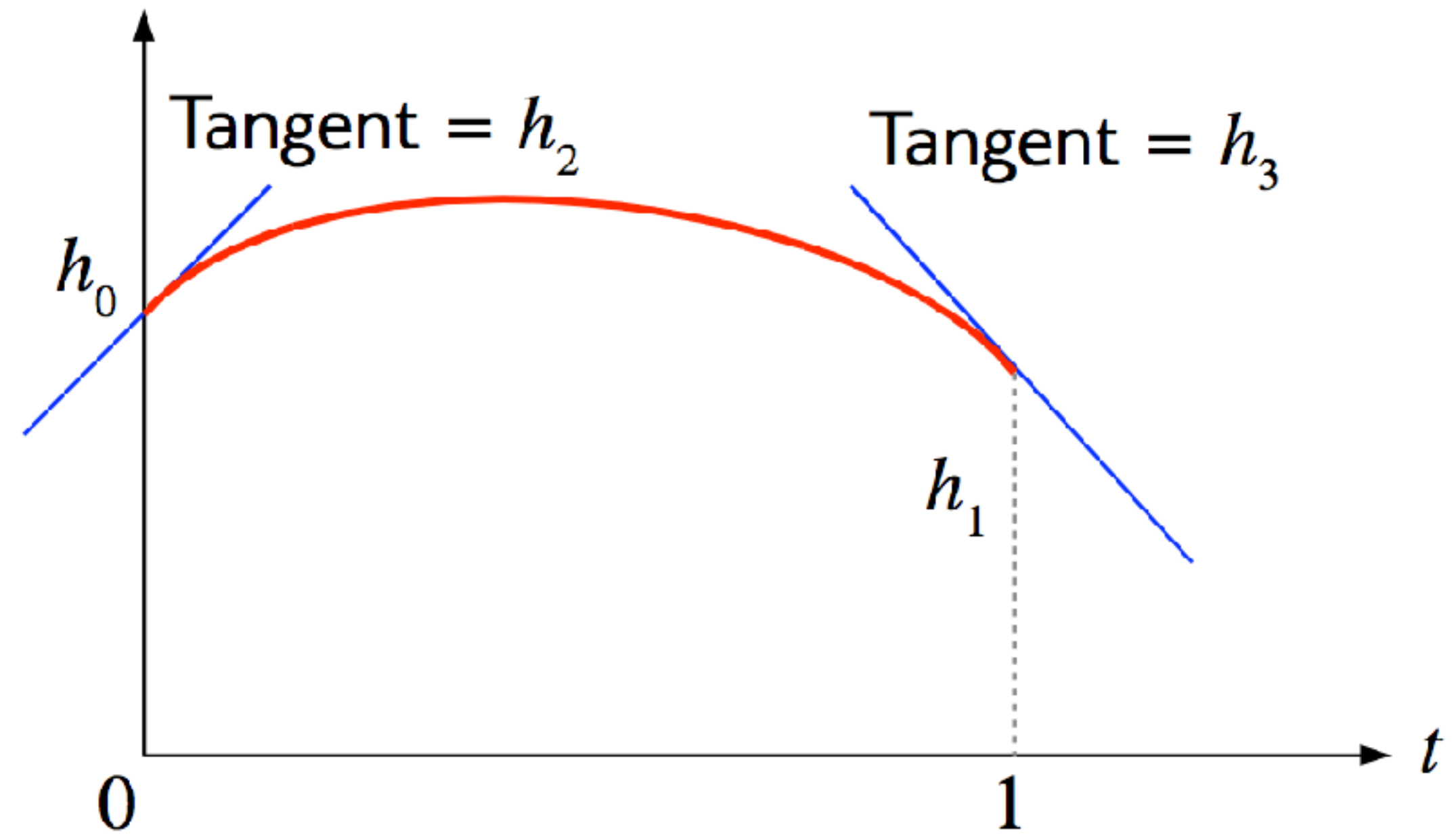
$$P'(t) = 3at^2 + 2bt + c$$

$$h_0 = P(0) = d$$

$$h_1 = P(1) = a + b + c + d$$

$$h_2 = P'(0) = c$$

$$h_3 = P'(1) = 3a + 2b + c$$



Matrix Representation

$$h_0 = d$$

$$h_1 = a + b + c + d$$

$$h_2 = c$$

$$h_3 = 3a + 2b + c$$

Hermite constraint matrix

$$\underbrace{\begin{bmatrix} h_0 \\ h_1 \\ h_2 \\ h_3 \end{bmatrix}}_{\mathbf{h}} = \underbrace{\begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix}}_{\mathbf{C}} \underbrace{\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}}_{\mathbf{a}}$$

Matrix Representation

$$\mathbf{h} = \mathbf{C}\mathbf{a} \Rightarrow \mathbf{a} = \mathbf{C}^{-1}\mathbf{h}$$

$$\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} h_0 \\ h_1 \\ h_2 \\ h_3 \end{bmatrix}$$

Hermite basis matrix

Matrix Representation of Polynomials

$$P(t) = \begin{bmatrix} a & b & c & d \end{bmatrix} \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix}$$

Matrix Representation of Polynomials

$$P(t) = \begin{bmatrix} a & b & c & d \end{bmatrix} \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix}$$

$$P(t) = \begin{bmatrix} h_0 & h_1 & h_2 & h_3 \end{bmatrix} \underbrace{\begin{bmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix}}_{(\mathbf{C}^{-1})^T} \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix}$$

Matrix Representation of Polynomials

$$P(t) = \begin{bmatrix} a & b & c & d \end{bmatrix} \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix}$$

$$P(t) = \begin{bmatrix} h_0 & h_1 & h_2 & h_3 \end{bmatrix} \underbrace{\begin{bmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix}}_{(\mathbf{C}^{-1})^T} \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix}$$

$$P(t) = \begin{bmatrix} h_0 & h_1 & h_2 & h_3 \end{bmatrix} \underbrace{\begin{bmatrix} H_0(t) \\ H_1(t) \\ H_2(t) \\ H_3(t) \end{bmatrix}}_{\text{Hermite basis functions}}$$

Matrix Representation of Polynomials

$$P(t) = \begin{bmatrix} a & b & c & d \end{bmatrix} \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix}$$

$$P(t) = \begin{bmatrix} h_0 & h_1 & h_2 & h_3 \end{bmatrix} \underbrace{\begin{bmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix}}_{(\mathbf{C}^{-1})^T} \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix}$$

$$P(t) = \begin{bmatrix} h_0 & h_1 & h_2 & h_3 \end{bmatrix} \underbrace{\begin{bmatrix} H_0(t) \\ H_1(t) \\ H_2(t) \\ H_3(t) \end{bmatrix}}_{\text{Hermite basis functions}}$$

$$P(t) = \sum_{i=0}^3 h_i H_i(t)$$

Matrix Representation of Polynomials

$$P(t) = \begin{bmatrix} a & b & c & d \end{bmatrix} \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix}$$

$$P(t) = \begin{bmatrix} h_0 & h_1 & h_2 & h_3 \end{bmatrix} \underbrace{\begin{bmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix}}_{(\mathbf{C}^{-1})^T} \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix}$$

$$P(t) = \begin{bmatrix} h_0 & h_1 & h_2 & h_3 \end{bmatrix} \underbrace{\begin{bmatrix} H_0(t) \\ H_1(t) \\ H_2(t) \\ H_3(t) \end{bmatrix}}_{\text{Hermite basis functions}}$$

$$P(t) = \sum_{i=0}^3 h_i H_i(t)$$

Matrix Representation of Polynomials

$$P(t) = [a \ b \ c \ d] \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix}$$

$$P(t) = [h_0 \ h_1 \ h_2 \ h_3] \underbrace{\begin{bmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix}}_{(\mathbf{C}^{-1})^T} \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix}$$

$$P(t) = [h_0 \ h_1 \ h_2 \ h_3] \underbrace{\begin{bmatrix} H_0(t) \\ H_1(t) \\ H_2(t) \\ H_3(t) \end{bmatrix}}$$

Hermite basis functions

$$P(t) = \sum_{i=0}^3 h_i H_i(t)$$

Matrix Representation of Polynomials

$$P(t) = \begin{bmatrix} a & b & c & d \end{bmatrix} \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix}$$

$$P(t) = \begin{bmatrix} h_0 & h_1 & h_2 & h_3 \end{bmatrix} \underbrace{\begin{bmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix}}_{(C^{-1})^T} \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix}$$

$$P(t) = \begin{bmatrix} h_0 & h_1 & h_2 & h_3 \end{bmatrix} \underbrace{\begin{bmatrix} H_0(t) \\ H_1(t) \\ H_2(t) \\ H_3(t) \end{bmatrix}}_{\text{Hermite basis functions}}$$

$$P(t) = \sum_{i=0}^3 h_i H_i(t)$$

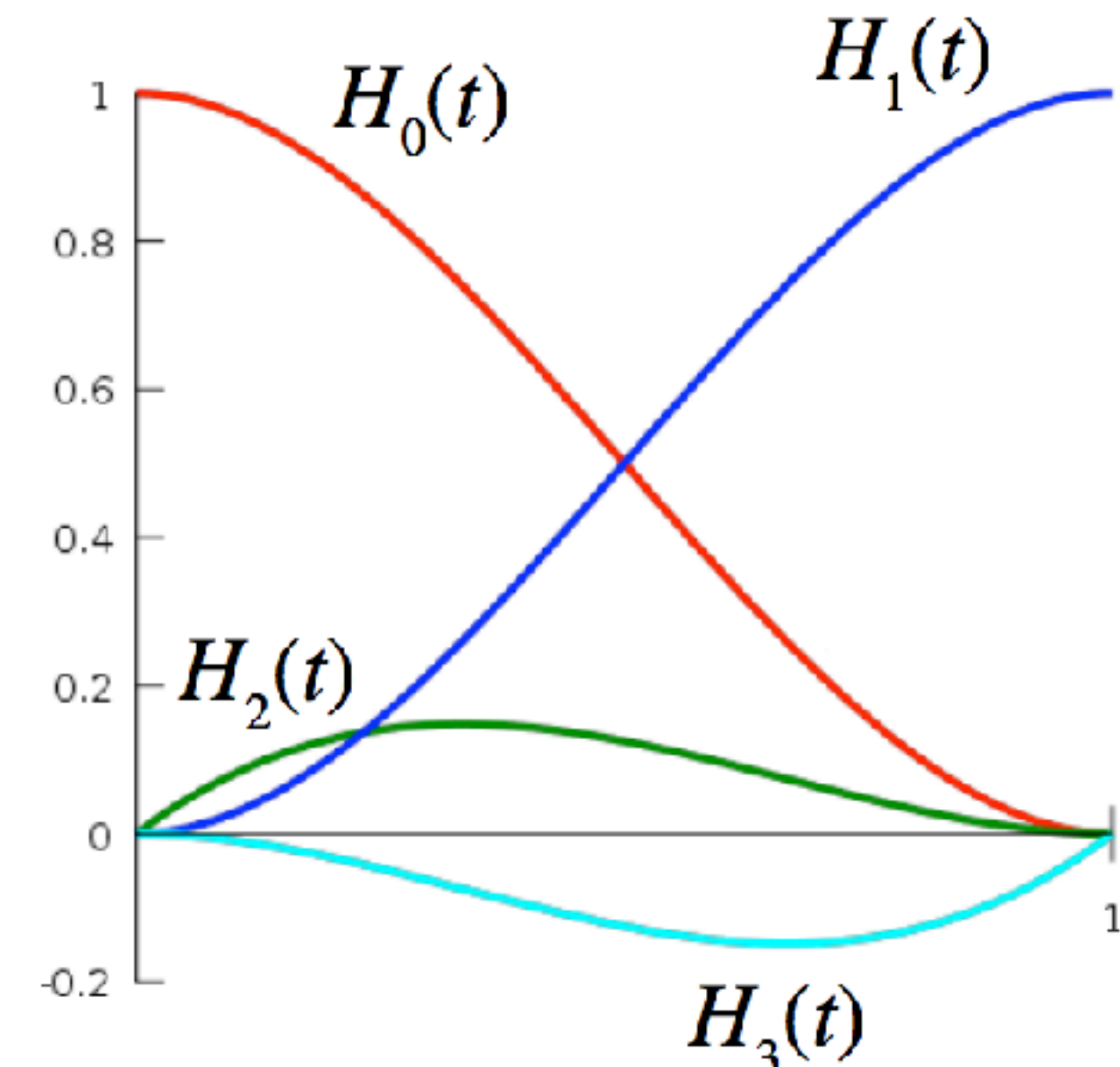
Hermite Basis Functions

$$H_0(t) = 2t^3 - 3t^2 + 1$$

$$H_1(t) = -2t^3 + 3t^2$$

$$H_2(t) = t^3 - 2t^2 + t$$

$$H_3(t) = t^3 - t^2$$



$$P(t) = \sum_{i=0}^3 h_i H_i(t)$$

Interpolación Hermite

Propiedades importantes

- Requiere de la disponibilidad de las primeras derivadas
- Si hay más de 2 puntos la continuidad se cumple de manera que la tangente a la curva i en el punto final es la tangente a la curva $i+1$ en el punto inicial.

Interpolación

Interpolación Hermite. Ejemplo 1.

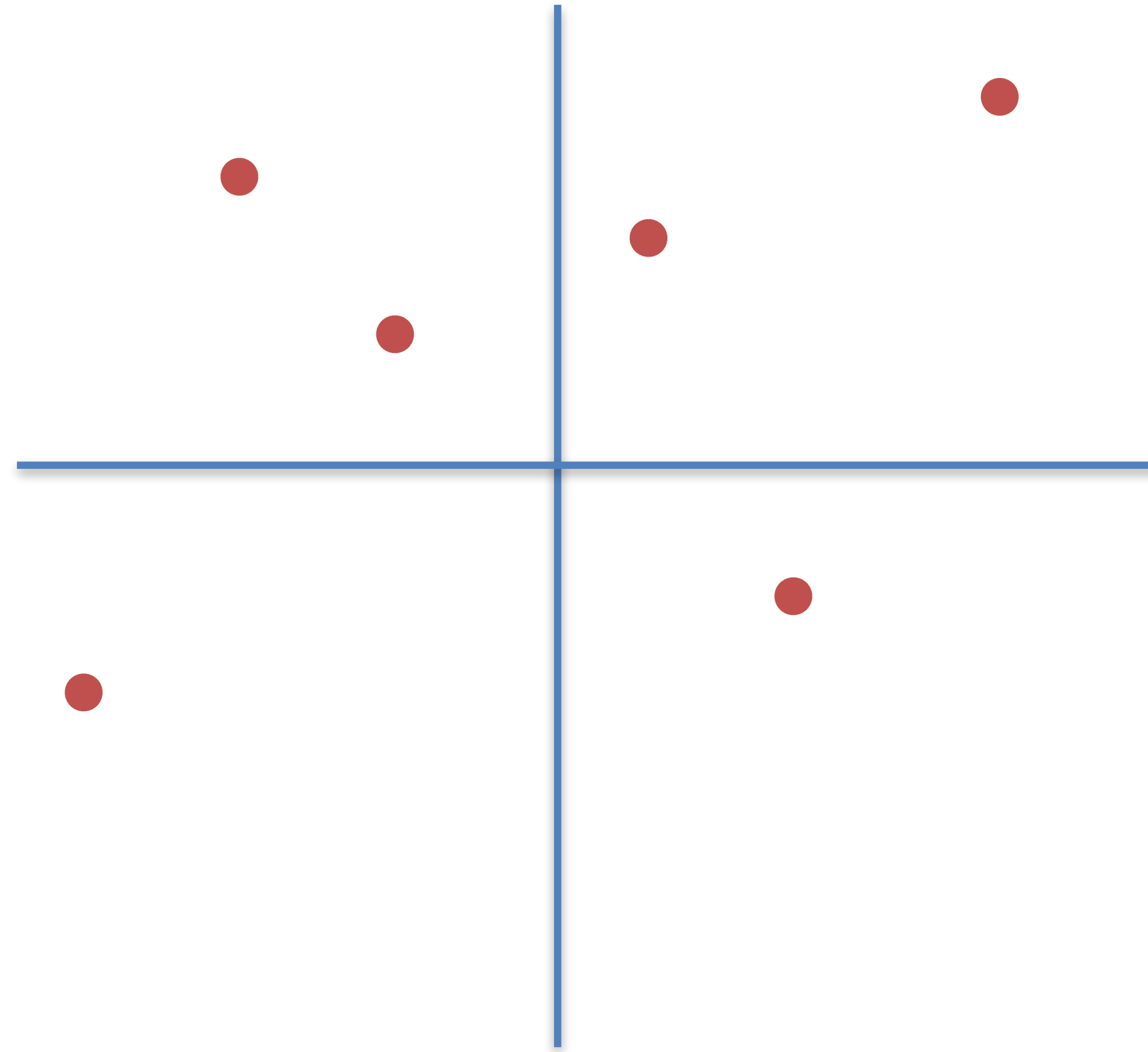
Encontrar un polinomio $P(x) \in \mathcal{P}_3(\mathbb{R})$ tal que:

$$P(0) = 1, \quad P(1) = 1, \quad P'(0) = 0, \quad P'(1) = 1.$$

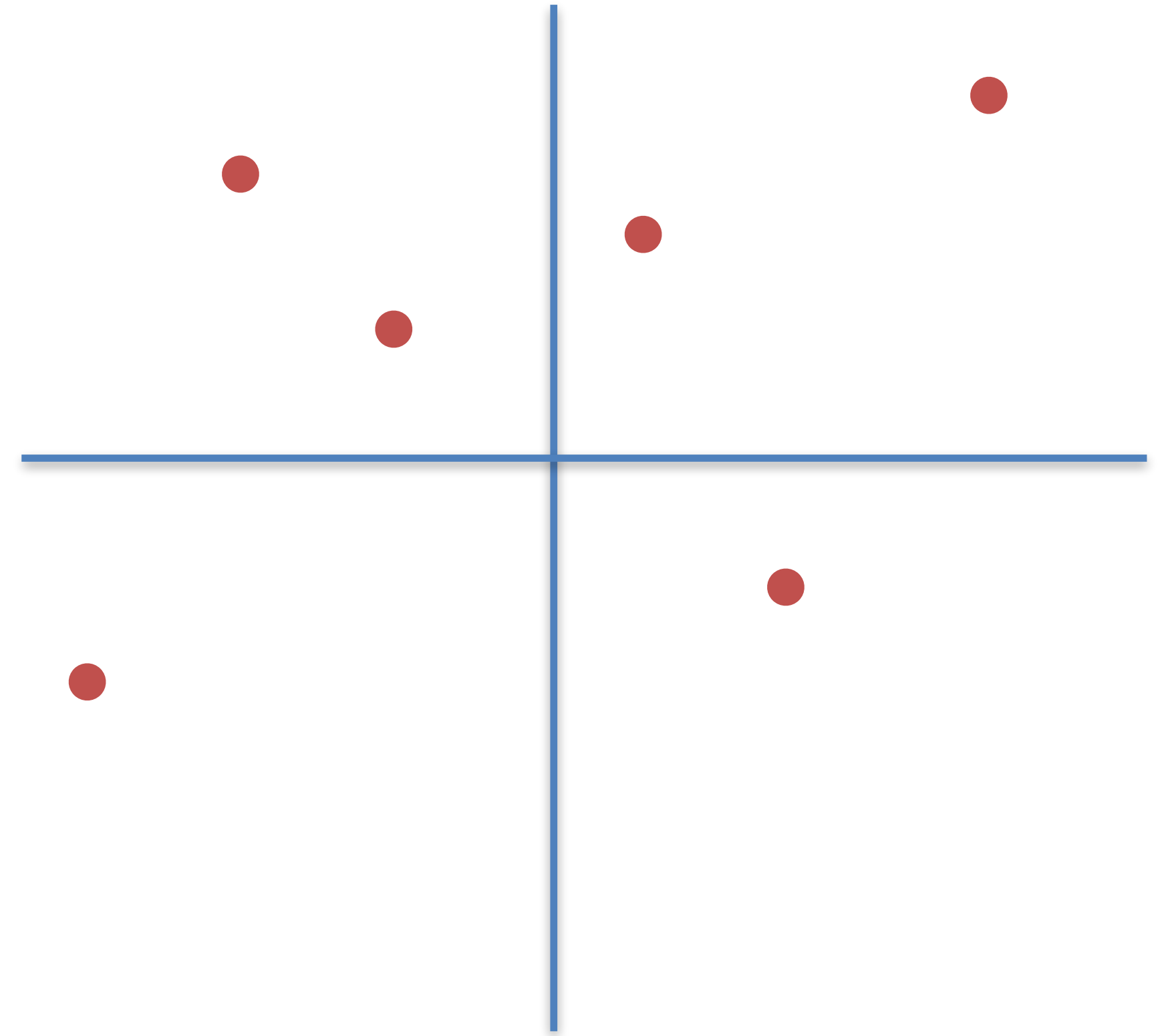
MATLAB demo

or <https://octave-online.net/>

Polinomio interpolado de Lagrange



Polinomio interpolado de Lagrange



Polinomio interpolado de Lagrange

Dada la tabla de valores

x	x_0	x_1	\cdots	x_n
y	y_0	y_1	\cdots	y_n

donde

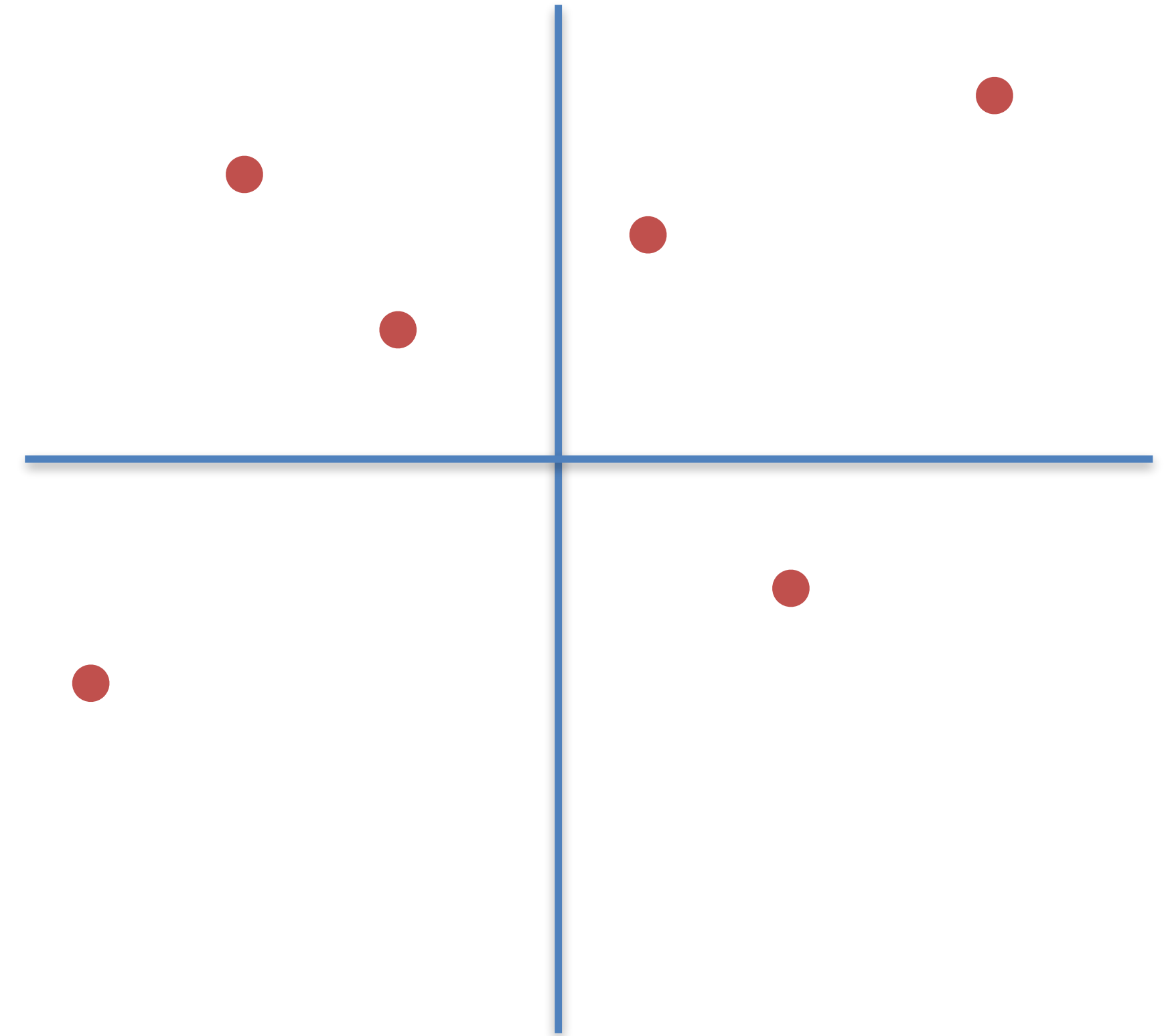
- x_0, x_1, \dots, x_n son $n + 1$ abscisas distintas.
- y_0, y_1, \dots, y_n son $n + 1$ valores arbitrarios.

Queremos determinar un polinomio de grado $\leq n$

$$P_n(x) = a_0 + a_1x + \cdots + a_nx^n,$$

que verifique las $n + 1$ condiciones

$$P_n(x_j) = y_j, \quad \text{para } j = 0, 1, \dots, n.$$



Polinomio interpolado de Lagrange

Forma de Lagrange

$$P_n(x) = l_0(x) y_0 + l_1(x) y_1 + \cdots + l_n(x) y_n.$$

Polinomio interpolado de Lagrange

Forma de Lagrange

$$P_n(x) = l_0(x)y_0 + l_1(x)y_1 + \cdots + l_n(x)y_n.$$

Polinomio interpolado de Lagrange

Forma de Lagrange

$$P_n(x) = l_0(x)y_0 + l_1(x)y_1 + \cdots + l_n(x)y_n.$$

Los polinomios $l_j(x)$ se denominan *polinomios componentes* y se caracterizan por las propiedades

$$l_j(x_k) = \delta_{jk} = \begin{cases} 1 & \text{si } j = k, \\ 0 & \text{si } j \neq k. \end{cases}$$

Es decir, el polinomio componente $l_j(x)$ vale 1 en su nodo x_j y se anula en los restantes.

Polinomio interpolado de Lagrange

Como $l_j(x)$ se anula para

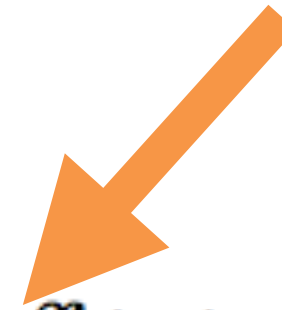
$$x_0, x_1, \dots, x_{j-1}, x_{j+1}, \dots, x_n,$$

Polinomio interpolado de Lagrange

Como $l_j(x)$ se anula para

$x_0, x_1, \dots, x_{j-1}, x_{j+1}, \dots, x_n,$

No se anula para x_j



Polinomio interpolado de Lagrange

Como $l_j(x)$ se anula para

$$x_0, x_1, \dots, x_{j-1}, x_{j+1}, \dots, x_n,$$

No se anula para x_j



y es de grado $\leq n$, debe ser de la forma

$$l_j(x) = C (x - x_0) (x - x_1) \cdots (x - x_{j-1}) (x - x_{j+1}) \cdots (x - x_n).$$

Polinomio interpolado de Lagrange

Como $l_j(x)$ se anula para

$$x_0, x_1, \dots, x_{j-1}, x_{j+1}, \dots, x_n,$$

No se anula para x_j



y es de grado $\leq n$, debe ser de la forma

$$l_j(x) = C (x - x_0) (x - x_1) \cdots (x - x_{j-1}) (x - x_{j+1}) \cdots (x - x_n).$$

Para cumplir $l_j(x_j) = 1$, debe ser

$$C = \frac{1}{(x_j - x_0) (x_j - x_1) \cdots (x_j - x_{j-1}) (x_j - x_{j+1}) (x_j - x_n)},$$

Polinomio interpolado de Lagrange

Como $l_j(x)$ se anula para

$$x_0, x_1, \dots, x_{j-1}, x_{j+1}, \dots, x_n,$$

No se anula para x_j



y es de grado $\leq n$, debe ser de la forma

$$l_j(x) = C (x - x_0) (x - x_1) \cdots (x - x_{j-1}) (x - x_{j+1}) \cdots (x - x_n).$$

Para cumplir $l_j(x_j) = 1$, debe ser

$$C = \frac{1}{(x_j - x_0) (x_j - x_1) \cdots (x_j - x_{j-1}) (x_j - x_{j+1}) (x_j - x_n)},$$

por lo tanto

$$l_j(x) = \frac{(x - x_0) (x - x_1) \cdots (x - x_{j-1}) (x - x_{j+1}) \cdots (x - x_n)}{(x_j - x_0) (x_j - x_1) \cdots (x_j - x_{j-1}) (x_j - x_{j+1}) (x_j - x_n)}.$$

Polinomio interpolado de Lagrange

Determina el polinomio interpolador de la tabla

x	1	2	3
y	1	$1/2$	$1/3$

Los polinomios componentes son

$$\begin{array}{|c|c|c|} \hline * & & \\ \hline x_0 = 1 & x_1 = 2 & x_2 = 3 \\ \hline \end{array} \Rightarrow l_0(x) = \frac{(x-2)(x-3)}{(1-2)(1-3)} = \frac{1}{2}(x-2)(x-3).$$

$$\begin{array}{|c|c|c|} \hline & * & \\ \hline x_0 = 1 & x_1 = 2 & x_2 = 3 \\ \hline \end{array} \Rightarrow l_1(x) = \frac{(x-1)(x-3)}{(2-1)(2-3)} = -(x-1)(x-3).$$

$$\begin{array}{|c|c|c|} \hline & & * \\ \hline x_0 = 1 & x_1 = 2 & x_2 = 3 \\ \hline \end{array} \Rightarrow l_2(x) = \frac{(x-1)(x-2)}{(3-1)(3-2)} = \frac{1}{2}(x-1)(x-2).$$

El polinomio interpolador es

$$P_2(x) = l_0(x) \cdot 1 + l_1(x) \cdot \frac{1}{2} + l_2(x) \cdot \frac{1}{3}.$$

Polinomio interpolado de Lagrange

El polinomio interpolador es

$$P_2(x) = l_0(x) \cdot 1 + l_1(x) \cdot \frac{1}{2} + l_2(x) \frac{1}{3}.$$

$$P_2(x) = \frac{1}{2} (x - 2) (x - 3) - \frac{1}{2} (x - 1) (x - 3) + \frac{1}{6} (x - 1) (x - 2). \quad (6)$$

La tabla de valores es la misma que la del Ejemplo 6.1, si operamos en (6), resulta

$$P_2(x) = \frac{1}{6}x^2 - x + \frac{11}{6},$$

Polinomio interpolado de Lagrange

El siguiente gráfico muestra la representación conjunta de la función

$$f(x) = 1/x$$

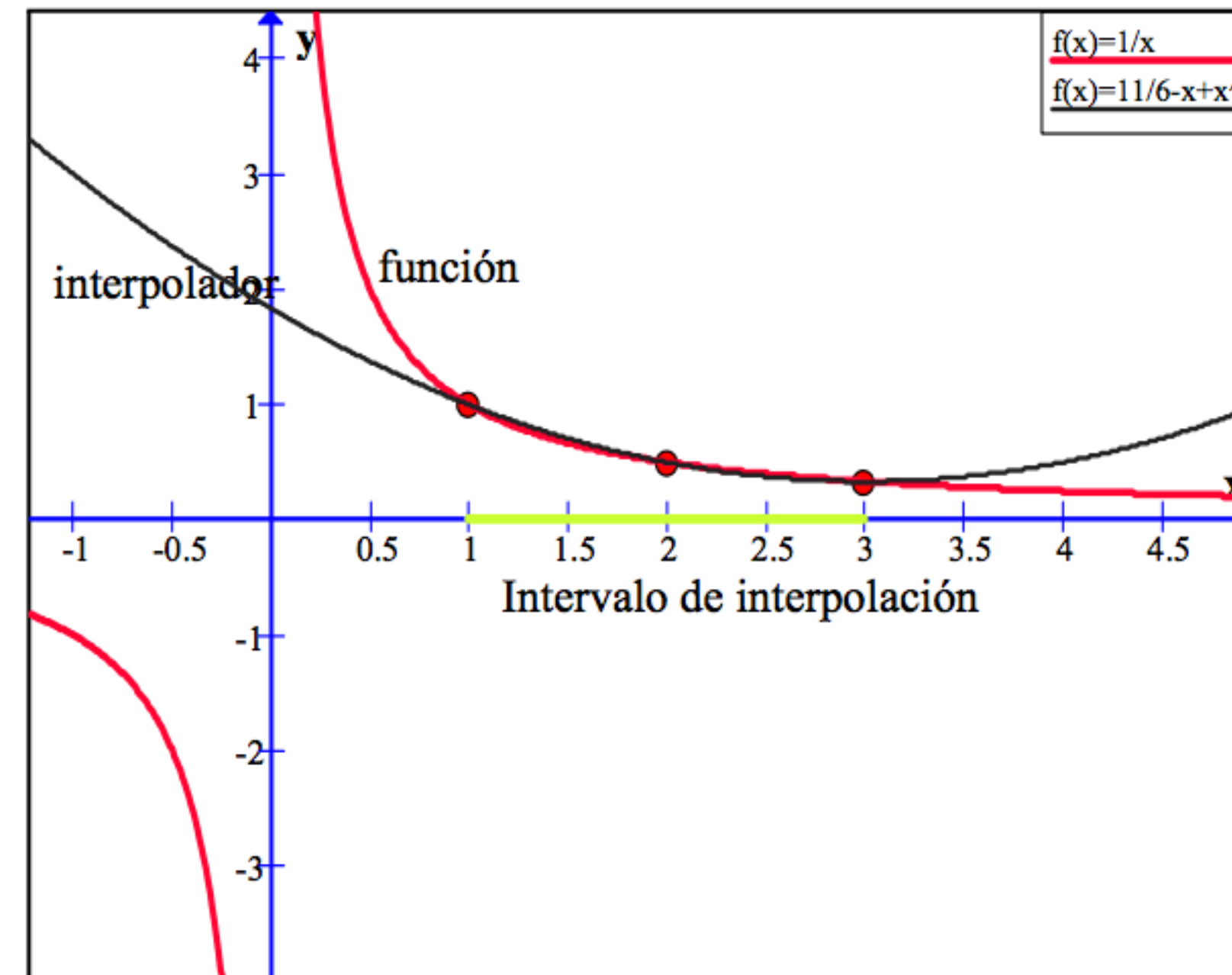


y el polinomio interpolador calculado en el Ejemplo 6.1

$$P_2(x) = \frac{11}{6} - x + \frac{x^2}{6}.$$

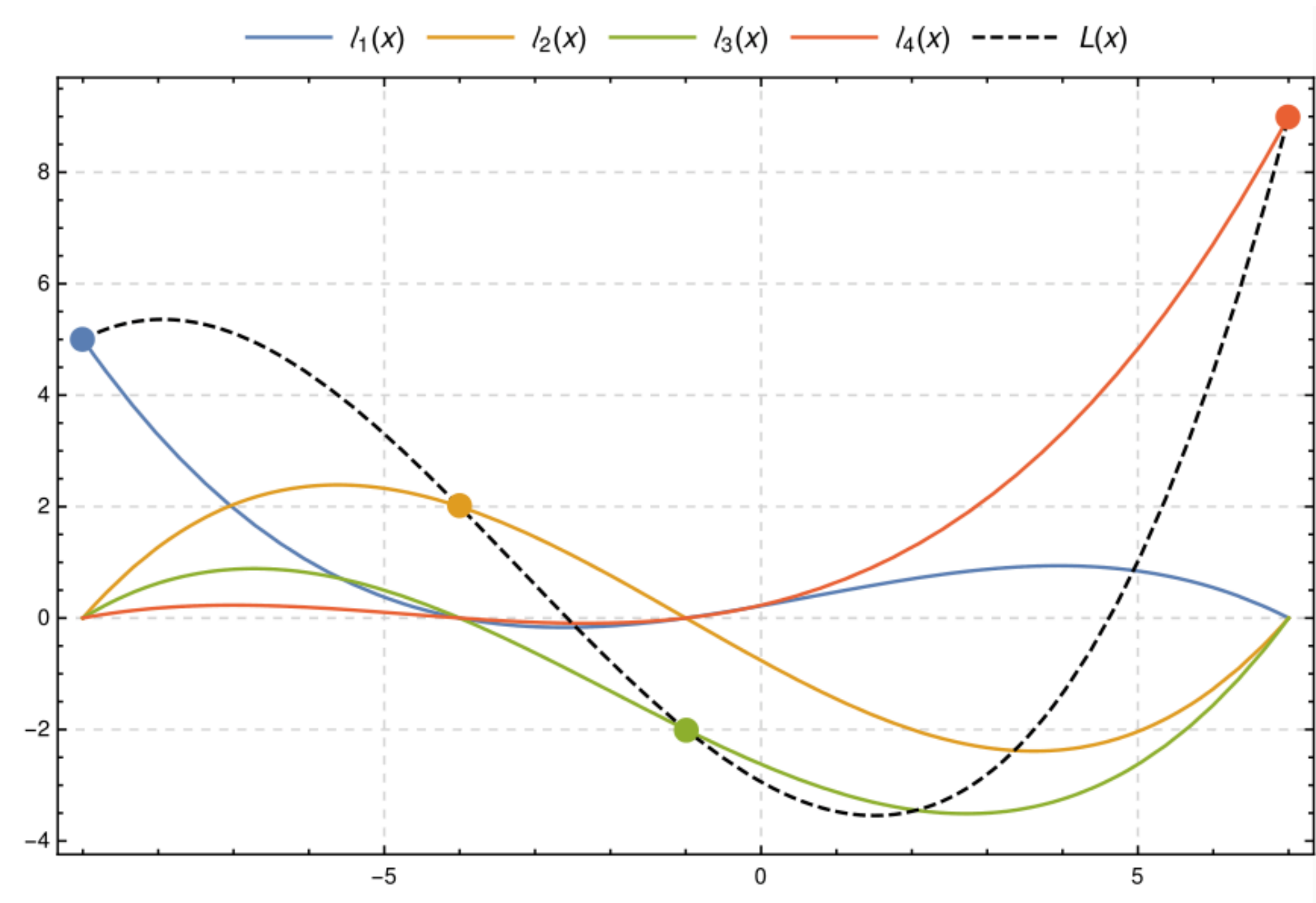
x	1	2	3
y	1	$1/2$	$1/3$

La tabla del problema anterior la generamos a partir de la función $f(x) = 1/x$



MATLAB demo

Polinomio interpolado de Lagrange



[L1\(x\)](#)

This image shows, for four points $((-9, 5), (-4, 2), (-1, -2), (7, 9))$, the (cubic) interpolation polynomial $L(x)$ (dashed, black), which is the sum of the *scaled* basis polynomials $y_0\ell_0(x)$, $y_1\ell_1(x)$, $y_2\ell_2(x)$ and $y_3\ell_3(x)$. The interpolation polynomial passes through all four control points, and each *scaled* basis polynomial passes through its respective control point and is 0 where x corresponds to the other three control points.

Interpolación Catmull-Rom

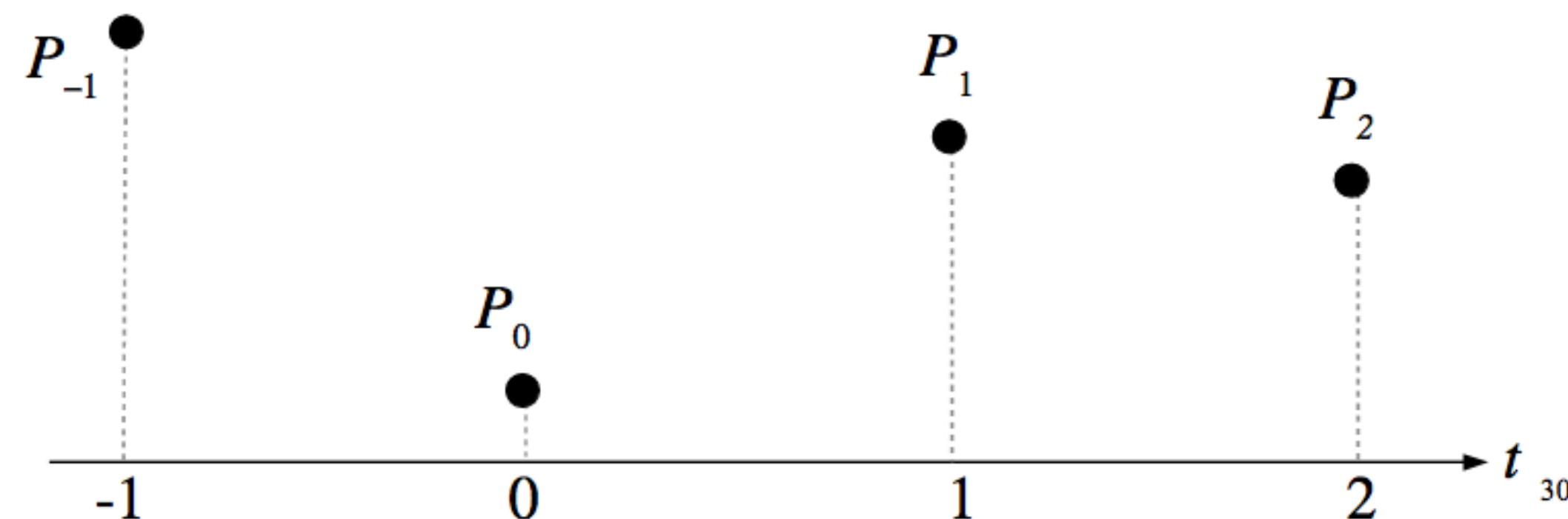
Catmull-Rom.

- Caso particular de Hermite.
- Tangentes desconocida.
- La tangente en cada punto p_i se calcula utilizando el punto anterior y el siguiente en la spline, p_{i-1} y p_{i+1} .

Interpolación Catmull-Rom

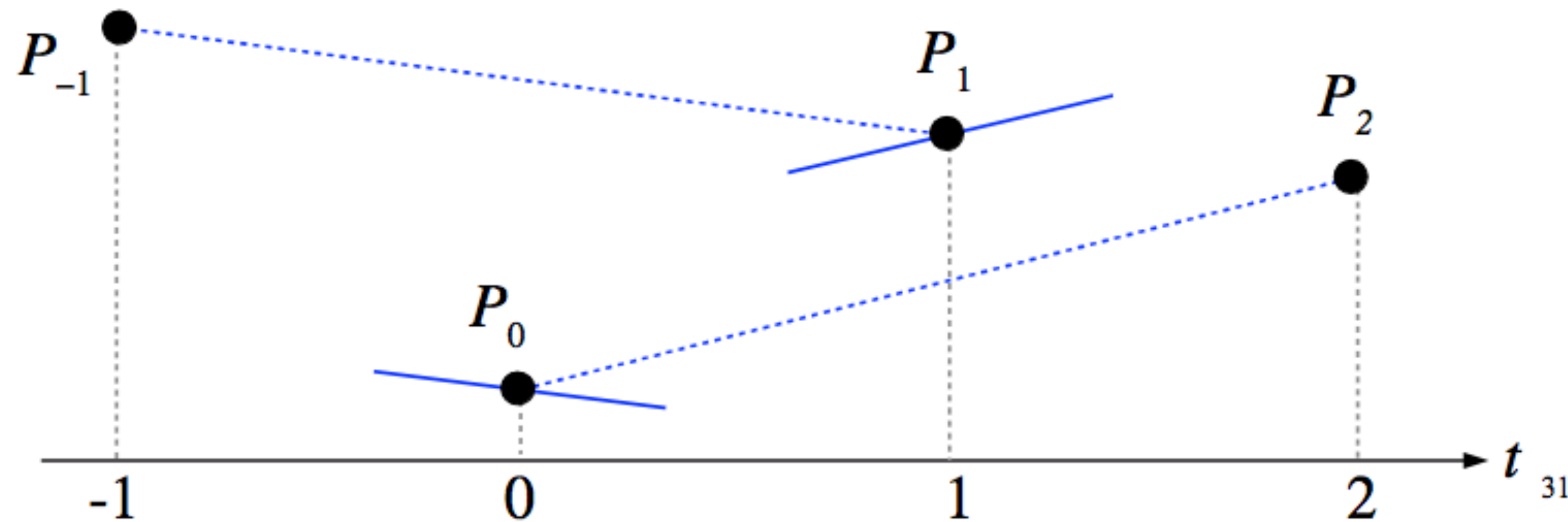
Catmull-Rom.

- Caso particular de Hermite.
- Tangentes desconocida.
- La tangente en cada punto p_i se calcula utilizando el punto anterior y el siguiente en la spline, p_{i-1} y p_{i+1} .
- **Intuition:** A plausible tangent at each point can be inferred directly from the data
 - Now use Hermite interpolation



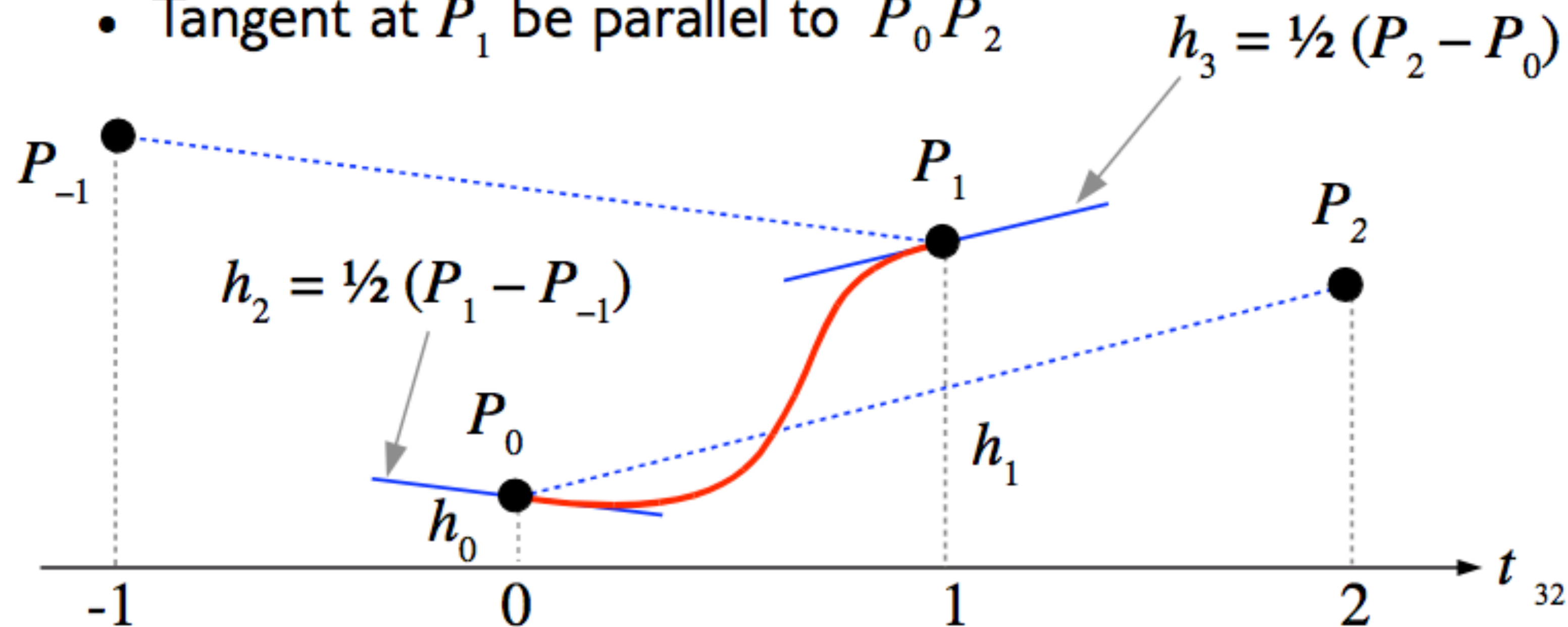
Catmull-Rom Interpolation

- For each segment (P_0, P_1) , use neighboring control points P_{-1}, P_2 and require that:
 - Tangent at P_0 be parallel to $\overline{P_{-1}P_1}$
 - Tangent at P_1 be parallel to $\overline{P_0P_2}$



Catmull-Rom Interpolation

- For each segment (P_0, P_1) , use neighboring control points P_{-1}, P_2 and require that:
 - Tangent at P_0 be parallel to $\overline{P_{-1}P_1}$
 - Tangent at P_1 be parallel to $\overline{P_0P_2}$



Catmull-Rom Interpolation

- In terms of Hermite constraints:

$$h_0 = P_0$$

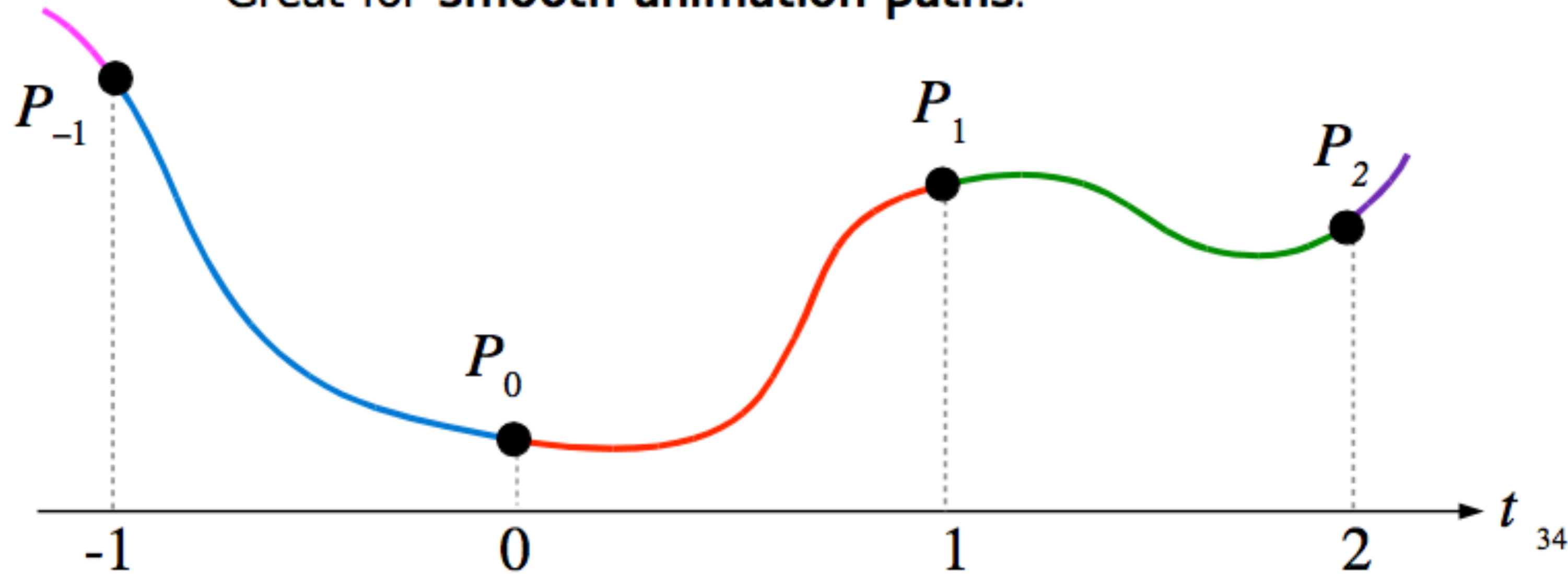
$$h_1 = P_1$$

$$h_2 = \frac{1}{2} (P_1 - P_{-1})$$

$$h_3 = \frac{1}{2} (P_2 - P_0)$$

Catmull-Rom Interpolation

- Repeat for every such interval
- Resulting curve is:
 - C_0 -continuous (segments meet end-to-end)
 - C_1 -continuous (C_0 + derivative is continuous)
 - Great for **smooth animation paths**!



Interpolación Catmull-Rom

Muy utilizadas interpolación suave entre keyframes.

Por ejemplo, trayectorias de la cámara generadas a partir de los keyframes.

Son populares principalmente por ser relativamente fáciles de calcular, principalmente las tangentes en puntos internos, garantizando que cada posición de keyframes se obtiene exactamente, y también garantizando que las tangentes de la curva generada son continuas sobre varios segmentos.

Interpolación Splines

- Un **spline** es una curva diferenciable definida en porciones mediante polinomios.
- Requiere solo polinomios de bajo grado, evitando así las oscilaciones, indeseables en la mayoría de las aplicaciones.

Interpolación Splines

Splines. Ejemplo 1. *Interpolar con splines $f(x) = 1 / x$, en los puntos en los que x vale 1, 2 y 4*

$$f(1) = 1; f(2) = 0.5; f(4) = 0.25$$

El primer segmento: $(1,1) \longrightarrow (2, 0.5)$

Ecuación lineal: $P_1(x) = ax + b$

$$(1) \quad 1 = a+b$$

$$(2) \quad 0.5 = 2a+b$$

De (1) se obtiene: $a=1-b$ (3)

Reemplazando (3) en (2) se obtiene: $0.5=2(1-b)+b$

luego $\longrightarrow b=1.5$

Interpolación

Splines. Ejemplo 1.

Reemplazando el valor de (b) en (1), se obtiene:

$$a = - 0.5$$

Se concluye que: $P1(x) = - 0.5x + 1.5$

El segundo segmento: $(2,0.5) \longrightarrow (4, 0.25)$

$$P2(x) = ax + b$$

Análogamente a lo hecho para $P1(x)$, en el caso de $P2(x)$ se obtiene:

$$(1) \quad 0.5 = 2a + b$$

$$(2) \quad 0.25 = 4a + b$$

$$a = - 0.125, b = 0.75$$

$$P2(x) = - 0.125x + 0.75$$

MATLAB demo

Interpolación

Splines. Ejemplo 2. *Interpolar con splines de grado 2*

$$f(3) = 2.5; f(4.5) = 1; f(7) = 2.5; f(9)=0.5$$

Primero que nada, vemos que se forman tres intervalos: $[3,4.5], [4.5,7], [7,9]$
En cada uno de estos intervalos, debemos definir una función polinomial de grado 2, como sigue:

Hacemos que la spline pase por los puntos de la tabla de datos, es decir, se debe cumplir que:

$$s(3)=2.5 \quad s(4.5)=1 \quad s(7)=2.5 \quad s(9)=0.5$$

Así, se forman las siguientes ecuaciones

$$s(3) = 2.5 \Rightarrow 9a_1 + 3b_1 + c_1 = 2.5$$

$$s(4.5) = 1 \Rightarrow \begin{cases} (4.5)^2 a_1 + 4.5b_1 + c_1 = 1 \\ (4.5)^2 a_2 + 4.5b_2 + c_2 = 1 \end{cases}$$

$$s(7) = 2.5 \Rightarrow \begin{cases} 49a_2 + 7b_2 + c_2 = 2.5 \\ 49a_3 + 7b_3 + c_3 = 2.5 \end{cases}$$

$$s(9) = 0.5 \Rightarrow 81a_3 + 9b_3 + c_3 = 0.5$$

Interpolación

Splines. Ejemplo 2.

Hasta aquí, tenemos un total de 6 ecuaciones con 9 incógnitas.
El siguiente paso es manejar la existencia de las derivadas continuas.
En el caso de las splines de grado 2, necesitamos que la spline tenga derivada continua de orden $k-1=1$, es decir, primera derivada continua.

Calculamos primero la primera derivada:

$$s'(x) = \begin{cases} 2a_1x + b_1 & \text{si } x \in [3, 4.5] \\ 2a_2x + b_2 & \text{si } x \in [4.5, 7] \\ 2a_3x + b_3 & \text{si } x \in [7, 9] \end{cases}$$

Vemos que esta derivada está formada por segmentos de rectas, que pudieran presentar discontinuidad en los cambios de intervalo. Es decir, las posibles discontinuidades son $x = 4.5$ y $x = 7$.

Interpolación

Splines. Ejemplo 2.

Por lo tanto para que $s'(x)$ sea continua, se debe cumplir que:

$$2a_1(4.5) + b_1 = 2a_2(4.5) + b_2 \Rightarrow 9a_1 + b_1 = 9a_2 + b_2$$

También debe cumplirse que:

$$2a_2(7) + b_2 = 2a_3(7) + b_3 \Rightarrow 14a_2 + b_2 = 14a_3 + b_3$$

Así, tenemos un total de 8 ecuaciones vs. 9 incógnitas; esto nos da un grado de libertad para elegir alguna de las incógnitas. Elegimos por simple conveniencia $a_1 = 0$.

De esta forma, tenemos un total de 8 ecuaciones con 8 incógnitas:

$$3b_1 + c_1 = 2.5 \qquad 49a_3 + 7b_3 + c_3 = 2.5$$

$$4.5b_1 + c_1 = 1 \qquad 81a_3 + 9b_3 + c_3 = 0.5$$

$$20.25a_2 + 4.5b_2 + c_2 = 1 \qquad b_1 = 9a_2 + b_2$$

$$49a_2 + 7b_2 + c_2 = 2.5 \qquad 14a_2 + b_2 = 14a_3 + b_3$$

Y las soluciones serían:

$b_1 = -1$	$a_2 = 0.64$	$a_3 = -1.6$
$c_1 = 5.5$	$b_2 = -6.76$	$b_3 = 24.6$
	$c_2 = 18.46$	$c_3 = -91.3$

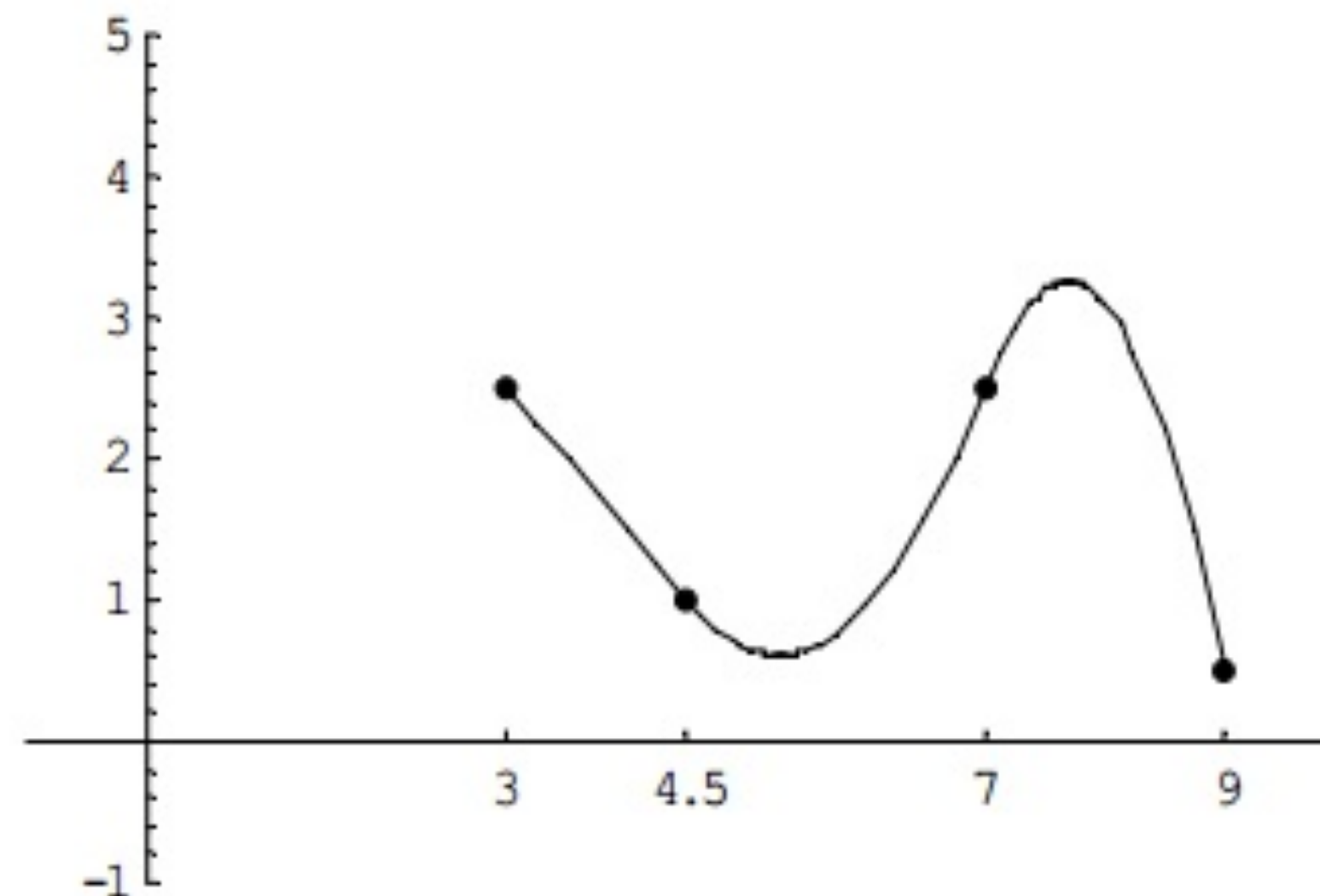
Interpolación

Splines. Ejemplo 2.

Sustituyendo los valores, obtenemos los splines:

$$s(x) = \begin{cases} -x + 5.5 & \text{si } x \in [3, 4.5] \\ 0.64x^2 - 6.76x + 18.46 & \text{si } x \in [4.5, 7] \\ -1.6x^2 + 24.6x - 91.3 & \text{si } x \in [7, 9] \end{cases}$$

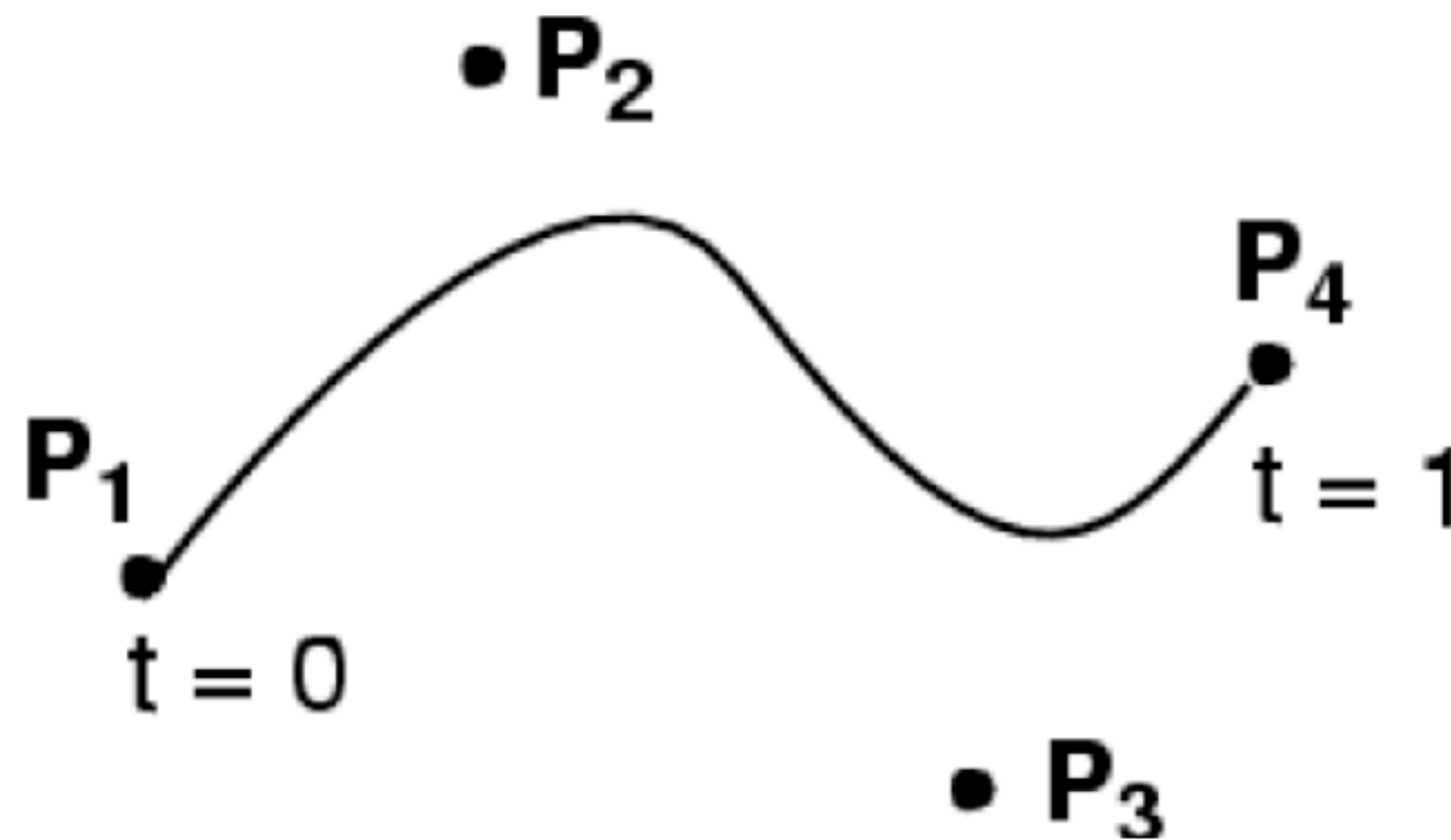
Cuya representación gráfica ser



MATLAB demo

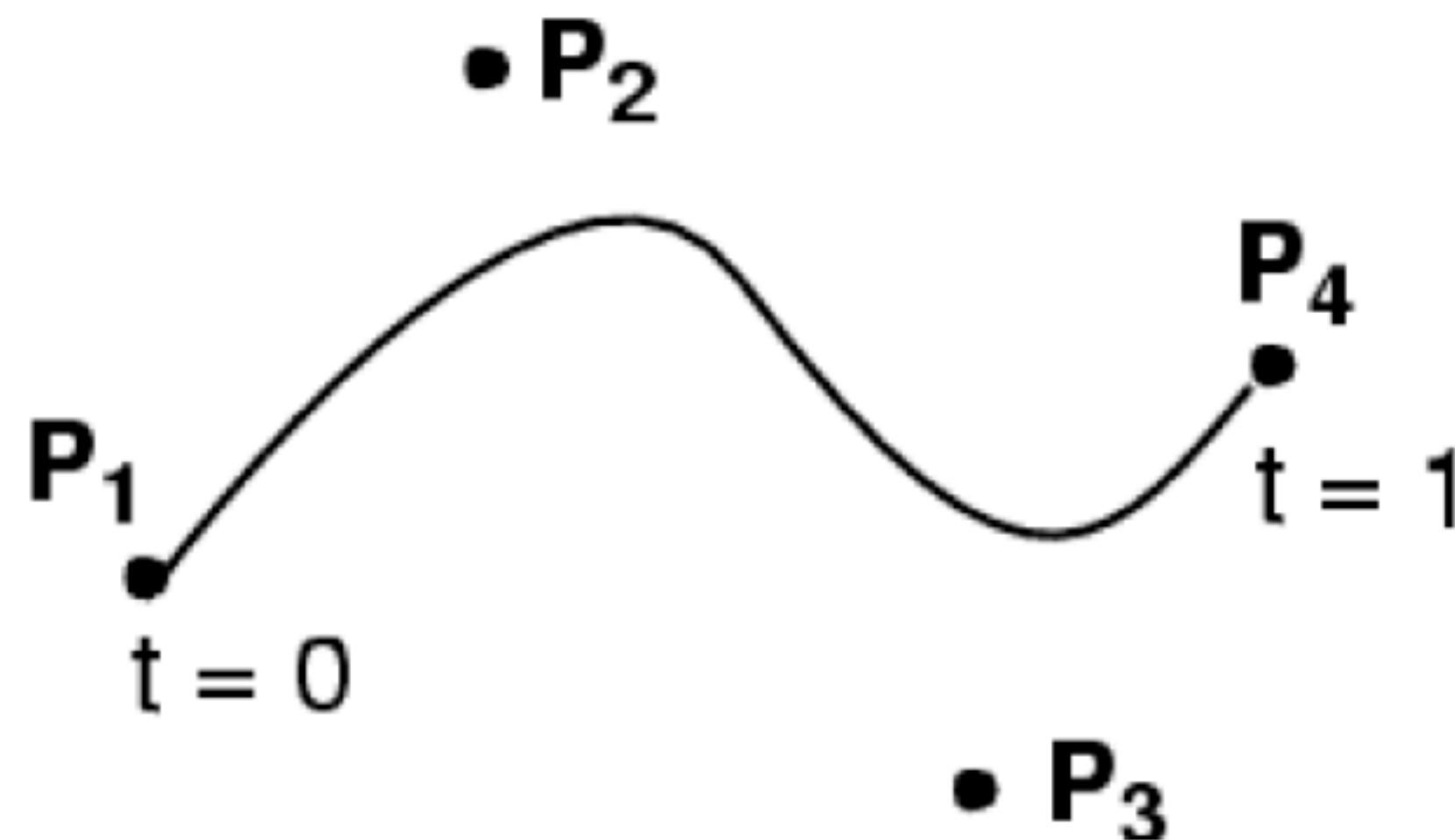
Curvas de Bezier

- User specifies 4 control points $P_1 \dots P_4$
- Curve goes through (interpolates) the ends P_1, P_4
- Approximates the two other ones
- Cubic polynomial



Curvas de Bezier

$$\begin{aligned} \bullet \quad P(t) = & (1-t)^3 \quad \mathbf{P}_1 \\ & + 3t(1-t)^2 \quad \mathbf{P}_2 \\ & + 3t^2(1-t) \quad \mathbf{P}_3 \\ & + t^3 \quad \mathbf{P}_4 \end{aligned}$$



That is,

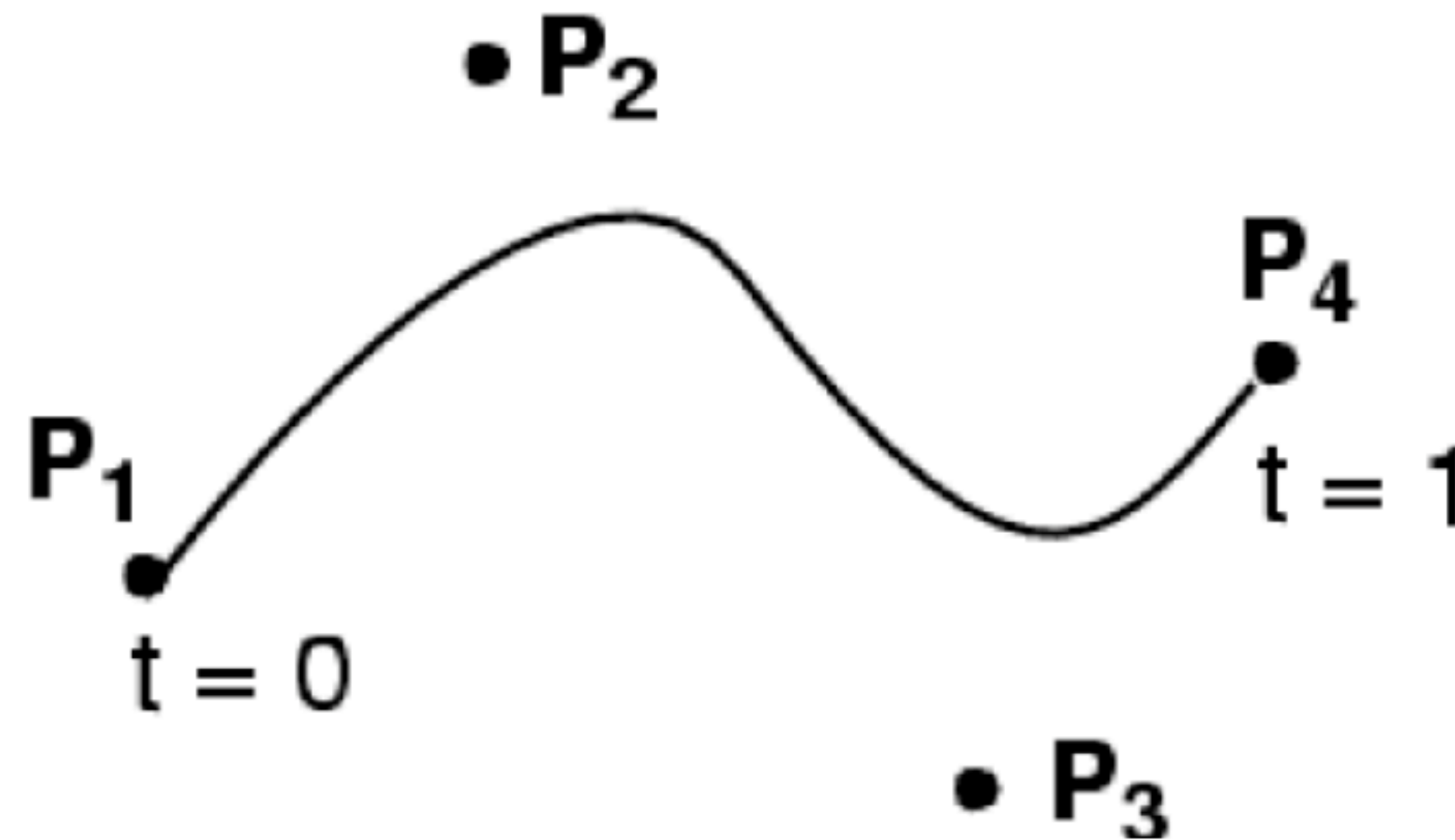
$$\begin{aligned} x(t) = & (1-t)^3 x_1 + \\ & 3t(1-t)^2 x_2 + \\ & 3t^2(1-t) x_3 + \\ & t^3 x_4 \end{aligned}$$

$$\begin{aligned} y(t) = & (1-t)^3 y_1 + \\ & 3t(1-t)^2 y_2 + \\ & 3t^2(1-t) y_3 + \\ & t^3 y_4 \end{aligned}$$

Curvas de Bezier

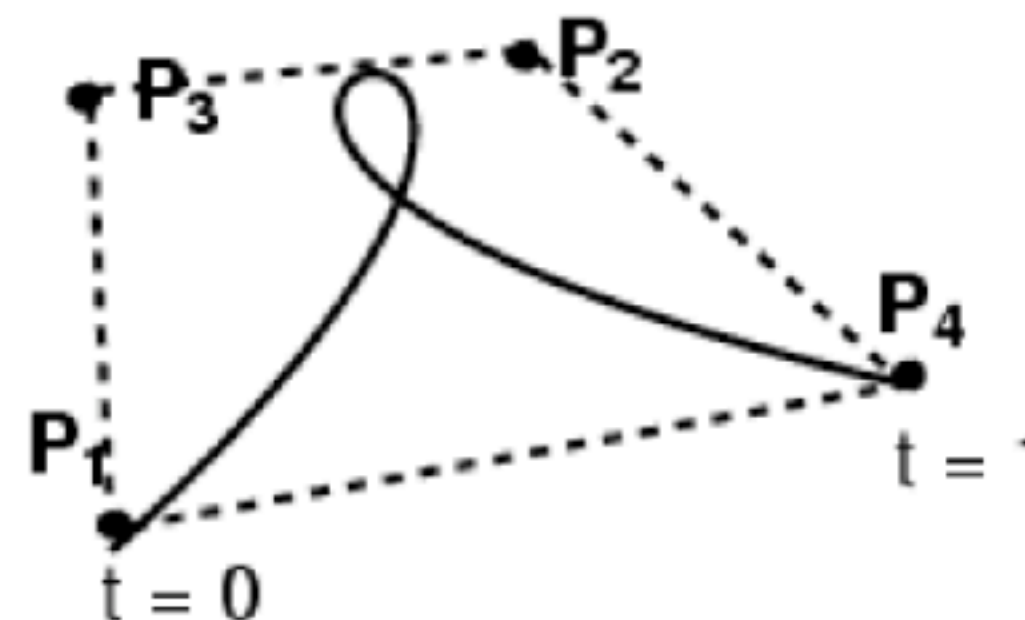
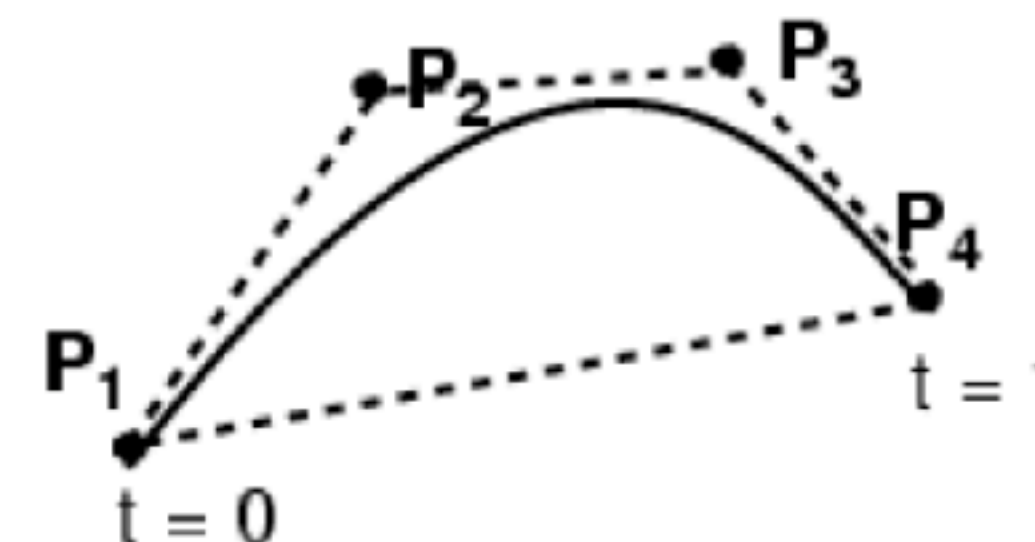
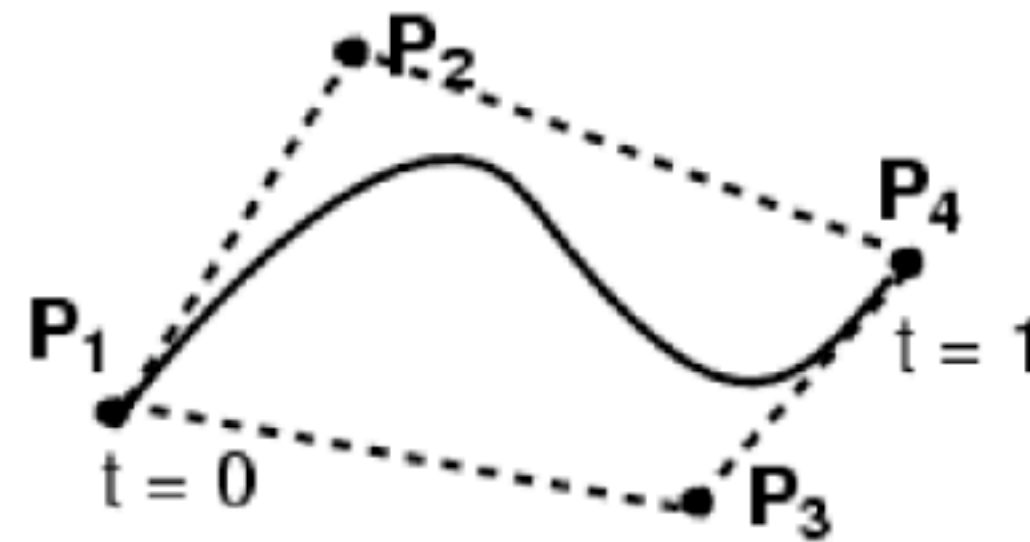
- $P(t) = (1-t)^3 \mathbf{P}_1$
+ $3t(1-t)^2 \mathbf{P}_2$
+ $3t^2(1-t) \mathbf{P}_3$
+ $t^3 \mathbf{P}_4$

Verify what happens
for $t=0$ and $t=1$



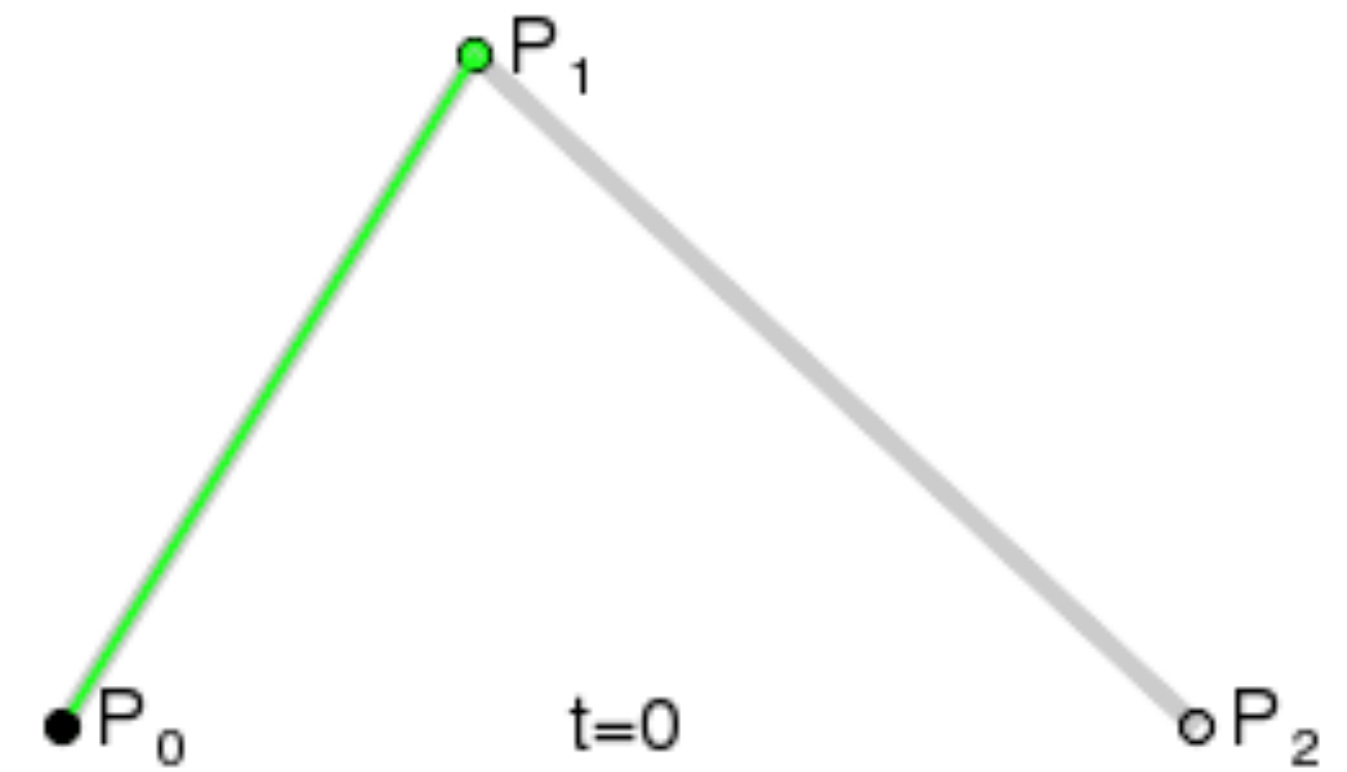
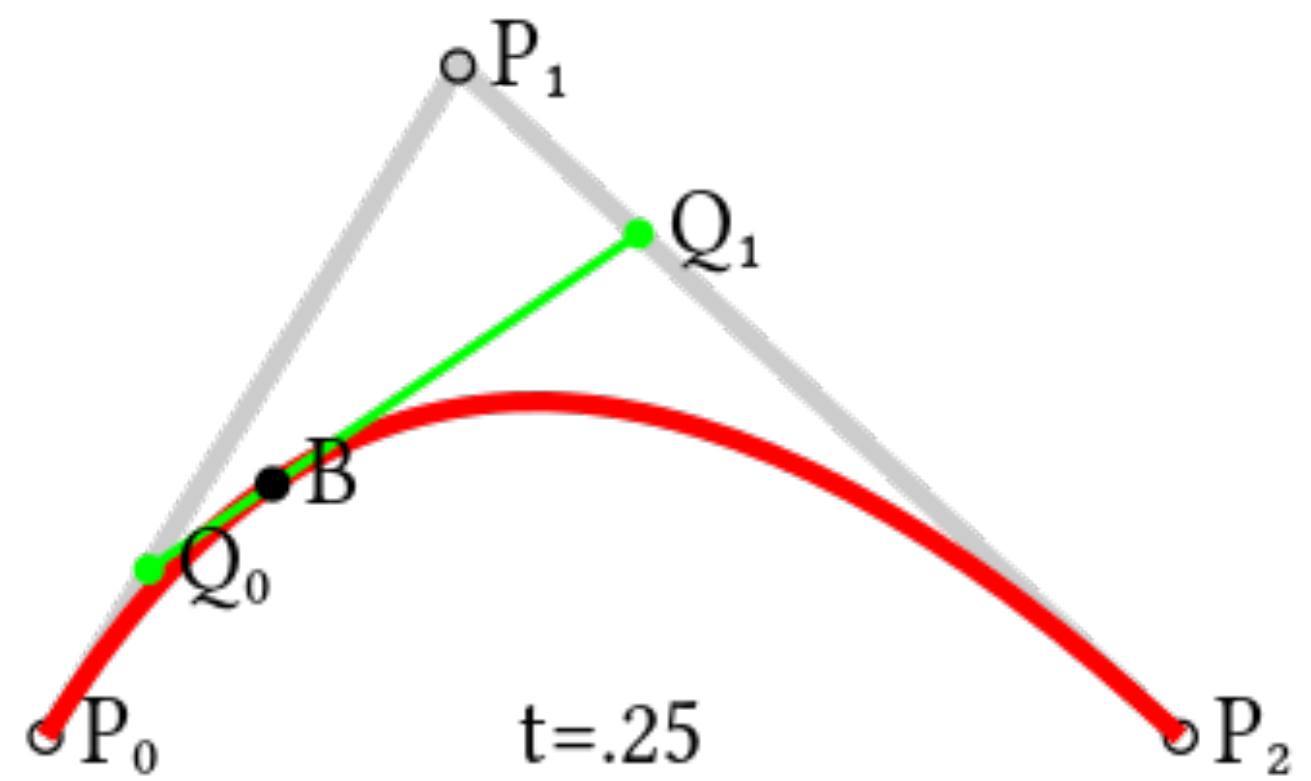
Curvas de Bezier

- 4 control points
- Curve passes through first & last control point

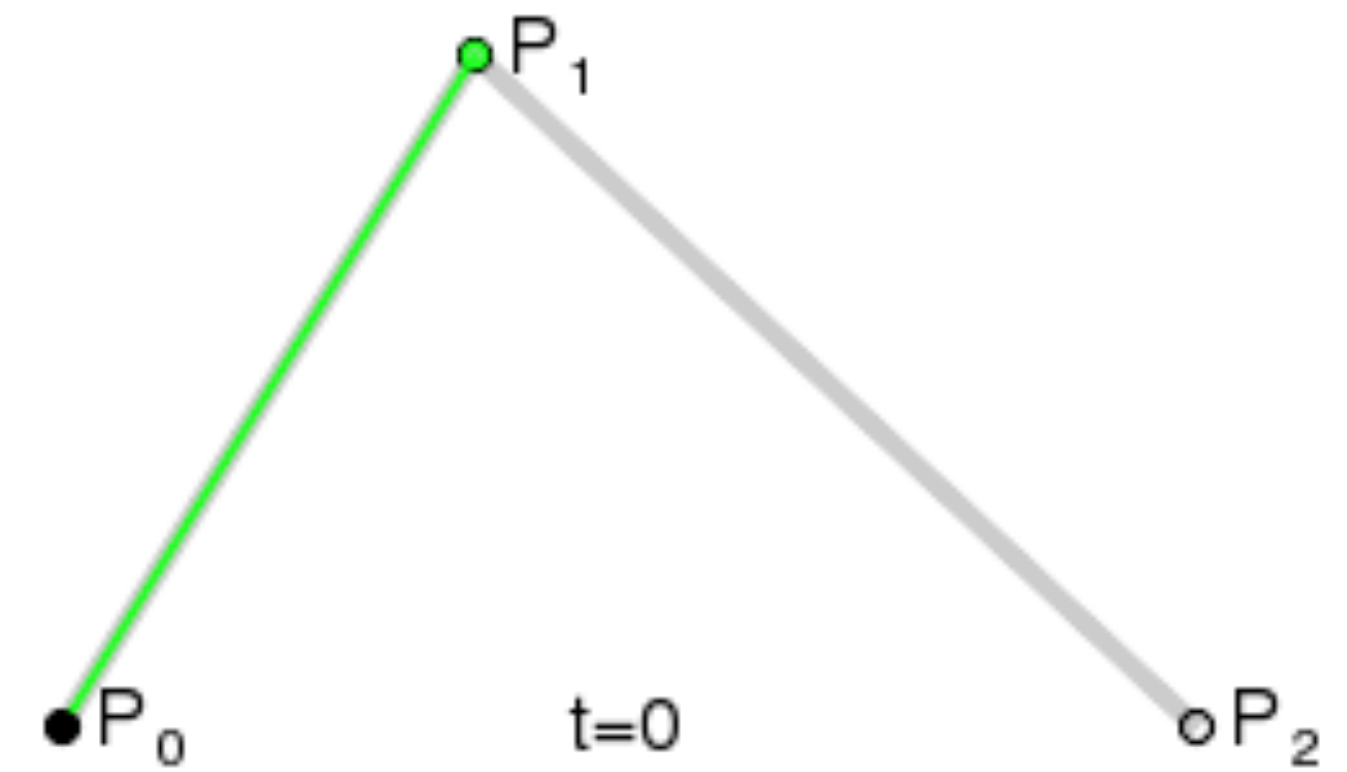
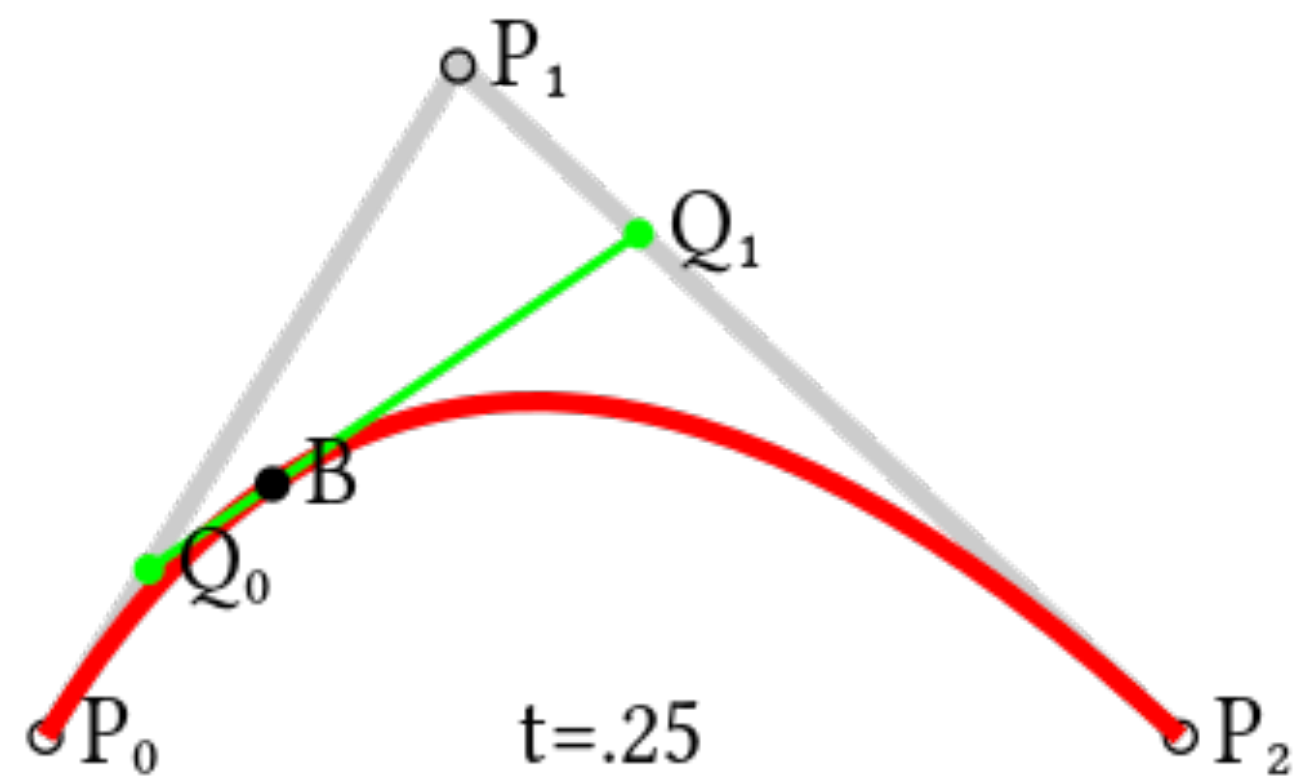


Courtesy of Seth Teller.
Used with permission.

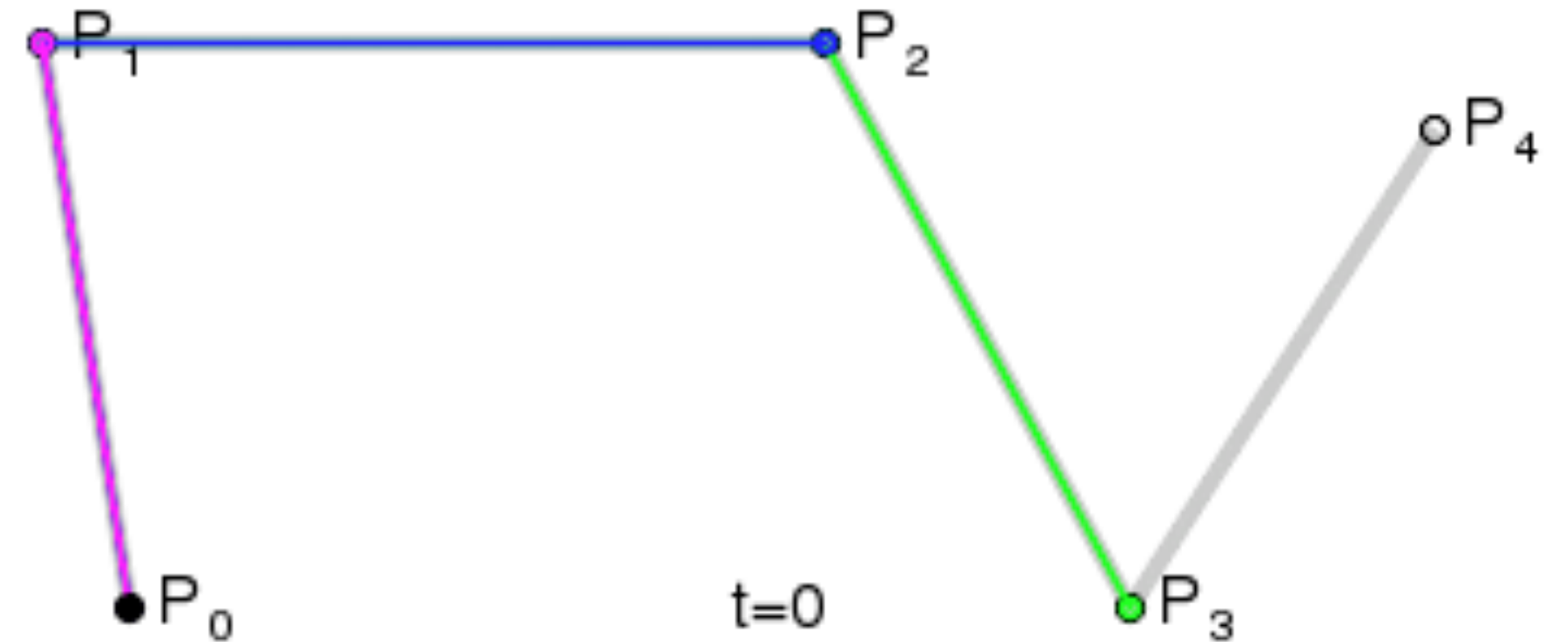
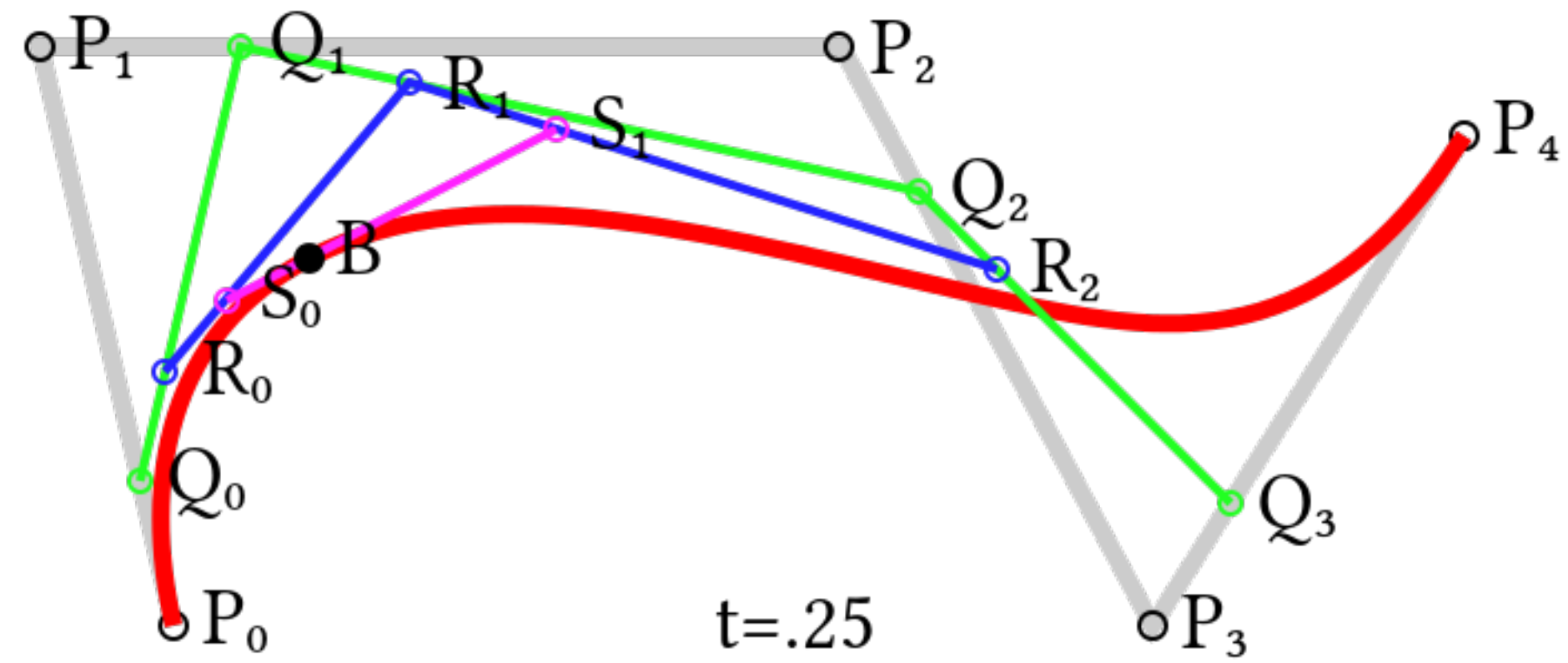
Curvas de Bezier



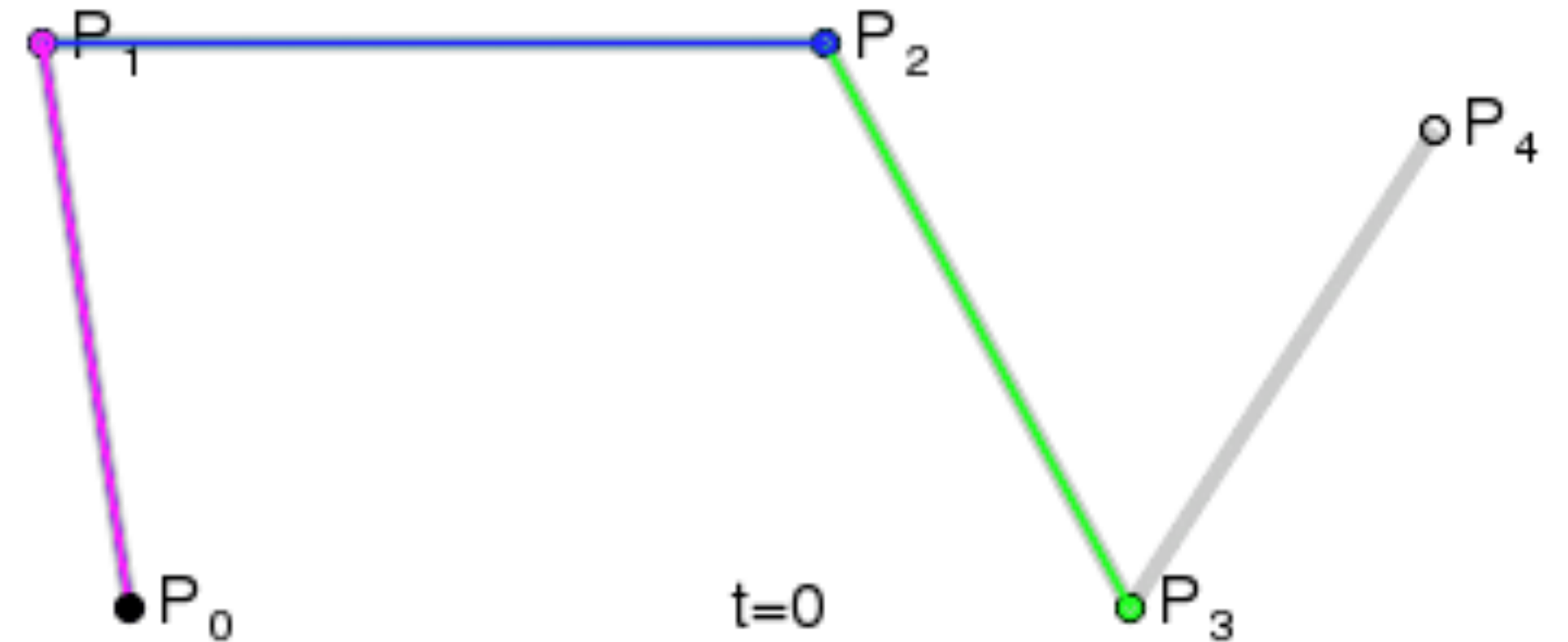
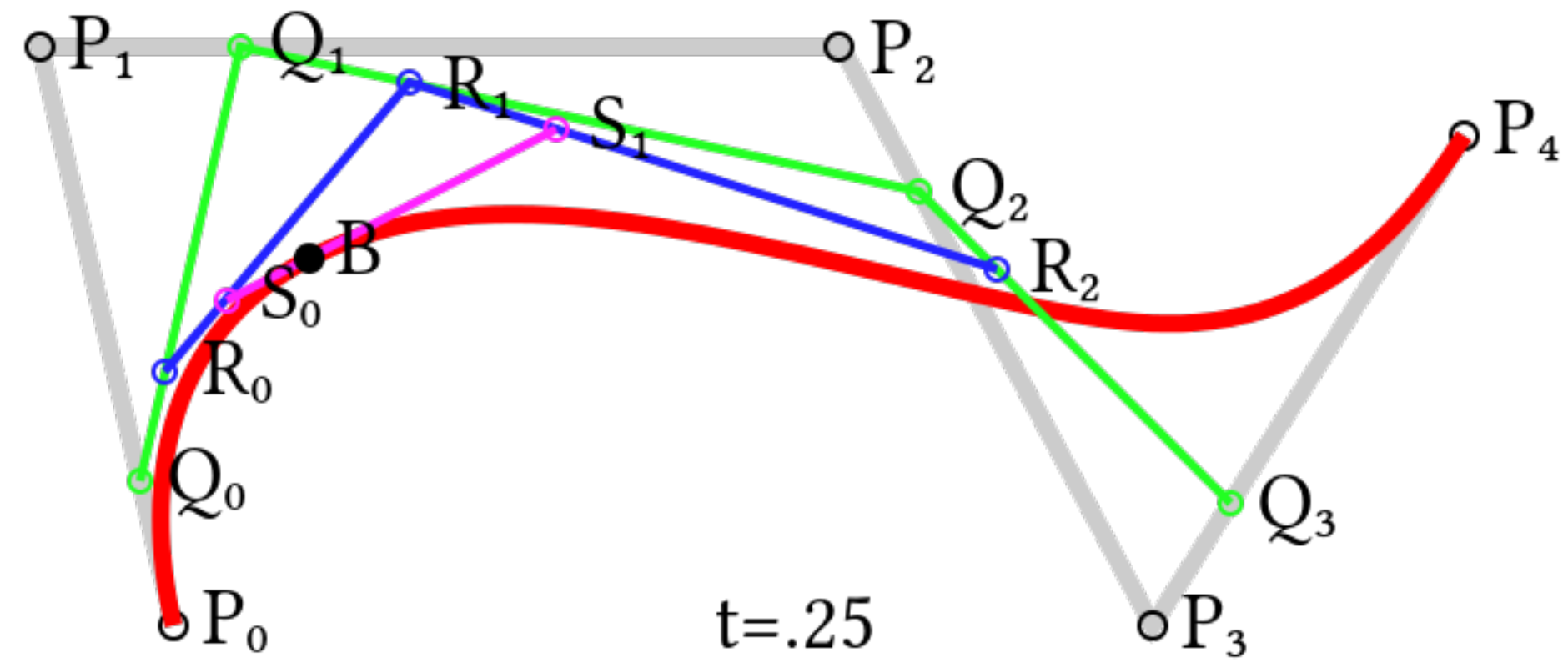
Curvas de Bezier



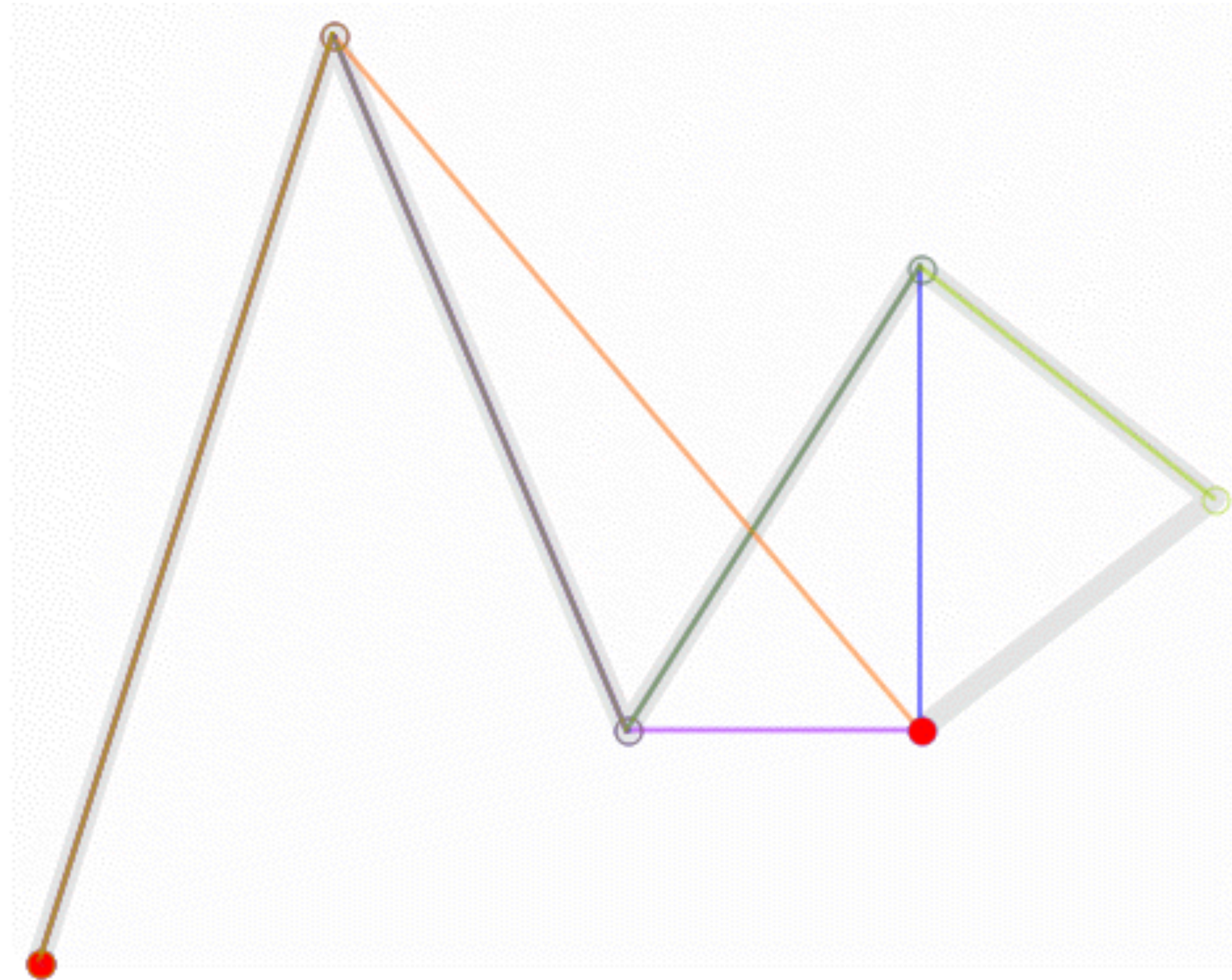
Curvas de Bezier



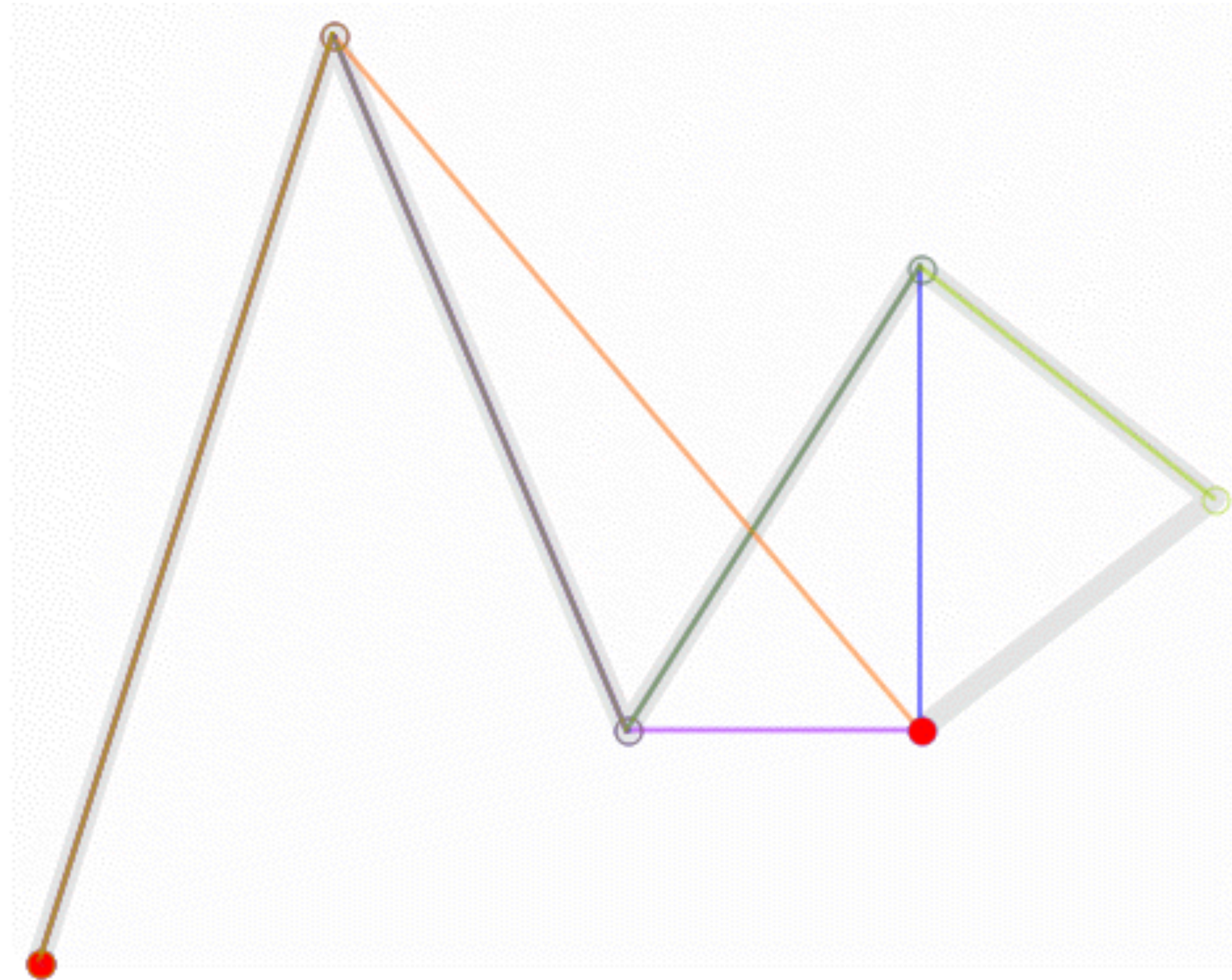
Curvas de Bezier



Curvas de Bezier

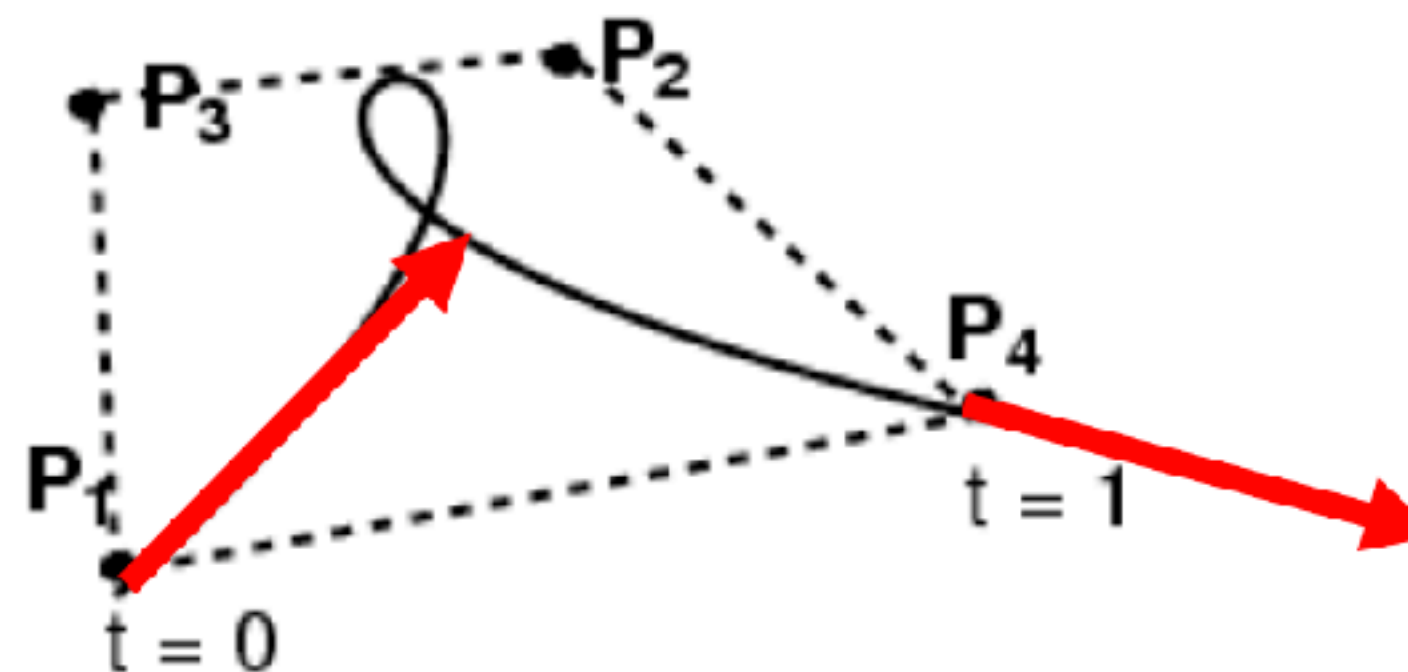
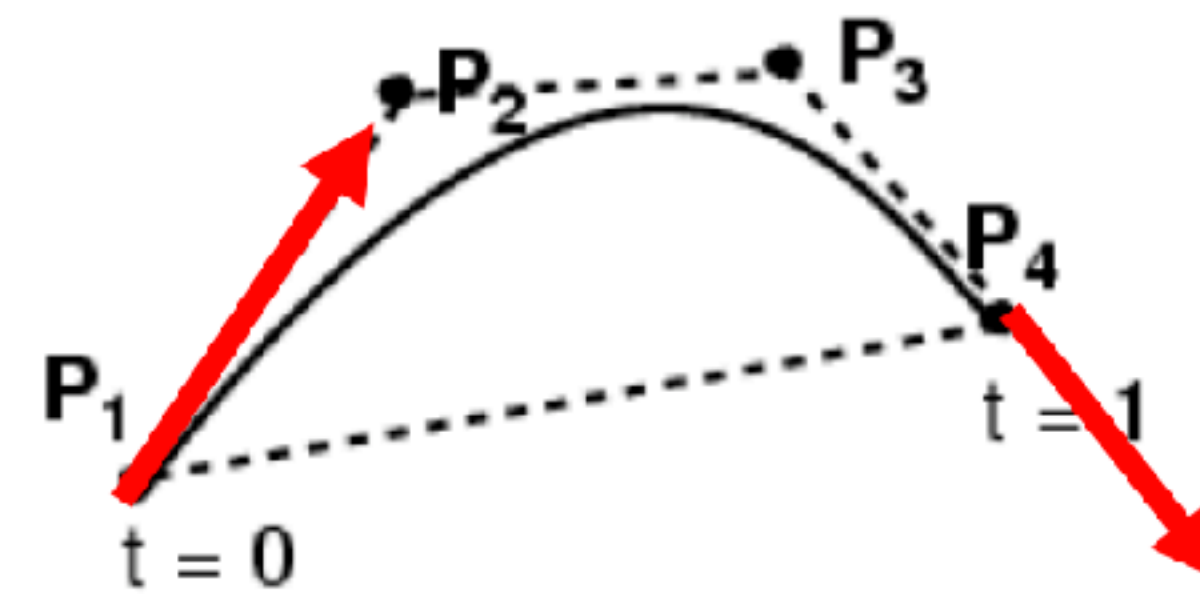
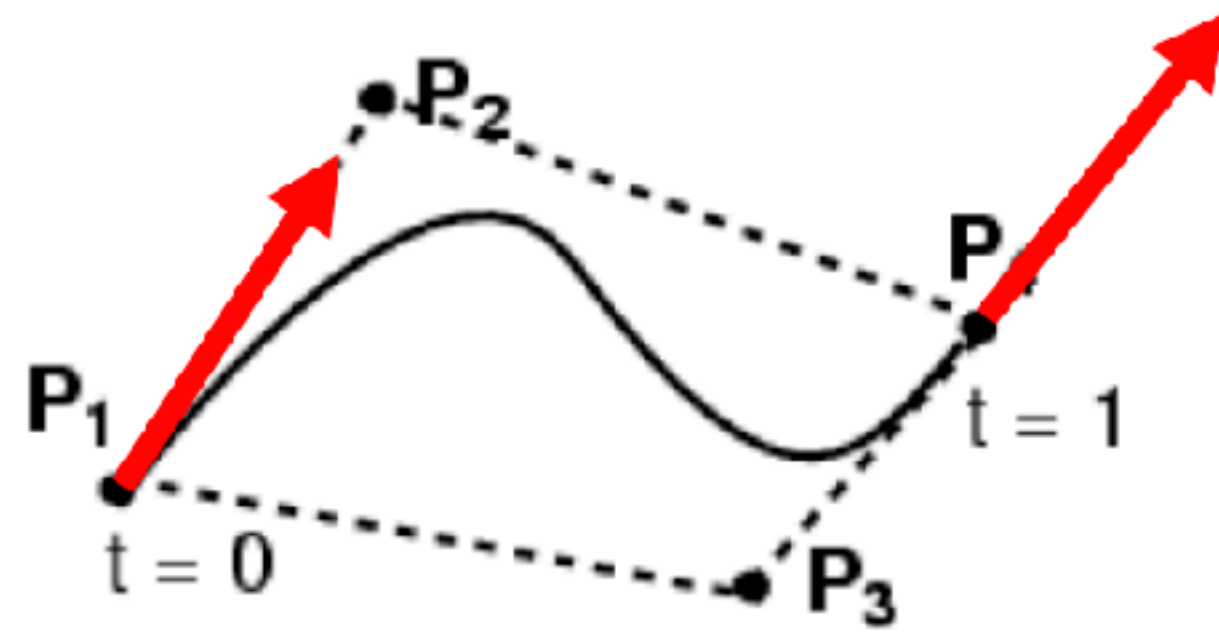


Curvas de Bezier



Curvas de Bezier

- 4 control points
- Curve passes through first & last control point
- Curve is tangent at **P1** to (P1-P2) and at **P4** to (P4-P3)



A Bézier curve is bounded by the **convex hull** of its control points.

Curvas de Bezier

$$\begin{aligned} \bullet \quad P(t) = & (1-t)^3 \quad \mathbf{P1} \\ & + 3t(1-t)^2 \quad \mathbf{P2} \\ & + 3t^2(1-t) \quad \mathbf{P3} \\ & + t^3 \quad \mathbf{P4} \end{aligned}$$

¿Por qué funciona?

Curvas de Bezier

$$\begin{aligned} \bullet \quad P(t) = & (1-t)^3 & P1 \\ & + 3t(1-t)^2 & P2 \\ & + 3t^2(1-t) & P3 \\ & + t^3 & P4 \end{aligned}$$

¿Por qué funciona?

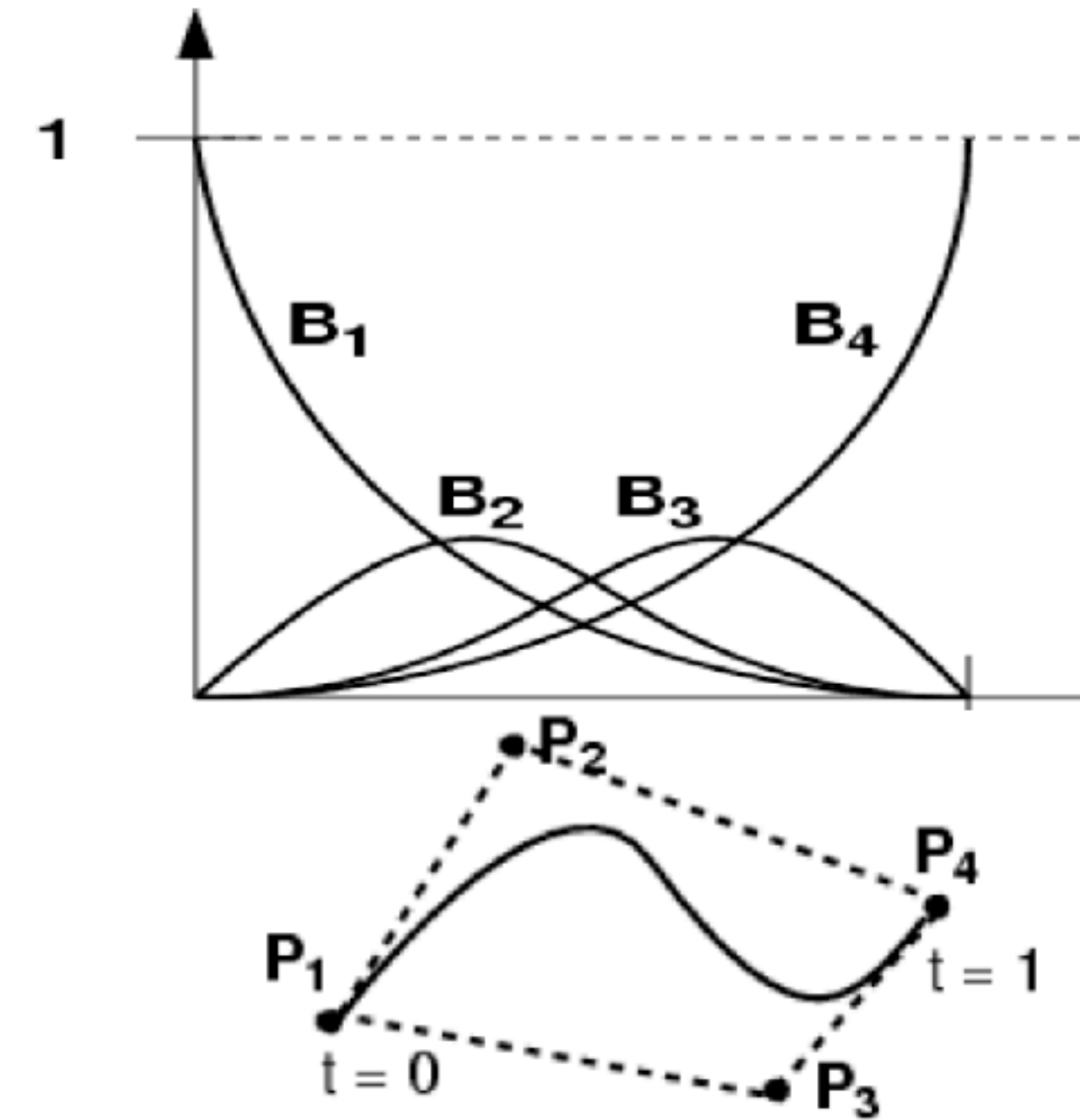


Combinación lineal de
polinomios base

Curvas de Bezier

- $P(t)$ is a weighted combination of the 4 control points with weights:
 - $B_1(t) = (1-t)^3$
 - $B_2(t) = 3t(1-t)^2$
 - $B_3(t) = 3t^2(1-t)$
 - $B_4(t) = t^3$
- First, P_1 is the most influential point, then P_2 , P_3 , and P_4

$$\begin{array}{rcl} P(t) = & (1-t)^3 & P_1 \\ + & 3t(1-t)^2 & P_2 \\ + & 3t^2(1-t) & P_3 \\ + & t^3 & P_4 \end{array}$$



Curvas de Bezier

$$P(t) = \sum_{i=0}^n B_i J_{n,i}(t)$$

Curvas de Bezier

$$P(t) = \sum_{i=0}^n B_i J_{n,i}(t)$$

Curvas de Bezier

$$P(t) = \sum_{i=0}^n B_i J_{n,i}(t)$$


Polinomio de Bernstein

Curvas de Bezier


$$P(t) = \sum_{i=0}^n B_i J_{n,i}(t)$$

Polinomio de Bernstein



$$J_{n,i}(t) = \frac{n!}{i!(n-i)!} t^i (1-t)^{n-i}$$

Grado



Curvas de Bezier


$$P(t) = \sum_{i=0}^n B_i J_{n,i}(t)$$

Polinomio de Bernstein



$$J_{n,i}(t) = \frac{n!}{i!(n-i)!} t^i (1-t)^{n-i} = \frac{6}{i!(3-i)!} t^i (1-t)^{n-i}$$

Grado



Curvas de Bezier


$$P(t) = \sum_{i=0}^n B_i J_{n,i}(t)$$

Polinomio de Bernstein



$$J_{n,i}(t) = \frac{n!}{i!(n-i)!} t^i (1-t)^{n-i} = \frac{6}{i!(3-i)!} t^i (1-t)^{n-i}$$

Grado



$$\begin{aligned} P(t) &= B_0 J_{3,0}(t) + B_1 J_{3,1}(t) + B_2 J_{3,2}(t) + B_3 J_{3,3}(t) \\ &= (1-t)^3 P_0 + 3t(1-t)^2 P_1 + 3t^2(1-t) P_2 + t^3 P_3 \end{aligned}$$

Curvas de Bezier

$$P(t) = \sum_{i=0}^n B_i J_{n,i}(t)$$

Polinomio de Bernstein

$$J_{n,i}(t) = \frac{n!}{i!(n-i)!} t^i (1-t)^{n-i} = \frac{6}{i!(3-i)!} t^i (1-t)^{n-i}$$

Grado

$$P(t) = B_0 J_{3,0}(t) + B_1 J_{3,1}(t) + B_2 J_{3,2}(t) + B_3 J_{3,3}(t)$$
$$= (1-t)^3 P_0 + 3t(1-t)^2 P_1 + 3t^2(1-t) P_2 + t^3 P_3$$

MATLAB demo

Inkscape demo

Curvas de Bezier

- ¿Podemos juntamos 2 o más curvas?
- ¿Podemos escalar, rotar o trasladar curvas?