

# Métodos Numéricos

**Doble Grado en Ingeniería Informática y Matemáticas**  
**Universidad Complutense de Madrid**

**Curso 2023/2024**

**Juan-Carlos Felipe-Navarro**

- Presentación de la Asignatura
- Métodos Directos para Sistemas Lineales
- Introducción a la aproximación
- Interpolación Polinómica
- Interpolación por Splines Polinómicos
- Diferenciación Numérica
- Integración Numérica

# Presentación de la Asignatura

## TEORÍA Y PROBLEMAS

- Profesor: Juan Carlos Felipe
- Mail: jcfelipe@ucm.es

- Horario de clase:

L-XJ- 10:00 - 10:50  
Aula S-116

## TEORÍA Y PROBLEMAS

- Profesor: Juan Carlos Felipe
- Mail: jcfelipe@ucm.es    ↷    **Uso del Foro de Consultas del CV**

- Horario de clase:

L-XJ- 10:00 - 10:50  
Aula S-116

## TEORÍA Y PROBLEMAS

- Profesor: Juan Carlos Felipe
- Mail: jcfelipe@ucm.es    ↷    Uso del Foro de Consultas del CV
- Despacho: QB-646 (Facultad de Químicas, Edificio B, planta 6<sup>a</sup>)
- Horario de clase:

L-XJ- 10:00 - 10:50  
Aula S-116


- Horario de tutorías
  - Miércoles 12:00-13:00 & 15:30-17:30
  - Jueves 11:30-13:00 & 15:00-16:30

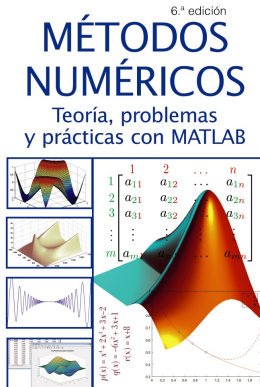
## PRÁCTICAS

- Profesor: Sergio Junquera
- Mail: sejunque@ucm.es     $\rightsquigarrow$     Uso del Foro de Consultas del CV
- Horario de clase:

-M--- 10:00 - 10:50

Aula INF-4

-  Infante, J.A. & Rey, J.M. *Métodos Numéricos. Teoría, problemas y prácticas con MATLAB*. 6ª edición. Ediciones Pirámide. 2022



Juan Antonio Infante del Río  
José María Rey Cabezas

PIRÁMIDE



- Entregas de Problemas
- Prácticas de Matlab
- Examen Final (14 Mayo)/ Extraordinario (12 Junio)

- Entregas de Problemas
- Prácticas de Matlab
- Examen Final (14 Mayo)/ Extraordinario (12 Junio)

$$\text{Nota Final} = \begin{cases} 15\% & \text{Problemas} \\ 15\% & \text{Prácticas} \\ 70\% & \text{Examen} \end{cases}$$

**¿Por qué estudiamos Métodos Numéricos?**

## ¿Por qué estudiamos Métodos Numéricos?

- Muchos resultados matemáticos demuestran la existencia de cierto objeto (la solución de nuestro problema) pero no nos dicen cómo construirlo

# Ejemplo 1

¿Cómo calcular  $\frac{1}{\sqrt{2}}$ ?

# Ejemplo 1

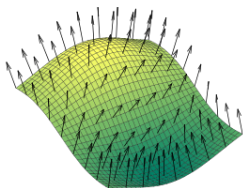
¿Cómo calcular  $\frac{1}{\sqrt{2}}$ ?

¿Por qué querríamos obtener eso?

¿Cómo calcular  $\frac{1}{\sqrt{2}}$ ?

¿Por qué queríamos obtener eso?

- Calcular de forma rápida y precisa  $\frac{1}{\sqrt{N}}$  es crucial en la programación de videojuegos 3D  $\rightsquigarrow$  La iluminación y sombreado de superficies requiere el cálculo de vector normal



¿Cómo calcular  $\frac{1}{\sqrt{2}}$ ?

¿Por qué querríamos obtener eso?

- Calcular de forma rápida y precisa  $\frac{1}{\sqrt{N}}$  es crucial en la programación de videojuegos 3D  $\rightsquigarrow$  La iluminación y sombreado de superficies requiere el cálculo de vector normal

$$v = (v_1, v_2, v_3) \rightsquigarrow \frac{v}{\|v\|} = \frac{1}{\sqrt{v_1^2 + v_2^2 + v_3^2}} (v_1, v_2, v_3)$$

¿Cómo calcular  $\frac{1}{\sqrt{2}}$ ?

¿Por qué querríamos obtener eso?

- Calcular de forma rápida y precisa  $\frac{1}{\sqrt{N}}$  es crucial en la programación de videojuegos 3D  $\rightsquigarrow$  La iluminación y sombreado de superficies requiere el cálculo de vector normal

$$v = (v_1, v_2, v_3) \rightsquigarrow \frac{v}{\|v\|} = \frac{1}{\sqrt{v_1^2 + v_2^2 + v_3^2}} (v_1, v_2, v_3)$$

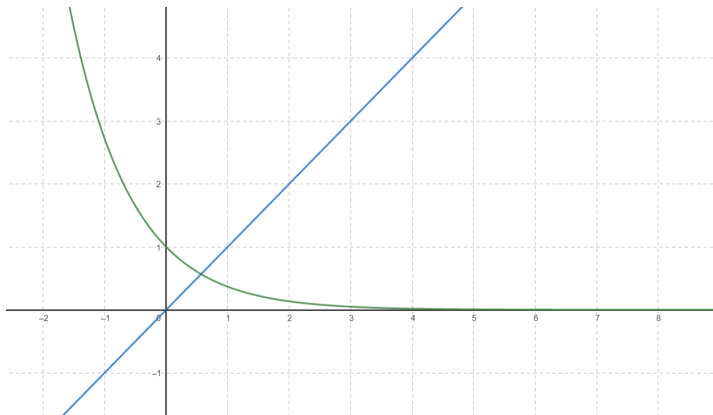
Algoritmo Fast Inverse Square Root

## Ejemplo 2

**¿Cómo encontrar la única solución de la ecuación  $e^{-x} = x$ ?**

## Ejemplo 2

¿Cómo encontrar la única solución de la ecuación  $e^{-x} = x$ ?



**¿Cómo encontrar la única solución de la ecuación  $e^{-x} = x$ ?**

- No hay solución en  $(-\infty, 0]$

**¿Cómo encontrar la única solución de la ecuación  $e^{-x} = x$ ?**

- No hay solución en  $(-\infty, 0]$
- Por Bolzano, hay al menos una solución en  $(0, +\infty)$

**¿Cómo encontrar la única solución de la ecuación  $e^{-x} = x$ ?**

- No hay solución en  $(-\infty, 0]$
- Por Bolzano, hay al menos una solución en  $(0, +\infty)$
- Por monotonía, hay como mucho una solución

**¿Cómo encontrar la única solución de la ecuación  $e^{-x} = x$ ?**

- No hay solución en  $(-\infty, 0]$
- Por Bolzano, hay al menos una solución en  $(0, +\infty)$
- Por monotonía, hay como mucho una solución

Método de bisección / Método de Newton / Método de Punto Fijo

## ¿Por qué estudiamos Métodos Numéricos?

- Muchos resultados matemáticos demuestran la existencia de cierto objeto (la solución de nuestro problema) pero no nos dicen cómo construirlo
- A veces, aun sabiendo como construir un objeto/solución, el procedimiento no es a priori factible en la práctica

## ¿Por qué estudiamos Métodos Numéricos?

- Muchos resultados matemáticos demuestran la existencia de cierto objeto (la solución de nuestro problema) pero no nos dicen cómo construirlo
- A veces, aun sabiendo como construir un objeto/solución, el procedimiento no es a priori factible en la práctica  $\rightsquigarrow$  **límite**

## Ejemplo 3

¿Cómo calcular la integral  $\int_0^{\infty} t^{-0,2} e^{-t} dt$ ?

## Ejemplo 3

¿Cómo calcular la integral  $\int_0^{\infty} t^{-0,2} e^{-t} dt$ ?

- 1 Por definición de integral impropia

$$\int_0^{\infty} t^{-0,2} e^{-t} dt = \lim_{\substack{\varepsilon \rightarrow 0 \\ L \rightarrow +\infty}} \int_{\varepsilon}^L t^{-0,2} e^{-t} dt$$

¿Cómo calcular la integral  $\int_0^{\infty} t^{-0,2} e^{-t} dt$ ?

- 1 Por definición de integral impropia

$$\int_0^{\infty} t^{-0,2} e^{-t} dt = \lim_{\substack{\varepsilon \rightarrow 0 \\ L \rightarrow +\infty}} \int_{\varepsilon}^L t^{-0,2} e^{-t} dt$$

- 2 Por la construcción de la integral de Riemman

$$\int_a^b f(t) dt = \lim_{N \rightarrow +\infty} \frac{b-a}{N} \sum_{i=1}^N f\left(a + i \frac{b-a}{N}\right)$$

## Ejemplo 3

¿Cómo calcular la integral  $\int_0^{\infty} t^{-0,2} e^{-t} dt$ ?

Por tanto,

$$\int_0^{\infty} t^{-0,2} e^{-t} dt = \lim_{\substack{\varepsilon \rightarrow 0 \\ L, N \rightarrow +\infty}} \frac{L - \varepsilon}{N} \sum_{i=1}^N \left( \varepsilon + i \frac{L - \varepsilon}{N} \right)^{-0,2} e^{-(\varepsilon + i \frac{L - \varepsilon}{N})}$$

## Ejemplo 3

¿Cómo calcular la integral  $\int_0^{\infty} t^{-0,2} e^{-t} dt$ ?

Por tanto,

$$\int_0^{\infty} t^{-0,2} e^{-t} dt = \lim_{\substack{\varepsilon \rightarrow 0 \\ L, N \rightarrow +\infty}} \frac{L - \varepsilon}{N} \sum_{i=1}^N \left( \varepsilon + i \frac{L - \varepsilon}{N} \right)^{-0,2} e^{-(\varepsilon + i \frac{L - \varepsilon}{N})}$$

Cuadraturas de Newton-Côtes / Cuadraturas de Newton

## ¿Por qué estudiamos Métodos Numéricos?

- Muchos resultados matemáticos demuestran la existencia de cierto objeto (la solución de nuestro problema) pero no nos dicen cómo construirlo
- A veces, aun sabiendo como construir un objeto/solución, el procedimiento no es a priori factible en la práctica  $\rightsquigarrow$  límite
- Para conseguir la solución se requiere un número de operaciones inasumible a mano pero si para un ordenador

## ¿Por qué estudiamos Métodos Numéricos?

- Muchos resultados matemáticos demuestran la existencia de cierto objeto (la solución de nuestro problema) pero no nos dicen cómo construirlo
- A veces, aun sabiendo como construir un objeto/solución, el procedimiento no es a priori factible en la práctica  $\rightsquigarrow$  límite
- Para conseguir la solución se requiere un número de operaciones inasumible a mano pero si para un ordenador  $\rightsquigarrow$  **Matlab**

**¿Cómo resolver un sistema lineal de un millón de ecuaciones?**

**¿Cómo resolver un sistema lineal de un millón de ecuaciones?**

**¿Por qué querríamos hacer eso?**

**¿Cómo resolver un sistema lineal de un millón de ecuaciones?**

**¿Por qué querríamos hacer eso?**

- En Procesamiento de Imágenes, la mayoría de transformaciones se reducen a la resolución de un sistema lineal

### ¿Cómo resolver un sistema lineal de un millón de ecuaciones?

#### ¿Por qué querríamos hacer eso?

- En Procesamiento de Imágenes, la mayoría de transformaciones se reducen a la resolución de un sistema lineal
- Cada pixel viene representado por 3 variables  $\rightsquigarrow$  Composición RGB

### ¿Cómo resolver un sistema lineal de un millón de ecuaciones?

#### ¿Por qué querríamos hacer eso?

- En Procesamiento de Imágenes, la mayoría de transformaciones se reducen a la resolución de un sistema lineal
- Cada pixel viene representado por 3 variables  $\rightsquigarrow$  Composición RGB



### ¿Cómo resolver un sistema lineal de un millón de ecuaciones?

#### ¿Por qué querríamos hacer eso?

- En Procesamiento de Imágenes, la mayoría de transformaciones se reducen a la resolución de un sistema lineal
- Cada pixel viene representado por 3 variables  $\rightsquigarrow$  Composición RGB

Métodos directos  $\rightsquigarrow$  Cramer / Gauss / Factorización LU / Cholesky

Métodos iterativos  $\rightsquigarrow$  Gauss-Seidel / Relajación

# Objetivo de la asignatura

Encontrar la solución (de forma exacta o aproximada) a problemas de Álgebra Lineal y Análisis

# Objetivo de la asignatura

Encontrar la solución (de forma exacta o aproximada) a problemas de Álgebra Lineal y Análisis

- Diseñar algoritmos
- Analizar algoritmos
- Implementar algoritmos

# Objetivo de la asignatura

Encontrar la solución (de forma exacta o aproximada) a problemas de Álgebra Lineal y Análisis

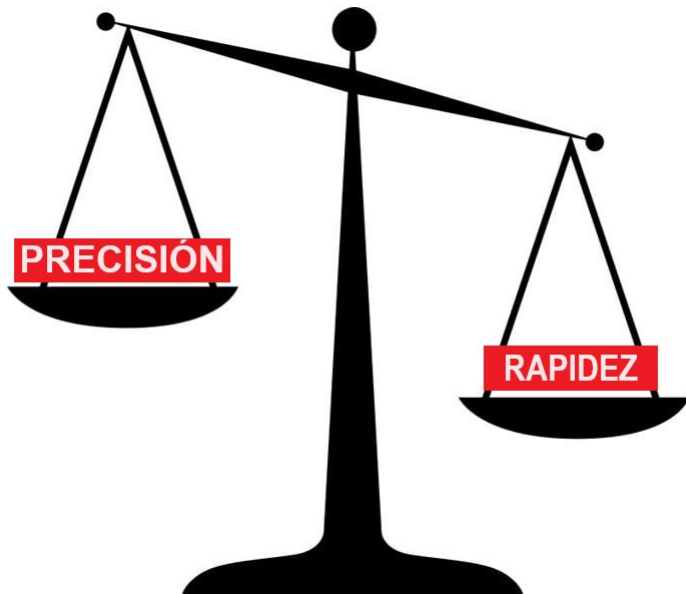
- Diseñar algoritmos
- Analizar algoritmos
- Implementar algoritmos  $\rightsquigarrow$  Matlab

# Objetivo de la asignatura

Encontrar la solución (de forma exacta o aproximada) a problemas de Álgebra Lineal y Análisis

- Diseñar algoritmos
- Analizar algoritmos
- Implementar algoritmos  $\rightsquigarrow$  Matlab





## Introducción y Análisis de Errores

## Introducción y Análisis de Errores

### Parte I: Álgebra Lineal Numérica

- 1 Complementos de Álgebra Lineal
- 2 Condicionamiento de Sistemas Lineales
- 3 Métodos Directos de Resolución de Sistemas Lineales
- 4 Métodos Iterativos de Resolución de Sistemas Lineales

## Introducción y Análisis de Errores

### Parte I: Álgebra Lineal Numérica

- 1 Complementos de Álgebra Lineal
- 2 Condicionamiento de Sistemas Lineales
- 3 Métodos Directos de Resolución de Sistemas Lineales
- 4 Métodos Iterativos de Resolución de Sistemas Lineales

### Parte II: Análisis Numérico

- 5 Interpolación (polinómica y splines)
- 6 Diferenciación Numérica
- 7 Integración Numérica
- 8 Métodos de Resolución de Ecuaciones NO Lineales  $\rightsquigarrow$  Polinómicas

# Métodos Directos para Sistemas Lineales

Dada una matriz invertible  $A \in \mathcal{M}_n(\mathbb{K})$  y un vector  $b \in \mathbb{K}^n$  queremos resolver el sistema

$$Ax = b$$

de la forma más eficiente posible.

# Coste computacional de operaciones básicas

- Producto de matrices (o vector):

$$\text{Dados } B \in \mathcal{M}_{n \times p}(\mathbb{K}) \text{ y } C \in \mathcal{M}_{p \times m}(\mathbb{K}) \rightsquigarrow (BC)_{ij} = \sum_{k=1}^p b_{ik} c_{kj}$$

$$nm[p + (p - 1)] = nm(2p - 1) \quad \text{operaciones}$$

- Cálculo de determinante:

$$\text{Dada } B \in \mathcal{M}_n(\mathbb{K}) \rightsquigarrow |B| = \sum_{\sigma \in S_n} \text{sign}(\sigma) \prod_i^n a_{i, \sigma(i)}$$

$$n!(n - 1) + (n! - 1) = n n! - 1 \quad \text{operaciones}$$

# Método basado en la inversa

## Proposición

Dada una matriz invertible  $A \in \mathcal{M}_n(\mathbb{K})$  tenemos que

$$(A^{-1})_{ij} = \frac{\alpha_{ij}}{|A|},$$

donde

$$\alpha_{ij} = (-1)^{i+j} \begin{vmatrix} a_{11} & \dots & a_{1,j-1} & a_{1,j+1} & \dots & a_{1n} \\ \vdots & & \vdots & \vdots & & \vdots \\ a_{i-1,1} & \dots & a_{i-1,j-1} & a_{i-1,j+1} & \dots & a_{i-1,n} \\ a_{i+1,1} & \dots & a_{i+1,j-1} & a_{i+1,j+1} & \dots & a_{i+1,n} \\ \vdots & & \vdots & \vdots & & \vdots \\ a_{n1} & \dots & a_{n,j-1} & a_{n,j+1} & \dots & a_{nn} \end{vmatrix}$$

El coste computacional de calcular la inversa así es

$$n^2 n! - n^2 - 1 \approx e^{n \log n} \text{ operaciones}$$

# Método basado en la inversa

Por tanto, si calculamos la solución del sistema

$$Ax = b \quad \text{como} \quad x = A^{-1}b$$

se necesitan:

- Cálculo de la inversa

$$n^2 n! - n^2 - 1 \quad \text{operaciones}$$

- Producto inversa por vector

$$n(2n - 1) = 2n^2 - n \quad \text{operaciones}$$

$$\text{Coste total: } n^2 n! + n^2 - n - 1 \quad \text{operaciones}$$

## Proposición

Dada una matriz invertible  $A \in \mathcal{M}_n(\mathbb{K})$ , la solución del sistema

$$Ax = b$$

viene dada por

$$x_i = \frac{|(a_1, \dots, a_{i-1}, b, a_{i+1}, \dots, a_n)|}{|A|}.$$

Por tanto, si calculamos la solución del sistema des esta forma se necesita

- Cálculo de  $(n + 1)$  determinantes  $n \times n \rightsquigarrow (n + 1)nn! - n - 1$  operac.
- Cálculo de  $n$  divisiones

Coste total:  $n(n + 1)! - 1$  operaciones

# Sistema con matriz diagonal

Cuando la matriz del sistema es diagonal  $A = D$ , entonces

$$\begin{pmatrix} d_1 & 0 & \dots & 0 \\ 0 & d_2 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & d_n \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ \vdots \\ \vdots \\ b_n \end{pmatrix}$$

es equivalente a resolver

$$d_i x_i = b_i \quad \text{para todo } i \in \{1, \dots, n\}.$$

Por tanto

$$x_i = b_i/d_i \quad \text{para todo } i \in \{1, \dots, n\}.$$

Coste total:  $n$  operaciones

# Sistema con matriz triangular inferior

Cuando la matriz del sistema es triangular inferior  $A = L$ , entonces

$$\begin{pmatrix} l_{11} & 0 & \dots & 0 \\ l_{21} & l_{22} & & 0 \\ \vdots & & \ddots & \vdots \\ l_{n1} & l_{n2} & \dots & l_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ \vdots \\ \vdots \\ b_n \end{pmatrix}$$

se puede resolver con el algoritmo de sustitución hacia adelante

$$x_1 = b_1 / l_{11}$$

$$x_i = \left( b_i - \sum_{j=1}^{i-1} l_{ij} x_j \right) / l_{ii}$$

$$\text{Coste total: } \sum_{i=1}^n (2i - 1) = n^2 \text{ operaciones}$$

# Sistema con matriz triangular superior

Cuando la matriz del sistema es triangular superior  $A = U$ , entonces

$$\begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & & u_{2n} \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & u_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ \vdots \\ \vdots \\ b_n \end{pmatrix}$$

se puede resolver con el algoritmo de sustitución hacia atrás

$$x_n = b_n / u_{nn}$$

$$x_i = \left( b_i - \sum_{j=i+1}^n u_{ij} x_j \right) / u_{ii}$$

$$\text{Coste total: } \sum_{i=1}^n (2n - 2i - 1) = n^2 \quad \text{operaciones}$$

# Método de Gauss

$$\boxed{Ax = b}$$

$$\begin{cases} A^{(1)} = A \\ b^{(1)} = b \end{cases} \rightsquigarrow \begin{cases} \mathcal{A}^{(1)} \\ \beta^{(1)} \end{cases} \rightsquigarrow \begin{cases} A^{(2)} \\ b^{(2)} \end{cases} \rightsquigarrow \dots \rightsquigarrow \begin{cases} A^{(n)} = U \\ b^{(n)} = \tilde{b} \end{cases}$$

$$\boxed{Ux = \tilde{b}}$$

Una vez transformado el sistema en uno triangular superior resolvemos con algoritmo de sustitución hacia atrás.

# Método de Gauss

Permutación (paso  $k \in \{1 : n - 1\}$ ):

$$(A^{(k)}, b^{(k)}) \rightsquigarrow (\mathcal{A}^{(k)}, \beta^{(k)})$$

Actualización de la matriz

$$\left\{ \begin{array}{l} \alpha_k^{(k)T} = a_{i_k}^{(k)T} \\ \alpha_{i_k}^{(k)T} = a_k^{(k)T} \\ \alpha_i^{(k)T} = a_i^{(k)T} \quad \text{si } i \neq k, i_k \end{array} \right.$$

Actualización del término independiente

$$\left\{ \begin{array}{l} \beta_k^{(k)} = b_{i_k}^{(k)} \\ \beta_{i_k}^{(k)} = b_k^{(k)} \\ \beta_i^{(k)} = b_i^{(k)} \quad \text{si } i \neq k, i_k \end{array} \right.$$

# Método de Gauss

Eliminación (paso  $k \in \{1 : n - 1\}$ ):  $(A^{(k)}, \beta^{(k)}) \rightsquigarrow (A^{(k+1)}, b^{(k+1)})$

$$\forall \text{ fila } i \in \{k+1 : n\} \quad \left\{ \begin{array}{l} l_{ik} = \frac{\alpha_{ik}^{(k)}}{\alpha_{kk}^{(k)}} \quad \text{multiplicador} \\ a_{ij}^{(k+1)} = \alpha_{ij}^{(k)} - l_{ik} \alpha_{kj}^{(k)} \quad \forall \text{ col. } j \in \{k+1 : n\} \\ b_i^{(k+1)} = \beta_i^{(k)} - l_{ik} \beta_k^{(k)} \end{array} \right.$$

Coste total:

$$\left( \sum_{k=1}^{n-1} (n-k) + 2(n-k)^2 + 2(n-k) \right) + n^2 = \frac{2}{3}n^3 + \frac{3}{2}n^2 - \frac{7}{6}n \quad \text{operaciones}$$



El proceso del Método de Gauss se puede resumir en

$$U = A^{(n)} = \left( E^{(n-1)} P^{(n-1)} \dots E^{(1)} P^{(1)} \right) A = MA$$

y por tanto

$$Ax = b \rightsquigarrow (MA)x = (Mb)$$

## Teorema

Sea  $A \in \mathcal{M}_n(\mathbb{K})$  una matriz invertible. Entonces existe  $M \in \mathcal{M}_n(\mathbb{K})$  también invertible tal que  $MA$  es triangular superior.

$$U = MA$$

## Teorema

Sea  $A \in \mathcal{M}_n(\mathbb{K})$  una matriz invertible. Entonces,

$$PA = LU,$$

donde  $L$  es una matriz triangular inferior con unos en la diagonal,  $U$  es una matriz triangular superior y  $P$  es una matriz de permutación.

$$L = \begin{pmatrix} 1 & & & & \\ l_{21} & 1 & & & \\ \vdots & & \ddots & & \\ l_{n-1,1} & & & 1 & \\ l_{n1} & \cdots & \cdots & l_{n,n-1} & 1 \end{pmatrix}, \quad U = \begin{pmatrix} u_{11} & u_{12} & \cdots & \cdots & u_{1n} \\ & u_{22} & & & u_{2n} \\ & & \ddots & & \vdots \\ & & & u_{n-1,n-1} & u_{n-1,n} \\ & & & & u_{nn} \end{pmatrix}$$

$$P = (e_{\sigma(1)}, e_{\sigma(2)}, \dots, e_{\sigma(n)}) \quad \text{con } \sigma \in S^n.$$

# Factorización PALU

## Lema [Problema 16 (Hoja 1)]

Sea

$$E_k = \begin{pmatrix} 1 & & & & & & \\ & 1 & & & & & \\ & & \ddots & & & & \\ & & & 1 & & & \\ & & & l_{k+1,k} & 1 & & \\ & & & \vdots & & \ddots & \\ & & & l_{n,k} & & & 1 \end{pmatrix} \in \mathcal{M}_n$$

con  $k \in \{1, 2, \dots, n-1\}$ . Entonces,

$$E_k = Id + l_k e_k^T \quad \text{y} \quad (E_k)^{-1} = Id - l_k e_k^T$$

donde  $l_k = (0, \dots, 0, l_{k+1,k}, \dots, l_{n,k})^T$ . Además

$$E_1 E_2 \cdots E_{n-1} = Id + \sum_{k=1}^{n-1} l_k e_k^T.$$

## Lema

Dados  $j \geq i > k$ , sean  $E_k = Id + l_k e_k^T$  y

$$P^{ij} = (e_1, e_2, \dots, \overset{i)}{e_j}, \dots, \overset{j)}{e_i}, \dots, e_n).$$

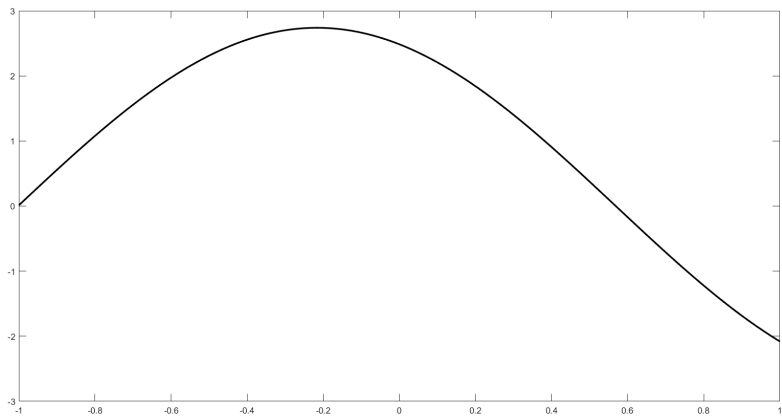
entonces

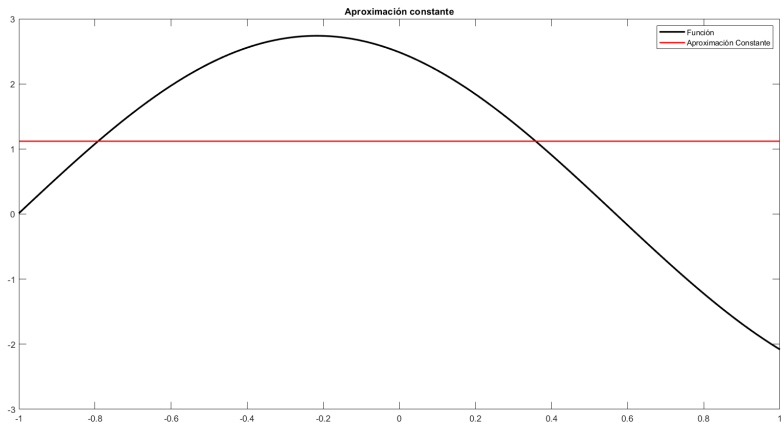
$$P^{ij} E_k P^{ij} = E'_k = Id + l'_k e_k^T,$$

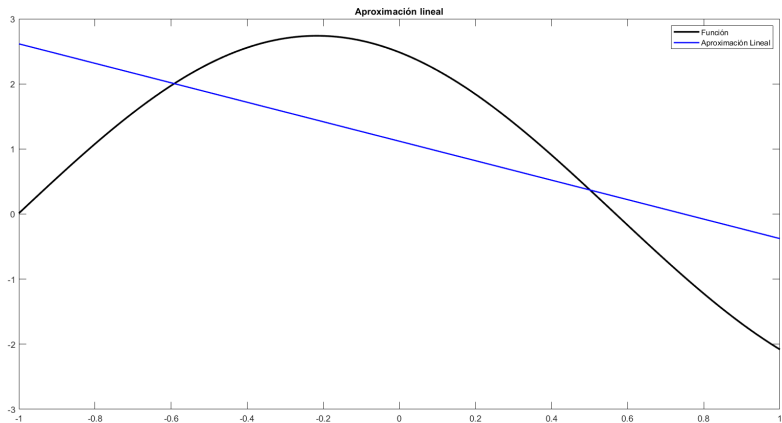
donde

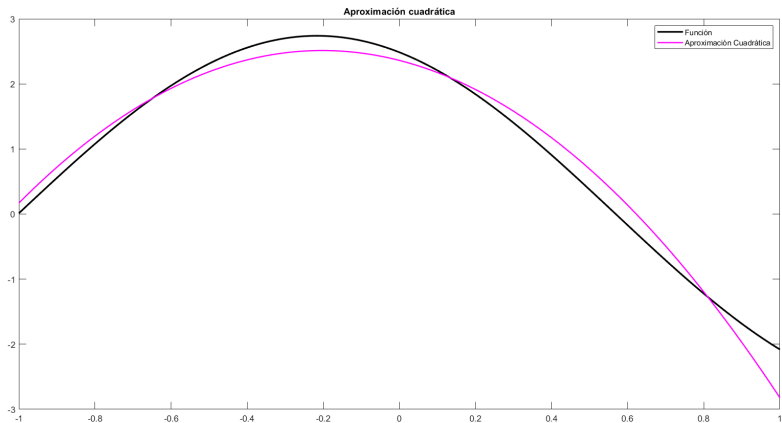
$$l'_k = P^{ij} l_k = (0, 0, \dots, \overset{k)}{0}, \overset{i)}{l_{k+1,k}}, \dots, \overset{j)}{l_{jk}}, \dots, l_{ik}, \dots, l_{nk})^T.$$

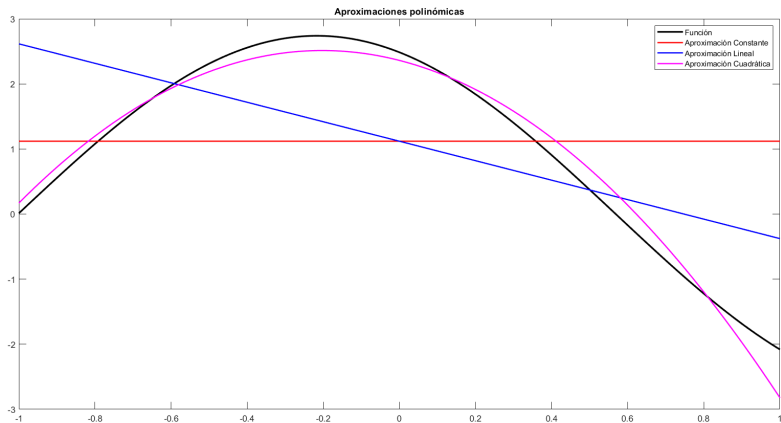
# Introducción a la aproximación

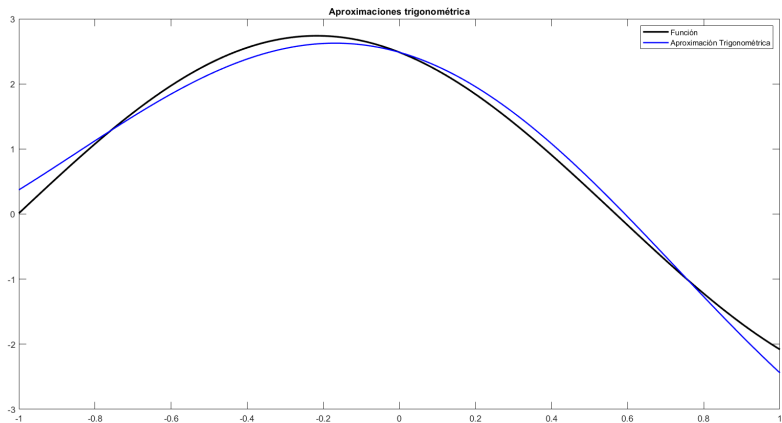


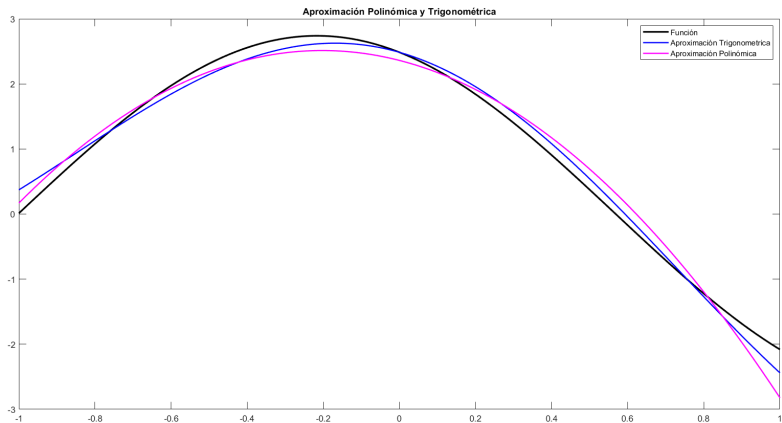


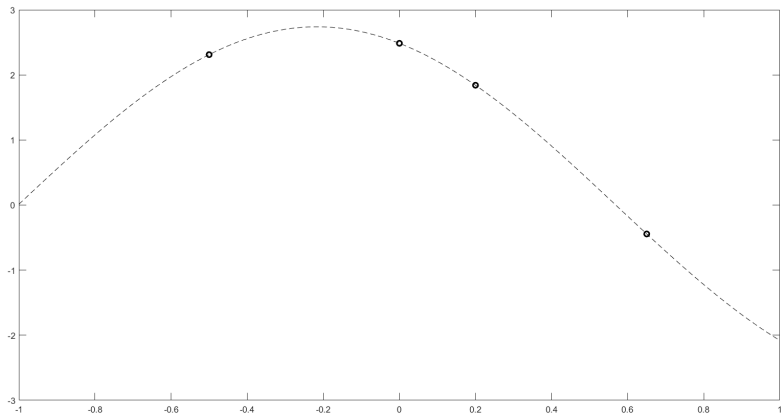


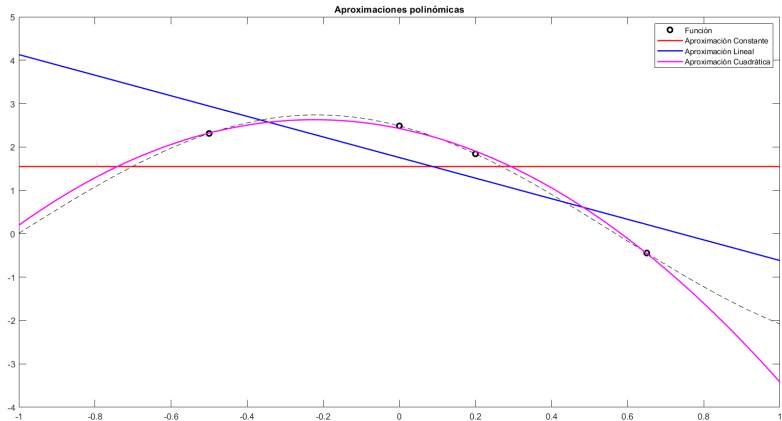


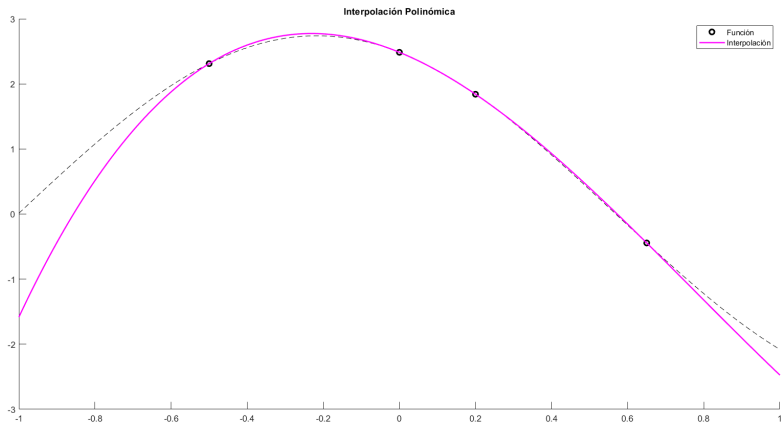


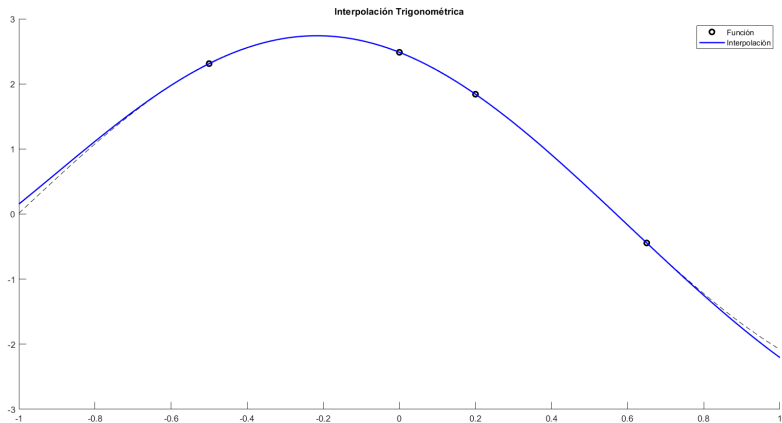


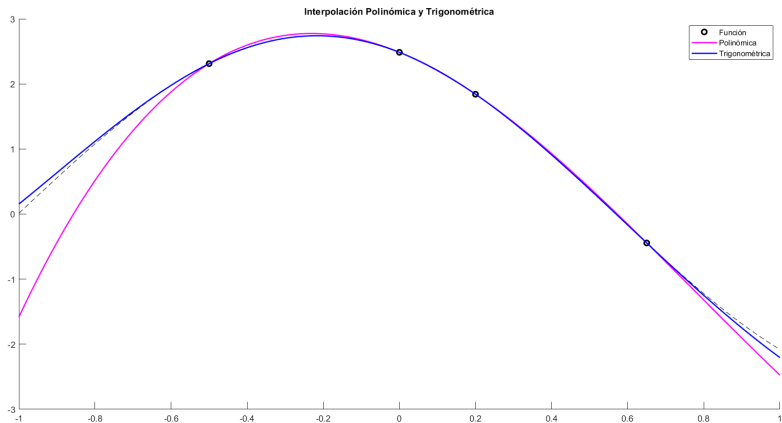












# Interpolación Polinómica

## Teorema

Dada  $f : [a, b] \rightarrow \mathbb{R}$  y  $\{x_0, \dots, x_n\}$ , existe un único  $p \in \mathcal{P}_n$  tal que

$$p(x_i) = f(x_i) \quad \text{para todo } i \in \{0, 1, \dots, n\}.$$

## Teorema

Dada  $f : [a, b] \rightarrow \mathbb{R}$  y  $\{x_0, \dots, x_n\}$ , existe un único  $p \in \mathcal{P}_n$  tal que

$$p(x_i) = f(x_i) \quad \text{para todo } i \in \{0, 1, \dots, n\}.$$

Además, viene dado por la expresión

$$p(x) = \sum_{i=0}^n f(x_i) L_i(x).$$

## Teorema

Dada  $f : [a, b] \rightarrow \mathbb{R}$  y  $\{x_0, \dots, x_n\}$ , existe un único  $p \in \mathcal{P}_n$  tal que

$$p(x_i) = f(x_i) \quad \text{para todo } i \in \{0, 1, \dots, n\}.$$

Además, viene dado por la expresión

$$p(x) = \sum_{i=0}^n f(x_i) L_i(x).$$

Recordamos que la base de Lagrange se define

$$L_i(x) = \prod_{\substack{k=0 \\ k \neq i}}^n \frac{x - x_k}{x_i - x_k}.$$

## Teorema

Dada  $f : [a, b] \rightarrow \mathbb{R}$  y  $\{x_0, \dots, x_n\}$ , el polinomio interpolador  $p \in \mathcal{P}_n$  es

$$p(x) = \sum_{i=0}^n f[x_0, \dots, x_i] N_i(x).$$

## Teorema

Dada  $f : [a, b] \rightarrow \mathbb{R}$  y  $\{x_0, \dots, x_n\}$ , el polinomio interpolador  $p \in \mathcal{P}_n$  es

$$p(x) = \sum_{i=0}^n f[x_0, \dots, x_i] N_i(x).$$

Recordamos que la base de Newton se define

$$N_i(x) = \prod_{k=0}^{i-1} (x - x_k),$$

y las diferencias divididas se obtienen recursivamente

$$\begin{cases} f[x_i] = f(x_i), \\ f[x_i, \dots, x_{i+k}] = \frac{f[x_i, \dots, x_{i+k-1}] - f[x_{i+1}, \dots, x_{i+k}]}{x_i - x_{i+k}}. \end{cases}$$

$$p(x) = \sum_{i=0}^n f[x_0, \dots, x_i] N_i(x)$$

¿Cómo vamos a calcular los coeficientes?

$$p(x) = \sum_{i=0}^n f[x_0, \dots, x_i] N_i(x)$$

¿Cómo vamos a calcular los coeficientes?

$f(x_0)$

$f(x_1)$

$f(x_2)$

$\vdots$

$\vdots$

$f(x_{n-1})$

$f(x_n)$

$$p(x) = \sum_{i=0}^n f[x_0, \dots, x_i] N_i(x)$$

¿Cómo vamos a calcular los coeficientes?

$f[x_0]$

$f[x_1]$

$f[x_2]$

$\vdots$

$\vdots$

$f[x_{n-1}]$

$f[x_n]$

$$p(x) = \sum_{i=0}^n f[x_0, \dots, x_i] N_i(x)$$

¿Cómo vamos a calcular los coeficientes?

$$\begin{array}{ll} f[x_0] & f[x_0, x_1] \\ f[x_1] & f[x_1, x_2] \\ f[x_2] & f[x_2, x_3] \\ \vdots & \vdots \\ \vdots & \vdots \\ f[x_{n-1}] & f[x_{n-1}, x_n] \\ f[x_n] & \end{array}$$

# Fórmula de Newton

$$p(x) = \sum_{i=0}^n f[x_0, \dots, x_i] N_i(x)$$

¿Cómo vamos a calcular los coeficientes?

$$\begin{array}{lll} f[x_0] & f[x_0, x_1] & f[x_0, x_1, x_2] \\ f[x_1] & f[x_1, x_2] & f[x_1, x_2, x_3] \\ f[x_2] & f[x_2, x_3] & f[x_2, x_3, x_4] \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & f[x_{n-2}, x_{n-1}, x_n] \\ f[x_{n-1}] & f[x_{n-1}, x_n] & \\ f[x_n] & & \end{array}$$

# Fórmula de Newton

$$p(x) = \sum_{i=0}^n f[x_0, \dots, x_i] N_i(x)$$

¿Cómo vamos a calcular los coeficientes de forma sencilla?

$$\begin{array}{cccc} f[x_0] & f[x_0, x_1] & f[x_0, x_1, x_2] & \cdots & f[x_0, \dots, x_{n-1}] \\ f[x_1] & f[x_1, x_2] & f[x_1, x_2, x_3] & \cdots & f[x_1, \dots, x_n] \\ f[x_2] & f[x_2, x_3] & f[x_2, x_3, x_4] & & \\ \vdots & \vdots & \vdots & \ddots & \\ \vdots & \vdots & f[x_{n-2}, x_{n-1}, x_n] & & \\ f[x_{n-1}] & f[x_{n-1}, x_n] & & & \\ f[x_n] & & & & \end{array}$$

# Fórmula de Newton

$$p(x) = \sum_{i=0}^n f[x_0, \dots, x_i] N_i(x)$$

¿Cómo vamos a calcular los coeficientes?

$$\begin{array}{ccccccc} f[x_0] & f[x_0, x_1] & f[x_0, x_1, x_2] & \cdots & f[x_0, \dots, x_{n-1}] & f[x_0, \dots, x_n] \\ f[x_1] & f[x_1, x_2] & f[x_1, x_2, x_3] & \cdots & f[x_1, \dots, x_n] & \\ f[x_2] & f[x_2, x_3] & f[x_2, x_3, x_4] & & & \\ \vdots & \vdots & \vdots & \ddots & & \\ \vdots & \vdots & f[x_{n-2}, x_{n-1}, x_n] & & & \\ f[x_{n-1}] & f[x_{n-1}, x_n] & & & & \\ f[x_n] & & & & & \end{array}$$

# Fórmula de Newton

$$p(x) = \sum_{i=0}^n f[x_0, \dots, x_i] N_i(x)$$

¿Cómo vamos a calcular los coeficientes?

$$\begin{array}{ccccccc} f[x_0] & f[x_0, x_1] & f[x_0, x_1, x_2] & \cdots & f[x_0, \dots, x_{n-1}] & f[x_0, \dots, x_n] \\ f[x_1] & f[x_1, x_2] & f[x_1, x_2, x_3] & \cdots & f[x_1, \dots, x_n] & \\ f[x_2] & f[x_2, x_3] & f[x_2, x_3, x_4] & & & \\ \vdots & \vdots & \vdots & \ddots & & \\ \vdots & \vdots & f[x_{n-2}, x_{n-1}, x_n] & & & \\ f[x_{n-1}] & f[x_{n-1}, x_n] & & & & \\ f[x_n] & & & & & \end{array}$$

# Fórmula de Newton

$$p(x) = \sum_{i=0}^n f[x_0, \dots, x_i] N_i(x) + f[x_0, \dots, x_{n+1}] N_{n+1}(x)$$

¿Cómo vamos a calcular los coeficientes?

$$\begin{array}{ccccccc} f[x_0] & f[x_0, x_1] & f[x_0, x_1, x_2] & \cdots & f[x_0, \dots, x_n] & & \\ f[x_1] & f[x_1, x_2] & f[x_1, x_2, x_3] & \ddots & & & \\ \vdots & \vdots & \ddots & & & & \\ f[x_{n-1}] & f[x_{n-1}, x_n] & & & & & \\ f[x_n] & & & & & & \\ f[x_{n+1}] & & & & & & \end{array}$$

Si añadimos un nuevo punto, actualizar el polinomio es muy sencillo

# Fórmula de Newton

$$p(x) = \sum_{i=0}^n f[x_0, \dots, x_i] N_i(x) + f[x_0, \dots, x_{n+1}] N_{n+1}(x)$$

¿Cómo vamos a calcular los coeficientes?

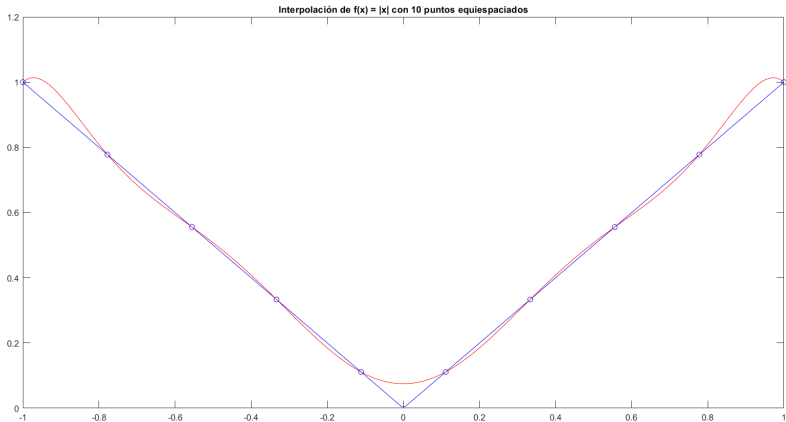
$f[x_0]$	$f[x_0, x_1]$	$f[x_0, x_1, x_2]$	$\dots$	$f[x_0, \dots, x_n]$	$f[x_0, \dots, x_{n+1}]$
$f[x_1]$	$f[x_1, x_2]$	$f[x_1, x_2, x_3]$	$\dots$	$f[x_1, \dots, x_{n+1}]$	
$\vdots$	$\vdots$	$\vdots$	$\ddots$		
$f[x_{n-1}]$	$f[x_{n-1}, x_n]$	$f[x_{n-1}, x_n, x_{n+1}]$			
$f[x_n]$	$f[x_n, x_{n+1}]$				
$f[x_{n+1}]$					

Si añadimos un nuevo punto, actualizar el polinomio es muy sencillo

¿Disminuye el error al aumentar el número de puntos de la interpolación?

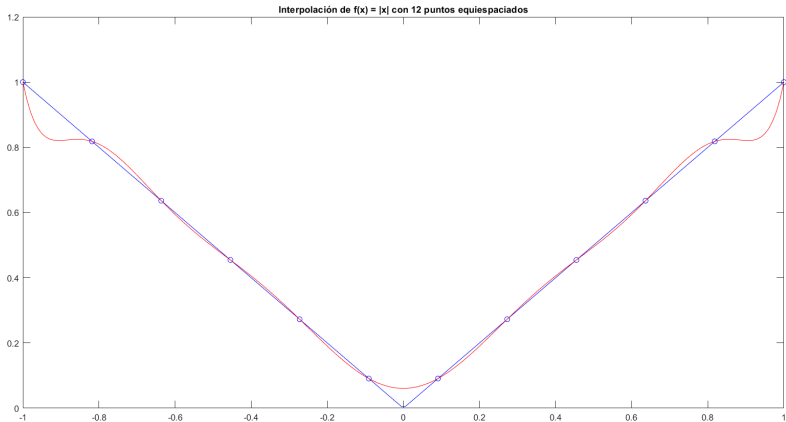
# Error en la aproximación

¿Disminuye el error al aumentar el número de puntos de la interpolación?



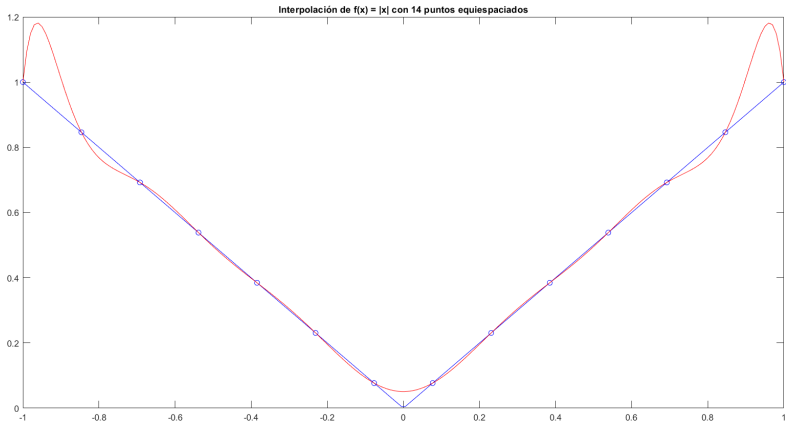
# Error en la aproximación

¿Disminuye el error al aumentar el número de puntos de la interpolación?



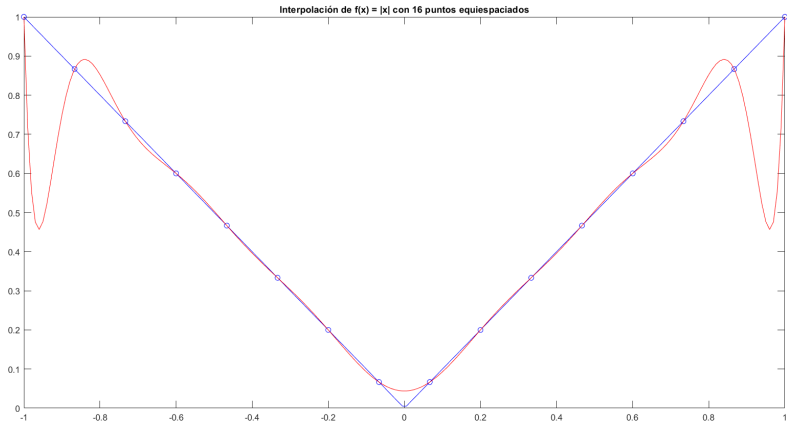
# Error en la aproximación

¿Disminuye el error al aumentar el número de puntos de la interpolación?



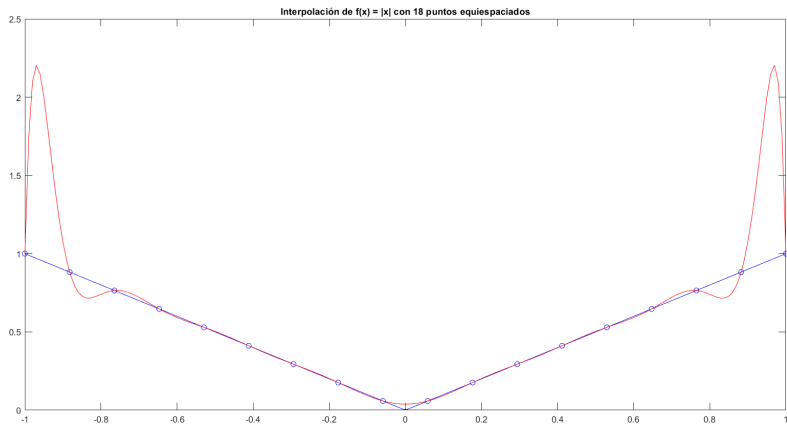
# Error en la aproximación

¿Disminuye el error al aumentar el número de puntos de la interpolación?



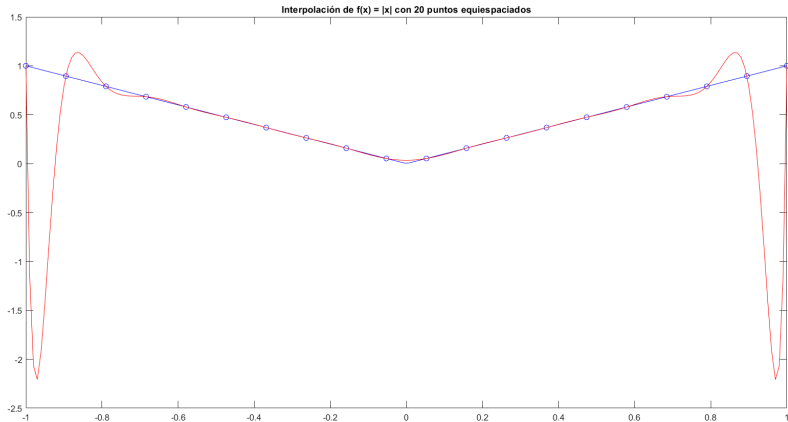
# Error en la aproximación

¿Disminuye el error al aumentar el número de puntos de la interpolación?



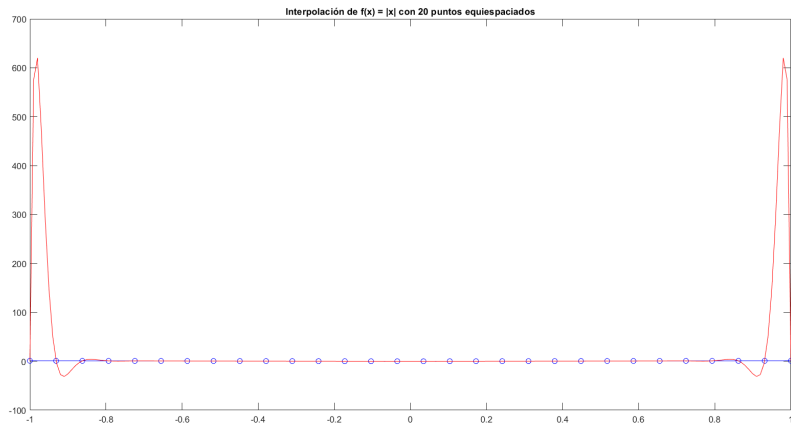
# Error en la aproximación

¿Disminuye el error al aumentar el número de puntos de la interpolación?



# Error en la aproximación

¿Disminuye el error al aumentar el número de puntos de la interpolación?



¿Podemos entonces hacer algo para disminuir el error de la interpolación?

¿Podemos entonces hacer algo para disminuir el error de la interpolación?

## Teorema

Sea  $f \in C^{n+1}([a, b])$  y  $p \in \mathcal{P}_n$  su polinomio interpolador en los puntos  $\{x_0, \dots, x_n\} \subset [a, b]$ . Para cada  $x \in [a, b]$ , existe  $\xi_x \in [a, b]$  tal que

$$f(x) - p(x) = \frac{f^{(n+1)}(\xi_x)}{(n+1)!} \prod_{k=0}^n (x - x_k).$$

¿Podemos entonces hacer algo para disminuir el error de la interpolación?

## Teorema

Sea  $f \in C^{n+1}([a, b])$  y  $p \in \mathcal{P}_n$  su polinomio interpolador en los puntos  $\{x_0, \dots, x_n\} \subset [a, b]$ . Para cada  $x \in [a, b]$ , existe  $\xi_x \in [a, b]$  tal que

$$f(x) - p(x) = \frac{f^{(n+1)}(\xi_x)}{(n+1)!} \prod_{k=0}^n (x - x_k).$$

Si tenemos posibilidad, elegir los puntos donde interpolar nos permite modificar el error

# Puntos para la interpolación

$$f(x) - p(x) = \frac{f^{(n+1)}(\xi_x)}{(n+1)!} \prod_{k=0}^n (x - x_k).$$

# Puntos para la interpolación

$$f(x) - p(x) = \frac{f^{(n+1)}(\xi_x)}{(n+1)!} \prod_{k=0}^n (x - x_k).$$

## Teorema

Dados los puntos  $\{x_0, \dots, x_n\} \subset [a, b]$ , entonces

$$\left\| \prod_{k=0}^n (x - x_k) \right\|_{L^\infty(a,b)} \geq \frac{(b-a)^{n+1}}{2^{2n+1}}.$$

Además, se alcanza la igualdad si escogemos los puntos de la forma

$$x_k = \frac{a+b}{2} + \frac{b-a}{2} \cos\left(\frac{(2k+1)\pi}{2(n+1)}\right).$$

# Puntos para la interpolación

$$f(x) - p(x) = \frac{f^{(n+1)}(\xi_x)}{(n+1)!} \prod_{k=0}^n (x - x_k).$$

## Teorema

Dados los puntos  $\{x_0, \dots, x_n\} \subset [a, b]$ , entonces

$$\left\| \prod_{k=0}^n (x - x_k) \right\|_{L^\infty(a,b)} \geq \frac{(b-a)^{n+1}}{2^{2n+1}}.$$

Además, se alcanza la igualdad si escogemos los puntos de la forma

$$x_k = \frac{a+b}{2} + \frac{b-a}{2} \cos\left(\frac{(2k+1)\pi}{2(n+1)}\right).$$



# Puntos para la interpolación

$$f(x) - p(x) = \frac{f^{(n+1)}(\xi_x)}{(n+1)!} \prod_{k=0}^n (x - x_k).$$

## Teorema

Dados los puntos  $\{x_0, \dots, x_n\} \subset [a, b]$ , entonces

$$\left\| \prod_{k=0}^n (x - x_k) \right\|_{L^\infty(a,b)} \geq \frac{(b-a)^{n+1}}{2^{2n+1}}.$$

Además, se alcanza la igualdad si escogemos los puntos de la forma

$$x_k = \frac{a+b}{2} + \frac{b-a}{2} \cos\left(\frac{(2k+1)\pi}{2(n+1)}\right).$$



# Puntos para la interpolación

$$f(x) - p(x) = \frac{f^{(n+1)}(\xi_x)}{(n+1)!} \prod_{k=0}^n (x - x_k).$$

## Teorema

Dados los puntos  $\{x_0, \dots, x_n\} \subset [a, b]$ , entonces

$$\left\| \prod_{k=0}^n (x - x_k) \right\|_{L^\infty(a,b)} \geq \frac{(b-a)^{n+1}}{2^{2n+1}}.$$

Además, se alcanza la igualdad si escogemos los puntos de la forma

$$x_k = \frac{a+b}{2} + \frac{b-a}{2} \cos\left(\frac{(2k+1)\pi}{2(n+1)}\right).$$



# Puntos para la interpolación

$$f(x) - p(x) = \frac{f^{(n+1)}(\xi_x)}{(n+1)!} \prod_{k=0}^n (x - x_k).$$

## Teorema

Dados los puntos  $\{x_0, \dots, x_n\} \subset [a, b]$ , entonces

$$\left\| \prod_{k=0}^n (x - x_k) \right\|_{L^\infty(a,b)} \geq \frac{(b-a)^{n+1}}{2^{2n+1}}.$$

Además, se alcanza la igualdad si escogemos los puntos de la forma

$$x_k = \frac{a+b}{2} + \frac{b-a}{2} \cos\left(\frac{(2k+1)\pi}{2(n+1)}\right).$$



# Polinomios de Tchebychev

Los puntos para la “mejor” interpolación son, después de un re-escalado, las raíces del polinomio de Tchebychev de orden  $n + 1$ .

## Definición (Polinomios de Tchebychev)

$$\begin{cases} T_0(x) = 1, \\ T_1(x) = x, \\ T_{n+1}(x) = 2x T_n(x) - T_{n-1}(x). \end{cases}$$

# Polinomios de Tchebychev

Los puntos para la “mejor” interpolación son, después de un re-escalado, las raíces del polinomio de Tchebychev de orden  $n + 1$ .

## Definición (Polinomios de Tchebychev)

$$\begin{cases} T_0(x) = 1, \\ T_1(x) = x, \\ T_{n+1}(x) = 2x T_n(x) - T_{n-1}(x). \end{cases}$$

## Proposición

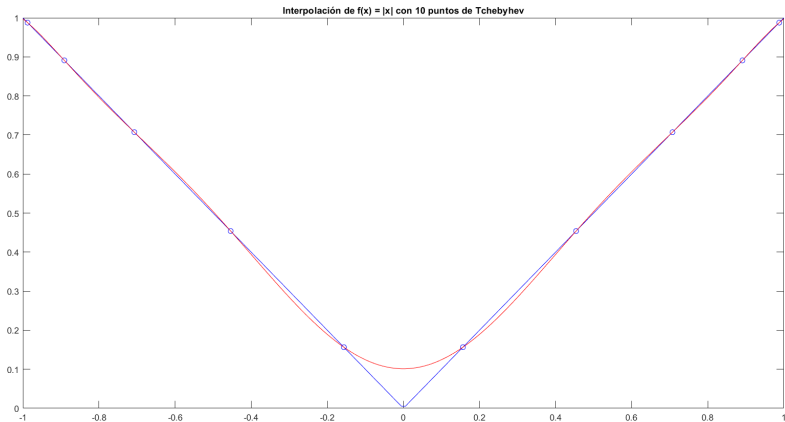
- a)  $T_n$  es un polinomio de grado  $n$  con coeficiente director  $2^{n-1}$  y la misma paridad que  $n$ .
- b)  $T_n(x) = \cos(n \arccos x)$  para todo  $x \in [-1, 1]$ .

# Interpolación en los puntos de Tchebychev

¿Evitamos la oscilación en los extremos (fenómeno de Runge) con estos puntos de interpolación?

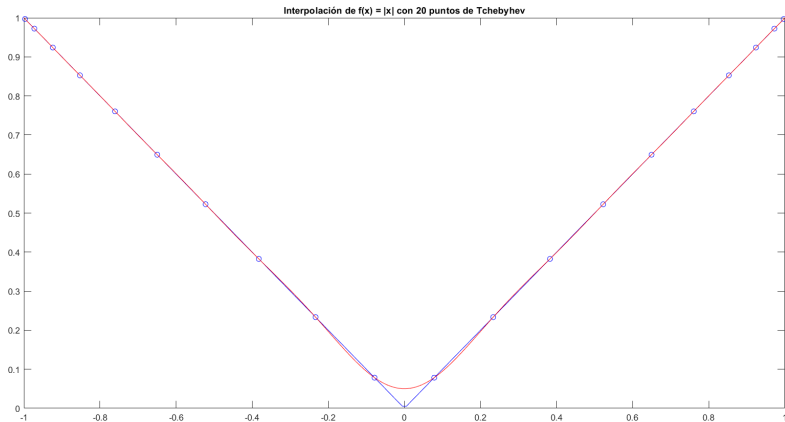
# Interpolación en los puntos de Tchebychev

¿Evitamos la oscilación en los extremos (fenómeno de Runge) con estos puntos de interpolación?



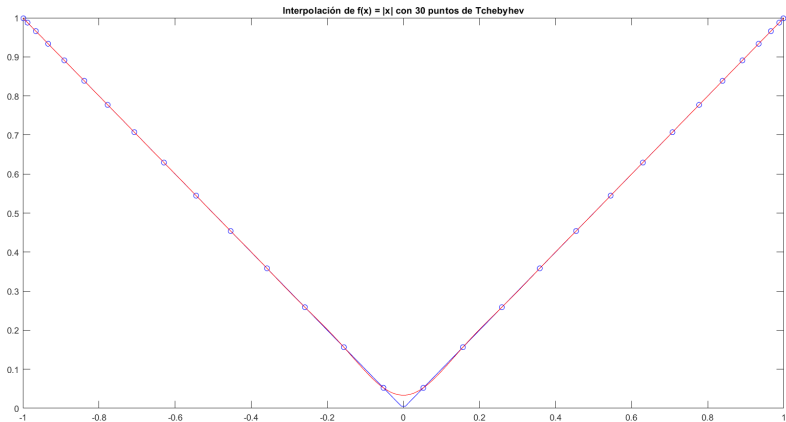
# Interpolación en los puntos de Tchebychev

¿Evitamos la oscilación en los extremos (fenómeno de Runge) con estos puntos de interpolación?



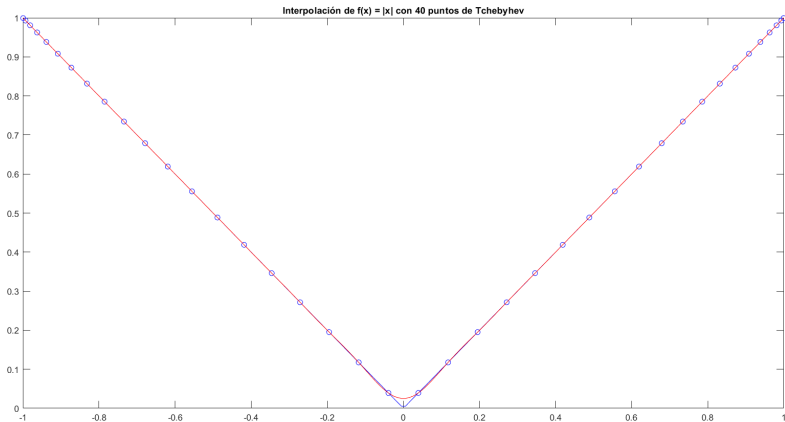
# Interpolación en los puntos de Tchebychev

¿Evitamos la oscilación en los extremos (fenómeno de Runge) con estos puntos de interpolación?



# Interpolación en los puntos de Tchebychev

¿Evitamos la oscilación en los extremos (fenómeno de Runge) con estos puntos de interpolación?



# Interpolación por Splines Polinómicos

El spline cúbico asociado a una partición  $\Delta$  queda determinado de manera única si conocemos las imágenes  $y_i = s(x_i)$  y los momentos  $M_i = s''(x_i)$

$$S(x) = y_j + \left( \frac{y_{j+1} - y_j}{x_{j+1} - x_j} - \frac{M_{j+1} + 2M_j}{6} (x_{j+1} - x_j) \right) (x - x_j) + \frac{M_j}{2} (x - x_j)^2 + \frac{M_{j+1} - M_j}{6(x_{j+1} - x_j)} (x - x_j)^3 \quad \text{en } x \in [x_j, x_{j+1}]$$

El spline cúbico asociado a una partición  $\Delta$  queda determinado de manera única si conocemos las imágenes  $y_i = s(x_i)$  y los momentos  $M_i = s''(x_i)$

$$S(x) = y_j + \left( \frac{y_{j+1} - y_j}{x_{j+1} - x_j} - \frac{M_{j+1} + 2M_j}{6} (x_{j+1} - x_j) \right) (x - x_j) \\ + \frac{M_j}{2} (x - x_j)^2 + \frac{M_{j+1} - M_j}{6(x_{j+1} - x_j)} (x - x_j)^3 \quad \text{en } x \in [x_j, x_{j+1}]$$

¿Cómo podemos calcular los momentos  $M_i$ ?

¿Cómo podemos calcular los momentos  $M_i$ ?

## Teorema

Dada  $f : [a, b] \rightarrow \mathbb{R}$  y una partición  $\Delta = \{x_0 = a, x_1, \dots, x_{n-1}, x_n = b\}$ , sea  $S$  un spline cúbico interpolador asociado a  $\Delta$  de **Tipo I ó II**.







$$S(x) = y_j + \left( \frac{y_{j+1} - y_j}{x_{j+1} - x_j} - \frac{M_{j+1} + 2M_j}{6} (x_{j+1} - x_j) \right) (x - x_j) \\ + \frac{M_j}{2} (x - x_j)^2 + \frac{M_{j+1} - M_j}{6(x_{j+1} - x_j)} (x - x_j)^3 \quad \text{en } x \in [x_j, x_{j+1}]$$

### Demostración (Idea)

Como el spline  $S \in C^2([a, b])$  entonces en particular satisface

$$S'(x_j^-) = S'(x_j^+) \quad \text{para todo } j = 1, \dots, n-1.$$

$$S(x) = y_j + \left( \frac{y_{j+1} - y_j}{x_{j+1} - x_j} - \frac{M_{j+1} + 2M_j}{6} (x_{j+1} - x_j) \right) (x - x_j) \\ + \frac{M_j}{2} (x - x_j)^2 + \frac{M_{j+1} - M_j}{6(x_{j+1} - x_j)} (x - x_j)^3 \quad \text{en } x \in [x_j, x_{j+1}]$$

### Demostración (Idea)

Como el spline  $S \in C^2([a, b])$  entonces en particular satisface

$$S'(x_j^-) = S'(x_j^+) \quad \text{para todo } j = 1, \dots, n-1.$$

Teniendo en cuenta la expresión de  $S$  obtenemos

$$\frac{x_j - x_{j-1}}{6} M_{j-1} + \frac{x_{j+1} - x_{j-1}}{3} M_j + \frac{x_{j+1} - x_j}{6} M_{j+1} = \frac{y_{j+1} - y_j}{x_{j+1} - x_j} - \frac{y_j - y_{j-1}}{x_j - x_{j-1}}.$$

$$S(x) = y_j + \left( \frac{y_{j+1} - y_j}{x_{j+1} - x_j} - \frac{M_{j+1} + 2M_j}{6} (x_{j+1} - x_j) \right) (x - x_j) + \frac{M_j}{2} (x - x_j)^2 + \frac{M_{j+1} - M_j}{6(x_{j+1} - x_j)} (x - x_j)^3 \quad \text{en } x \in [x_j, x_{j+1}]$$

### Demostración (Idea)

Como el spline  $S \in C^2([a, b])$  entonces en particular satisface

$$S'(x_j^-) = S'(x_j^+) \quad \text{para todo } j = 1, \dots, n-1.$$

Teniendo en cuenta la expresión de  $S$  obtenemos

$$\frac{x_j - x_{j-1}}{6} M_{j-1} + \frac{x_{j+1} - x_{j-1}}{3} M_j + \frac{x_{j+1} - x_j}{6} M_{j+1} = \frac{y_{j+1} - y_j}{x_{j+1} - x_j} - \frac{y_j - y_{j-1}}{x_j - x_{j-1}}.$$

Finalmente imponiendo las condiciones de **Tipo I, II ó III** llegamos al sistema deseado.

## Corolario

Sea  $f : [a, b] \rightarrow \mathbb{R}$  y una partición  $\Delta = \{x_0 = a, x_1, \dots, x_{n-1}, x_n = b\}$ .

Entonces **existe un único spline cúbico interpolador** de Tipo I, II ó III.

## Corolario

Sea  $f : [a, b] \rightarrow \mathbb{R}$  y una partición  $\Delta = \{x_0 = a, x_1, \dots, x_{n-1}, x_n = b\}$ .  
Entonces existe un único spline cúbico interpolador de Tipo I, II ó III.

## Demostración

La matriz de los momentos es de **diagonal estrictamente dominante**.

## Corolario

Sea  $f : [a, b] \rightarrow \mathbb{R}$  y una partición  $\Delta = \{x_0 = a, x_1, \dots, x_{n-1}, x_n = b\}$ .  
Entonces existe un único spline cúbico interpolador de Tipo I, II ó III.

## Demostración

La matriz de los momentos es de diagonal estrictamente dominante.

## Teorema

Sea  $f \in C^4([a, b])$  y  $\Delta = \{x_0 = a, x_1, \dots, x_{n-1}, x_n = b\}$  una partición.  
Entonces, existen constante  $C \geq 0$  independiente de  $\Delta$  y  $f$  tal que

$$\|f^{(k)} - s^{(k)}\|_{L^\infty} \leq C \|f^{(iv)}\|_{L^\infty} \frac{\max |x_{j+1} - x_j|^{5-k}}{\min |x_{j+1} - x_j|} \quad \text{para } k = 0, 1, 2, 3.$$

## Corolario

Sea  $f : [a, b] \rightarrow \mathbb{R}$  y una partición  $\Delta = \{x_0 = a, x_1, \dots, x_{n-1}, x_n = b\}$ .  
Entonces existe un único spline cúbico interpolador de Tipo I, II ó III.

## Demostración

La matriz de los momentos es de diagonal estrictamente dominante.

## Teorema

Sea  $f \in C^4([a, b])$  y **partición regular**  $\Delta = \{x_i = a + i \frac{b-a}{n}\}$ . Entonces, existen constante  $C \geq 0$  independiente de  $\Delta$  y  $f$  tal que

$$\|f^{(k)} - s^{(k)}\|_{L^\infty} \leq C \|f^{(4)}\|_{L^\infty} \left(\frac{b-a}{n}\right)^{4-k} \quad \text{para } k = 0, 1, 2, 3.$$

# Diferenciación Numérica

# Teorema de las Medias

## Teorema (de las medias)

Dada  $f \in C([a, b])$  sean  $\xi_i \in (a, b)$  y  $c_i \geq 0$  para todo  $i \in \{0, \dots, n\}$  tal que

$$\sum_{i=0}^n c_i = 1.$$

## Teorema (de las medias)

Dada  $f \in C([a, b])$  sean  $\xi_i \in (a, b)$  y  $c_i \geq 0$  para todo  $i \in \{0, \dots, n\}$  tal que

$$\sum_{i=0}^n c_i = 1.$$

Entonces, existe  $\bar{\xi} \in (a, b)$  tal que

$$\sum_{i=0}^n c_i f(\xi_i) = f(\bar{\xi}).$$

# Teorema de las Medias

## Teorema (de las medias)

Dada  $f \in C([a, b])$  sean  $\xi_i \in (a, b)$  y  $c_i \geq 0$  para todo  $i \in \{0, \dots, n\}$  tal que

$$\sum_{i=0}^n c_i = 1.$$

Entonces, existe  $\bar{\xi} \in (a, b)$  tal que

$$\sum_{i=0}^n c_i f(\xi_i) = f(\bar{\xi}).$$

demo (idea):

Inducción en  $n$  + Bolzano

# Aproximaciones a partir del límite

A partir de la propia definición de derivada obtenemos

$$f'(z) = \lim_{h \rightarrow 0} \frac{f(z+h) - f(z)}{h}$$

# Aproximaciones a partir del límite

A partir de la propia definición de derivada obtenemos

$$f'(z) = \lim_{h \rightarrow 0} \frac{f(z+h) - f(z)}{h} \approx \frac{f(z+h_0) - f(z)}{h_0} \quad \text{si } h_0 \text{ pequeño.}$$

# Aproximaciones a partir del límite

A partir de la propia definición de derivada obtenemos

$$f'(z) = \lim_{h \rightarrow 0} \frac{f(z+h) - f(z)}{h} \approx \frac{f(z+h_0) - f(z)}{h_0} \quad \text{si } h_0 \text{ pequeño.}$$

Alternativamente, también tenemos

$$f'(z) = \lim_{h \rightarrow 0} \frac{f(z+h) - f(z-h)}{2h}$$

# Aproximaciones a partir del límite

A partir de la propia definición de derivada obtenemos

$$f'(z) = \lim_{h \rightarrow 0} \frac{f(z+h) - f(z)}{h} \approx \frac{f(z+h_0) - f(z)}{h_0} \quad \text{si } h_0 \text{ pequeño.}$$

Alternativamente, también tenemos

$$f'(z) = \lim_{h \rightarrow 0} \frac{f(z+h) - f(z-h)}{2h} \approx \frac{f(z+h_0) - f(z-h_0)}{2h_0} \quad \text{si } h_0 \text{ pequeño.}$$

# Aproximaciones a partir del límite

A partir de la propia definición de derivada obtenemos

$$f'(z) = \lim_{h \rightarrow 0} \frac{f(z+h) - f(z)}{h} \approx \frac{f(z+h_0) - f(z)}{h_0} \quad \text{si } h_0 \text{ pequeño.}$$

Alternativamente, también tenemos

$$f'(z) = \lim_{h \rightarrow 0} \frac{f(z+h) - f(z-h)}{2h} \approx \frac{f(z+h_0) - f(z-h_0)}{2h_0} \quad \text{si } h_0 \text{ pequeño.}$$

¿Son buenas estas aproximaciones?

$$f'(z) \approx \frac{f(z+h) - f(z)}{h} \quad \text{con } h \text{ pequeño}$$

$$f'(z) \approx \frac{f(z+h) - f(z)}{h} \quad \text{con } h \text{ pequeño}$$

Por el Teorema de Taylor (hasta orden 1) sabemos

$$f(z+h) = f(z) + hf'(z) + \frac{h^2}{2}f''(\xi), \quad \text{para algún } \xi \in (z, z+h).$$

# Error de la aproximación

$$f'(z) \approx \frac{f(z+h) - f(z)}{h} \quad \text{con } h \text{ pequeño}$$

Por el Teorema de Taylor (hasta orden 1) sabemos

$$f(z+h) = f(z) + hf'(z) + \frac{h^2}{2}f''(\xi), \quad \text{para algún } \xi \in (z, z+h).$$

Obtenemos, reorganizando términos, una expresión para el **error** de la aproximación

$$f'(z) - \frac{f(z+h) - f(z)}{h} = -\frac{h}{2}f''(\xi), \quad \text{para algún } \xi \in (z, z+h).$$

## Error de la aproximación

$$f'(z) \approx \frac{f(z+h) - f(z)}{h} \quad \text{con } h \text{ pequeño}$$

Por el Teorema de Taylor (hasta orden 1) sabemos

$$f(z+h) = f(z) + hf'(z) + \frac{h^2}{2}f''(\xi), \quad \text{para algún } \xi \in (z, z+h).$$

Obtenemos, reorganizando términos, una expresión para el error de la aproximación

$$f'(z) - \frac{f(z+h) - f(z)}{h} = -\frac{h}{2}f''(\xi), \quad \text{para algún } \xi \in (z, z+h).$$

El error de la aproximación es **lineal** en  $h$

$$f'(z) \approx \frac{f(z+h) - f(z-h)}{2h} \quad \text{con } h \text{ pequeño}$$

# Error de la aproximación

$$f'(z) \approx \frac{f(z+h) - f(z-h)}{2h} \quad \text{con } h \text{ pequeño}$$

Por el Teorema de Taylor (hasta orden 2) sabemos

$$f(z+h) = f(z) + hf'(z) + \frac{h^2}{2}f''(z) + \frac{h^3}{6}f'''(\xi_1), \quad \text{para algún } \xi_1 \in (z, z+h)$$

y

$$f(z-h) = f(z) - hf'(z) + \frac{h^2}{2}f''(z) - \frac{h^3}{6}f'''(\xi_2), \quad \text{para algún } \xi_2 \in (z-h, z).$$

## Error de la aproximación

$$f'(z) \approx \frac{f(z+h) - f(z-h)}{2h} \quad \text{si } h \text{ pequeño.}$$

Por el Teorema de Taylor (hasta orden 2) sabemos

$$f(z+h) = f(z) + hf'(z) + \frac{h^2}{2}f''(z) + \frac{h^3}{6}f'''(\xi_1), \quad \text{para algún } \xi_1 \in (z, z+h)$$

y

$$f(z-h) = f(z) - hf'(z) + \frac{h^2}{2}f''(z) - \frac{h^3}{6}f'''(\xi_2), \quad \text{para algún } \xi_2 \in (z-h, z).$$

**Restando** ambas expresiones

$$f(z+h) - f(z-h) = 2hf'(z) + \frac{h^3}{6}f'''(\xi_1) + \frac{h^3}{6}f'''(\xi_2), \quad \text{con } \xi_1, \xi_2 \in (z-h, z+h)$$

## Error de la aproximación

$$f'(z) \approx \frac{f(z+h) - f(z-h)}{2h} \quad \text{con } h \text{ pequeño}$$

Reorganizando los términos

$$f'(z) - \frac{f(z+h) - f(z-h)}{2h} = -\frac{h^2}{6} \frac{f'''(\xi_1) + f'''(\xi_2)}{2}, \quad \text{con } \xi_1, \xi_2 \in (z-h, z+h)$$

## Error de la aproximación

$$f'(z) \approx \frac{f(z+h) - f(z-h)}{2h} \quad \text{con } h \text{ pequeño}$$

Reorganizando los términos

$$f'(z) - \frac{f(z+h) - f(z-h)}{2h} = -\frac{h^2}{6} \frac{f'''(\xi_1) + f'''(\xi_2)}{2}, \quad \text{con } \xi_1, \xi_2 \in (z-h, z+h).$$

Y por el Teorema de las Medias,

$$f'(z) - \frac{f(z+h) - f(z-h)}{2h} = -\frac{h^2}{6} f'''(\xi), \quad \text{con } \xi \in (z-h, z+h).$$

El error de la aproximación es cuadrático en  $h$

# Error de la aproximación

$$f'(z) \approx \frac{f(z+h) - f(z-h)}{2h} \quad \text{con } h \text{ pequeño}$$

Reorganizando los términos

$$f'(z) - \frac{f(z+h) - f(z-h)}{2h} = -\frac{h^2}{6} \frac{f'''(\xi_1) + f'''(\xi_2)}{2}, \quad \text{con } \xi_1, \xi_2 \in (z-h, z+h).$$

Y por el Teorema de las Medias,

$$f'(z) - \frac{f(z+h) - f(z-h)}{2h} = -\frac{h^2}{6} f'''(\xi), \quad \text{con } \xi \in (z-h, z+h).$$

El error de la aproximación es cuadrático en  $h$

# Derivadas de orden superior

Con desarrollos de Taylor también podemos encontrar aproximaciones de derivadas de **orden superior**.

# Derivadas de orden superior

Con desarrollos de Taylor también podemos encontrar aproximaciones de derivadas de orden superior. Así es,

$$f(z+h) = f(z) + hf'(z) + \frac{h^2}{2}f''(z) + \frac{h^3}{6}f'''(z) + \frac{h^4}{24}f^{(4)}(\xi_1), \quad \xi_1 \in (z, z+h),$$

y

$$f(z-h) = f(z) - hf'(z) + \frac{h^2}{2}f''(z) - \frac{h^3}{6}f'''(z) + \frac{h^4}{24}f^{(4)}(\xi_2), \quad \xi_2 \in (z-h, z).$$

## Derivadas de orden superior

Con desarrollos de Taylor también podemos encontrar aproximaciones de derivadas de orden superior. Así es,

$$f(z+h) = f(z) + hf'(z) + \frac{h^2}{2}f''(z) + \frac{h^3}{6}f'''(z) + \frac{h^4}{24}f^{(4)}(\xi_1), \quad \xi_1 \in (z, z+h),$$

y

$$f(z-h) = f(z) - hf'(z) + \frac{h^2}{2}f''(z) - \frac{h^3}{6}f'''(z) + \frac{h^4}{24}f^{(4)}(\xi_2), \quad \xi_2 \in (z-h, z).$$

Sumando ambas expresiones y reorganizando

$$f''(z) = \frac{f(z+h) - 2f(z) + f(z-h)}{h^2} - \frac{h^2}{12}f^{(4)}(\xi), \quad \text{con } \xi \in (z-h, z+h).$$

## Derivadas de orden superior

Con desarrollos de Taylor también podemos encontrar aproximaciones de derivadas de orden superior. Así es,

$$f(z+h) = f(z) + hf'(z) + \frac{h^2}{2}f''(z) + \frac{h^3}{6}f'''(z) + \frac{h^4}{24}f^{(4)}(\xi_1), \quad \xi_1 \in (z, z+h),$$

y

$$f(z-h) = f(z) - hf'(z) + \frac{h^2}{2}f''(z) - \frac{h^3}{6}f'''(z) + \frac{h^4}{24}f^{(4)}(\xi_2), \quad \xi_2 \in (z-h, z).$$

Sumando ambas expresiones y reorganizando

$$f''(z) = \frac{f(z+h) - 2f(z) + f(z-h)}{h^2} - \frac{h^2}{12}f^{(4)}(\xi), \quad \text{con } \xi \in (z-h, z+h).$$

Aproximación de orden 2 para la segunda derivada

## Otras aproximaciones

No es necesario que los puntos donde evaluemos la función tengan que estar equiespaciados.

## Otras aproximaciones

No es necesario que los puntos donde evaluemos la función tengan que estar equiespaciados.

Por ejemplo, con

$$f(z + 2h) = f(z) + 2hf'(z) + 2h^2f''(\xi_1), \quad \xi_1 \in (z, z + 2h),$$

y

$$f(z - h) = f(z) - hf'(z) + \frac{h^2}{2}f''(\xi_2), \quad \xi_2 \in (z - h, z),$$

podemos obtener

$$f'(z) = \frac{f(z + 2h) - f(z - h)}{3h} - \frac{10}{12}hf''(\xi), \quad \text{con } \xi \in (z - h, z + 2h).$$

No es necesario que los puntos donde evaluemos la función tengan que estar equiespaciados.

Por ejemplo, con

$$f(z + 2h) = f(z) + 2hf'(z) + 2h^2f''(\xi_1), \quad \xi_1 \in (z, z + 2h),$$

y

$$f(z - h) = f(z) - hf'(z) + \frac{h^2}{2}f''(\xi_2), \quad \xi_2 \in (z - h, z),$$

podemos obtener

$$f'(z) = \frac{f(z + 2h) - f(z - h)}{3h} - \frac{10}{12}hf''(\xi), \quad \text{con } \xi \in (z - h, z + 2h).$$

# Aproximación por interpolación

Dada  $f : [a, b] \rightarrow \mathbb{R}$  y los puntos  $\{x_0, \dots, x_n\} \subset [a, b]$ , si  $P \in \mathcal{P}_n$  es el **polinomio interpolador**, entonces podemos hacer la aproximación

$$f'(z) \approx P'(z).$$

# Aproximación por interpolación

Dada  $f : [a, b] \rightarrow \mathbb{R}$  y los puntos  $\{x_0, \dots, x_n\} \subset [a, b]$ , si  $P \in \mathcal{P}_n$  es el polinomio interpolador, entonces podemos hacer la aproximación

$$f'(z) \approx P'(z).$$

Como

$$f(z) \approx P(z) = \sum_{i=1}^n f(x_i) L_i(z),$$

entonces obtenemos

$$f'(z) \approx P'(z) = \sum_{i=1}^n f(x_i) L'_i(z).$$

Es decir, aproximamos la derivada con una combinación lineal de las imágenes de  $f$ .

# Aproximación por interpolación

Dada  $f : [a, b] \rightarrow \mathbb{R}$  y los puntos  $\{x_0, \dots, x_n\} \subset [a, b]$ , si  $P \in \mathcal{P}_n$  es el polinomio interpolador, entonces podemos hacer la aproximación

$$f'(z) \approx P'(z).$$

Como

$$f(z) \approx P(z) = \sum_{i=1}^n f(x_i) L_i(z),$$

entonces obtenemos

$$f'(z) \approx P'(z) = \sum_{i=1}^n f(x_i) L'_i(z).$$

Es decir, aproximamos la derivada con una **combinación lineal** de las imágenes de  $f$ .

# Fórmulas numéricas para la derivada

- Dos puntos ( $n = 1$ ):

# Fórmulas numéricas para la derivada

- Dos puntos ( $n = 1$ ): El polinomio interpolador es la recta

$$P(x) = f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0)$$

# Fórmulas numéricas para la derivada

- Dos puntos ( $n = 1$ ): El polinomio interpolador es la recta

$$P(x) = f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0)$$

Si tomamos  $x_0 = z$  y  $x_1 = z + h$  nos queda

$$P'(x) = \frac{f(z + h) - f(z)}{h},$$

# Fórmulas numéricas para la derivada

- Dos puntos ( $n = 1$ ): El polinomio interpolador es la recta

$$P(x) = f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0)$$

Si tomamos  $x_0 = z$  y  $x_1 = z + h$  nos queda

$$P'(x) = \frac{f(z+h) - f(z)}{h},$$

por lo que evaluando en  $x = z$  obtenemos

$$f'(z) \approx P'(z) = \frac{f(z+h) - f(z)}{h}.$$

# Fórmulas numéricas para la derivada

- Dos puntos ( $n = 1$ ): El polinomio interpolador es la recta

$$P(x) = f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0)$$

Si tomamos  $x_0 = z$  y  $x_1 = z + h$  nos queda

$$P'(x) = \frac{f(z+h) - f(z)}{h},$$

por lo que evaluando en  $x = z$  obtenemos

$$f'(z) \approx P'(z) = \frac{f(z+h) - f(z)}{h}.$$

Recuperamos la aproximación de **primer orden**

# Fórmulas numéricas para la derivada

- Tres puntos ( $n = 2$ ):

# Fórmulas numéricas para la derivada

- Tres puntos ( $n = 2$ ): El polinomio interpolador es la parábola

$$P(x) = f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0) + \left( \frac{f(x_1) - f(x_0)}{(x_1 - x_0)(x_0 - x_2)} - \frac{f(x_2) - f(x_1)}{(x_2 - x_1)(x_0 - x_2)} \right) (x - x_0)(x - x_1)$$

# Fórmulas numéricas para la derivada

- Tres puntos ( $n = 2$ ): El polinomio interpolador es la parábola

$$P(x) = f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0) + \left( \frac{f(x_1) - f(x_0)}{(x_1 - x_0)(x_0 - x_2)} - \frac{f(x_2) - f(x_1)}{(x_2 - x_1)(x_0 - x_2)} \right) (x - x_0)(x - x_1)$$

Si tomamos  $x_0 = z$ ,  $x_1 = z + h$  y  $x_2 = z - h$  nos queda

$$P'(x) = \frac{f(z+h) - f(z)}{h} + \frac{f(z+h) - 2f(z) + f(z-h)}{2h^2}(2x - x_0 - x_1)$$

# Fórmulas numéricas para la derivada

- Tres puntos ( $n = 2$ ): El polinomio interpolador es la parábola

$$P(x) = f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0) + \left( \frac{f(x_1) - f(x_0)}{(x_1 - x_0)(x_0 - x_2)} - \frac{f(x_2) - f(x_1)}{(x_2 - x_1)(x_0 - x_2)} \right) (x - x_0)(x - x_1)$$

Si tomamos  $x_0 = z$ ,  $x_1 = z + h$  y  $x_2 = z - h$  nos queda

$$P'(x) = \frac{f(z+h) - f(z)}{h} + \frac{f(z+h) - 2f(z) + f(z-h)}{2h^2}(2x - x_0 - x_1),$$

por lo que evaluando en  $x = z$  obtenemos

$$f'(z) \approx P'(z) = \frac{f(z+h) - f(z-h)}{2h}.$$

# Fórmulas numéricas para la derivada

- Tres puntos ( $n = 2$ ): El polinomio interpolador es la parábola

$$P(x) = f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0) + \left( \frac{f(x_1) - f(x_0)}{(x_1 - x_0)(x_0 - x_2)} - \frac{f(x_2) - f(x_1)}{(x_2 - x_1)(x_0 - x_2)} \right) (x - x_0)(x - x_1)$$

Si tomamos  $x_0 = z$ ,  $x_1 = z + h$  y  $x_2 = z - h$  nos queda

$$P'(x) = \frac{f(z+h) - f(z)}{h} + \frac{f(z+h) - 2f(z) + f(z-h)}{2h^2}(2x - x_0 - x_1),$$

por lo que evaluando en  $x = z$  obtenemos

$$f'(z) \approx P'(z) = \frac{f(z+h) - f(z-h)}{2h}.$$

Recuperamos la aproximación de **segundo orden**

## Teorema

Si  $f \in C^{n+2}([a, b])$ , entonces existen  $\xi_1, \xi_2 \in (a, b)$  tal que

$$f'(z) - P'(z) = \frac{f^{(n+1)}(\xi_1)}{(n+1)!} \left( \prod_{i=0}^n (z - x_i) \right)' + \frac{f^{(n+2)}(\xi_2)}{(n+2)!} \prod_{i=0}^n (z - x_i).$$

## Teorema

Si  $f \in C^{n+2}([a, b])$ , entonces existen  $\xi_1, \xi_2 \in (a, b)$  tal que

$$f'(z) - P'(z) = \frac{f^{(n+1)}(\xi_1)}{(n+1)!} \left( \prod_{i=0}^n (z - x_i) \right)' + \frac{f^{(n+2)}(\xi_2)}{(n+2)!} \prod_{i=0}^n (z - x_i).$$

En particular, si  $z = x_j$  se tiene

$$f'(z) - P'(z) = \frac{f^{(n+1)}(\xi_1)}{(n+1)!} \prod_{\substack{i=0 \\ i \neq j}}^n (x_j - x_i).$$

## Teorema

Si  $f \in C^{n+2}([a, b])$ , entonces existen  $\xi_1, \xi_2 \in (a, b)$  tal que

$$f'(z) - P'(z) = \frac{f^{(n+1)}(\xi_1)}{(n+1)!} \left( \prod_{i=0}^n (z - x_i) \right)' + \frac{f^{(n+2)}(\xi_2)}{(n+2)!} \prod_{i=0}^n (z - x_i).$$

En particular, si  $z = x_j$  se tiene

$$f'(z) - P'(z) = \frac{f^{(n+1)}(\xi_1)}{(n+1)!} \prod_{\substack{i=0 \\ i \neq j}}^n (x_j - x_i).$$

Cuando los puntos están **equiespaciados**,  $x_i = x_0 + hi$  se obtiene una aproximación de **orden  $n$  en  $h$**  ya que

$$\prod_{\substack{i=0 \\ i \neq j}}^n (x_j - x_i) = h^n \prod_{\substack{i=0 \\ i \neq j}}^n (j - i).$$

# Integración Numérica

## Teorema [Valor Medio Integral Generalizado]

Sea  $f \in C([a, b])$  y  $g : [a, b] \rightarrow (0, +\infty)$ . Entonces, existe  $\theta \in [a, b]$  tal que

$$\int_a^b f(x) g(x) dx = f(\theta) \int_a^b g(x) dx.$$

## Teorema [Valor Medio Integral Generalizado]

Sea  $f \in C([a, b])$  y  $g : [a, b] \rightarrow (0, +\infty)$ . Entonces, existe  $\theta \in [a, b]$  tal que

$$\int_a^b f(x) g(x) dx = f(\theta) \int_a^b g(x) dx.$$

## Demostración

Como  $f$  es continua, entonces  $f([a, b]) = [m, M]$ .

## Teorema [Valor Medio Integral Generalizado]

Sea  $f \in C([a, b])$  y  $g : [a, b] \rightarrow (0, +\infty)$ . Entonces, existe  $\theta \in [a, b]$  tal que

$$\int_a^b f(x) g(x) dx = f(\theta) \int_a^b g(x) dx.$$

### Demostración

Como  $f$  es continua, entonces  $f([a, b]) = [m, M]$ .

Además, como  $g$  no cambia de signo es fácil de comprobar que

$$\frac{\int_a^b f(x) g(x) dx}{\int_a^b g(x) dx} \in [m, M] = \text{Im}(f).$$

## Teorema [Valor Medio Integral Generalizado]

Sea  $f \in C([a, b])$  y  $g : [a, b] \rightarrow (0, +\infty)$ . Entonces, existe  $\theta \in [a, b]$  tal que

$$\int_a^b f(x) g(x) dx = f(\theta) \int_a^b g(x) dx.$$

### Demostración

Como  $f$  es continua, entonces  $f([a, b]) = [m, M]$ .

Además, como  $g$  no cambia de signo es fácil de comprobar que

$$\frac{\int_a^b f(x) g(x) dx}{\int_a^b g(x) dx} \in [m, M] = \text{Im}(f).$$

Por tanto, existe  $\theta \in [a, b]$  tal que

$$f(\theta) = \frac{\int_a^b f(x) g(x) dx}{\int_a^b g(x) dx}.$$

$$\int_a^b f(x) dx \approx \int_a^b p(x) dx \quad \text{con } p \in \mathcal{P}_n$$

$$\int_a^b f(x) dx \approx \int_a^b p(x) dx \quad \text{con } p \in \mathcal{P}_n$$

- Fórmula del trapecio cerrada ( $n = 1$ , dos puntos)

$$\int_a^b f(x) dx = \frac{f(a) + f(b)}{2} (b-a) - \frac{(b-a)^3}{12} f''(\xi), \quad \xi \in [a, b]$$

- Fórmula del trapecio abierta ( $n = 1$ , dos puntos)

$$\int_a^b f(x) dx = \frac{f(a + \frac{b-a}{3}) + f(a + 2\frac{b-a}{3})}{2} (b-a) + \frac{(b-a)^3}{36} f''(\xi), \quad \xi \in [a, b]$$

# Formulas de Newton-Cotes

- Fórmula del Simpson cerrada ( $n = 2$ , tres puntos)

$$\int_a^b f(x) dx = \frac{f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)}{6} (b-a) - \frac{(b-a)^5}{2880} f^{(iv)}(\xi), \quad \xi \in [a, b]$$

- Fórmula del Simpson abierta ( $n = 2$ , tres puntos)

$$\int_a^b f(x) dx = \frac{2f\left(a + \frac{b-a}{4}\right) - f\left(a + \frac{b-a}{2}\right) + 2f\left(a + 3\frac{b-a}{4}\right)}{3} (b-a) - 7\frac{(b-a)^5}{23040} f^{(iv)}(\xi), \quad \xi \in [a, b]$$

# Formulas de Newton-Cotes

- Fórmula del Simpson cerrada ( $n = 2$ , tres puntos)

$$\int_a^b f(x) dx = \frac{f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)}{6} (b-a) - \frac{(b-a)^5}{2880} f^{(iv)}(\xi), \quad \xi \in [a, b]$$

- Fórmula del Simpson abierta ( $n = 2$ , tres puntos)

$$\int_a^b f(x) dx = \frac{2f\left(a + \frac{b-a}{4}\right) - f\left(a + \frac{b-a}{2}\right) + 2f\left(a + 3\frac{b-a}{4}\right)}{3} (b-a) - 7\frac{(b-a)^5}{23040} f^{(iv)}(\xi), \quad \xi \in [a, b]$$

- Fórmula del Simpson 3/8 ( $n = 3$ , cuatro puntos)

$$\int_a^b f(x) dx = \frac{f(a) + 3f\left(a + \frac{b-a}{3}\right) + 3f\left(a + 2\frac{b-a}{3}\right) + f(b)}{8} (b-a) - \frac{(b-a)^5}{6480} f^{(iv)}(\xi), \quad \xi \in [a, b]$$

# Error de la aproximación

## Teorema

Sea  $f \in C^{n+1}([a, b])$  y  $\{z_0, \dots, z_n\} \in [0, 1]$ . Si  $P \in \mathcal{P}_n$  es el polinomio interpolador de  $f$  en los puntos  $\{x_i = a + (b - a)z_i\}$ , entonces existe  $\theta \in [a, b]$  tal que

$$\int_a^b f(x) dx = \int_a^b P(x) dx + \frac{C (b - a)^{n+2}}{(n + 1)!} f^{(n+1)}(\theta).$$

## Teorema

Sea  $f \in C^{n+1}([a, b])$  y  $\{z_0, \dots, z_n\} \in [0, 1]$ . Si  $P \in \mathcal{P}_n$  es el polinomio interpolador de  $f$  en los puntos  $\{x_i = a + (b - a)z_i\}$ , entonces existe  $\theta \in [a, b]$  tal que

$$\int_a^b f(x) dx = \int_a^b P(x) dx + \frac{C (b - a)^{n+2}}{(n + 1)!} f^{(n+1)}(\theta).$$

Tomando los puntos equiespaciados  $z_i = i/n$  (Newton-Cotes)

- Si  $n$  es impar:

$$\int_a^b f(x) dx = \int_a^b P(x) dx + \frac{C (b - a)^{n+2}}{(n + 1)!} f^{(n+1)}(\theta)$$

- Si  $n$  es par:

$$\int_a^b f(x) dx = \int_a^b P(x) dx + \frac{C (b - a)^{n+3}}{(n + 2)!} f^{(n+2)}(\theta)$$