ALGORÍTMICA NUMÉRICA

TEMARIO

T1. COMA FLOTANTE Y ERRORES
T2. INTERPOLACIÓN
T3. AJUSTE DE DATOS
T4. RESOLUCIÓN DE ECS. NO LINEALES.

T1. COMA FLOTANTE Y ERRORES

Tenemos dos escenarios: Las matemáticas en el mundo real y lo numérico en el mundo digital

NÚMEROS REALES	NÚMEROS MÁQUINA
número real → ∞	nº máq + redondeo → ∄∞
· Nº reales	· Nº máquina SRN
· Valores exactos	· Valores aproximados. Errores
· ∞ nºs	· Cdad finita de nums
· ∞ decimales	· Cdad finita de decimales
· ∞ operaciones: series...	· Cdad finita de operaciones
· Operaciones exactas *, /...	· Operaciones aproximadas
· Funciones reales. Gráficas	· Tablas / matrices. Gráfica
	· Código

Nºs MÁQUINA

Como los números reales son ∞ y el ordenador sólo puede manejar una cantidad finita de números. Se utilizan nºs máquina

$\boxed{x = \text{nº real}}$ $\boxed{\hat{x} = \text{nº máquina}}$

- Todos los números reales se aproximan a números máquina
  - Si un número real coincide con un número máquina. = situación óptima (podemos almacenar el nº en memoria y operar con él).

$\boxed{\text{Nº real} = \text{Nº máquina}}$ $\boxed{x = \tilde{x}}$

  - Si un número real no coincide con ningún nº máquina, ese nº real estará entre dos nº máquina consecutivos, lo aproximamos / identificamos por el nº máquina más próximo

$\boxed{\text{Nº real} \simeq \text{Nº máquina}}$ $\boxed{x \simeq \tilde{x}}$

  - Cuando no coinciden se utilizarán errores de redondeo

$\boxed{\text{Error redondeo} = |\text{nº real} - \text{nº máquina}|}$

REPRESENTACIÓN Nºs MÁQUINA

- La representación de números máquina es el conjunto de números y las reglas de uso.
- Los nºs máquina son un conjunto finito de nºs por lo que
  - Tienen un valor mínimo positivo (Vmin) y un valor máximo positivo (Vmax)
  - Existe el siguiente a un nº máquina
  - Existe una distancia entre dos nº consecutivos

- Cada nº real está entre dos nºs máquina y uno de ellos será su representación
- Se puede operar con dos nºs máquina y el resultado será otro nº máquina

El ordenador maneja un subconjunto de los números (Nºs MÁQUINA)

¿QUÉ nºs son los afortunados? → DEPENDE de REPRESENTACIÓN USADA

- REPRESENTACIONES COMA FIJA
- REPRESENTACIONES COMA FLOTANTE

En el resto de casos hay errores de partida y para cuantificar todo esto usaremos el error entre lo obtenido y lo deseado

- ABSOLUTO: Asociado a decimales correctos
- RELATIVO: Asociado a cifras correctas

ERROR ABSOLUTO

El error absoluto conserva las mismas unidades que x.

Se basa en la diferencia decimal

$$\boxed{E_{abs} = |\hat{x} - x|}$$

$x = 3,14159265359$

$\tilde{x} = 3,14159265350$

$E_{abs} = 0,00000000009$

Nº de decimales correctos: $\boxed{-\log 10 \, (\text{Error absoluto})}$

$-\log_{10} 0,00000000009 \approx 10$

10^{-10}

ERROR RELATIVO

El error relativo es adimensional.

Está asociado a las cifras correctas que hay entre los dos nums

$$\boxed{E_{rel} = \frac{|\hat{x} - x|}{|\hat{x}|}}$$

$$\frac{0{,}00000000009}{3{,}14159265350} = 2{,}8647 \cdot 10^{-11}$$

CIFRAS SIGNIFICATIVAS

Se pueden medir contando desde la izquierda, a partir del primer nº distinto de 0, las cifras iguales entre las dos cifras

$$\boxed{E_{rel} \approx 10^{-(\text{cifras significativas})}} \qquad \boxed{\text{cifras.sig} = -\log_{10}(E_{rel})}$$

$$-\log_{10}(2{,}8647 \cdot 10^{-11}) = 10{,}5429\ldots \approx 10$$

REPRESENTACIÓN N°S MÁQUINA

RANGO DE VALORES

La mitad de los valores son positivos y la otra mitad negativos

- $V_{max}, -V_{max}$: Valor máximo en el eje positivo y su opuesto
- $V_{min}, -V_{min}$: Valor mínimo en el eje positivo y su opuesto.

DISTANCIA

- EPS: Distancia entre un nº máquina y su consecutivo.

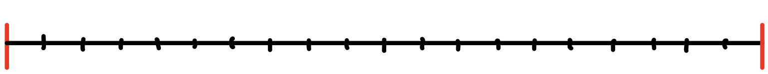
Los nºs máquina NO son equidistantes

COMA FIJA

Equivale a usar enteros y acordar el valor del dígito menos significativo

Todos los nº tienen la coma en el mismo lugar

PARÁMETROS: • Tipo de datos a usar • Rango a cubrir • Espaciado



$$\boxed{\#\,\text{Números} = \frac{\text{Rango}}{\text{Salto}}}$$

ERROR MÁXIMO

$$\boxed{E_{abs} \leq 1/2 \cdot \text{salto}}$$

El error absoluto es constante pero el error relativo NO lo es

COMA FLOTANTE

Un nº en coma flotante combina una representación en coma fija y una información sobre la escala.

La **MANTISA** cubre un cierto intervalo y el **EXPONENTE** indica qué escala utilizar → ((cuantos más bits de mantisa, mayor precisión y cuanto mayor exponente mayor escala)

Se impone un intervalo para la mantisa. El más común es $[1,2)$.

La base que se utiliza es 2.

$$\boxed{\hat{x} = s \cdot m \cdot B^e}$$

- s: Signo
- m: mantisa
- B: Base
- e: Exponente

Los nº máquina vienen dados por la siguiente expresión $\boxed{\hat{x} = \pm m \cdot 2^e}$

lo que vendría a verse representado como

$$\boxed{\hat{x} = (-1)^s \cdot (d_1 . d_2 d_3 d_4 \ldots d_p)_2 \cdot 2^{(e_1 e_2 e_3 e_4 \ldots e_q)_2}}$$

CONVERSIÓN DECIMAL → BINARIO

- **PARTE ENTERA**

Ejemplo:

14 | 2
0   7 | 2
1   3 | 2
1   1

$= 14 = 1110_2$

$0 \cdot 2^0 + 1 \cdot 2^1 +$

$1 \cdot 2^2 + 1 \cdot 2^3 = 14$

- **PARTE DECIMAL**

Ejemplo:

$0,1 \cdot 2 = 0,2 < 1$

$0,2 \cdot 2 = 0,4 < 1$

$0,4 \cdot 2 = 0,8 < 1$

$0,8 \cdot 2 = 1,6 > 1$

$0,6 \cdot 2 = 1,2 > 1$

$0,2 \cdot 2 = 0,4 < 1$

$x = 0,1 = (0,0001100110011)$

SALTO

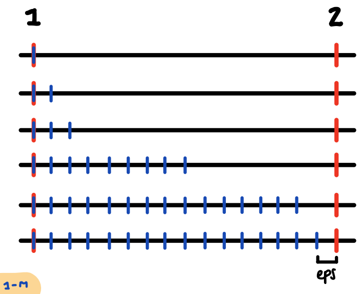
El salto en coma flotante se obtiene con el bit menos representativo

$$\boxed{eps = 2^{-(M-1)} = 2^{1-M}}$$

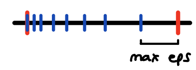
$1.1111 = 1.9375$
$- \; 1.1110 = 1.875$
$0.0001 = 0{,}0625 = X.XXXX$ ←

0 1 2 3 4 → 2^{-4}

bit 0	bit 1	bit 2	bit 3				bit M-1
1.	0	0	0	...	0	0	0
1.	0	0	0	...	0	0	1
1.	0	0	0	...	0	1	0
1.	...	...	...	...	...	...	...
1.	1	1	1	...	1	1	0
1.	1	1	1	...	1	1	1
2^0	2^{-1}	2^{-2}	2^{-3}				$2^{-(M-1)} \to 2^{1-M}$



Máximo espaciado: $\boxed{Max.eps = eps \cdot 2^{max\_exponente}}$



Cifras binarias y decimales: $\boxed{1 \text{ cifra decimal} = 3{,}32 \text{ cifras binarias} = \log_2 10}$

ERRORES

Error relativo

$$\boxed{E_{rel} = \frac{E_{abs}}{|x|} \leq \frac{2^{-M}}{|x|} \leq 2^{-M} \rightarrow E_{rel} \leq 2^{-M}}$$

$|x| \geq 1$

Error absoluto

$$\boxed{E_{abs} \leq 1/2 \cdot salto = 1/2 \cdot 2^{1-M} = 2^{-M}} \quad \boxed{E_{abs} \leq 2^{-M}}$$

AMPLIAR RANGO

Para ampliar el rango del intervalo se aumenta o disminuye el exponente

rango $[1, 2)$ → exponente $= 0$

rango $[2, 4)$ → exponente $= 1$

rango $[0'5, 1)$ → exponente $= -1$

RESUMEN

- La coma flotante mantiene constante el error relativo.
- Ese error relativo o precisión (eps) depende del nº de dígitos empleados en la mantisa
- El rango Vmin, Vmax depende de los bits usados en el EXP
- Cada intervalo tiene el mismo nº de nº máquina. Solo varía el eps y el Eabs. El Erel y el nº de cifras de precisión es constante.
- El exponente determina el intervalo de los nº máquina

NÚMEROS DENORMALIZADOS

- Al tener una representación del tipo $\hat{x} = m \cdot 2^e$, $m \in [1,2)$, no podemos llegar al 0. Por esto se necesitan nums denormalizados
- Su mantisa empieza por 0 y el exponente se queda en el último exponente válido.
- Con esto se puede obtener el 0 y valores muy cercanos a él.

IEEE STANDARD FOR BINARY FLOATING-POINT ARITHMETIC

Define varios formatos de coma flotante

1. Precisión simple (4 bytes = 32 bits) (single, float)
2. Precisión doble (8 bytes = 64 bits) (double)
3. Precisión extendida (80 bits)

El standard especifica entre otras cosas:

- Reparto de los bits entre: signo / mantisa / exponente
- Precisión exigida en operaciones elementales
- Nums especiales (NaN, ∞) y operaciones entre ellos, manejo de excepciones (dividir por 0, etc).

SIMPLE PRECISIÓN

BITS

- **Palabra**: 4 bytes (32 bits)

  ↳ 8 bits exponente + 23 bits mantisa + 1 bit signo

TIPOS

- **Nº NORMALIZADOS**: $\hat{x} = (1.d_1d_2d_3\ldots d_{23}) \cdot 2^{(e_1e_2\ldots e_8)_2 - 127}$
- **Nº DENORMALIZADOS**: $\hat{x} = (0.d_1d_2d_3\ldots d_{23}) \cdot 2^{-126}$ si $e = (00000000)_2$
- **Nº NO NUMÉRICOS**: ∞, NaN si $e = (11111111)_2$

RANGO

- **Nº MAX NORMALIZADO**: $\hat{x}_{max} = (1.1\ldots1) \cdot 2^{(1\ldots1)-127} \cong 10^{38}$
- **Nº MIN NORMALIZADO**: $\hat{x}_{min} = (1.0\ldots0) \cdot 2^{(0\ldots0)-127} \cong 10^{-38}$
- **Nº MAX DENORMALIZADO**: $\hat{x}^d_{max} = (0.1\ldots1) \cdot 2^{-126} \cong 10^{-38}$
- **Nº MIN DENORMALIZADO**: $\hat{x}^d_0 = (0.0\ldots0) \cdot 2^{-126} \cong 0$

  $\hat{x}^d_{min} = (0.0\ldots1) \cdot 2^{-126} \cong 10^{-45}$

- $[V^d_{min}, V^d_{max}] = [10^{-45}, 10^{-38}]$
- $[V_{min}, V_{max}] = [10^{-38}, 10^{38}]$

DOBLE PRECISIÓN

BITS

- **Palabra**: 8 bytes (64 bits)

  ↳ 11 bits exponente + 52 bits mantisa + 1 bit signo

TIPOS

- **Nº NORMALIZADOS:** $\hat{x} = (1.d_1 d_2 d_3 \dots d_{52}) \cdot 2^{(e_1 e_2 \dots e_{11})_2 - 1023}$
- **Nº DENORMALIZADOS:** $\hat{x} = (0.d_1 d_2 d_3 \dots d_{52}) \cdot 2^{-1022}$ si $e = (00000000000)_2$
- **Nº NO NUMÉRICOS:** ∞, NaN si $e = (11111111111)_2$

RANGO

- **Nº MAX NORMALIZADO:** $\hat{x}_{max} = (1.1\dots1) \cdot 2^{(1\dots1)-1023} \cong 10^{308}$
- **Nº MIN NORMALIZADO:** $\hat{x}_{min} = (1.0\dots0) \cdot 2^{(0\dots0)-1023} \cong 10^{-307}$
- **Nº MAX DENORMALIZADO:** $\hat{x}^d_{max} = (0.1\dots1) \cdot 2^{-1022} \cong 10^{-307}$
- **Nº MIN DENORMALIZADO:** $\hat{x}^d_0 = (0.0\dots0) \cdot 2^{-1022} \cong 0$

  $\hat{x}^d_{min} = (0.0\dots1) \cdot 2^{-1022} \cong 10^{-323}$

- $[V^d_{min}, V^d_{max}] = [10^{-323}, 10^{-307}]$
- $[V_{min}, V_{max}] = [10^{-307}, 10^{308}]$

T2. INTERPOLACIÓN

INTRODUCCIÓN

La interpolación se basa en conseguir una función que cumpla una serie de condiciones

ENTRADA

x_i	0	0.25	0.5	0.75
f_i	1	-1	2	0

+

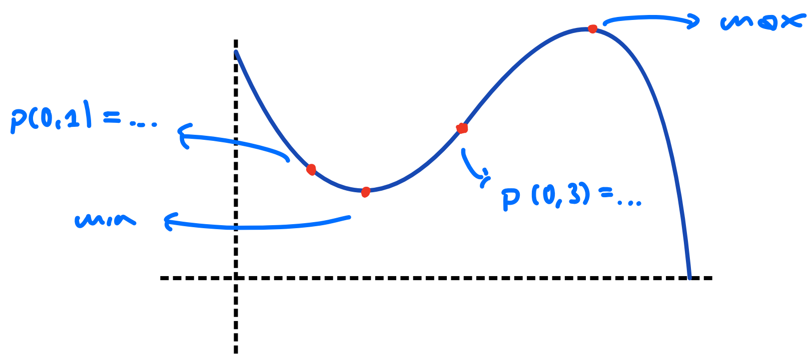
CALCULAR EL POLINOMIO MÁS SENCILLO

→

SALIDA

$$p(x) = 1 - 31.33x + 120x^2 - 106,66x^3$$

APLICACIONES



EJEMPLO:

Dados los datos y valores

x_i	0	0.25	0.5	0.75
f_i	1	-1	2	0

construir el polinomio más sencillo que pasa por los puntos ⟶ $p(x) = a_0 + a_1x + a_2x^2 + a_3x^3$ y evaluar el polinomio en $x = 0,3$

SOLUCIÓN

$$p(x_i) = f_i \rightarrow \begin{cases} p(0) = 1 \\ p(0{,}25) = -1 \\ p(0{,}5) = 2 \\ p(0{,}75) = 0 \end{cases} \rightarrow \begin{cases} a_0 + a_1 0 + a_2 0^2 + a_3 0^3 = 1 \\ a_0 + a_1 0{,}25 + a_2 0{,}25^2 + a_3 0{,}25^3 = -1 \\ a_0 + a_1 0{,}5 + a_2 0{,}5^2 + a_3 0{,}5^3 = 2 \\ a_0 + a_1 0{,}75 + a_2 0{,}75^2 + a_3 0{,}75^3 = 0 \end{cases} \rightarrow$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0{,}25 & 0{,}25^2 & 0{,}25^3 \\ 1 & 0{,}5 & 0{,}5^2 & 0{,}5^3 \\ 1 & 0{,}75 & 0{,}75^2 & 0{,}75^3 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \\ 2 \\ 0 \end{pmatrix} \rightarrow \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} 1 \\ -31{,}33 \\ 120 \\ -106{,}66 \end{pmatrix} \rightarrow$$

$$\boxed{p(x) = 1 - 31{,}33x + 120x^2 + 106{,}66x^3}$$

$$p(0{,}3) = 1 - 31{,}33 \cdot 0{,}3 + 120 \cdot 0{,}3^2 - 106{,}66 \cdot 0{,}3^3 = -0{,}48 \rightarrow p(0{,}3) = -0{,}48$$

En todos los problemas hay que:

- Identificar elementos

1. Tipo función interpoladora $u(x)$: la pueden dar de las siguientes maneras:

   - Forma descriptiva:

     EJ: Polinomio de grado 4

   - Forma general:

     EJ: $u(x) = a + bx + ce^{-x} + d\cos(x)$

   - Dando espacio interpolador

     EJ: $\langle 1, x, x^2, x^3, x^4 \rangle \quad \langle 1, x, e^{-x}, \cos(x) \rangle$

2. Condiciones a cumplir

- Directamente los datos (tabla):

Ej:

x	0	1	π
u(x)	1	0	-1

- Indirectamente (Función a interpolar)

Ej: interpolar $f(x) = \text{sen}(x)$ en los puntos $0, 1, \pi$

x	0	1	π
f(x)	0	0,8	0

← $f(x) = \text{sen}(x)$

- Verificar $\boxed{\text{nº condiciones} = \text{nº coeficientes}}$

→ coef = grados libertad

→ cond = restricciones

- Escribir forma más general de la función:

$$\boxed{u(x) = A \cdot b_1(x) + B \cdot b_2(x) + C \cdot b_3(x) + \dots}$$

- Imponer condiciones en u(x) → Obtener ecuaciones

- Construir el sistema lineal $Hc = b$ (cuadrado)

- Resolver el sistema lineal

- Obtener la función interpoladora u(x):

↳ Gráfica / Evaluarla / Derivada...

INTERPOLACIÓN POLINOMIAL CLÁSICA

Se basa en conseguir un polinomio grado $n-1$ (n coeficientes) que pase por los n puntos de una tabla

RESOLUCIÓN

1. Construir matriz H y vector V

$$\text{nodos}\left\{ \underbrace{\begin{pmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ \vdots & \vdots & \vdots & \vdots & \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{pmatrix}}_{\text{funciones}} \cdot \underbrace{\begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix}}_{\text{coef}} = \underbrace{\begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{pmatrix}}_{\text{valores}} \right.$$

2. Resolver sistema $\boxed{Hc = V}$

3. El vector c son los coeficientes del polinomio

4. La solución será

$$\boxed{p(x) = c_0 + c_1 x + c_2 x^2 + \dots + c_n x^n}$$

EJ:

x_i	0	1	-1
y_i	1	2	3

$p(x) = a + bx + cx^2$

$$\left\{\begin{matrix} 0 \\ 1 \\ -1 \end{matrix}\right\} \underbrace{\begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & -1 & 1 \end{pmatrix}}_{\{1,\, x,\, x^2\}} \cdot \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$$

Resolvemos y obtenemos = $c = \begin{pmatrix} 1 \\ -1/2 \\ 3/2 \end{pmatrix}$

$$\boxed{p(x) = 1 - 1/2\, x + 3/2\, x^2}$$

GENERALIZACIÓN

En lugar de usar un polinomio, se pueden usar funciones de otros tipos.

EJ: Interpolar con $p(x) = C_0 + C_1 \cos(2\pi x) + C_2 \operatorname{sen}(2\pi x) + C_3 \cos(4\pi x)$

x_i	0	1/4	1/2	3/4
y_i	1	-1	2	0

$$\begin{array}{c} \\ 0 \\ 1/4 \\ 1/2 \\ 3/4 \end{array} \begin{array}{c} \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & -1 \\ 1 & -1 & 0 & \cos 2\pi \\ 1 & \cos\frac{3\pi}{2} & \operatorname{sen}\frac{3\pi}{2} & \cos 3\pi \end{pmatrix} \\ \{\, 1 \quad \cos 2\pi x \quad \operatorname{sen} 2\pi x \quad \cos 4\pi x \,\} \end{array} \begin{pmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \\ 2 \\ 0 \end{pmatrix}$$

Resolvemos y obtenemos = $c = \begin{pmatrix} 1/2 \\ -1/2 \\ -1/2 \\ 1 \end{pmatrix}$

$$\boxed{p(x) = 1/2 - 1/2 \cos 2\pi x - 1/2 \operatorname{sen} 2\pi x + \cos 4\pi x}$$

INTERPOLACIÓN DE TAYLOR

Sirve para interpolar un sólo punto con sus derivadas

Es parecido al anterior

Dada la función interpoladora de Taylor, se generan sus derivadas y se hace la matriz con el valor del punto en ellas.

$$\begin{array}{c} u(x) \\ \left\{\begin{array}{c} u(x) \\ u'(x) \\ \vdots \\ u^n(x) \end{array}\right\} \end{array} \begin{pmatrix} \cdot & \cdot & \cdot \end{pmatrix} \cdot \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} f(x_0) \\ f'(x_0) \\ \vdots \\ f^n(x_0) \end{pmatrix}$$

EJ: Hallar $u(x) = C_0 + C_1 e^{x} + C_2 \cdot e^{-x}$ función interpoladora de Taylor en $x_0 = 0$ de $f(x) = \operatorname{sen}(x)$

$$\begin{Bmatrix} u(x) \\ u'(x) \\ u''(x) \end{Bmatrix} \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & -1 \\ 0 & 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} C_0 \\ C_1 \\ C_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$

Resolvemos y obtenemos = $c = \begin{pmatrix} 0 \\ 1/2 \\ -1/2 \end{pmatrix}$

$$u(x) = \langle 1 \quad e^{x} \quad e^{-x} \rangle$$
$$u'(x) = \langle 0 \quad e^{x} \quad -e^{-x} \rangle$$
$$u''(x) = \langle 0 \quad e^{x} \quad e^{-x} \rangle$$

$$\boxed{u(x) = (e^{x} - e^{-x})/2}$$

MÉTODO DE NEWTON

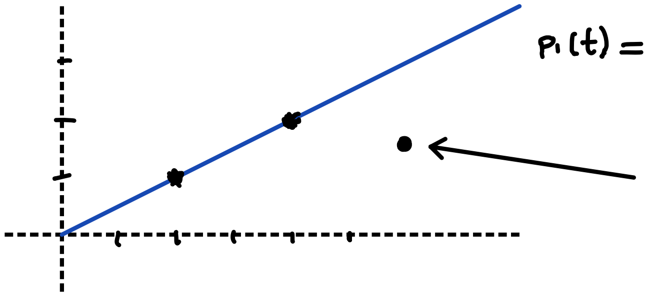
Interpolar con un polinomio una tabla de datos

t_0	t_1	t_2	$\cdots$	t_n
f_0	f_1	f_2	$\cdots$	f_n

Empezamos con el polinomio que pasa por (t_0, f_0), a partir de él construimos el que pasa por (t_0, f_0) y $(t_1; f_1)$, etc

Lo que buscamos con el método de Newton es aprovechar el trabajo que ya tenemos con puntos anteriores para formar un nuevo polinomio que pase por un punto nuevo

Supongamos que nos dan una tabla interpoladora con 2 puntos y hemos construido el polinomio (recta) que pasa por ambos puntos

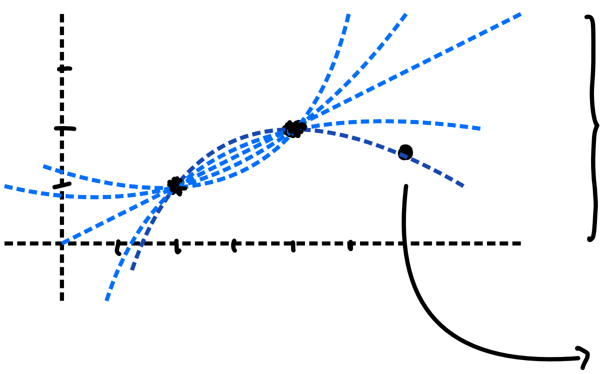


$p_1(t) = a_0 + a_1(t - t_0)$

Si añadimos un tercer punto t_2, y_2 a la tabla, obviamente la recta anterior ya no pasa por el nuevo punto.

Con el método de Newton OBTENEMOS un polinomio que pasa por los 3 puntos y no estropea el trabajo anterior de la forma:

$$\boxed{p_2(t) = p_1(t) + a_2(t - t_0)(t - t_1)}$$



Independientemente del valor de a_2, todos los polinomios resultantes siguen pasando por los dos primeros puntos (t_0, t_1)

Hay un valor de P_2 que hace que el nuevo polinomio pase por el nuevo punto

Obtenemos, por tanto, un polinomio de las siguientes características

$$p(t) = a_0 + a_1(t - t_0) + a_2(t - t_0)(t - t_1) + \dots + a_n(t - t_0)(t - t_1)\dots(t - t_{n-1})$$

No estropea los puntos anteriores $t_0, t_1, \dots, t_{n-1}$.

$a_0 = f[t_0] = f_0$

No estropea t_0

No estropea (t_0, t_1)

Ajusta último punto (t_n, f_n) $\quad a_n = \dots = f[t_0, t_1, \dots, t_n]$

Escojo a_2 para pasar por nuevo punto (t_2, f_2) $\quad a_2 = \dots = f[t_0, t_1, t_2]$

Escojo a_1 para pasar por (t_1, f_1) $\quad a_1 = f[t_0, t_1]$

CÁLCULO DE LOS COEFICIENTES $a(x)$

- Tabla con 1 punto (t_0): $\rightarrow$ $p(t) = a_0$; $p(t_0) = a_0 = f_0$; $a_0 = f_0$

- Tabla con 2° punto (t_1) $\rightarrow$ $p(t) = f_0 + a_1(t_1 - t_0)$

  $p(t_1) = f_0 + a_1(t_1 - t_0) = f_1$

  $a_1 = (f_1 - f_0)/(t_1 - t_0)$

- Tabla con 3° punto (t_2) $\rightarrow$ $p(t) = f_0 + \frac{f_1 - f_0}{t_1 - t_0}(t_1 - t_0) + a_2(t_1 - t_0)(t - t_1)$

  $p(t_2) = f_0 + \frac{f_1 - f_0}{t_1 - t_0}(t_1 - t_0) + a_2(t_1 - t_0)(t - t_1) = f_2$

- $a(k)$ = DIFERENCIAS DIVIDIDAS

  $$a_2 = \frac{(f_2 - f_0) - \left(\frac{f_1 - f_0}{t_1 - t_0}\right)(t_2 - t_0)}{(t_2 - t_0)(t_2 - t_1)}$$

DIFERENCIAS DIVIDIDAS

$$\boxed{f[t_0, t_1] = \frac{f_1 - f_0}{t_1 - t_0}}$$

$$\boxed{f[t_0, t_1, \ldots, t_K] = \frac{f[t_1, t_2, \ldots, t_K] - f[t_0, t_1, \ldots t_{K-1}]}{(t_K - t_0)}}$$

Si la función es derivable, la diferencia dividida de un mismo punto es su derivada

$$\boxed{f[t_K, t_K] = f'(t_K)}$$

Para más derivadas:

$$\boxed{f[t_0, \ldots, t_n] = \frac{f^n(t_n)}{n!}}$$

$t_0 = t_1 = \ldots = t_n$

TRIÁNGULO DIFERENCIAS DIVIDIDAS

t_i	f_i				
t_0	f_0				
		$f[t_0, t_1]$			
t_1	f_1		$f[t_0, t_1, t_2]$		
		$f[t_1, t_2]$		$\dots$	
t_2	f_2		$f[t_1, t_2, t_3]$		
		$f[t_2, t_3]$			$f[t_0, t_1, t_2, \dots, t_n]$
$\dots$	$\dots$		$\dots$	$\dots$	
t_{n-1}	f_{n-1}		$f[t_{n-2}, t_{n-1}, t_n]$		
		$f[t_{n-1}, t_n]$			
t_n	f_n				

MÉTODO NEWTON

Se realiza el triángulo de diferencias divididas y la diagonal superior son los coeficientes del polinomio

Si se quiere interpolar una derivada, se repite ese nodo en la tabla → **INTERPOLACIÓN DE HERMITE**

$$\boxed{p(t) = a_0 + a_1 (t - t_0) + a_2 (t - t_0)(t - t_1) + \cdots}$$

EJ: Polinomio de Newton que interpole:

x_i	0	1	2
f_i	0	0	0
f'_i	1	-1	1

x_i	f_i					
0	0					
		$f[0,0] = f'(0) = \underline{1}$				
0	0		$f[0,0,1] = \underline{-1}$			
		$f[0,1] = 0$		$\underline{0}$		
1	0		$f[0,1,1] = -1$		$\underline{1/2}$	
		$f[1,1] = f'(1) = -1$		1		$\underline{-1/2}$
1	0		$f[1,1,2] = 1$		$-1/2$	
		$f[1,2] = 0$		0		
2	0		$f[1,2,2] = 1$			
		$f[2,2] = f'(2) = 1$				
2	0					

$$p(x) = f[0] + f[0,0](x-0) + f[0,0,1](x-0)^2 + f[0,0,1,1](x-0)^2 \cdot (x-1) +$$

$$+ f[0,0,1,1,2] \cdot (x-0)^2 (x-1)^2 + f[0,0,1,1,2,2] \cdot (x-0)^2 (x-1)^2 (x-2) =$$

$$= \boxed{x - x^2 + 1/2\, x^2 (x-1)^2 - 1/2\, x^2 (x-1)^2 (x-2)}$$

SPLINES

Un spline es una función interpoladora a trozos

Para conseguir un spline se tiene que cumplir la continuidad, por tanto, la derivada por la derecha y por la izquierda deben ser iguales en los extremos de los intervalos

EJ:

$$s(x) = \begin{cases} p(x) & x \in [-1, 0] \\ a + bx & x \in [0, 1] \\ c + d(x+1) + e(x-1)^2 & x \in [1, 2] \end{cases}$$

$s(x)$ es de grado 2 tq $s'(-1) = 0$ e interpola la tabla

x_i	-1	0	1	2
$s(x_i)$	2	5	7	14

1. $p(x) = A + Bx + Cx^2$ en $[-1, 0]$

$p(-1) = 2$; $p(0) = 5$; $p'(-1) = 0$

x_i	f_i		
-1	2		
		0	
-1	2		$\frac{3-0}{0-(-1)} = \underline{3}$
		$\frac{5-2}{0-(-1)} = 3$	
0	5		

$$\boxed{p(x) = 2 + 3(x+1)^2}$$

2. $a + bx$ en $[0,1]$

$s'(0^-) = s'(0^+)$

$6(x+1) = b$

$x=0 \to \underline{6=b}$

$a+bx = a+6x$

$s(0) = a+6\cdot 0 = 5$; $\underline{a=5}$

$a+bx \to \boxed{5+6x}$; $5+6\cdot 1 = \lambda \longrightarrow \boxed{\lambda = 11}$

3. $c + d(x-1) + e(x-1)^2$ en $[1,2]$

$s'(1^-) = s(1^+)$

$6 = d+2e(x-1)$;

$6 = d+2\cdot e(1-1)$;

$\underline{d=6}$

$s(1) = c = \lambda = 11 \longrightarrow \underline{c = 11}$

$s(2) = c+6+e(2-1)^2 = 14$;

$11+6+e = 14$; $\underline{e=-3}$

$$\boxed{11+6(x-1)-3(x-1)^2}$$

Con lo que obtenemos:

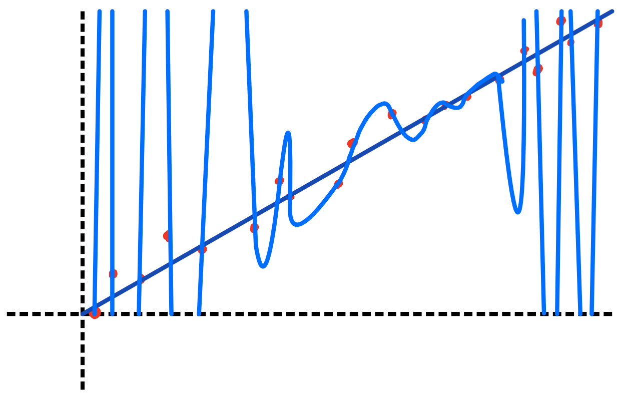
$$s(x) = \begin{cases} \boxed{p(x) = 2+3(x+1)^2} & x \in [-1,0] \\ \boxed{5+6x} & x \in [0,1] \\ \boxed{11+6(x-1)-3(x-1)^2} & x \in [1,2] \end{cases}$$

ERROR DE INTERPOLACIÓN

$\boxed{E(x) = |f(x) - p(x)|}$ $\boxed{|\text{Función original} - \text{Función interpolante}|}$

$E(x_i) = 0$, el error de interpolación es cero en los puntos x_i en los que $p(x)$ interpola a la función $f(x)$

T3. AJUSTE DE DATOS



* INTERPOLACIÓN
* AJUSTE DE DATOS

En interpolación: $\boxed{\text{N° coeficientes (incógnitas) = n° datos (ecuaciones)}}$

En ajuste: $\boxed{\text{N° coeficientes < n° datos}}$

Dado que n° incognitas < n° ecuaciones → No podremos verificar exactamente (=) las ecuaciones (=), nos conformaremos con que más o menos cumplan (≈). → Tras resolver NO pasaremos por los puntos sino cerca de ellos.

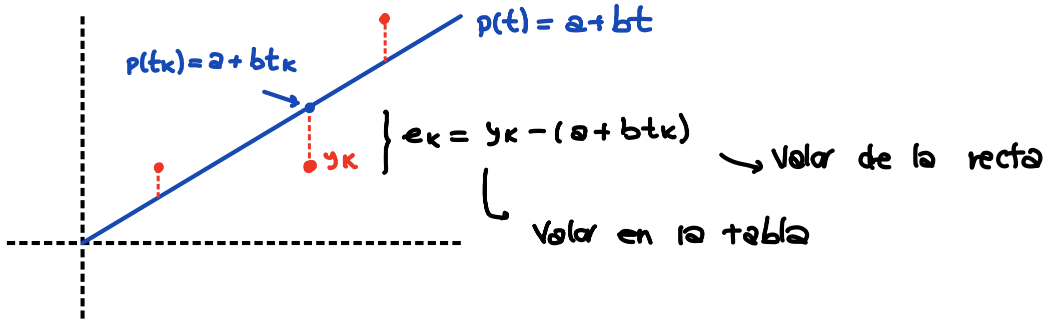
Intentaremos resolver el sistema sobredeterminado que resulta

$$\text{N ecuaciones} \updownarrow \underbrace{\begin{pmatrix} H \end{pmatrix}}_{(n<N) \text{ incógnitas}} \cdot \begin{pmatrix} c \end{pmatrix} \approx \begin{pmatrix} b \end{pmatrix}$$

Es un problema muy importante en la práctica, pues:

- Reducimos ruido en las medidas y detectamos un patrón general
- Describimos muchos datos con menos parámetros
- Reducimos la información

Queremos encontrar el mejor ajuste de una recta a una nube. Gráficamente:



Para ello, tratamos de encontrar los parámetros (a, b) que minimicen el error global del ajuste

$$\boxed{E = \sum_{k=1}^{N} e_k^2 = \sum_{k=1}^{N} (y_k - p(t_k))^2 = \sum_{k=1}^{N} (y_k - a - bt_k)^2}$$

Aquí hacen algo que ni entiendo ni vale porque es muy tedioso, por lo que pasamos a utilizar el método de mínimos cuadrados.

Así, garantizamos que el problema se reduce a RESOLVER UN SISTEMA LINEAL

Planteamos el problema de la recta como si fuese una interpolación

$a + bt_1 \approx y_1$

$a + bt_2 \approx y_2$

→ Esto es lo que podemos hacer con interpolación

$a + bt_3 \approx y_3$

→ Imposible de plantear con 2 coeficientes, cambiamos $= \to \approx$

...

$a + bt_N \approx y_N$

$$\underbrace{\begin{pmatrix} 1 & t_1 \\ 1 & t_2 \\ 1 & t_3 \\ \cdots & \cdots \\ 1 & t_N \end{pmatrix}}_{H} \cdot \underbrace{\begin{pmatrix} a \\ b \end{pmatrix}}_{\text{coefs}} \approx \underbrace{\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \cdots \\ y_N \end{pmatrix}}_{b}$$

Muy similar al método tipo ⟶

- Interpolación → H cuadrada
- Ajuste → H rectangular

Para resolver $H\cdot c \approx b$ definimos el vector de residuos o errores como:

$$\boxed{\bar r = \bar b - H\cdot c}$$

En ajuste no vamos a conseguir anular todas las componentes (no pasamos por los puntos)

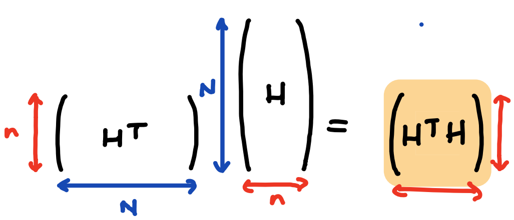
Usaremos el criterio de MÍNIMOS CUADRADOS para hacerlo pequeño

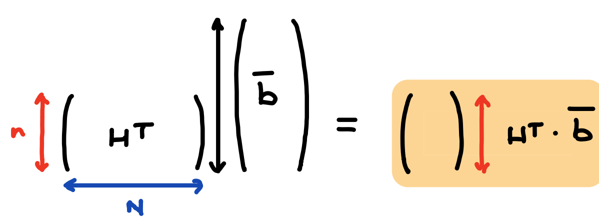
↳ La solución c debe MINIMIZAR la norma al cuadrado del v. residuos

$$\boxed{\|\bar r\|_2^2 = \|\bar b - H\cdot \bar c\|_2^2}$$

La solución c. que minimiza esta norma del vector de residuos r es la solución de las llamadas ECUACIONES NORMALES

$$\boxed{(H^T H)\cdot \bar c = H^T\cdot \bar b}$$





- No importa el número (N) de puntos iniciales presentes en el ajuste, que al final terminaremos con el num de coeficientes n (E) anterior: 2 coefs (a,b) → terminamos con un sistema 2x2)
- La K-ésima comp de r es: $r_K = (\bar b - H\bar c) = y_K - (a + b t_K) = e_K$
- La norma de r al cuadrado es: $\|r\|_2^2 = \|\bar b - H\bar c\|_2^2 = \sum_K (\bar b - H\bar c)_K^2 = \sum_K e_K^2$

T4. RESOLUCIÓN DE ECS. NO LINEALES

INTRODUCCIÓN

- Una ecuación no lineal es una ecuación para la que no existen reglas generales que ayuden a resolverla.
- Para resolverlas, se pasa la ecuación a una forma $f(x)=0$ y se calcula una 's' tal que $f(s)=0$
- A esa 's' también se la llama raíz de la ecuación $f(x)=0$ o cero de la función $f(x)$

TEOREMA DE BOLZANO

- El teorema de Bolzano se utiliza para localizar raíces
- Si una función $f(x)$ es continua en un intervalo $[a,b]$ y $f(a)\cdot f(b)<0$ entonces la función tiene al menos una raíz en el intervalo $[a,b]$
- Si $f(x)$ es monótona en el intervalo, la raíz es única
- Que $f(a)\cdot f(b)>0$ no quiere decir que no haya raíces en el intervalo. Podría haberlas o no.

$$\boxed{\left.\begin{array}{l} f(x) \text{ continua } [a,b] \\ f(a)\cdot f(b)<0 \end{array}\right\} \exists s\in[a,b],\ f(s)=0}$$

Ej: $f(x)=x^2-e^{-x}$ $[0,1]$

$$\left.\begin{array}{l} f(x) \text{ continua en } [0,1] \\ f(0)=-1 \\ f(1)=1-1/e>0 \end{array}\right\} \exists s\in[0,1],\ f(s)=0$$

$f'(x) = 2x + e^{-x}$. $f'(x)$ no cambia de signo en $[0,1]$ por lo que $f(x)$ es monótona en el intervalo $\rightarrow$ s es única

MÉTODOS ITERATIVOS

En este tema usaremos métodos iterativos para hallar la sol. de $f(x)=0$

Construir una sucesión $\langle x_n \rangle$ convergente que verifique que $\lim_{n\to\infty} x_n = s$

REGLA ITERATIVA

$$\boxed{\begin{cases} x_0 & \text{punto inicial} \\ x_{n+1} = g(x_n) \end{cases}}$$

- Se saca una función $g(x)$ de la función $f(x)$ (suele despejarse una 'x').
- Las funciones $f(x)$ y $g(x)$ tienen una relación en la que $s: f(s)=0 \rightarrow s = g(s)$ punto fijo de $g(x)$
- Esta regla se usa para iterar sobre ella y encontrar una raíz lo más precisa a la real

EJ: $f(x) = x^2 - e^{-x} = 0$

- $x^2 - 1/e^x = 0$; $e^x = 1/x^2$; $\ln e^x = \ln 1/x^2$; $x = -\ln x^2$
- $x^2 = e^{-x}$; $x = \sqrt{e^{-x}}$

Método 1: $x_{n+1} = -\ln x_n^2$

Método 2: $x_{n+1} = \sqrt{e^{-x_n}}$

ALGORITMO

- Para el algoritmo se da un punto de arranque 'x_0' y un criterio de parada q puede ser un nº máximo de iteraciones o una precisión a la que llegar

ERROR

- Si se conoce 's': $\boxed{e_n = |x_n - s|}$
- Si no se conoce 's': $\boxed{e_n \approx |x_{n+1} - x_n|}$

CONVERGENCIA

- Un método es CONVERGENTE si el error tiende a 0 para cualquier $x_0 \in [a,b]$
  - VELOCIDAD DE LA CONVERGENCIA: En caso de convergencia, el orden de convergencia es p.

$$\boxed{\frac{e_{n+1}}{e_n^p} \leq K \neq 0}$$

(para n suficientemente grande)

REMINDER

$$\boxed{|e| \approx 10^{-\text{cifras\_ok}}}$$

- Orden 1 o convergencia lineal: Se gana 1 cifra en cada itera.

$$\boxed{\frac{e_{n+1}}{e_n} \leq K \neq 0} \rightarrow \boxed{\log_{10}(e_{n+1}) \approx -\log_{10}(e_n) + \log_{10}(1/K)}$$

cifras dec. en iter n+1 — cifras dec en iter n — ganancia en cada iteración

- Orden 2 o convergencia cuadrática: Duplican cifras OK por iteración

$$\boxed{\frac{e_{n+1}}{e_n^2} \leq K \neq 0} \rightarrow \boxed{\log_{10}(e_{n+1}) \approx -2\log_{10}(e_n) + \log_{10}(1/K)}$$

↓

doble de cifras

MÉTODO DE LA BISECCIÓN

- Este método solo funciona si en el intervalo $[a,b]$ hay cambio de signo.
- Se coge una mitad del intervalo y se comprueba si hay cambio de signo.
- Si hay cambio, se establece ese intervalo para la iteración. Si no hay cambio se coge el otro intervalo.

EJ: $f(x) = x^3 + 2x^2 + 3x + 4 = 0$ en $[-2, 0]$

Hallar intervalo de longitud 0,5 que contenga a 's'.

$$\left.\begin{array}{l} f(x) \text{ continua} \\ f(-2) = -8+8-6+4 = -2 \\ f(0) = 4 \end{array}\right\} \exists s \in [-2,0],\ f(s) = 0$$

Método bisección

- Se coge intervalo $[-2,-1]$ y se comprueba cambio de signo:

  $f(-2) = -2$; $f(-1) = 2$

  Hay cambio de signo por lo que se sigue con el intervalo $[-2,-1,5]$
- Se utiliza el intervalo $[-2,-1,5]$

  $f(-1,5) = 0,625$; $f(-2) = -2$

  Hay cambio de signo por lo que hay raíz y además este intervalo ya tiene longitud 0,5.

  $\underline{\underline{[-2, -1,5]}}$

MÉTODO NEWTON - RAPHSON

OBJETIVO: Hallar el cero de una función $f(x)$ no lineal → No sabemos

SOLUCIÓN: Cambiar $f(x)$ por una función más sencilla y hallar su cero la sol. NO será la correcta pero será una "mejor" aprox de la hipot. inic.

¿QUÉ FUNCIÓN SENCILLA USAREMOS PARA APROX $f(x)$? → SU RECTA TANGENTE

1. Dado el pto de partida x_0, calculamos la recta tg a $f(x)$ en x_0: $\boxed{r(x)=f(x_0)+f'(x_0)(x-x_0)}$
2. Hallamos el cero de $r(x)$: $r(x)=0 \rightarrow \boxed{x = x_0 - f(x_0)/f'(x_0)}$
3. Obviamente $x \neq s$ pero esperamos que sea mejor aprox que x_0
4. Repetimos el proceso desde el nuevo punto $x_0 = x$

ALGORITMO

$$\boxed{x_{k+1} = x_k - f(x_k)/f'(x_k)}$$

EJ: $f(x) = x^2 - 2 = 0$

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} = x_k - \frac{x_k^2 - 2}{2x_k} = \underline{\underline{\frac{x_k^2}{2} + \frac{1}{x_k}}}$$

ERROR

$$\boxed{e_n = |x_n - s| \approx |x_{n+1} - x_n|}$$

CONTRARRESTAR

$$\boxed{e_n \leq \underbrace{m \cdot e_{n-1}^2}_{1} \leq \underbrace{\frac{1}{m}(m\, e_0)^{2^n}}_{2}}$$

1. Relaciona el error en n con la etapa anterior
2. Relaciona el error en la etapa n con el inicial

$$\boxed{m = \frac{\max_{x \in V_s} |f''(x)|}{2 \cdot \min_{x \in V_s} |f'(x)|}}$$ V_s es un entorno en el que está s.

CONVERGENCIA

- Se inicializa el método con x_0 tq $me_0 < 1$

$\boxed{e_n \leq 1/m\,(m \cdot e_0)^{2^n} \xrightarrow[n\to\infty]{} 0}$ el método converge

- Si hay convergencia y la raíz es simple, el método es de orden 2

Nº DE ITERACIONES QUE GARANTIZAN UN ERROR MENOR QUE UNA TOLERANCIA 'tol'

$$\boxed{e_n \leq 1/m\,(me_0)^{2^n} < tol}$$

$2^n \cdot \log(me_0) < \log(tol) + \log(m)$; $\quad 2^n > \dfrac{\log(m \cdot tol)}{\log(m \cdot e_0)}$

$$\boxed{n > \log\left(\frac{\log(m \cdot tol)}{\log(m \cdot e_0)}\right) / \log(2)}$$

EJ: Calcular raíz de $f(x) = x^2 \cdot e^{-x} - 1 = 0$ en $[1,2]$

1. Comprobación T. Bolzano

$$\left.\begin{array}{l} f(x) \text{ continua en } [1,2] \\ f(1) = -e^{-x} < 0 \\ f(2) = 1 - 1/e^2 > 0 \end{array}\right\} \exists s \in [1,2],\ f(s) = 0$$

2. Newton con $x_0 = 1{,}5$

$$x_{n+1} = x_n - f(x_n)/f'(x_n) = x_n - \frac{x_n^2 - e^{-x_n} - 1}{2x_n + e^{-x_n}}$$

x_k	e_k
1,1814060280444023	$3,19 \cdot 10^{-1}$
1,1481159331009358	$3,33 \cdot 10^{-2}$
1,147757673947452 41	$3,58 \cdot 10^{-4}$
1,1477576321447440	$4,13 \cdot 10^{-8}$
1,147757632144743 6	$4,44 \cdot 10^{-16}$

3. Estudiar convergencia si $x_0 = 1,5$

$$m \cdot e_0 < 1$$

$$m = \frac{\max_{x\in[1,2]} |f''(x)|}{2 \cdot \min_{x\in[1,2]} |f'(x)|} = \frac{2 - 1/e^2}{2\cdot(2+1/e)} = 0,3937$$

$$f''(x) = 2 - e^{-x} \rightarrow \left.\begin{array}{l} f''(1) = 2 - 1/e \ ; \\ f''(2) = 2 - 1/e^2 \end{array}\right\} f''(2) > f''(1)$$

$$f'(x) = 2x + e^{-x} \rightarrow \begin{array}{l} f'(1) = 2 + 1/e \ ; \\ f'(2) = 4 + 1/e^2 \end{array}$$

$e_0 = |1,5 - s| < 0,5$ por estar s en $[1,2]$ $\left(e_n = |x_n - s| \leq \frac{1}{m}(me_0)^{2^n}\right)$

$me_0 < 0,5 \cdot 0,3937 = 0,1968 < 1$; $me_0 < 1$ CONVERGE