Análisis Numérico para Ingeniería

Clase Nro. 10

Aproximación de Funciones

Temas a tratar:

- Interpolación por Splines Cúbicos.
- Aproximación por Mínimos Cuadrados.
- Criterios de elección:
  - Tipo de Aproximación
  - Polinomio resultante
- Ventajas y desventajas de los métodos.

Aproximación por Splines Cúbicos

SPLINES

Interpolación por Splines Cúbicos

Dada una función **f** definida en **[a, b]** y un conjunto de números, llamados nodos, $\mathbf{a = x_0 < x_1 < x_2 < \ldots < x_n = b}$, un polinomio interpolante cúbico spline **S**, para **f**, es una función que satisface las siguientes condiciones:

a) **S** es un polinomio cúbico denotado $\mathbf{S_j}$ en el sub-intervalo $[\mathbf{x_j}, \mathbf{x_{j+1}}]$, para cada j=0, 1,..., n-1.

b) $S(\mathbf{x_j}) = f(\mathbf{x_j})$, para cada j=0, 1,..., n.

c) $S_{j+1}(\mathbf{x_{j+1}}) = S_j(\mathbf{x_{j+1}})$, para cada j=0, 1,..., n-2.

d) $S'_{j+1}(\mathbf{x_{j+1}}) = S'_j(\mathbf{x_{j+1}})$, para cada j=0, 1,..., n-2.

e) $S''_{j+1}(\mathbf{x_{j+1}}) = S''_j(\mathbf{x_{j+1}})$, para cada j=0, 1,..., n-2.

Interpolación por Splines Cúbicos

Además dicha función debe satisfacer **alguna** de las siguientes condiciones de frontera:

i. ***FRONTERA LIBRE***

- $S''(\mathbf{x_0}) = S''(\mathbf{x_n}) = 0$

ii. ***FRONTERA SUJETA***

- $S'(\mathbf{x_0}) = f'(\mathbf{x_0})$ y $S'(\mathbf{x_n}) = f'(\mathbf{x_n})$

Polinomio de Spline Cúbico

La forma general de un polinomio de **Spline Cúbico** es:

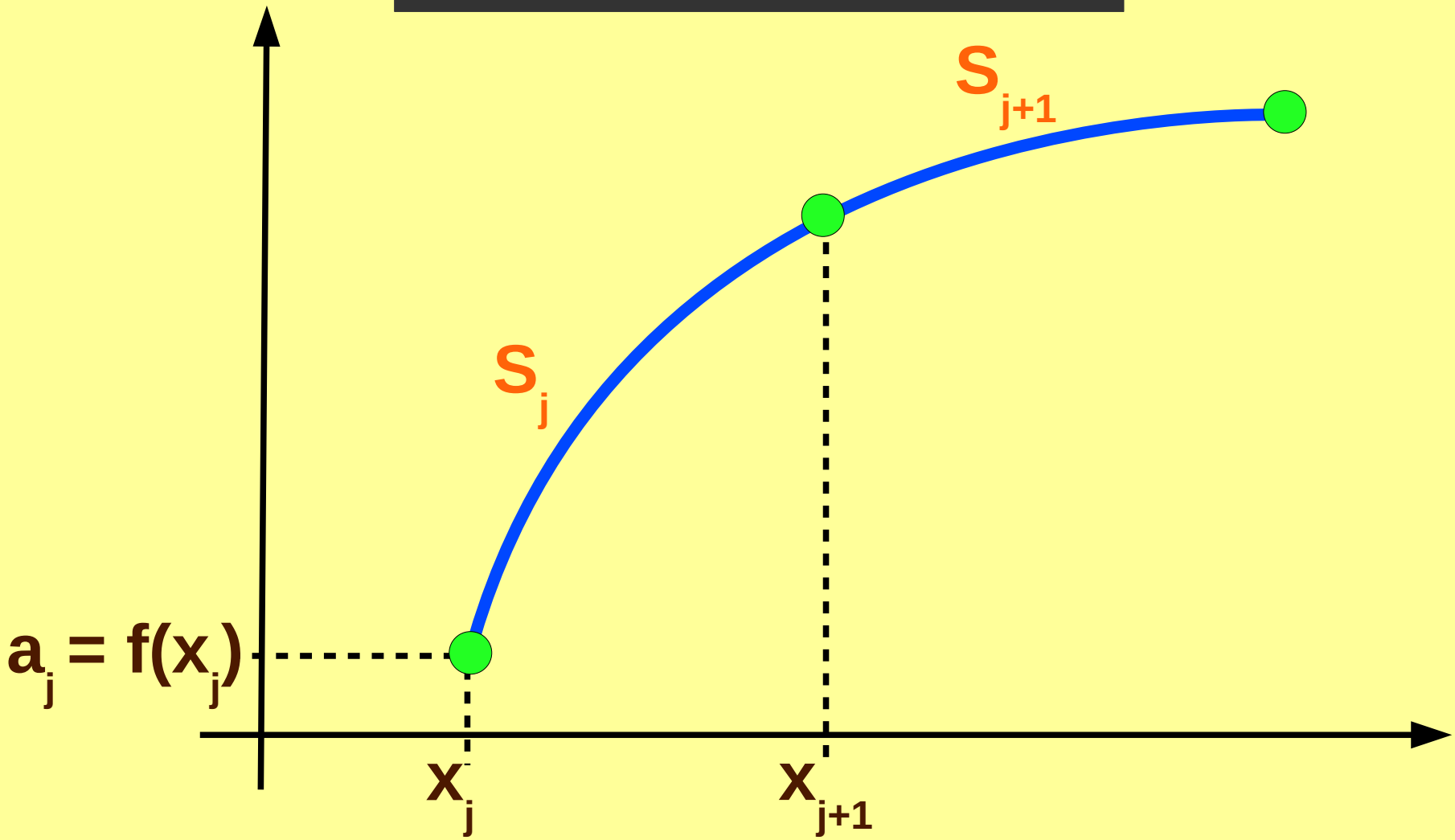
$$S_j(x) = a_j + b_j \cdot (x - x_j) + c_j \cdot (x - x_j)^2 + d_j \cdot (x - x_j)^3$$

Aplicando la condición **$S(x_j) = f(x_j)$**, es evidente que:

$$S_j(x_j) = a_j = f(x_j)$$

Interpolación por Splines Cúbicos

$$S_j(x_j) = f(x_j) = a_j$$



Polinomio de Spline Cúbico

Luego, aplicando la condición $\mathbf{S_j(x_{j+1}) = S_{j+1}(x_{j+1})}$, se tiene que:

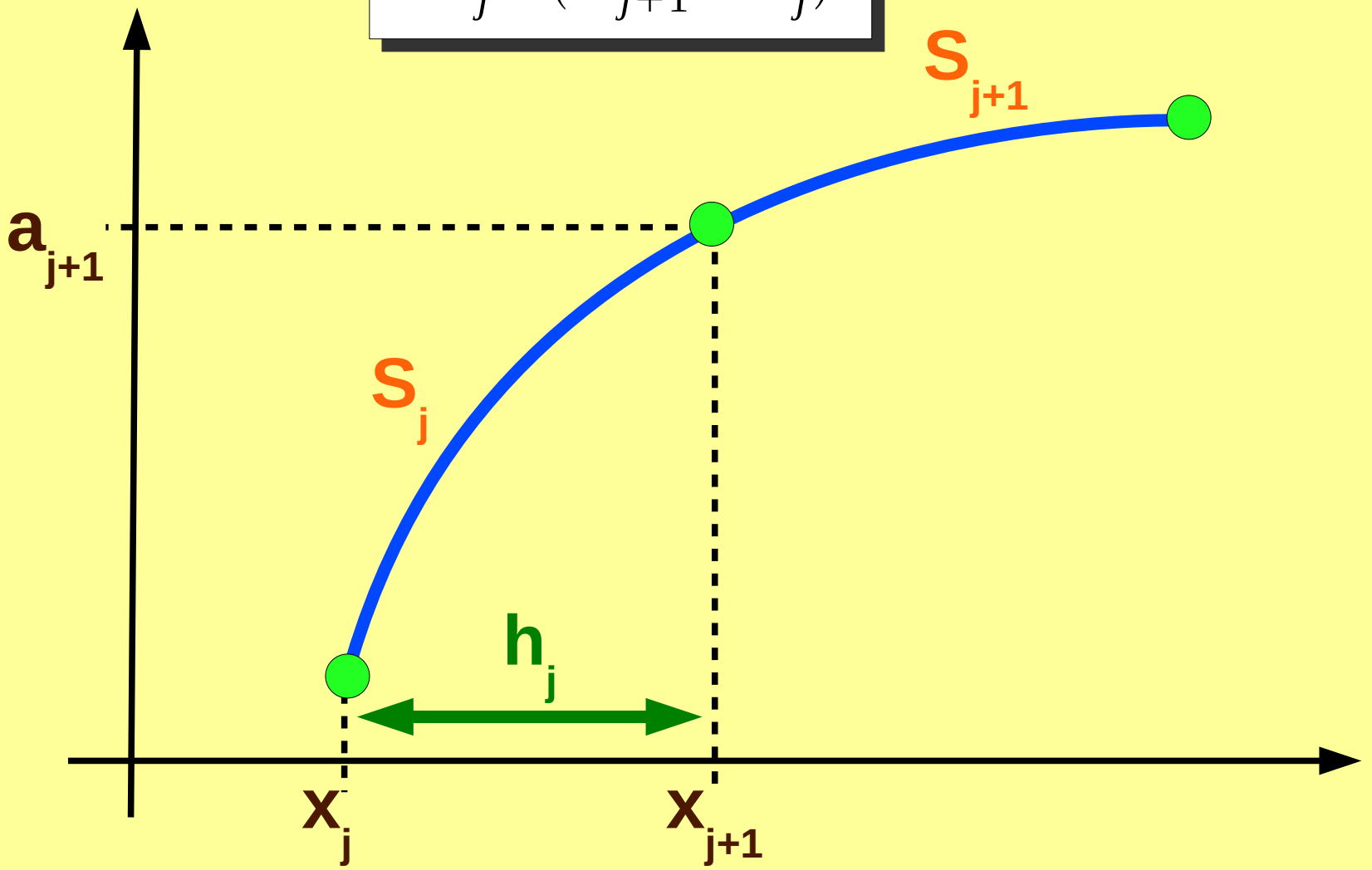
$$a_{j+1} = S_j(x_{j+1}) = S_{j+1}(x_{j+1}) = \\ = a_j + b_j \cdot (x_{j+1} - x_j) + c_j \cdot (x_{j+1} - x_j)^2 + d_j \cdot (x_{j+1} - x_j)^3$$

Como utilizaremos frecuentemente $(\mathbf{x_{j+1} - x_j})$, adoptaremos la siguiente notación para simplificar:

$$h_j = (x_{j+1} - x_j)$$

Polinomio de Spline Cúbico

$$h_j = (x_{j+1} - x_j)$$



Polinomio de Spline Cúbico

Siendo $\mathbf{a_n = f(x_n)}$, entonces para **j=0, 1, 2,…, n-1**:

$$a_{j+1} = S_j(x_{j+1}) = S_{j+1}(x_{j+1}) =$$
$$= a_j + b_j \cdot (x_{j+1} - x_j) + c_j \cdot (x_{j+1} - x_j)^2 + d_j \cdot (x_{j+1} - x_j)^3$$

Y reemplazando, nos queda de la siguiente forma:

$$a_{j+1} = S_j(x_{j+1}) = S_{j+1}(x_{j+1}) =$$
$$= a_j + b_j \cdot h_j + c_j \cdot h_j^2 + d_j \cdot h_j^3$$

Polinomio de Spline Cúbico

Derivando el polinomio de Splines, obtenemos:

$$S'_j(x) = b_j + 2 \cdot c_j \cdot (x - x_j) + 3 \cdot d_j \cdot (x - x_j)^2$$

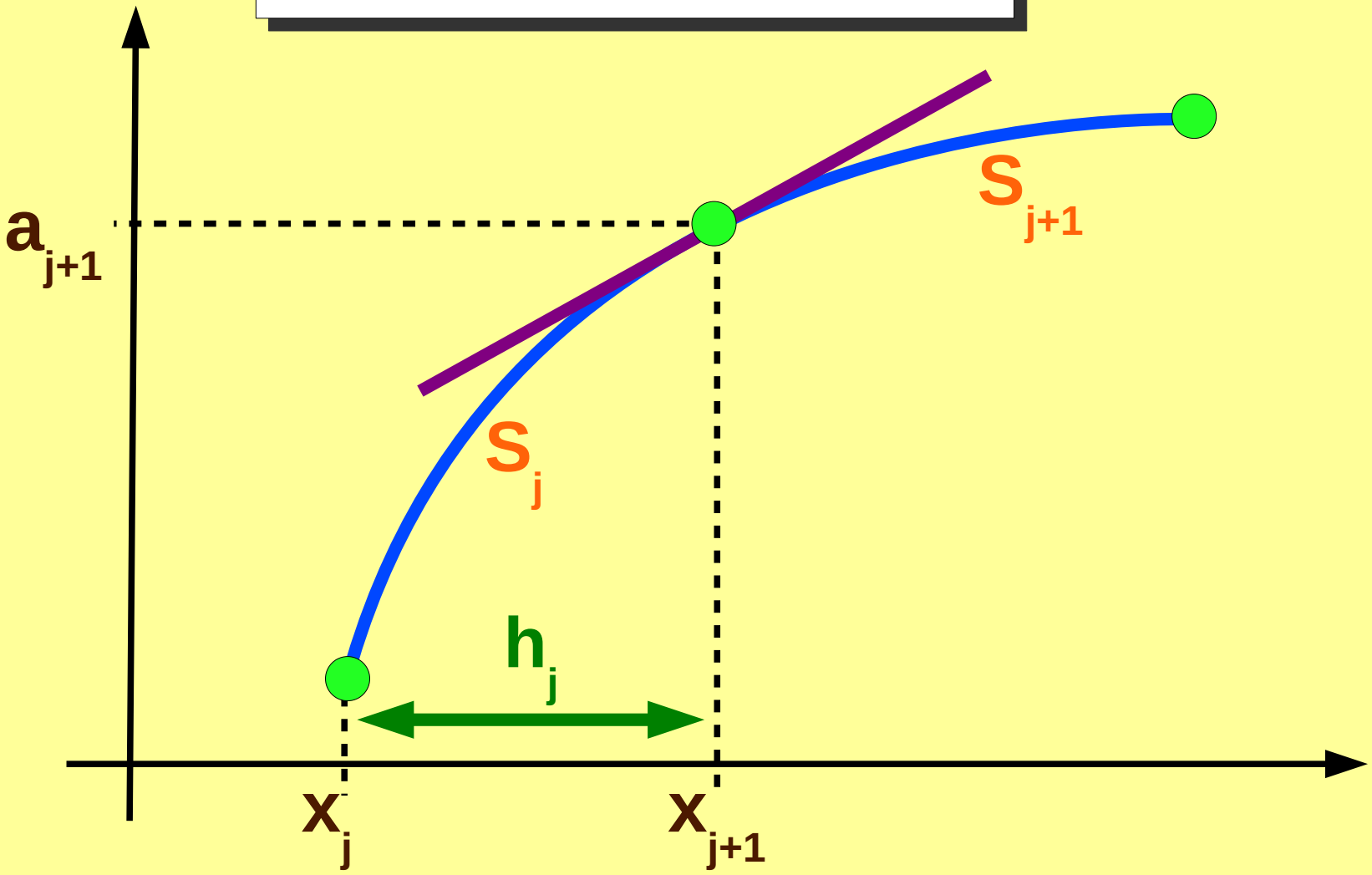
Lo cual implica :

$$S'_j(x_j) = b_j$$

para j = 0, 1, 2, …, n-1

Polinomio de Spline Cúbico

$$S'_j(x_{j+1}) = S'_{j+1}(x_{j+1})$$



Polinomio de Spline Cúbico

Luego, aplicando la condición $\mathbf{S'_{j+1}(x_{j+1}) = S'_{j}(x_{j+1})}$, se tiene que:

$$b_{j+1} = S'_j(x_{j+1}) = S'_{j+1}(x_{j+1}) = \\ = b_j + 2 \cdot c_j \cdot (x_{j+1} - x_j) + 3 \cdot d_j \cdot (x_{j+1} - x_j)^2$$

Es decir:

$$b_{j+1} = S'_j(x_{j+1}) = S'_{j+1}(x_{j+1}) = \\ = b_j + 2 \cdot c_j \cdot h_j + 3 \cdot d_j \cdot h_j^2$$

Polinomio de Spline Cúbico

Derivando nuevamente la expresión **S'$_j$(x)**, obtenemos:

$$S''_j(x) = 2 \cdot c_j + 6 \cdot d_j \cdot (x - x_j)$$

Por lo tanto:

$$c_j = \frac{S''_j(x_j)}{2}$$

Polinomio de Spline Cúbico

Por lo tanto, si:

$$S_j''(x_{j+1}) = 2 \cdot c_j + 6 \cdot d_j \cdot (x_{j+1} - x_j)$$

Y como $\mathbf{S''_{j+1}(x_{j+1}) = 2.c_{j+1} = S''_j(x_{j+1})}$, nos queda:

$$c_{j+1} = c_j + 3 \cdot d_j \cdot h_j$$

para j = 0, 1, 2, …, n-1

Polinomio de Spline Cúbico

Ahora despejando $\mathbf{d_j}$ de la ecuación anterior,

$$c_{j+1} = c_j + 3 \cdot d_j \cdot h_j$$

Nos queda:

$$d_j = \frac{c_{j+1} - c_j}{3 \cdot h_j}$$

para j = 0, 1, 2, …, n-1

Polinomio de Spline Cúbico

Y reemplazando $\mathbf{d_j}$ en $\mathbf{a_{j+1}}$ tenemos:

$$a_{j+1} = S_j(x_{j+1}) = S_{j+1}(x_{j+1}) = \\ = a_j + b_j \cdot h_j + c_j \cdot h_j^2 + d_j \cdot h_j^3$$

Por lo tanto :

$$a_{j+1} = a_j + b_j \cdot h_j + \frac{(2 \cdot c_j + c_{j+1}) \cdot h_j^2}{3}$$

para **j = 0, 1, 2, …, n-1**

Polinomio de Spline Cúbico

Y reemplazando $\mathbf{d_j}$ en $\mathbf{b_{j+1}}$ tenemos:

$$b_{j+1} = S'_j(x_{j+1}) = S'_{j+1}(x_{j+1}) = \\ = b_j \cdot h_j + 2 \cdot c_j \cdot h_j + 3 \cdot d_j \cdot h_j^2$$

Por lo tanto :

$$b_{j+1} = b_j + h_j \cdot (c_j + c_{j+1})$$

para **j = 0, 1, 2, …, n-1**

Polinomio de Spline Cúbico

Por último, a partir de la ecuación de $\mathbf{a_{j+1}}$ obtenemos :

$$a_{j+1} = a_j + b_j \cdot h_j + \frac{(2 \cdot c_j + c_{j+1}) \cdot h_j^2}{3}$$

Y despejando $\mathbf{b_j}$, obtenemos:

$$b_j = \frac{(a_{j+1} - a_j)}{h_j} - \frac{(2 \cdot c_j + c_{j+1}) \cdot h_j}{3}$$

Polinomio de Spline Cúbico

Y a partir de la ecuación de $\mathbf{b_j}$ anteriormente obtenida:

$$b_j = \frac{(a_{j+1} - a_j)}{h_j} - \frac{(2 \cdot c_j + c_{j+1}) \cdot h_j}{3}$$

Si disminuimos en **1**, el índice de $\mathbf{b_j}$, nos queda:

$$b_{j-1} = \frac{(a_j - a_{j-1})}{h_{j-1}} - \frac{(2 \cdot c_{j-1} + c_j) \cdot h_{j-1}}{3}$$

Polinomio de Spline Cúbico

Realizamos la misma disminución en la siguiente expresión:

$$b_{j+1} = b_j + h_j \cdot (c_j + c_{j+1})$$

Con lo que obtenemos:

$$b_j = b_{j-1} + h_{j-1} \cdot (c_{j-1} + c_j)$$

Polinomio de Spline Cúbico

Y reemplazando en :

$$b_{j-1} = \frac{(a_j - a_{j-1})}{h_{j-1}} - \frac{(2 \cdot c_{j-1} + c_j) \cdot h_{j-1}}{3}$$

Arribamos al siguiente sistema de ecuaciones lineales:

$$h_{j-1} \cdot c_{j-1} + 2 \cdot (h_{j-1} + h_j) \cdot c_j + h_j \cdot c_{j+1} = \frac{3 \cdot (a_{j+1} - a_j)}{h_j} - \frac{3 \cdot (a_j - a_{j-1})}{h_{j-1}}$$

Polinomio de Spline Cúbico

El sistema planteado posee **n ecuaciones** y **n+1 incógnitas**, por lo tanto, es preciso utilizar las condiciones de frontera. Por ejemplo, para el caso de **frontera libre** o **SPLINE CÚBICO NATURAL**, tenemos :

$$S''(x_0) = S''(x_n) = 0$$

$$c_n = \frac{S''(x_n)}{2} = 0 \quad \text{porque} \quad S''(x_n)=0$$

$$S''(x_0) = 2 \cdot c_0 + 6 \cdot (x_0 - x_0) = 0 \quad \text{por lo tanto} \quad c_0 = 0$$

Coeficientes de Spline Cúbico

De esta forma, queda conformada una matriz tri-diagonal :

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & \dots & 0 \\ h_0 & 2(h_0+h_1) & h_1 & 0 & \dots & 0 \\ 0 & h_1 & 2(h_1+h_2) & h_2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & h_{n-2} & 2(h_{n-2}+h_{n-1}) & h_{n-1} \\ 0 & 0 & \dots & 0 & 0 & 1 \end{bmatrix}$$

Coeficientes de Spline Cúbico

Y su correspondiente vector de términos independientes :

$$b = \begin{bmatrix} 0 \\ \frac{3(a_2 - a_1)}{h_1} - \frac{3(a_1 - a_0)}{h_0} \\ \vdots \\ \frac{3(a_n - a_{n-1})}{h_{n-1}} - \frac{3(a_{n-1} - a_{n-2})}{h_0} \\ 0 \end{bmatrix}$$

Procedimiento de Cálculo

1. Los valores de $\mathbf{a_j}$ y $\mathbf{h_j}$ se obtienen a partir de los datos y con ellos se arma el sistema.

2. Resolviendo el **sistema tri-diagonal** planteado, se obtienen los valores de $\mathbf{c_j}$.

3. Con los valores de $\mathbf{a_j}$ y $\mathbf{c_j}$ se calculan los valores de $\mathbf{b_j}$.

4. Con los valores de $\mathbf{c_j}$ y $\mathbf{h_j}$ se calculan los valores de $\mathbf{d_j}$.

Subrutina DGTSV

```
      SUBROUTINE DGTSV( N, NRHS, DL, D, DU, B, LDB, INFO )
*
*
*     .. Argumentos Escalares ..
      INTEGER            INFO, LDB, N, NRHS
*     ..
*     .. Argumentos Matriciales ..
      DOUBLE PRECISION   B( LDB, * ), D( * ), DL( * ), DU( * )
*
*
*  Propósito
*  =======
*
*  DGTSV  resuelve el sistema de ecuaciones lineales  A*X = B,
*  donde  A  es una matriz  tri-diagonal de NxN,  por eliminación
*  Gaussiana con pivoteo parcial.
```

Código de Splines Cúbicos

```
SUBROUTINE splineCubico(a, b, c, d, h)
REAL(8) a(0:), h(0:)
REAL(8), ALLOCATABLE :: b(:), c(:), d(:), uD(:), dD(:), lD(:)
INTEGER n, i

  n = SIZE(a)-1
  print *, n
  ALLOCATE(b(0:n), c(0:n), d(0:n))
  ALLOCATE(uD(0:n-1), dD(0:n), lD(0:n-1))

  uD(0) = 0
  dD(0) = 1
  lD(0) = h(0)
  c(0) = 0
```

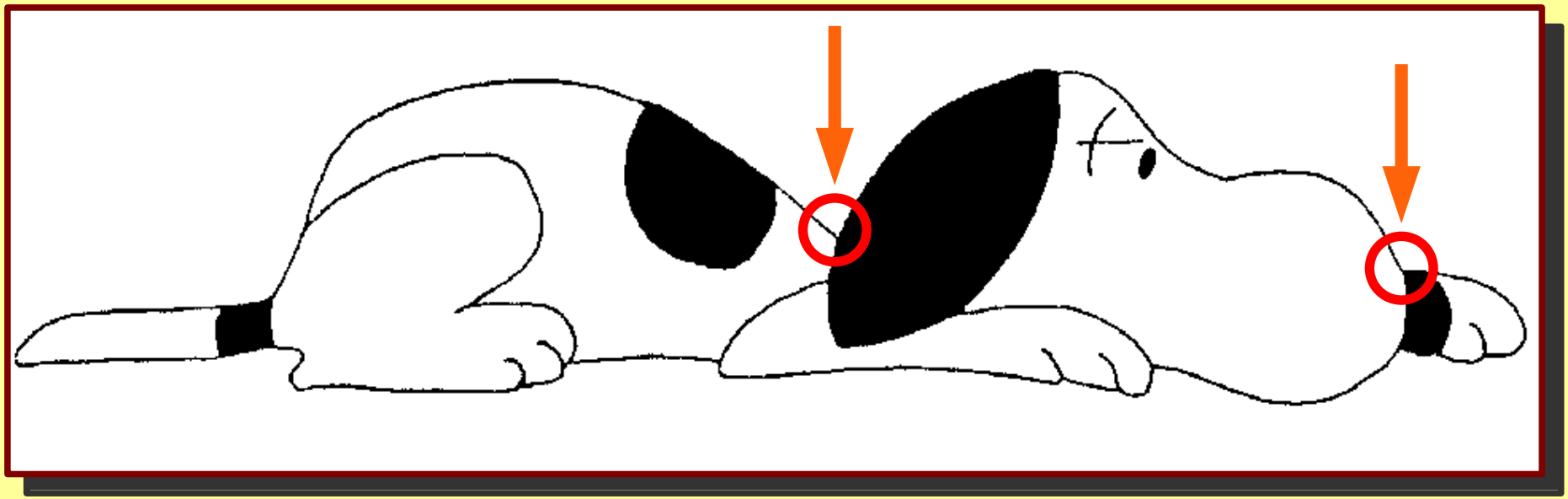
Sique en la próxima página

Código de Splines Cúbicos

```
    DO i=1, n-1
       uD(i) = h(i)
       dD(i) = 2*(h(i-1)+h(i))
       lD(i) = h(i)
       c(i) = 3*((a(i+1)-a(i))/h(i) - (a(i)-a(i-1))/h(i-1))
    END DO

    dD(n) = 1
    lD(n-1) = 0
    c(n) = 0

    CALL DGTSV( n, 1, lD, dD, uD, c, n, INFO )

    DO i=0, n-1
       b(i) = (a(i+1)-a(i))/h(i) - h(i)*(2*c(i)+c(i+1))/3
       d(i) = (c(i+1)-c(i))/(3*h(i))
    ENDDO

    DEALLOCATE(uD, dD, lD)

END SUBROUTINE
```

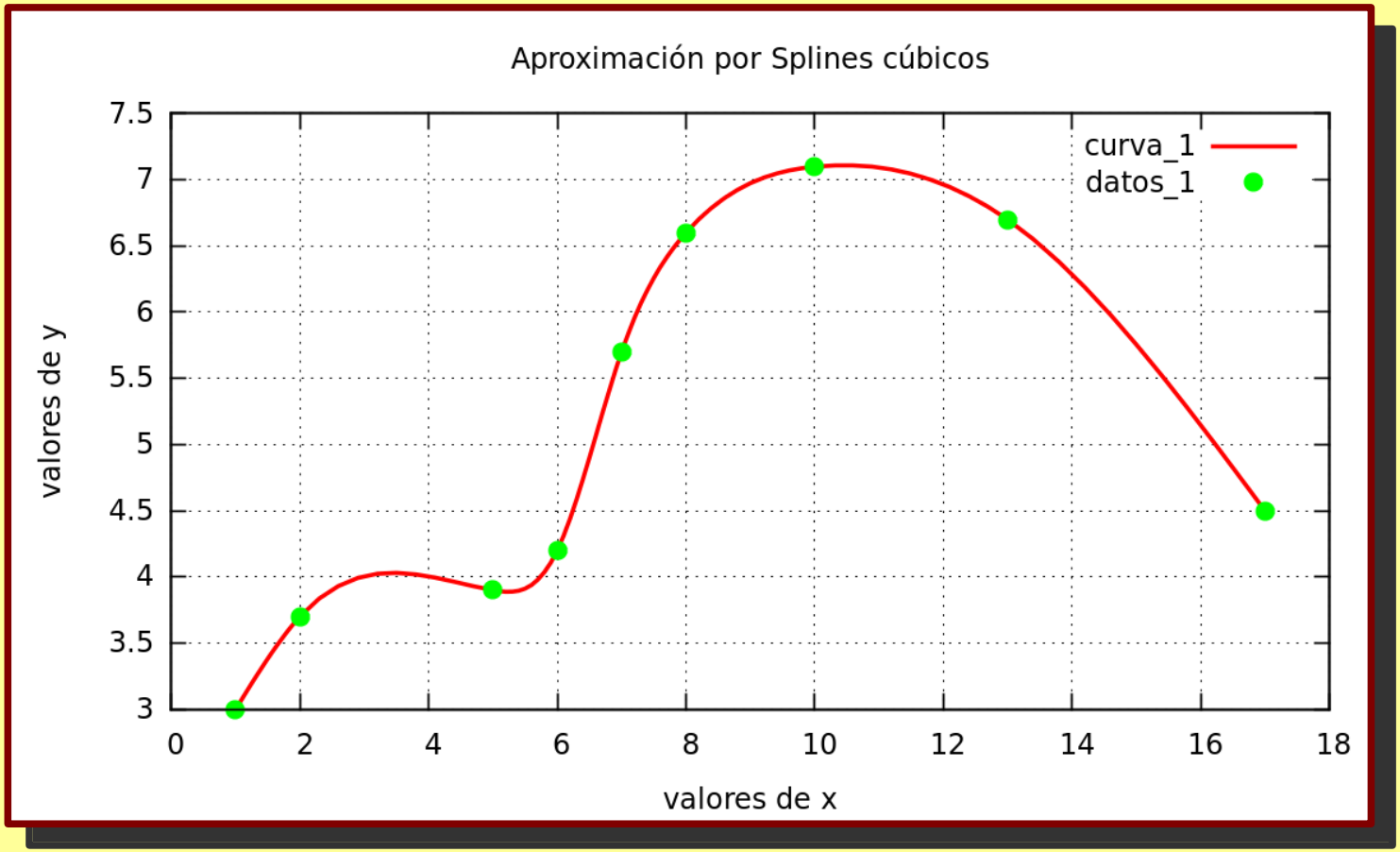
Ejemplo: Perfil de Snoopy

El perfil de la figura se divide en **tres trazadores independientes,** para **evitar** aquellos puntos en los que la **derivada cambia abruptamente**.



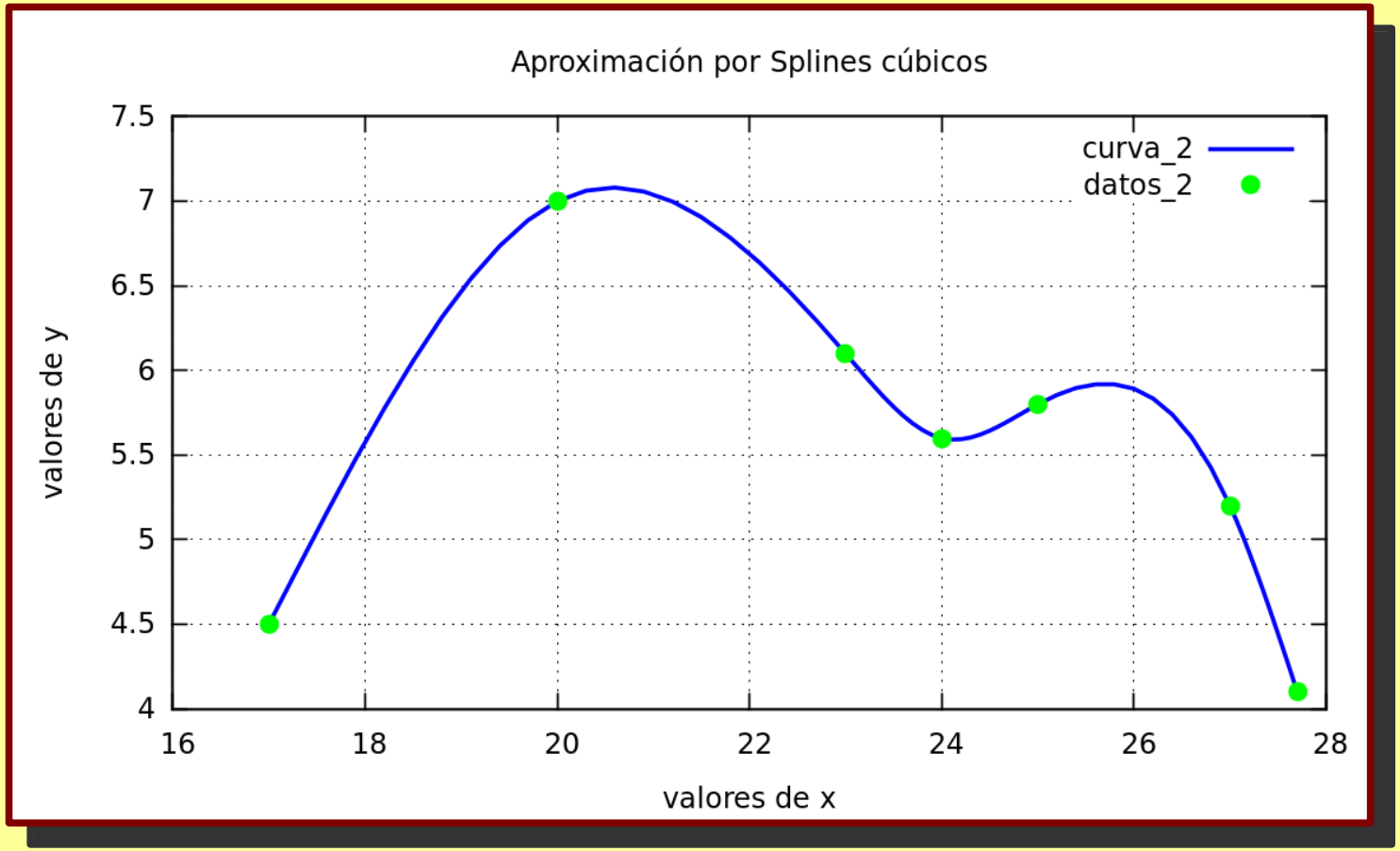
Ejemplo: Trazador 1

x_i	$f(x_i)$
1,00	3,00
2,00	3,70
5,00	3,90
6,00	4,20
7,00	5,70
8,00	6,60
10,00	7,10
13,00	6,70
17,00	4,50



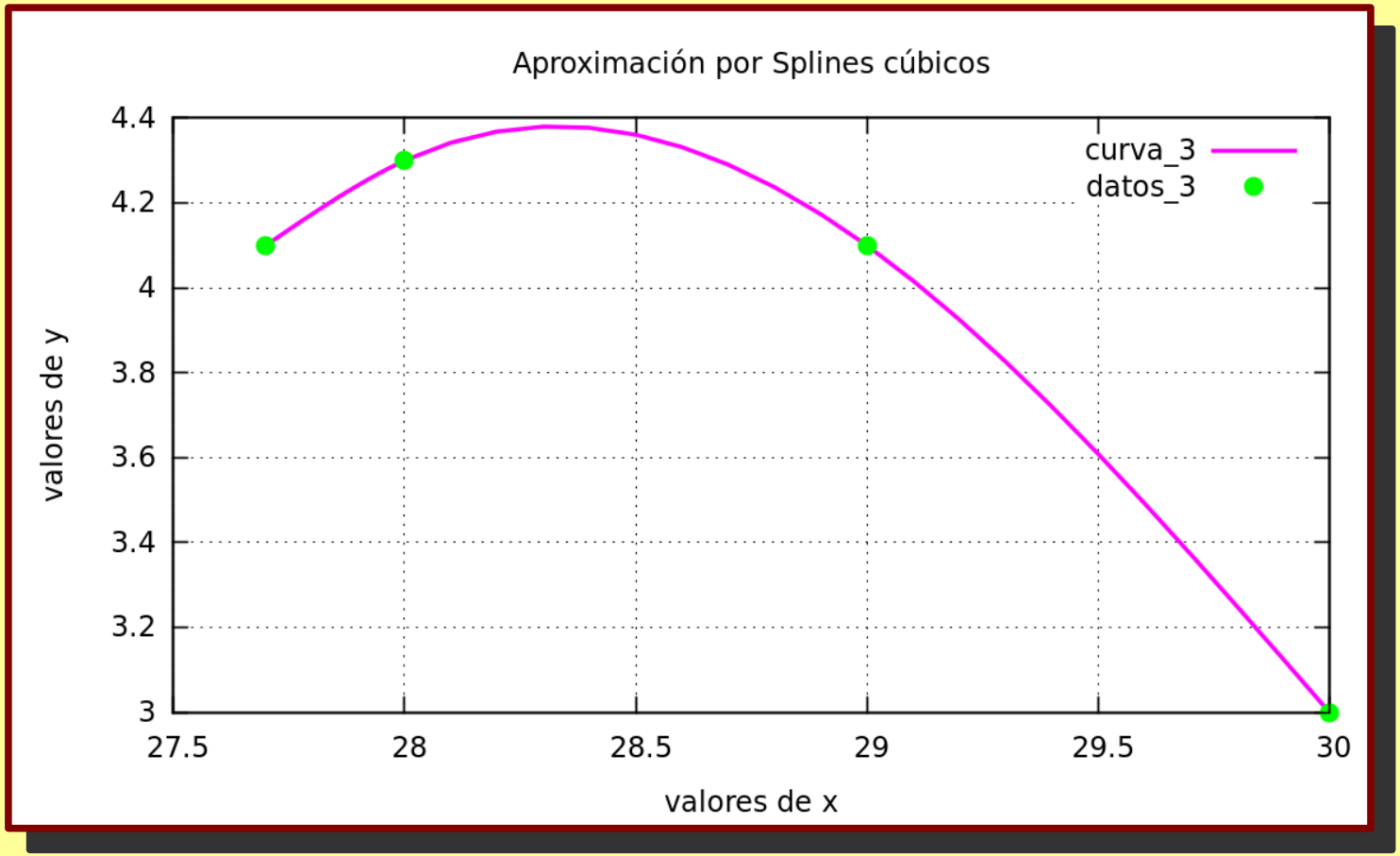
Ejemplo: Trazador 2

x_i	$f(x_i)$
17,00	4,50
20,00	7,00
23,00	6,10
24,00	5,60
25,00	5,80
27,00	5,20
27,70	4,10

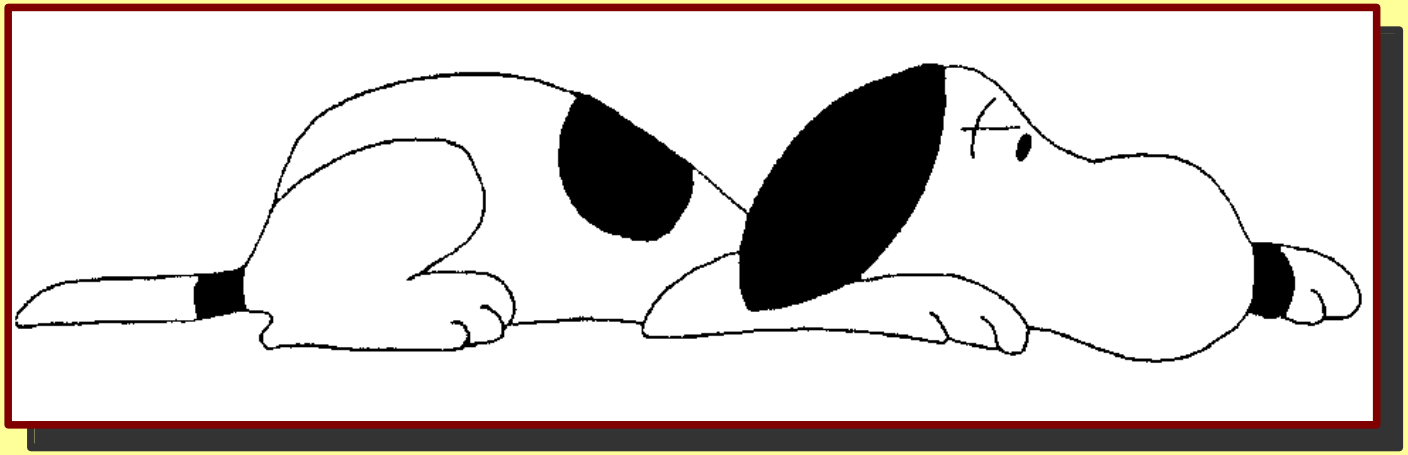


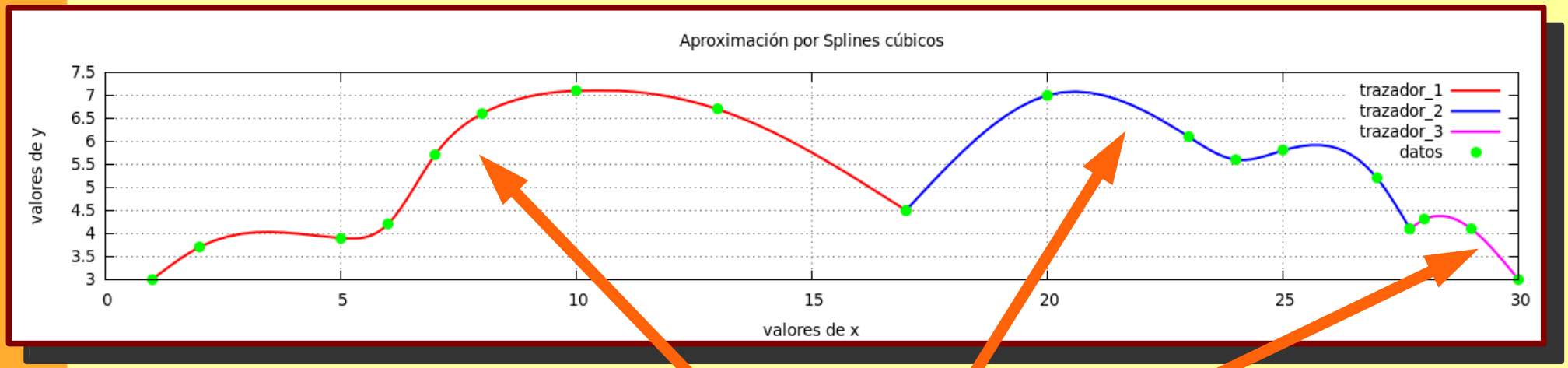
Ejemplo: Trazador 3

x_i	$f(x_i)$
27,70	4,10
28,00	4,30
29,00	4,10
30,00	3,00



Ejemplo: Perfil de Snoopy





TRAZADORES

Aplicaciones de Splines

- Se utilizan cuando es necesario que la/las funciones de interpolación pasen no solamente por los puntos dato, sino que además, copien fielmente la forma enmarcada por los mismos.

- Por esta razón son ampliamente utilizados en programas de dibujo y sistemas de Diseño Asistido por Computadora (CAD).

- Su principal ventaja es que los polinomios resultantes son de grado bajo. La desventaja es que se trata de un tipo de interpolación segmentada.

Aproximación Mínimo-Cuadrática

- Se utiliza un **polinomio de grado n** como función aproximante.
- Para determinar los **coeficientes del polinomio**, se establece como **criterio de aproximación**, que la diferencia entre los valores de la función y los del polinomio evaluado en cada uno de los puntos dato **debe ser mínima**.
- El grado del polinomio **se elige** y **no está determinado** por la cantidad de puntos dato.

Polinomio de Mínimos Cuadrados

Función aproximante :

$$p(x) = a_0 + a_1 \cdot x + a_2 \cdot x^2 + \cdots + a_n \cdot x^n$$

Criterio de aproximación: **Minimizar la siguiente función**

$$F(a_{0,} a_{1,} \cdots, a_n) = \sum_{k=1}^{M} (y_k - p(x_k))^2 = \sum_{k=1}^{M} \varepsilon_k^2$$

Polinomio de Mínimos Cuadrados

Dada la función **F** :

$$F(a_0, a_1, \cdots, a_n) = \sum_{k=1}^{M} (y_k - p(x_k))^2 = \sum_{k=1}^{M} \varepsilon_k^2$$

Para obtener el mínimo, calculamos las derivadas de **F**, con respecto a cada coeficiente y las igualamos a cero.

$$\frac{\partial F}{\partial a_i} = 0$$

para **i = 0, 1, 2, …, n**

Polinomio de Mínimos Cuadrados

Para i = 0 :

$$\frac{\partial}{\partial a_0}\left[\sum_{k=1}^{M}\left(y_k-\sum_{i=0}^{n} a_i\cdot x_k^i\right)^2\right] =$$

$$= \frac{\partial}{\partial a_0}\left[\sum_{k=1}^{M}\left(y_k-\left(a_0+a_1\cdot x_k+a_2\cdot x_k^2+\cdots+a_n\cdot x_k^n\right)\right)^2\right] =$$

$$= \sum_{k=1}^{M} -2\cdot\left[y_k-\left(a_0+a_1\cdot x_k+a_2\cdot x_k^2+\cdots+a_n\cdot x_k^n\right)\right] = 0$$

Polinomio de Mínimos Cuadrados

Para i = 1 :

$$\frac{\partial}{\partial a_1}\left[\sum_{k=1}^{M}\left(y_k-\sum_{i=0}^{n} a_i\cdot x_k^i\right)^2\right] =$$

$$= \frac{\partial}{\partial a_1}\left[\sum_{k=1}^{M}\left(y_k-\left(a_0+a_1\cdot x_k+a_2\cdot x_k^2+\cdots+a_n\cdot x_k^n\right)\right)^2\right] =$$

$$= \sum_{k=1}^{M} -2\cdot\left[y_k-\left(a_0+a_1\cdot x_k+a_2\cdot x_k^2+\cdots+a_n\cdot x_k^n\right)\cdot x_k\right] = 0$$

Polinomio de Mínimos Cuadrados

Para i = n :

$$\frac{\partial}{\partial a_n}\left[\sum_{k=1}^{M}\left(y_k-\sum_{i=0}^{n} a_i\cdot x_k^i\right)^2\right] =$$

$$= \frac{\partial}{\partial a_n}\left[\sum_{k=1}^{M}\left(y_k-\left(a_0+a_1\cdot x_k+a_2\cdot x_k^2+\cdots+a_n\cdot x_k^n\right)\right)^2\right] =$$

$$= \sum_{k=1}^{M} -2\cdot\left[y_k-\left(a_0+a_1\cdot x_k+a_2\cdot x_k^2+\cdots+a_n\cdot x_k^n\right)\cdot x_k^n\right] = 0$$

Polinomio de Mínimos Cuadrados

Generalizando para cualquier valor de i, nos queda:

$$\frac{\partial F}{\partial a_i} = -2 \cdot \sum_{k=1}^{M} \left[y_k - \left(a_0 + a_1 \cdot x_k + a_2 \cdot x_k^2 + \cdots + a_n \cdot x_k^n \right) \cdot x_k^i \right] = 0$$

Para i = 0, 1, 2, …, n

Polinomio de Mínimos Cuadrados

Operando algebraicamente, obtenemos:

$$\sum_{k=1}^{M}\left(y_k-\sum_{j=0}^{n}a_j\cdot x_k^j\right)\cdot x_k^i \;=\; 0$$

$$\sum_{k=1}^{M}\left(y_k\cdot x_k^i-\sum_{j=0}^{n}a_j\cdot x_k^j\cdot x_k^i\right) \;=\; 0$$

$$\sum_{k=1}^{M}y_k\cdot x_k^i-\sum_{k=1}^{M}\sum_{j=0}^{n}a_j\cdot x_k^{j+i} \;=\; 0$$

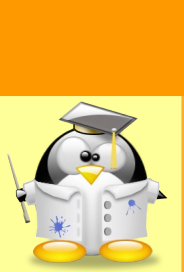
$$\sum_{k=1}^{M}y_k\cdot x_k^i-\sum_{j=0}^{n}a_j\cdot\left(\sum_{k=1}^{M}x_k^{j+i}\right) \;=\; 0$$

Polinomio de Mínimos Cuadrados

Con lo que finalmente obtenemos el siguiente sistema de ecuaciones lineales :

$$\sum_{j=0}^{n} a_j \cdot \left(\sum_{k=1}^{M} x_k^{j+i} \right) = \sum_{k=1}^{M} y_k \cdot x_k^i$$

Para i = 0, 1, 2, …, n



Sistema resultante

$$\begin{bmatrix} M & \sum_{k=1}^{M} x_k & \sum_{k=1}^{M} x_k^2 & \cdots & \sum_{k=1}^{M} x_k^n \\ \sum_{k=1}^{M} x_k & \sum_{k=1}^{M} x_k^2 & \sum_{k=1}^{M} x_k^3 & \cdots & \sum_{k=1}^{M} x_k^{n+1} \\ \sum_{k=1}^{M} x_k^2 & \sum_{k=1}^{M} x_k^3 & \sum_{k=1}^{M} x_k^4 & \cdots & \sum_{k=1}^{M} x_k^{n+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \sum_{k=1}^{M} x_k^n & \sum_{k=1}^{M} x_k^{n+1} & \sum_{k=1}^{M} x_k^{n+2} & \cdots & \sum_{k=1}^{M} x_k^{2n} \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} \sum_{k=1}^{M} y_k \\ \sum_{k=1}^{M} y_k \cdot x_k \\ \sum_{k=1}^{M} y_k \cdot x_k^2 \\ \vdots \\ \sum_{k=1}^{M} y_k \cdot x_k^n \end{bmatrix}$$

Subrutina DPPSV

```
SUBROUTINE DPPSV( UPLO, N, NRHS, AP, B, LDB, INFO )
*
*  Propósito
*  =======
*
*  DPPSV  calcula la solución de un sistema lineal de ecuaciones real
*  A * X = B, donde A es una matriz definida simétrica positiva de NxN
*  almacenada  en  formato empaquetado siendo  X  y  B  matrices de
*  NxNRHS.
*
*  La descomposición de  Cholesky se utiliza para  factorizar  A  como
*     A = U**T* U,  if UPLO = 'U', o
*     A = L * L**T,  if UPLO = 'L',
*  donde  U  es una  triangular superior y  L  es una  triangular inferior.
*
*  La forma factorizada de A es posteriormente utilizada para resolver
*  el sistema de ecuaciones A * X = B.
```

Código de Mínimos Cuadrados

```
SUBROUTINE min_cuad(x, y, grado, coef)
! Calculo de coef
REAL(8) x(:), y(:)
REAL(8), ALLOCATABLE :: aux(:), diag(:), b(:), ap(:), coef(:)
INTEGER n, i, j, grado, l, m, c, INFO

  m = SIZE(x)
  n = grado+1
! Calcula la longitud de ap (La matriz A comprimida)
  l = 0
  DO i=1, n
     l=l+i
  ENDDO
  ALLOCATE(aux(m), diag(2*n-1), b(n), coef(n), ap(l))

  diag(1) = m

  aux = 1
```

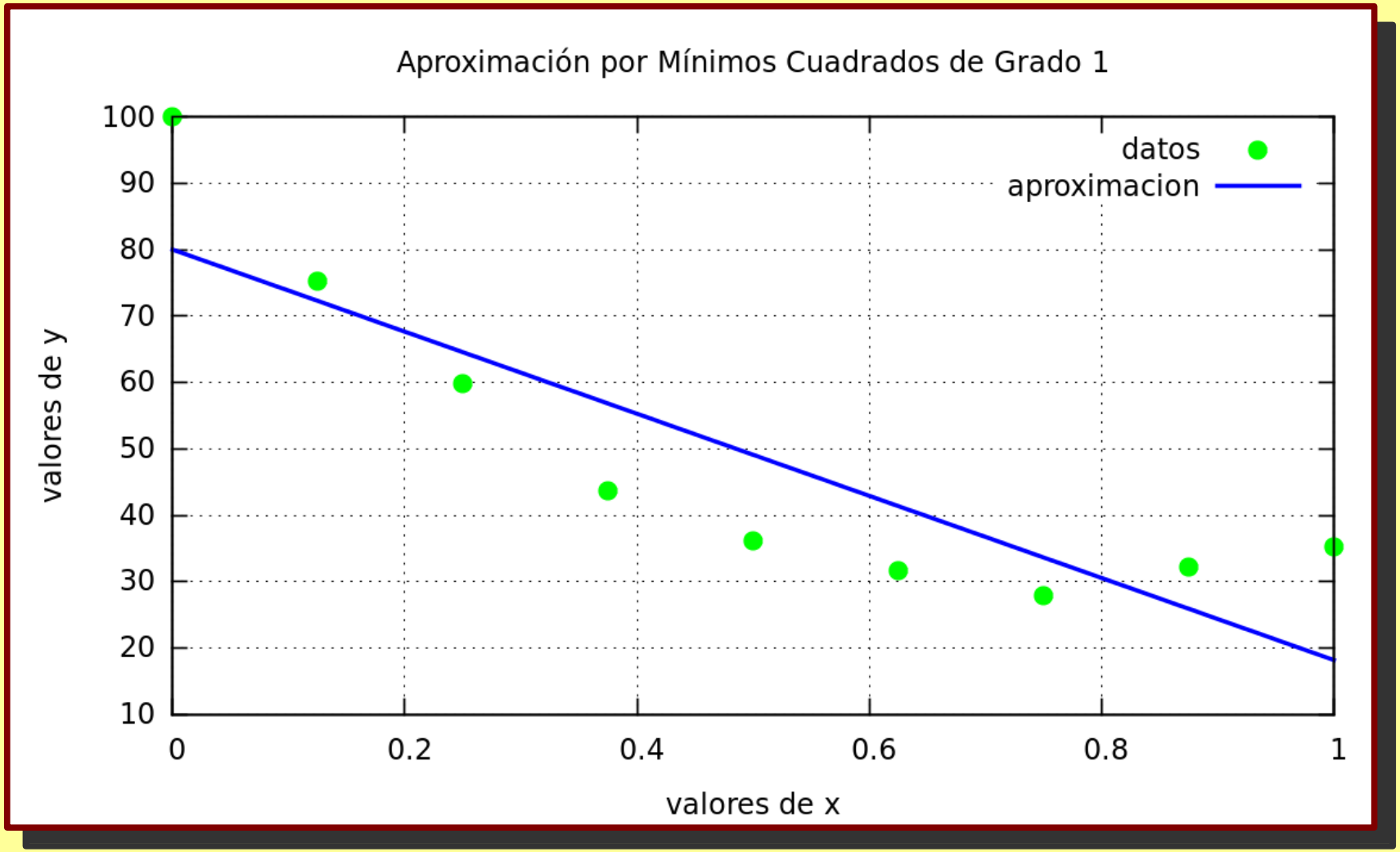
Sique en la próxima página

Código de Mínimos Cuadrados

```
    DO i=1, n
       diag(i+1) = DOT_PRODUCT(aux, x)
       b(i) = DOT_PRODUCT(aux,y)
       aux = aux*x
    ENDDO
    DO i=n+1, 2*n-2
       diag(i+1) = DOT_PRODUCT(aux, x)
       aux = aux*x
    ENDDO
! Arma la matriz A comprimida
    c=1
    DO i=1, n
       DO j=0, i-1
         ap(c) = diag(j+i)
         c=c+1
       ENDDO
    ENDDO
    CALL DPPSV( 'U', n, 1, ap, b, n, INFO )
    coef = b
    DEALLOCATE(aux, diag, b, ap)
END SUBROUTINE
```

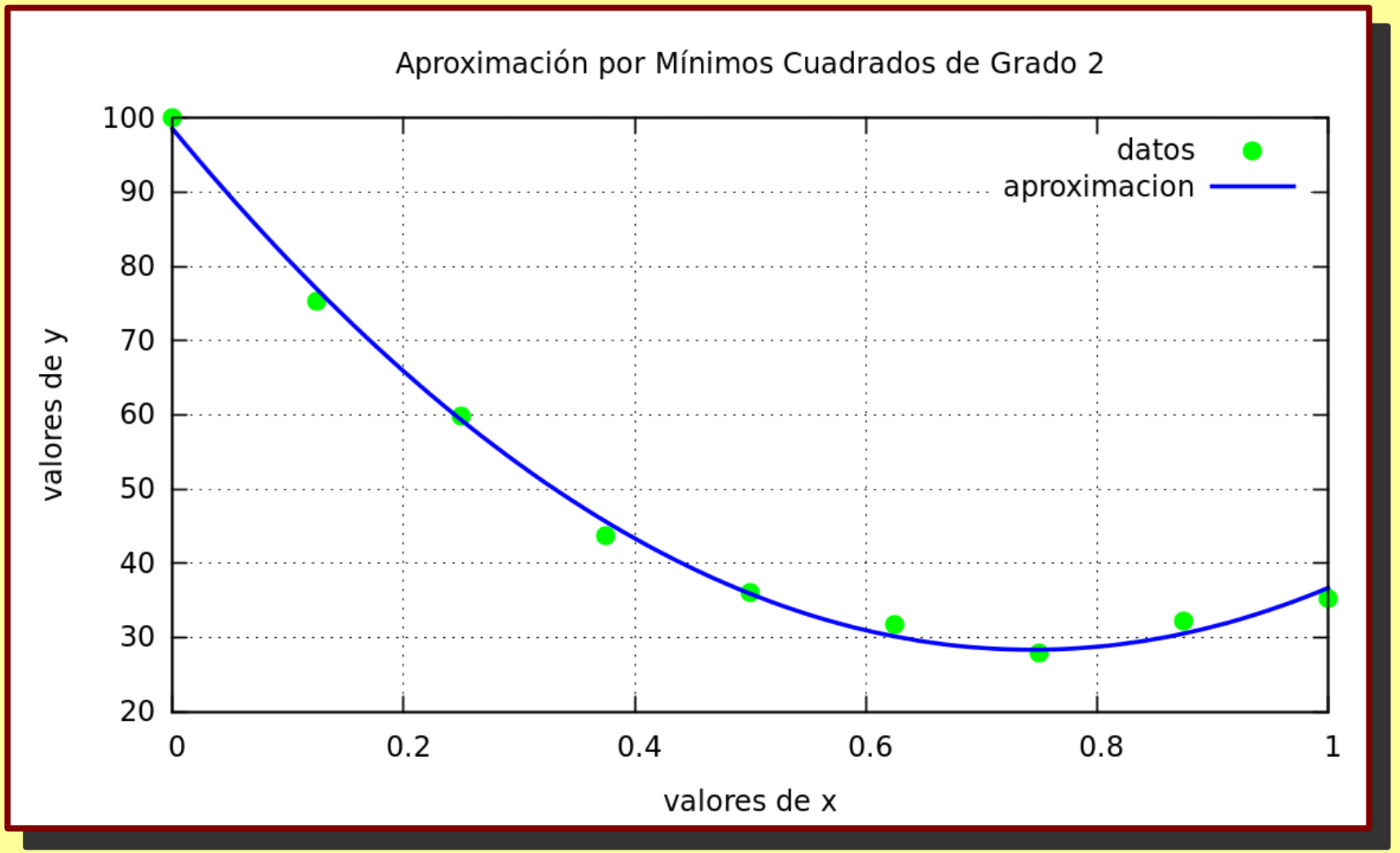
Aproximación de Grado 1

$$p_1(x) = -61.882678 \cdot x + 80.06244$$



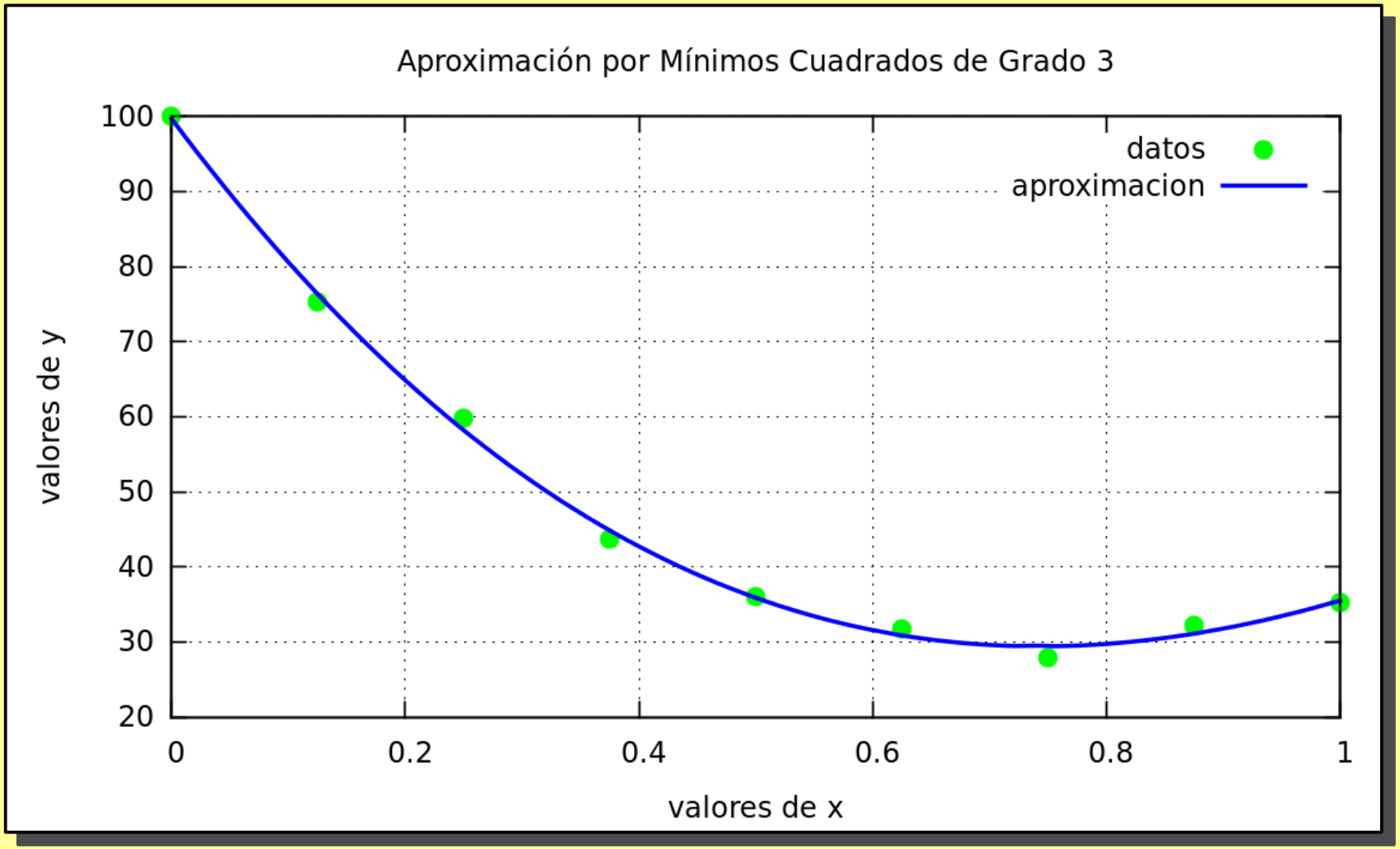
Aproximación de Grado 2

$$p_2(x) = 126.91394 \cdot x^2 - 188.79661 \cdot x + 98.57073$$



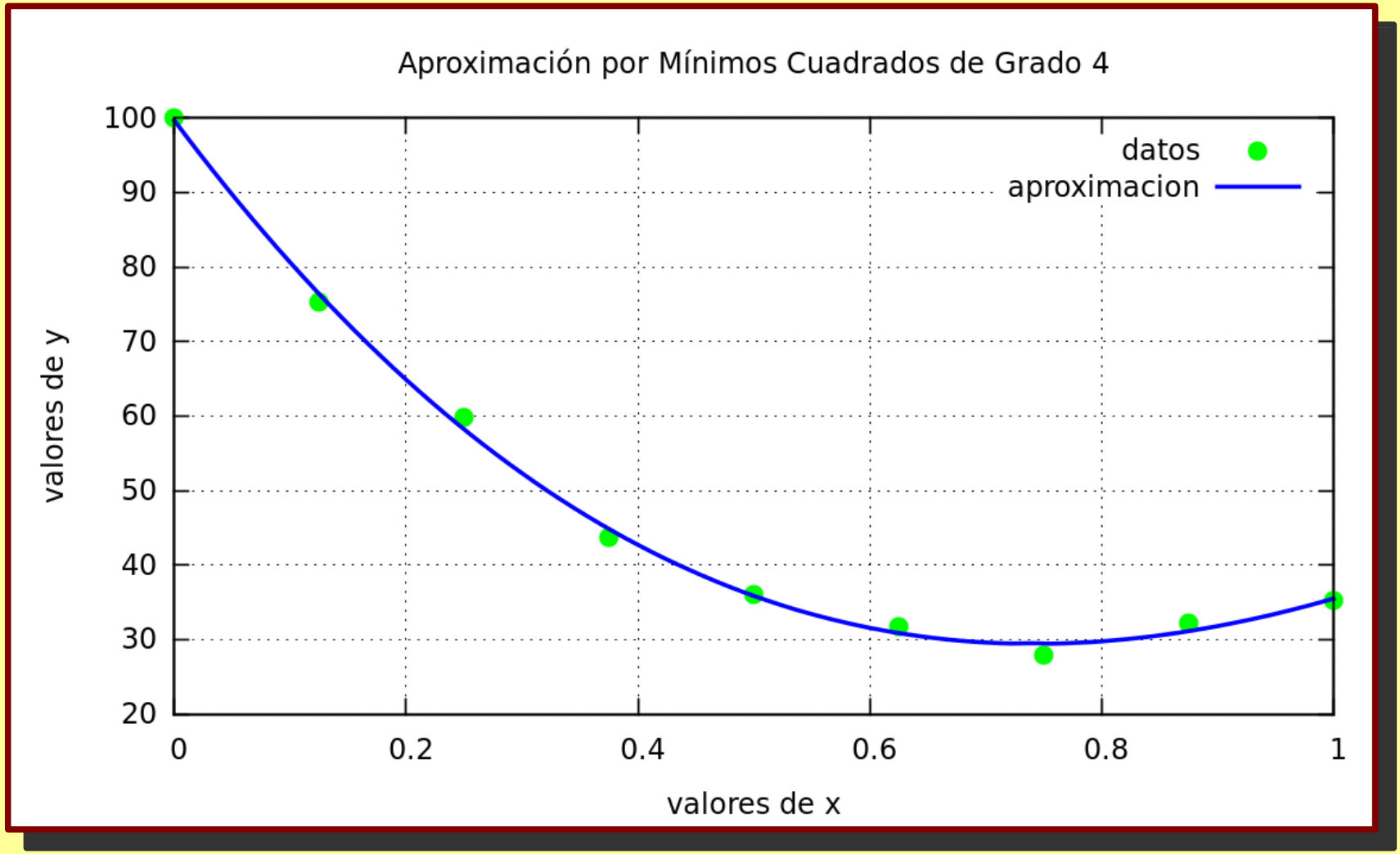
Aproximación de Grado 3

$$p_3(x) = -35.30559 \cdot x^3 + 179.87232 \cdot x^2 - 208.76633 \cdot x + 99.72919$$



Aproximación de Grado 4

$$p_4(x) = -3.00755 \cdot x^4 - 29.29048 \cdot x^3 + 176.03302 \cdot x^2 - 208.03458 \cdot x + 99.71157$$



Datos Infectados COVID-19

Datos oficiales correspondientes al período desde el 03/03/2020 hasta el 19/05/2020. Estos datos fueron recopilados y gentilmente compartidos por el Prof. Roberto Tait

DIA	INFEC
1	0
2	1
3	6
4	8
5	12
6	19
7	23
8	34
9	55
10	57
11	62
12	68
13	78
14	83
15	98
16	110
17	119
18	128
19	158
20	266

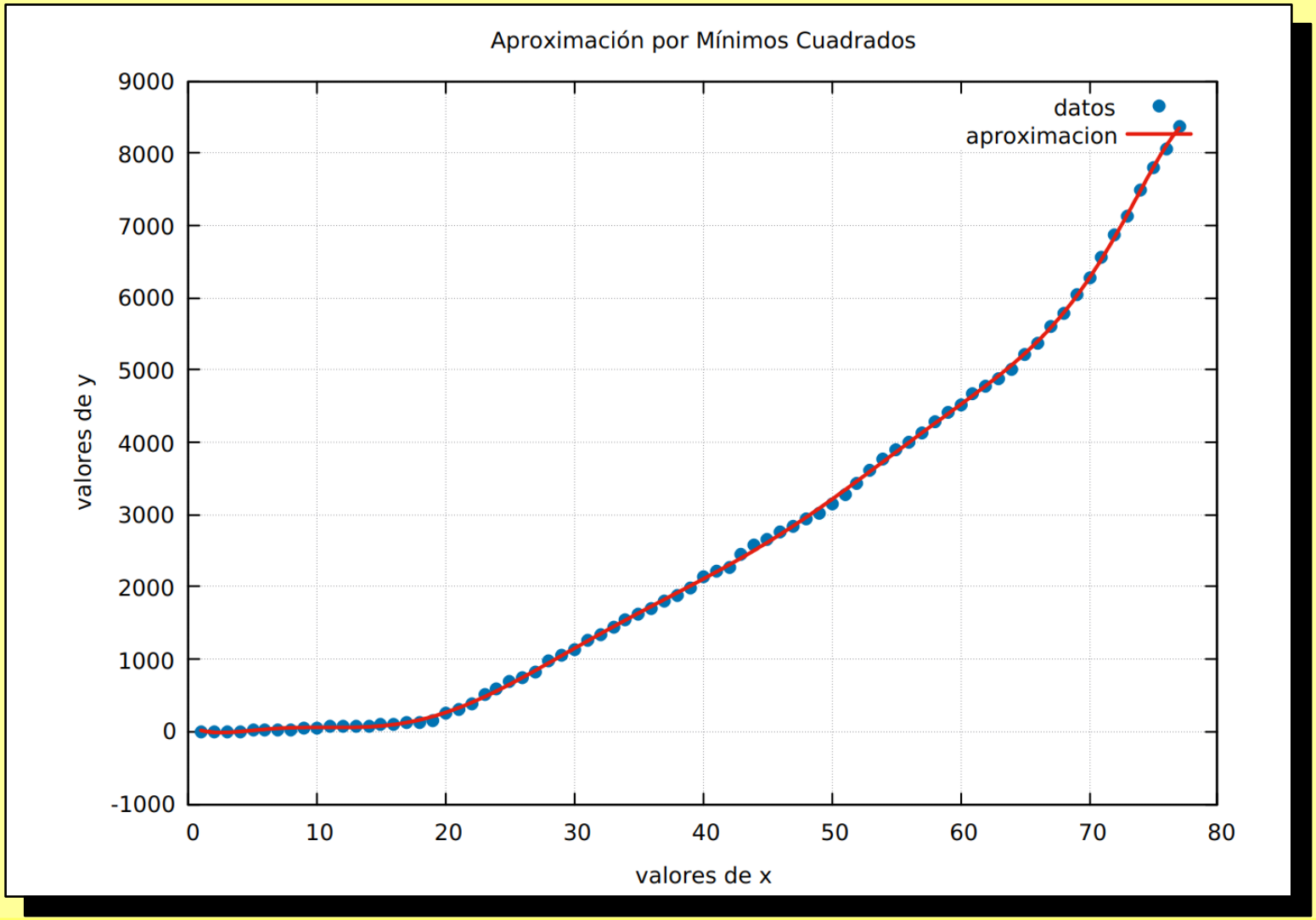
DIA	INFEC
21	301
22	387
23	502
24	589
25	690
26	745
27	820
28	966
29	1054
30	1133
31	1265
32	1353
33	1452
34	1554
35	1628
36	1715
37	1795
38	1894
39	1975
40	2142

DIA	INFEC
41	2208
42	2277
43	2443
44	2571
45	2669
46	2758
47	2839
48	2941
49	3031
50	3144
51	3288
52	3435
53	3607
54	3780
55	3892
56	4003
57	4127
58	4285
59	4428
60	4532

DIA	INFEC
61	4681
62	4783
63	4887
64	5020
65	5208
66	5371
67	5611
68	5776
69	6034
70	6278
71	6563
72	6879
73	7134
74	7479
75	7805
76	8068
77	8371

Aproximación de Grado 10

$$p_{10}(x) = -4.1\cdot10^{-13}\cdot x^{10} + 7.2\cdot10^{-11}\cdot x^{9} + 3.56\cdot10^{-9}\cdot x^{8} - 1.86\cdot10^{-6}\cdot x^{7} + 1.95\cdot10^{-4}\cdot x^{6} - 1.01\cdot10^{-2}\cdot x^{5} + 2.84\cdot10^{-1}\cdot x^{4} - 4.26\cdot x^{3} + 31.9\cdot x^{3} - 96.6\cdot x + 84.5$$



Valor Medio Cuadrático (RMS)

Una vez calculados los coeficientes del polinomio de aproximación de **orden n**, podemos calcular su **Valor Medio Cuadrático.**

$$RMS = \sqrt{\frac{\sum_{k=1}^{M} \varepsilon_k^2}{M}}$$

El Valor Medio Cuadrático, o **Error Medio Cuadrático,** mide el **"ajuste"** del polinomio con referencia a los datos **únicamente**.

Varianza al cuadrado

Una vez calculados los coeficientes del polinomio de aproximación de **orden n**, podemos calcular su **varianza al cuadrado.**

$$\sigma^2(n) = \frac{\sum_{k=1}^{M} \varepsilon_k^2}{M-n-1}$$

Este valor no sólo mide cuánto se aproxima el polinomio a los datos, sino que tiene en cuenta el **esfuerzo computacional involucrado.**

Aplicaciones de Mínimos Cuadrados

- Sirven para aproximar **grandes** conjuntos de puntos, pudiendo **elegir el grado** del polinomio aproximante.

- Esto permite obtener un modelo matemático simplificado del fenómeno representado por los datos.

- Como el criterio de aproximación del método se basa en minimizar el error entre los datos y el aproximante, es posible que dicho polinomio, no coincida exactamente con ninguno de los puntos dato.

PREGUNTAS ...