1.1 Calcular el error absoluto y relativo de las siguientes aproximaciones a $p$ por $p^*$:

a) $p = e^{10}$, $p^* = 22000$,

b) $p = 10^{\pi}$, $p^* = 1400$,

c) $p = 8!$, $p^* = 39900$,

d) $p = 9!$, $p^* = \sqrt{18\pi}(9/e)^9$.

$\varepsilon_{ABS} = |p - p^*| \qquad \varepsilon_{REL} = \left|\frac{p-p^*}{p}\right|$

CALCULADO EN PYTHON

a) $\varepsilon_{ABS} = |e^{10} - 22000| = 26{,}46$

$\varepsilon_{REL} = 0{,}0012 \rightsquigarrow \approx 0{,}1\%$

b) $\varepsilon_{ABS} = 14{,}54$

$\varepsilon_{REL} = 0{,}0105 \rightsquigarrow \approx 1\%$

c) $\varepsilon_{ABS} = 420$

$\varepsilon_{REL} = 0{,}0104 \rightsquigarrow \approx 1\%$

d) $\varepsilon_{ABS} = 3343{,}13$

$\varepsilon_{REL} = 0{,}0092 \rightsquigarrow \approx 1\%$

1.2 Encuentre el intervalo más grande en donde $p^*$ puede estar de forma que el error relativo al aproximar los siguientes $p$ no sea mayor a $10^{-4}$.

a) $\pi$,

b) $e$,

c) $\sqrt{2}$,

d) $7^{1/3}$.

$$\varepsilon_{REL} = \left| \frac{p - p^*}{p} \right| < 10^{-4}$$

$$\Rightarrow \quad -10^{-4} < \frac{p - p^*}{p} < 10^{-4}$$

$$\Rightarrow \quad -10^{-4}p - p < -p^* < 10^{-4}p - p$$

$$10^{-4}p + p > p^* > -10^{-4}p + p$$

$$\rightarrow \quad p \in (-10^{-4}p + p, 10^{-4}p + p)$$

.... PYTHON

$\Downarrow$

a) $\rightsquigarrow p^* \in (3{,}1412\ldots, 3{,}1419\ldots)$

b) $\rightsquigarrow p^* \in (2{,}7180\ldots, 2{,}7185\ldots)$

c) $\rightsquigarrow p^* \in (1{,}4140\ldots, 1{,}4143\ldots)$

d) $\rightsquigarrow p^* \in (1{,}9127\ldots, 1{,}9131\ldots)$

1.3 El número $\pi$ se puede obtener a partir de las siguientes igualdades.

a) $\pi = 4\left[\arctan\left(\frac{1}{2}\right) + \arctan\left(\frac{1}{3}\right)\right]$,

b) $\pi = 16\arctan\left(\frac{1}{5}\right) - 4\arctan\left(\frac{1}{239}\right)$.

Para cada uno de los casos, calcule una aproximación a $\pi$ considerando los primeros 3 términos no nulos de la serie de Taylor de la arcotangente

$$\arctan(x) \approx x - (1/3)x^3 + (1/5)x^5,$$

Usando todos los dígitos disponibles en la calculadora. ¿Cuál fórmula llevó a un menor error? Argumente la razón de la diferencia.

PYTHON.-

LA VERSION b) TIENE MUCHO MENOS ERROR DE REDONDEO ES DE $O(x^7)$, A MENOR VALOR DEL X ESTIMADO, MENOR ERROR DE REDONDEO, SE VE QUE $\left(\frac{1}{3}\right)^7 << \left(\frac{1}{239}\right)^7$, ENTONCES b TIENE MENOR ERROR

# 1.4 Demostrar que:

a) $\cos(x) = 1 - \frac{x^2}{2} + O(x^4)$

b) $\sin(x) = x - \frac{x^3}{6} + O(x^5)$

Uso $f(x) = \sum_{k=0}^{n} \frac{f^{(k)}(0)}{k!} x^k + \frac{f^{(n+1)}(\xi)}{(n+1)!} x^{n+1}$

→ a) TAYLOR DE ORDEN 3 DE COS.

$\Rightarrow = \frac{f^{(0)}(0)}{0!} x^0 + \frac{f^{1}(0)}{1!} x^1 + \frac{f^{2}(0)}{2!} x^2 + \frac{f^{3}(0)}{3!} x^3 + \overbrace{\frac{f^{4}(\xi)}{4!} x^4}^{R_4}$

$= \frac{\cos(0)}{1} 1 + \frac{-\operatorname{sen}(0)}{1} x + \frac{-\cos(0)}{2} x^2 + \frac{\operatorname{sen}(0)}{6} x^3 + \frac{f^{4}(\xi)}{4!} x^4$

$= 1 - \frac{1}{2} x^2 + R_4 \rightsquigarrow \cos''''(\xi) = \cos(\xi) \leq 1$

$\Rightarrow \frac{f^{4}(\xi)}{4!} x^4 \leq \frac{1 \cdot x^4}{4!} \Rightarrow R_4(x) = O(x^4)$

$\Rightarrow \cos(x) = 1 - \frac{1}{2} x^2 + O(x^4)$

b) MISMA IDEA QUE EN a) HAGO TAYLOR $n=4$

$\rightarrow \operatorname{sen}(x) = \cancel{0} + \frac{1}{1!} x + \cancel{\frac{0}{2!} x^2} + \frac{-1}{3!} x^3 + \cancel{\frac{0}{4!} x^4} + \frac{f^5(\xi)}{5!} x^5$

$= x - \frac{1}{6} x^3 + \frac{f^5(\xi)}{5!} x^5$

$f^5(\xi) = \cos(\xi) \leq 1$

$\Rightarrow \frac{f^5(\xi)}{5!} x^5 \leq \frac{x^5}{5!} \Rightarrow R_5(x) = O(x^5)$

$\Rightarrow \operatorname{sen}(x) = x - \frac{1}{6} x^3 + O(x^5)$

1.5 Encuentre la tasas de convergencia de las siguientes secuencias cuando $n \to \infty$.

a) $\lim_{n\to\infty} \sin \frac{1}{n} = 0$,

b) $\lim_{n\to\infty} \sin \frac{1}{n^2} = 0$,

c) $\lim_{n\to\infty} \left(\sin \frac{1}{n}\right)^2 = 0$,

d) $\lim_{n\to\infty}[\ln(n+1) - \ln(n)] = 0$.

$e_n = |X_n - L| \qquad \lim_{n\to\infty} \frac{e_{n+1}}{e_n} = c$

a) $X_n = \operatorname{sen}(\frac{1}{n}) \quad L=0$

$e_n = |\operatorname{sen}(\frac{1}{n})|$

POR TAYLOR DE ORDEN 4:

$\operatorname{sen}(x) = x - \frac{1}{6}x^3 + O(x^5)$

$\Rightarrow e_n = |\operatorname{sen}(\frac{1}{n})| = |\frac{1}{n} - \frac{1}{6}\frac{1}{n^3} + O(\frac{1}{n^5})|$

$= \frac{1}{n} + \frac{1}{6}\frac{1}{n^3} + O(\frac{1}{n^5})$

$\Rightarrow e_n \sim \frac{1}{n}$

$\frac{e_{n+1}}{e_n} = \frac{1}{n+1} \cdot n \Rightarrow \lim_{n\to\infty} \frac{n}{n+1} = 1 \rightarrow$ CONVERGENCIA LINEAL

b) $X_n = \operatorname{sen}(\frac{1}{n^2}) \quad L=0$

MISMO TAYLOR $\rightarrow x - \frac{1}{6}x^3 + O(x^5)$

$e_n = \operatorname{sen}(1/n^2) = \frac{1}{n^2} - \frac{1}{6n^6} + O(\frac{1}{n^{10}})$

A MAYOR ORDEN $e_n \sim \frac{1}{n^2}$

$\lim_{n\to\infty} \frac{e_{n+1}}{e_n} = \frac{n^2}{(n+1)^2} \Rightarrow \lim_{n\to\infty} \frac{n^2}{n^2+2n+1} = 1 \rightarrow$ CONVERGENCIA LINEAL

c)

$X_n = \operatorname{sen}(\frac{1}{n})^2 \quad L=0$

$\operatorname{sen}(x) = x - \frac{1}{6}x^3 + O(x^5)$

MAS COSAS DE ORDEN MAYOR

$\rightarrow \operatorname{sen}(x)^2 = x^2 - \frac{1}{6}x^4 + \underbrace{O(x^6) + \ldots}$

COSAS MUY CHICAS

$\rightarrow \operatorname{sen}(\frac{1}{n})^2 = \frac{1}{n^2} - \frac{1}{6}\frac{1}{n^4} + \underbrace{O(\frac{1}{n^6}) + \ldots}$

$e_n \sim \frac{1}{n^2}$ ) IGUAL QUE b) $\Rightarrow$ CONVERGENCIA LINEAL

d)

$X_n = \ln(n+1) - \ln(n) \qquad L=0$

$= \ln(1 + 1/n)$

TAYLOR

$\ln(1+x) = x - \frac{1}{2}x^2 + O(x^3)$

$\Rightarrow \ln(1+\frac{1}{n}) = \frac{1}{n} - \frac{1}{2}\frac{1}{n^2} + O(\frac{1}{n^3})$

ME QUEDO A MAYOR ORDEN

$e_n \sim \frac{1}{n}$ ) IGUAL QUE a) CONVERGENCIA LINEAL

1.6 Encuentre numéricamente en *python* el valor del epsilon de máquina en *doble precision*, *simple precision* y *half precision*.

```python
# Ej1.6

doublePrecision = np.float64(1)
simplePrecision = np.float32(1)
halfPrecision = np.float16(1)

def getMachineEpsilon(precision):
    epsilon = precision
    while 1 + epsilon > 1:
        epsilon /= 2
    return epsilon

print(f"Double Precision: {getMachineEpsilon(doublePrecision)}")
print(f"Simple Precision: {getMachineEpsilon(simplePrecision)}")
print(f"Half Precision: {getMachineEpsilon(halfPrecision)}")
```

✓ 0.0s

```
Double Precision: 1.1102230246251565e-16
Simple Precision: 5.960464477539063e-08
Half Precision: 0.00048828125
```

1.7 Utilice aritmética de punto flotante redondeando a tres dígitos para realizar los siguientes cálculos. Calcule el error absoluto y relativo, con el valor exacto determinado a al menos cinco dígitos.

a) $133 + 0{,}921$,

b) $133 - 0{,}499$,

c) $(121 - 0{,}327) - 119$,

d) $(121 - 119) - 0{,}327$.

a) $133 + 0{,}921$

$= 1{,}33 \times 10^2 + 0{,}00921 \times 10^2 = 1{,}33921 \times 10^2 = 1{,}34 \times 10^2 = 134$

$\mathcal{E}_{ABS} = |134 - 133{,}921| = 0{,}79$ $\quad \mathcal{E}_{REL} \approx 3{,}766 \times 10^{-3}$

b) $133 - 0{,}499$

$= 1{,}33 \times 10^2 - 0{,}00499 \times 10^2 = 1{,}32501 \times 10^2 = 133$

$\mathcal{E}_{ABS} = 0{,}499$ $\quad \mathcal{E}_{REL} \approx 3{,}766 \times 10^{-3}$

c) $\underbrace{\underbrace{(121 - 0{,}327)}_{2} - 119}_{2}$

$121 - 0{,}327 = 1{,}21 \times 10^2 - 0{,}00327 \times 10^2 = 1{,}20673 \times 10^2 = 1{,}21$

$\Rightarrow 121 - 119 = 1{,}21 \times 10^2 - 1{,}19 \times 10^2 = 0{,}02 \times 10^2 = 2$

$\mathcal{E}_{ABS} = 0{,}327$ $\quad \mathcal{E}_{REL} \approx 1{,}9546 \times 10^{-1}$

d) $(121 - 119) - 0{,}327$

$121 - 119 = \dots = 2$

$\rightarrow 2 - 0{,}327 = 0{,}02 \times 10^2 - 0{,}00327 \times 10^2 = 0{,}01673 \times 10^2 = 1{,}67$

$\mathcal{E}_{ABS} = 0{,}003$ $\quad \mathcal{E}_{REL} \approx 1{,}793 \times 10^{-3}$

1.8 Suponga que $fl(y)$ es una aproximación de redondeo de $k$-dígitos a $y$. Muestre que

$$\left|\frac{y - fl(y)}{y}\right| \leq 0{,}5 \times 10^{-k+1}.$$

Ayuda: si $d_{k+1} < 5$, entonces $fl(y) = 0.d_1 d_2 \cdots d_k \times 10^n$, si $d_{k+1} \geq 5$ entonces $fl(y) = 0.d_1 d_2 \cdots d_k \times 10^n + 10^{n-k}$.

$$\varepsilon_{REL} = \left|\frac{y - fl(y)}{y}\right| \leq 0{,}5 \times 10^{-k+1} \qquad fl(y) = \begin{cases} 0{,}d_1 d_2 \ldots d_k \times 10^n & \text{si } d_{k+1} < 5 \\ 0{,}d_1 d_2 \ldots d_k \times 10^n + 10^{n-k} & \text{si } d_{k+1} \geq 5 \end{cases}$$

$$y = 0.d_1 d_2 \ldots d_k d_{k+1} \ldots 10^n$$

PRIMEROS K DECIMALES

SEA $\overset{\frown}{T} = 0{,}d_1 d_2 \ldots d_k \times 10^n$

$\Rightarrow r = y - T = 0{,}d_{k+1} d_{k+2} \ldots 10^{n-k}$
$\hookrightarrow \in [0, 10^{n-k})$

Si $d_{k+1} < 5 \Rightarrow fl(y) = T$

$\Rightarrow |y - fl(y)| = |r| = |0{,}\underbrace{d_{k+1}}_{<5} d_{k+2} \ldots \times 10^{n-k}| < \frac{1}{2} 10^{n-k}$

Si $d_{k+1} \geq 5 \Rightarrow fl(y) = T + 10^{n-k}$

$\Rightarrow |y - fl(y)| = |y - (T + 10^{n-k})| = |r - 10^{n-k}| = |0{,}d_{k+1} d_{k+2} \ldots \times 10^{n-k} - 10^{n-k}|$

$= |10^{n-k} (0{,}\overset{\geq 5}{d_{k+1}} \ldots - 1)|$

$= 10^{n-k} (\overline{0{,}d_{k+1} \ldots})$ "NEGADO"

COMO RESTE 1 Y HAGO MODULO, ME QUEDA EL COMPLEMENTO DE LOS DIGITOS, SI $d_{k+1} = c$

$\Rightarrow$ AHORA $d_{k+1} = 1 - c$, COMO $d_{k+1} \geq 5$ ACA $\Rightarrow d_{k+1} \leq 5$

$\Rightarrow (0{,}d_{k+1} \ldots) \leq 0{,}5$

OSEA QUE $|y - fl(y)| \leq 0{,}5 \; 10^{n-k}$

TENGO EL ABSOLUTO, BUSCO EL RELATIVO:

TENGO $y = 0{,}d_1 d_2 \ldots d_k \ldots 10^n$ TOMO $d_1 \neq 0 \Rightarrow y \geq 0{,}1 . 10^n$ $= 10^{-1} . 10^n = 10^{n-1}$

$\Rightarrow |y| \geq |0{,}1 . 10^n| = 0{,}1 . 10^n$

$\Rightarrow$ Si $|y - fl(y)| \leq 0{,}5 . 10^{n-k}$

$$\Rightarrow \frac{|y - fl(y)|}{|y|} \leq \frac{0{,}5 . 10^{n-k}}{10^{n-1}}$$

$$\Rightarrow \frac{|y - fl(y)|}{|y|} \leq 0{,}5 . 10^{-k+1}$$

1.9 El desarrollo de Taylor de la función $e^x$ proporciona una forma muy inestable de calcular este valor cuando $x$ es negativo. Evalúe numéricamente el desarrollo de Taylor hasta grado $n$ de la función $e^x$ en $x = -12$, para $n = 1, .., 100$. Comparar con el valor exacto: 0,000006144212353328210... ¿Cuáles son las principales fuentes de error? Proponer un método alternativo para estimar $e^{-12}$. Verificar si la aproximación obtenida es mejor.

USANDO $e^{-12} \sum_{k=1}^{n} \frac{(-12)^k}{k!}$

```
Valor de e^-12:  6.14421235332821e-06
Menor error hallado con n =  53 es:  2.521159347371356e-13
Valor encontrado:  6.1442126054441445e-06
Mayor error hallado con n =  11 es:  9504.789097053303
Valor encontrado:  -9504.789090909091
```

ERRORES

ESTO SE PUEDE DEBER PRINCIPALMENTE A ORDENES DE SUMAS, ARRASTRES DE TRUNCADO, REDONDEO, CANCELACION CATASTROFICA, ETC.

DATO SUGIERE APROXIMAR $e^1 \sim \sum_{k=1}^{n} \frac{1}{k!}$ PARA $n$ MUY GRANDE

Y DESPUES ELEVAR... ES MEJOR!

1.10 Considere la función $f(x) = \frac{1-\cos(x)}{x^2}$ y analice su comportamiento numérico para valores de $x$ cercanos a cero, $x = 10^{-k}$, $k = 1, \ldots$ Evalúe la función en doble precisión para una serie de valores decrecientes de $x$ y compare los resultados con el valor límite $\lim_{x\to 0} f(x) = \frac{1}{2}$ (no olvide usar escala logarítmica para visualizar). Justifique la pérdida de precisión observada y reformule la expresión de $f(x)$ de manera algebraicamente equivalente que mejore la estabilidad numérica. Compare los resultados obtenidos con ambas expresiones.

$f(x) = \frac{1-\cos(x)}{x^2}$ → VALORES $\approx 0$ $\Rightarrow x = 10^{-k}$, $k = 1, 2, \ldots$

| k | x=10^{-k} | f | err |
|---|---|---|---|
| 1 | 1.0e-01 | 4.9958347219741783e-01 | 4.165e-04 |
| 2 | 1.0e-02 | 4.9999583334736641e-01 | 4.167e-06 |
| 3 | 1.0e-03 | 4.9999995832550326e-01 | 4.167e-08 |
| 4 | 1.0e-04 | 4.9999999696126451e-01 | 3.039e-09 |
| 5 | 1.0e-05 | 5.0000004137018539e-01 | 4.137e-08 |
| 6 | 1.0e-06 | 5.0004445029117051e-01 | 4.445e-05 |
| 7 | 1.0e-07 | 4.9960036108132050e-01 | 3.996e-04 |
| 8 | 1.0e-08 | 0.0000000000000000e+00 | 5.000e-01 |
| 9 | 1.0e-09 | 0.0000000000000000e+00 | 5.000e-01 |
| 10 | 1.0e-10 | 0.0000000000000000e+00 | 5.000e-01 |
| 11 | 1.0e-11 | 0.0000000000000000e+00 | 5.000e-01 |
| 12 | 1.0e-12 | 0.0000000000000000e+00 | 5.000e-01 |
| 13 | 1.0e-13 | 0.0000000000000000e+00 | 5.000e-01 |
| 14 | 1.0e-14 | 0.0000000000000000e+00 | 5.000e-01 |
| 15 | 1.0e-15 | 0.0000000000000000e+00 | 5.000e-01 |
| 16 | 1.0e-16 | 0.0000000000000000e+00 | 5.000e-01 |

MENOR ERROR ENCONTRADO

LLEGA UN PUNTO QUE SE PIERDE LA PRECISION Y SE ESTIMA $1-\cos(x) \approx \frac{x^2}{2}$ Y ARRANCA A DAR CERO

TERMINO PRINCIPAL DE LA EXPANSION DE TAYLOR

PARA ALGO MAS ESTABLE PODEMOS USAR QUE $1-\cos(x) = 2\operatorname{sen}^2\left(\frac{x}{2}\right)$

$$\Rightarrow f(x) = \frac{1}{2}\left(\frac{\operatorname{sen}(x/2)}{x/2}\right)^2$$

| k | x=10^{-k} | f | err |
|---|---|---|---|
| 1 | 1.0e+00 | 4.9958347219742344e-01 | 4.165e-04 |
| 2 | 2.0e+00 | 4.9999583334722214e-01 | 4.167e-06 |
| 3 | 3.0e+00 | 4.9999995833333466e-01 | 4.167e-08 |
| 4 | 4.0e+00 | 4.9999999958333341e-01 | 4.167e-10 |
| 5 | 5.0e+00 | 4.9999999999583322e-01 | 4.167e-12 |
| 6 | 6.0e+00 | 4.9999999999995826e-01 | 4.174e-14 |
| 7 | 7.0e+00 | 4.9999999999999956e-01 | 4.441e-16 |
| 8 | 8.0e+00 | 5.0000000000000000e-01 | 0.000e+00 |
| 9 | 9.0e+00 | 5.0000000000000000e-01 | 0.000e+00 |
| 10 | 1.0e+01 | 5.0000000000000000e-01 | 0.000e+00 |
| 11 | 1.1e+01 | 5.0000000000000000e-01 | 0.000e+00 |
| 12 | 1.2e+01 | 5.0000000000000000e-01 | 0.000e+00 |
| 13 | 1.3e+01 | 5.0000000000000000e-01 | 0.000e+00 |
| 14 | 1.4e+01 | 5.0000000000000000e-01 | 0.000e+00 |
| 15 | 1.5e+01 | 5.0000000000000000e-01 | 0.000e+00 |
| 16 | 1.6e+01 | 5.0000000000000000e-01 | 0.000e+00 |

ASI NOS EVITAMOS RESTAR NUMEROS CASI IGUALES

MUY BUENO!

1.11 ¿Cuántas multiplicaciones y cuántas sumas se necesitan para realizar la siguiente operación en la forma en la que está dada?

$$\sum_{i=1}^{n}\sum_{j=1}^{i} a_i b_j.$$

Modifique la suma de modo tal que reduzca el número de operaciones necesarias. Compare la *performance* de ambos algoritmos en *python* usando un vector aleatorio de *numpy*.

$$\sum_{i=1}^{n}\sum_{j=1}^{i} a_i b_j$$

Si $n=4$

$$a_1b_1 + a_2b_1 + a_2b_2 + a_3b_1 + a_3b_2 + a_3b_3 + a_4b_1 + a_4b_2 + a_4b_3 + a_4b_4$$

PRODUCTOS: $1+2+3+4 \Rightarrow \frac{n(n+1)}{2}$

SUMAS: $0+2+3+4 \Rightarrow \frac{n(n+1)}{2} - 1$

TIENE SENTIDO SI TENES LA SUMA DE $n$ PRODUCTOS, TENES $N-1$ SUMAS

FORMA EFICIENTE:

$$\Rightarrow a_1b_1 + a_2b_1 + a_2b_2 + a_3b_1 + a_3b_2 + a_3b_3 + a_4b_1 + a_4b_2 + a_4b_3 + a_4b_4$$

$$= b_1(a_1+a_2+a_3+a_4) + b_2(a_2+a_3+a_4) + b_3(a_3+a_4) + a_4b_4$$

SI TE VAS GUARDANDO LA SUMA PARCIAL DE $\vec{a}$:

$$= a_4b_4 + b_3(\overbrace{S_1}^{S_1}+a_3) + b_2(\overbrace{S_2}^{S_2}+a_2) + b_1(\overbrace{S_3}^{S_3}+a_1)$$

ENTONCES TENGO 4 PRODUCTOS, 6 SUMAS

$\Rightarrow$ $n$ PRODUCTOS $2n-2$ SUMAS.

PARA $n=5$

$$\Rightarrow a_5b_5 + \overbrace{b_4(S_1+a_4) + b_3(S_2+a_3) + b_2(S_3+a_2) + b_1(S_4+a_1)}^{N-1 \text{ SUMAS}}$$

$\underbrace{\qquad\qquad}_{N-1 \text{ SUMAS}}$

$n$ PRODUCTOS

ESTO ES $O(n)$ CONTRA $O(n^2)$

```
n = 9000
MALA       :  2.477725e+03   tiempo = 3.686s
Optima     :  2.477725e+03   tiempo = 0.000110s

Conteos teóricos de operaciones:
  MALA      mult = 40,504,500  |  sumas = 40,504,499
  Optim.:   mult = 9,000       |  sumas = 17,998
```

CLARITO EL EJEMPLO CHE

1.12 **Método de Horner**: Dado un polinomio $p(x) = a_n x^n + a_{n-1}x^{n-1} + ... + a_1 x + a_0$. ¿Cuántos productos y cuántas sumas se realizan al evaluar el poliomio en un cierto $x_0$ ? Horner propone como alternativa escribir a $p$ como $p(x) = a_0 + x(a_1 + x(a_2 + ... + x(a_{n-1} + xa_n)))$. ¿Cuántos productos y cuántas sumas se realizan al evaluar $p$ bajo esta forma?

$P(x) = a_n x^n + a_{n-1}x^{n-1} + \ldots + a_1 x_1 + a_0$ → N-1 SUMAS

Si TENGO $x^2$ → ES 1 PRODUCTO

⇒ $x^n$ → $n-1$ PRODUCTOS

⇒ $ax^n$ → $n$ PRODUCTOS

ENTONCES EN PRODUCTOS TENGO: $n + n-1 + n-2 + \ldots + 1 \Rightarrow \frac{n(n+1)}{2}$ PRODUCTOS

⇝ $\frac{n(n-1)}{2}$ PRODUCTOS N-1 SUMAS

Si $P(x) = a_0 + x(a_1 + x(a_2 + \ldots + x(a_{n-1} + xa_n)))$

USAMOS EJEMPLO CHICO $n=5$

$P(x) = a_0 + x(a_1 + x(a_2 + x(a_3 + x(a_4 + xa_5))))$

5 SUMAS

5 PRODUCTOS

Si AGRANDO $n$ POR 1, SOLAMENTE AGREGARÍA 1 SUMA Y 1 PRODUCTO.

⇒ $n$ SUMAS $n$ PRODUCTOS

2.1 Realice 3 iteraciones del método de bisección en papel sobre las siguientes funciones. Luego, cree una función en *python* que tome como entrada $f$ (cualquier función $f(x)$), $a$ y $b$ (intervalo inicial), $N$ (número máximo de iteraciones), y $\mathrm{tol}$ (tolerancia, que será la cota superior del error). Encuentre aproximaciones a las raíces tomando la tolerancia $\mathrm{tol} = 10^{-5}$ y $N = 1000$. Puede elegir tomar el criterio de terminación en la iteración $i$ como

$$|p_i - p_{i-1}| < \mathrm{tol}, \quad \frac{|p_i - p_{i-1}|}{|p_i|} < \mathrm{tol}, \text{ o } \quad |f(p_i)| < \mathrm{tol}. \tag{1}$$

a) $3x - e^x = 0$ con $x \in [1, 2]$,

b) $2x + 3\cos x - e^x = 0$ con $x \in [1, 2]$,

c) $x^2 - 4x + 4 - \ln x = 0$ con $x \in [1, 2]$ y $x \in [2, 4]$,

d) $x + 1 - 2\sin \pi x = 0$ con $x \in [0, 0{,}5]$ y $x \in [0{,}5, 1]$.

a) $f(x) = 3x - e^x$ CONTINUA ✓ $f(1)f(2) \approx -0{,}39 < 0$ ✓

I1) $a = 1$ $b = 2$

$c = \frac{1+2}{2} = 1{,}5$

$f(1)f(1{,}5) \approx 0{,}005$

I2) $a = 1{,}5$ $b = 2$

$c = \frac{1{,}5+2}{2} = \frac{7}{4}$

$f(1{,}5)f(7/4) \approx -0{,}004$

I3) $a = 7/4$ $b = 2$

$c = \frac{7/4 + 8/4}{2} = \frac{15}{8}$

$f(15/8)f(7/4) \approx 0{,}45$

I4)...

AHORA CON PYTHON

(1.5121307373046875, ~~15~~ 16) ) a
(1.239715576171875, ~~14~~ 15) ) b
(1.4123916625976562, ~~16~~ 17)
(3.057098388671875, ~~15~~ 16) ) c
(0.20603561401367188, ~~16~~ 17)
(0.681976318359375, ~~13~~ 14) ) d

2.2 Sea $f(x) = (x+2)(x+1)^2 x(x-1)^3(x-2)$. ¿A qué cero de $f$ converge el método de la bisección cuando se aplica en los siguientes intervalos?

a) $[-1{,}5, 2{,}5]$,

b) $[-0{,}5, 2{,}4]$,

c) $[-0{,}5, 3]$,

d) $[-3, -0{,}5]$.

(0.0, ~~2~~ 3) a [-1,5, 2,5]

(9.536743162531158e-08, ~~19~~ 20) b [-0,5, 2,4]

(2.000000596046448, ~~22~~ 23) c [-0,5, 3]

(-1.9999999850988388, ~~24~~ 25) D [-3, -0,5]

2.3 Una partícula cae desde el reposo por un plano inclinado cuyo ángulo $\theta$ cambia a una tasa constante

$$\frac{d\theta}{dt} = \omega < 0.$$

La distancia desde el punto de partida de la partícula a tiempo $t$ está dado por la siguiente fórmula

$$x(t) = -\frac{g}{2\omega^2}\left(\frac{e^{\omega t} - e^{-\omega t}}{2} - \sin \omega t\right).$$

Suponga que la partícula se movió $0{,}5\,\mathrm{m}$ en $1\,\mathrm{s}$ y asuma que $g = 9{,}8\,\mathrm{m\,s^{-2}}$. Encuentre una aproximación de $\omega$ con error menor a $10^{-5}$ (elegir uno de los 3 criterios).

$$X(t) = -\frac{g}{2\omega^2}\left(\frac{e^{\omega t} - e^{-\omega t}}{2} - sen(\omega t)\right)$$

$$X(1) = -\frac{9{,}8}{2\omega^2}\left(\frac{e^{\omega} - e^{-\omega}}{2} - sen(\omega)\right) = 0{,}5$$

$$\Rightarrow \underbrace{-\frac{9{,}8}{2\omega^2}\left(\frac{e^{\omega} - e^{-\omega}}{2} - sen(\omega)\right) - 0{,}5 = 0}_{\text{CONTINUA}} = f(\omega)$$

CALCULO EN PYTHON

$f(-1) \approx 1{,}1 \quad f(0) = -0{,}5$

$[a,b] = [-1,0]$ HAY SOLUCION

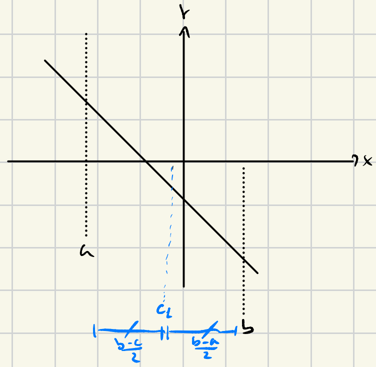
HAGO BISECCION ACA

$$\Rightarrow \omega \approx -0{,}7891 \ \frac{RAD}{S}$$

2.4 Sea $f \in C[a,b]$ y $f(a)f(b) < 0$. Muestre que el método de la bisección genera una secuencia $\{p\}_{n=1}^{\infty}$ que aproxima a $p$ tal que

$$|p_n - p| \leq \frac{b-a}{2^n},$$

cuando $n \geq 1$. Muestre que esta cota de error converge linealmente a $0$.




$I_1)$ $|p_1 - p| \leq \frac{b-a}{2}$

EN EL INTERVALO $[a,b]$, SIEMPRE PARO MI $P_i$ EN LA MITAD, LA DISTANCIA NUNCA PUEDE SER MAYOR A $\frac{b-a}{2}$

MIENTRAS AGARRO MAS $P_i$'S, LA DISTANCIA SE ACHICA A LA MITAD $\Rightarrow$ $|p_n - p| \leq \frac{b-a}{2^n}$

CUANDO $n \to \infty$

$$\Rightarrow \lim_{n \to \infty} \frac{b-a \; )\, c}{2^n \; )\, \infty} = 0$$

PARA VER CONVERGENCIA:

$X_n = \frac{b-a}{2^n}$ $\quad L = 0$

$e_n = \frac{b-a}{2^n}$

$$\frac{e_{n+1}}{e_n} = \frac{\cancel{b-a}}{2^{n+1}} \cdot \frac{2^n}{\cancel{b-a}} = 2^{-1} = \frac{1}{2}$$ ) CONVERGENCIA LINEAL

2.5 Encuentre una cota al número de iteraciones necesarias para lograr una aproximación con error menor a $10^{-3}$ a la solución de $x^3 = -x + 4$ en el intervalo $[1, 4]$. Encuentre la aproximación.

INTERVALO $[1, 4]$

$f(x) = x^3 + x - 4$ ) CONTINUA $f(1)f(4) < 0$

SE DE ANTES QUE

$$\underbrace{|P_n - P|}_{\text{DIST ENTRE ESTIMADO Y SOLUCION}} \leq \frac{b-a}{2^n}$$

$$\Rightarrow \text{BUSCO } \overbrace{|P_n - P|}^{\text{DISTANCIA}} \leq \underbrace{\frac{b-a}{2^n} \overbrace{<}^{\text{MENOR A}} \overbrace{10^{-3}}^{\text{TOLERANCIA}}}_{\text{TRABAJO ESTA PARTE}}$$

$$\Rightarrow \frac{4-1}{2^n} < \frac{1}{1000}$$

$$\Rightarrow 3000 < 2^n$$

$$\Rightarrow n > \log_2(3000) \approx 11{,}5$$

$$\Rightarrow n \geq 12$$

2.6 Sea $f(x) = (x-1)^{10}$, $p = 1$ y $p_n = 1 + 1/n$. Muestre que $|f(p_n)| < 10^{-3}$ para cualquier $n > 1$ pero que para tener $|p - p_n| < 10^{-3}$ se necesita $n > 1000$.

PARTIMOS DE:

$$\underbrace{|p_n - p|}_{} \leq \frac{b-a}{2^n} \underbrace{< 10^{-3}}_{}$$

ME QUEDO CON ESTO

$$|1 + \tfrac{1}{n} - 1| \leq 10^{-3} \Rightarrow \left|\tfrac{1}{n}\right| < 10^{-3}$$

$>0$

$$\Rightarrow \frac{1}{n} < \frac{1}{1000} \Rightarrow n > 1000$$ SE NECESITA $n > 1000$

POR OTRO LADO

SEA $f(x) = (x-1)^{10}$

BUSCO

$$\Rightarrow |f(p_n)| = \left|\left(1 + \tfrac{1}{n} - 1\right)^{10}\right| = \left|\tfrac{1}{n^{10}}\right| = \frac{1}{n^{10}} = n^{-10} < 10^{-3}$$

$$\Rightarrow \frac{1}{n^{10}} < \frac{1}{1000}$$

VALE PARA CUALQUIER $n > 1$

$$\Rightarrow n^{10} > 1000$$

$\approx 1{,}99$

$$\Rightarrow n > \sqrt[10]{1000}$$

$$\Rightarrow n > 1{,}99 \Rightarrow n > 1$$

2.7 Sea $f(x) = x^3 - 2x + 1$. Para resolver $f(x) = 0$ se proponen los siguientes cuatro problemas de punto fijo. Verifique que los puntos fijos de las funciones $g(x)$ se corresponden con raíces de $f(x)$. Escriba una función de *python* que tome como entrada $g$ (cualquier función $g(x)$), $p_0$ (valor inicial), $N$ (número máximo de iteraciones), y $\mathrm{tol}$ (tolerancia). Encuentre aproximaciones a las raíces tomando la tolerancia $\mathrm{tol} = 10^{-8}$ y $N = 1000$.

a) $g(x) = \frac{1}{2}(x^3 + 1)$, $p_0 = \frac{1}{2}$,

b) $g(x) = \frac{2}{x} - \frac{1}{x^2}$, $p_0 = \frac{1}{2}$,

c) $g(x) = \sqrt{2 - \frac{1}{x}}$, $p_0 = \frac{1}{2}$,

d) $g(x) = -\sqrt[3]{1 - 2x}$, $p_0 = \frac{1}{2}$,

a) $x = \frac{1}{2}(x^3+1) \Rightarrow 2x = x^3 + 1 \Rightarrow x^3 - 2x + 1 = 0$ ✓

b) $x = \frac{2}{x} - \frac{1}{x^2} \Rightarrow x^3 = 2x - 1 \Rightarrow x^3 - 2x + 1 = 0$ ✓

c) $x = \sqrt{2 - \frac{1}{x}} \Rightarrow x^2 = 2 - \frac{1}{x} \Rightarrow x^3 = 2x - 1 \Rightarrow x^3 - 2x + 1 = 0$ ✓

d) $x = -\sqrt[3]{1 - 2x} \Rightarrow x^3 = -1 + 2x \Rightarrow x^3 - 2x + 1 = 0$ ✓

(np.float64(0.6180339753258253), ~~28~~ 24)
(np.float64(nan), ~~999~~ 1000)
(np.float64(nan), ~~999~~ 1000)
(np.float64(-1.618033985292107), ~~16~~ 17)

EL METODO FALLA PARA b y c.

2.8 Para las funciones de punto fijo del ejercicio anterior, encuentre cuáles cumplen las condiciones de convergencia de punto fijo y en qué intervalo.

Busco $|g'(x)| < 1$ $\quad g(I) \subset I \quad L = \max(g'(x))$

$f(x) = x^3 - 2x + 1 \quad C_0 = \{\underset{P_1}{\approx -1{,}618}, \underset{P_2}{\approx 0{,}618}, \underset{P_3}{1}\}$

a) $g(x) = \frac{1}{2}(x^3+1)$

$\Rightarrow g'(x) = \frac{3}{2}x^2 \Rightarrow$ PARA $|x| < \overbrace{\sqrt{2/3}}^{\approx 0{,}8165}$

BUSCO UN INTERVALO QUE SEPA MAS O MENOS QUÉ CONTIENE ALGUN CERO DE $f(x)$ Y QUE NO SE PASE DE $\sqrt{2/3}$ COMO MÁXIMO

PROPONGO $I = [0{,}5, 0{,}8]$

$g(0{,}5) = 0{,}5625$, $g(0{,}8) = 0{,}756 \Rightarrow g(I) \subset I$ ✓

$|g'(x)|$ EN $I < 1$ ✓

$L = \text{Max}(g'(x)) = g'(0{,}8) = 0{,}96 < 1$ ✓

CONVERGE A $P_2$ EN $I$ (LINEAL)

b) $g(x) = \frac{2}{x} - \frac{1}{x^2}$

$\Rightarrow g'(x) = -\frac{2}{x^2} + \frac{2}{x^3} = \frac{2(1-x)}{x^3} \Rightarrow \frac{2|1-x|}{x^3} < 1$ DECRECIENTE

PROPONGO $I = [0{,}8, 1{,}25]$

$\Rightarrow g(0{,}8) = 0{,}9375 \quad g(1{,}25) = 0{,}96 \Rightarrow g(I) \subset I$

$\Rightarrow |g'(0{,}8)| = 0{,}78125 < 1$

EN $I$ CONVERGE LINEALMENTE

c) $g(x) = \sqrt{2 - \frac{1}{x}}$

$\Rightarrow g'(x) = \frac{1}{2x^2\sqrt{2 - 1/x}} \rightsquigarrow < 1$

DECRECIENTE

PROPONGO $I = [0{,}8, 1{,}5]$

$\Rightarrow g(0{,}8) \approx 0{,}866 \quad g(1{,}5) \approx 1{,}155 \Rightarrow g(I) \subset I$

$\Rightarrow |g'(0{,}8)| \approx 0{,}902 < 1$

EL INTERVALO PROPUESTO VALE Y EL METODO CONVERGE

d) $g(x) = -\sqrt[3]{1-2x}$

$g'(x) = \frac{2}{3(1-2x)^{2/3}}$ DECRECIENTE

PROPONGO $I = [0{,}9, 1{,}1]$

$g(0{,}9) \approx 0{,}928 \quad g(1{,}1) \approx 1{,}062 \Rightarrow g(I) \subset I$

$|g'(0{,}9)| \approx 0{,}775 < 1$

EL INTERVALO PROPUESTO VALE Y EL METODO CONVERGE

2.9 a) Demuestre el teorema de punto fijo (*Burden Faires 2.4*).

b) Demuestre que si $g$ satisface las hipótesis del teorema de punto fijo, entonces las cotas de error de la iteración de punto fijo son

$$|p_n - p| \leq k^n \max\{p_0 - a,\ b - p_0\},$$

y

$$|p_n - p| \leq \frac{k^n}{1-k}|p_1 - p_0|, \quad \text{para todo } n \geq 1.$$

a)

SEA $f(x)$ CONTINUA Y DERIVABLE EN $[a,b]$ /

- $f(x) \in (a,b)$ $\forall x \in (a,b)$
- $|f'(x)| < 1$ $\forall x \in (a,b)$

$\Rightarrow \exists p \,/\, f(p) = p$

DEMO:

Si $f(a)=a$ ó $f(b)=b \rightarrow$ PUNTO FIJO

SINO:

SEA $h(x) = f(x) - x \rightsquigarrow$ Si $h(p)=0 \Rightarrow f(p)=p$

Si $h(a)h(b) < 0$

$\Rightarrow \left(f(a)>a \wedge f(b)<b\right) \vee \left(f(a)<a \wedge f(b)>b\right)$

ENTONCES POR BOLZANO $\Rightarrow \exists\, c \,/\, h(c)=0 \rightarrow$ PUNTO FIJO

b)...

2.10 Los siguientes cuatro métodos son propuestos para computar $21^{1/3}$. Asumiendo $p_0 = 1$, ordénelos de mayor a menor con respecto a su velocidad aparente de convergencia.

a) $p_n = \dfrac{20p_{n-1} + 21/p_{n-1}^2}{21}$

b) $p_n = p_{n-1} - \dfrac{p_{n-1}^3 - 21}{3p_{n-1}^2}$

c) $p_n = p_{n-1} - \dfrac{p_{n-1}^4 - 21p_{n-1}}{p_{n-1}^2 - 21}$

d) $p_n = \left(\dfrac{21}{p_{n-1}}\right)^{1/2}$

Si $a > b$ DIGO QUE $a$ ES MÁS RAPIDO

$\Rightarrow d > a > b > c$

Resultado de la ejecución:

(np.float64(2.758924124121746), ~~105~~ 106)

(np.float64(2.7589241763811208), ~~7~~ 8)

(np.float64(0.0), ~~1~~ 2)

(np.float64(2.7589241737735395), ~~29~~ 30)

LE PIFIE, AL FINAL SE VE QUE ERA

$b > d > a > c$

↳ Ni CONVERGE ESTA CACA

2.11 Utilice un método de iteración de punto fijo para encontrar la solución de las siguientes ecuaciones con una precisión de $10^{-2}$.

a) $x^3 - x - 1 = 0$, intervalo de interés: $[1, 2]$ y $p_0 = 1$.

b) $e^{-x} - x = 0$, intervalo de interés: $[0, 1]$ y $p_0 = 1$.

c) $\sin(\ln(x)) - (x^3 - x^2) = 0$, intervalo de interés: $[0{,}1,\ 1]$ y $p_0 = 0{,}1$.

a) $f(x) = x^3 - x - 1$

$\Rightarrow x^3 = x + 1 \Rightarrow x = \sqrt[3]{x+1}$

$\Rightarrow g(x) = \sqrt[3]{x+1} \rightarrow g(1) = \sqrt[3]{2} \approx 1{,}2 \quad g(2) = \sqrt[3]{3} \approx 1{,}44 \Rightarrow g(I) \subset I$ ✓

$g'(x) = \dfrac{1}{3\sqrt[3]{(x+1)^2}}$ DECRECIENTE $\rightsquigarrow |g'(1)| \approx 0{,}2 < 1$ ✓

LA g(x) PROPUESTA VALE Y CONVERGE, RESULTADO DE PYTHON:

```
(np.float64(1.324268744551578), 4)
```

b) $f(x) = e^{-x} - x$

$\Rightarrow x = e^{-x} \Rightarrow g(x) = e^{-x} \Rightarrow g'(x) = -e^{-x}$ $\quad -\dfrac{1}{e^x}$ } DECRECIENTE

$\rightsquigarrow g(0) = 1 \quad g(1) = \dfrac{1}{e} < 1 \Rightarrow g(I) \subset I$ ✓

$|g'(0)| = 1$ NO ES ESTRICTAMENTE MENOR A UNO PERO ME IMPORTA POCO, QUE SE OCUPE LA COMPU. ✓

LA g(x) PROPUESTA VALE Y CONVERGE, RESULTADO DE PYTHON:

```
(np.float64(0.5648793473910495), 9)
```

c)

$f(x) = \operatorname{sen}(\ln(x)) - (x^3 - x^2)$

$x = e^{\operatorname{sen}^{-1}(x^3 - x^2)} = g(x)$

$g(0{,}1) \approx 0{,}99 \quad g(1) = 1$

WTF ENCONTRE UN PUNTO FIJO

FIN.

2.12 Muestre que $g(x) = 2^{-x}$ tiene un único punto fijo en $[1/3, 1]$. Utilice iteración de punto fijo para encontrarlo con una precisión de $10^{-4}$. Encuentre una cota al número de iteraciones necesarias para llegar a esta tolerancia partiendo de $p_0 = 1$. y compárela con el que obtuvo empíricamente.

$g(x) = 2^{-x} \Rightarrow g'(x) = -\ln(2)2^{-x} \rightsquigarrow \max(|g'|) \approx 0,55$

PUEDO VOLVER A $f(x) = 2^{-x} - x$

$\Rightarrow f'(x) = -\underbrace{\ln(2)2^{-x}}_{\text{TIENDE A CERO}} - 1$ ] ES ESTRICTAMENTE DECRECIENTE $\Rightarrow$ A LO SUMO UNA RAIZ

$\rightarrow f(1/3) \approx 0,4 \quad f(1) = -0,5 \rightarrow Sg(f(1/3)) \neq Sg(f(1)) \Rightarrow f$ TIENE UNA UNICA RAIZ

$\Rightarrow g$ TIENE UN UNICO PUNTO FIJO.

DEL 2.9 TENEMOS $|p_n - p| \leq \underbrace{K^n \max\{p_0 - a, b - p_0\}}$

$K^n \max\{\underbrace{1 - 1/3}_{2/3}, 1 - 1\}$

max(|g'|)

$K^n$

$(0,55)^n \frac{2}{3} < 10^{-4} \Rightarrow (0,55)^n < 10^{-4} \frac{3}{2}$

$\Rightarrow n > \log_{0,55}\left(10^{-4} \cdot \frac{3}{2}\right) \approx 14,73$

$\Rightarrow n \geq 15$

```
(np.float64(0.6437186417228692), 6)
```

EMPIRICAMENTE LO HICE EN 6 ITERACIONES.
DOMADO PATO DOMADO.

2.13 Encuentre un valor aproximado iterando 3 veces para $\sqrt[3]{25}$ utilizando el método de la bisección y el método de iteración de punto fijo. Compare los resultados. Busque el número de iteraciones necesarias para tener una precisión de $10^{-4}$.

BUSCO RESOLVER $f(x) = x^3 - 25 = 0$

REQUISITOS:

→ BISECCION → NO HAY ✓

→ PTO. FIJO:

$g(x) = x - \lambda(x^3 - 25)$ $\quad \lambda = \frac{1}{27}$

PROPONGO $I = [2,3]$ QUE SE QUE $P \in I$ $\quad \underbrace{2^3}_{8} < \underbrace{P}_{\sqrt[3]{25}} < \underbrace{3^3}_{27}$

$g(2) \approx 2{,}6 \quad g(3) \approx 2{,}8 \Rightarrow g(I) \subset I$ ✓

$g'(x) = 1 - \frac{3}{27}x^2 = 1 - \frac{1}{9}x^2$

$|g'(2)| = 0{,}5 \quad |g'(3)| = 0 \Rightarrow |g'(x)| < 1$ ✓

ENTONCES $g$ ES VALIDA EN $I$

CANTIDAD TEORICA DE ITERACIONES:

BISECCION

$\frac{\overset{3}{b} - \overset{2}{a}}{2^n} < 10^{-4} \Rightarrow \frac{1}{2^n} < \frac{1}{10000} \Rightarrow 10000 < 2^n \Rightarrow n > \log_2(10000) \approx 13{,}28$

$\Rightarrow n \geq 14$

PTO FIJO $(P_0 = 2{,}5)$

$\underbrace{K^n}_{\max(|g'|)^n = (0{,}5)^n} \cdot \max\{P_0 - a, b - P_0\} = (0{,}5)^n \cdot \max\{0{,}5, 0{,}5\} = \frac{1}{2^{n+1}} < \frac{1}{10000}$

$\Rightarrow 10000 < 2^{n+1}$

$\Rightarrow n > \log_2(10000) - 1 \approx 12{,}28$

$n \geq 13$

RESULTADOS EMPIRICOS (PYTHON)

```
(2.875, 3)
(np.float64(2.9237199593977308), 3)
```

CON 3 ITERS MAX

```
(2.9240188598632812, 17)
(np.float64(2.924016991854739), 5)
```

SIN LIMITE DE ITERS

⇒ REFLEXION: PTO. FIJO GOAT.

2.14 Sea $g \in C^1[a,b]$ y $p \in (a,b)$ con $g(p) = p$ y $|g'(p)| > 1$. Muestre que existe un $\delta > 0$ tal que si $0 < |p_0 - p| < \delta$, entonces $|p_0 - p| < |p_1 - p|$. Es decir, no importa que tan cerca esté $p_0$ de $p$, la siguiente iteración $p_1$ siempre estará más lejos, por lo que el método de iteración de punto fijo no converge si $p_0 \neq p$.

## Enunciado

Sea $g \in C^1[a,b]$ y $p \in (a,b)$ tal que $g(p) = p$ y $|g'(p)| > 1$. Probar que existe $\delta > 0$ tal que, si $0 < |p_0 - p| < \delta$ y $p_1 = g(p_0)$, entonces

$$|p_1 - p| > |p_0 - p|.$$

En particular, el método de punto fijo $p_{n+1} = g(p_n)$ no converge a $p$ si $p_0 \neq p$.

FUE.

---

## Prueba

1. **Elección de una cota estricta $c$ por debajo de $|g'(p)|$.**
   Como $|g'(p)| > 1$, existe un número $c$ tal que
   $$1 < c < |g'(p)|. \tag{1}$$
2. **Continuidad de $g'$ ⇒ control local del módulo.**
   Dado $\varepsilon := |g'(p)| - c > 0$, por continuidad de $g'$ en $p$ existe $\delta > 0$ tal que
   $$|x - p| < \delta \quad \Rightarrow \quad |g'(x) - g'(p)| < \varepsilon.$$
   Entonces, para todo $x$ con $|x - p| < \delta$,
   $$|g'(x)| \;\geq\; |g'(p)| - |g'(x) - g'(p)| \;>\; |g'(p)| - \varepsilon \;=\; c, \tag{2}$$
   donde usamos la desigualdad triangular y la definición de $\varepsilon$.
3. **Aplicación del Teorema del Valor Medio (TVM).**
   Sea $p_0$ con $0 < |p_0 - p| < \delta$ y defínase $p_1 := g(p_0)$. Por el TVM aplicado a $g$ en el segmento $[p, p_0]$, existe $\xi$ entre $p$ y $p_0$ tal que
   $$g(p_0) - g(p) = g'(\xi)\,(p_0 - p).$$
   Como $g(p) = p$, resulta
   $$p_1 - p = g'(\xi)\,(p_0 - p).$$
   Tomando módulos y usando (2) (nótese que $|\xi - p| < \delta$),
   $$|p_1 - p| = |g'(\xi)|\,|p_0 - p| \;\geq\; c\,|p_0 - p|.$$
   Por (1), $c > 1$; luego
   $$\boxed{|p_1 - p| > |p_0 - p|}.$$

2.15 Encuentre una función $g$ definida en $[0,1]$ que no cumple ninguna de las hipótesis del teorema de punto fijo pero igual tiene un único punto fijo en $[0,1]$.

Si $g(x) = x + \sqrt{x}$ $\rightarrow$ TIENE PUNTO FIJO $g(0) = 0$

PERO:

$g(I) \not\subset I \Rightarrow g(1/2) \approx 1{,}2$ $\times$

$g'(x) = 1 + \frac{1}{2\sqrt{x}}$ $\rightsquigarrow$ $|g'(x)|$ NO ES $< 1$ EN $[0,1]$

ADEMAS EN 0 NO ESTA DEFINIDA.

FIN.

2.16 Utilice el método de Newton para encontrar soluciones con de los siguientes problemas

a) $e^x + 2^{-x} + 2\cos x - 6 = 0$, con $x \in [1, 2]$,

b) $\ln(x-1) + \cos(x-1) = 0$, con $x \in [1{,}3, 2]$,

c) $(x-2)^2 - \ln x = 0$, con $x \in [1, 2]$.

Primero resuelva un par de iteraciones a mano, partiendo de algún $p_0$ dentro de los intervalos, y luego implemente una función de *python* para converger con una tolerancia de $10^{-5}$.

a) $f(x) = e^x - 2^{-x} + 2\cos(x) - 6$

$f'(x) = e^x + \ln(2)2^{-x} - 2\,\mathrm{sen}(x)$

$x \in [1,2]$

```
(np.float64(1.9903464716135102), 5)
```

b) $f(x) = \ln(x-1) + \cos(x-1)$

$f'(x) = \frac{1}{x-1} - \mathrm{sen}(x-1)$

$x \in [1{,}3, 2]$

```
(np.float64(1.3977478369912424), 4)
```

c) $f(x) = (x-2)^2 - \ln(x)$

$f'(x) = 2(x-2) - \frac{1}{x}$

$x \in [1,2]$

```
(np.float64(1.4123911717250053), 4)
```

2.17 La ecuación $x^2 - 10\cos x = 0$ tiene dos soluciones $\pm 1{,}3793646$. Utilice el método de Newton para aproximar las soluciones con los siguientes $p_0$

a) $p_0 = -100$,

b) $p_0 = -50$,

c) $p_0 = -25$,

d) $p_0 = 25$,

e) $p_0 = 50$,

f) $p_0 = 100$.

BUENO...

$f(x) = x^2 - 10\cos(x) \Rightarrow f'(x) = 2x + 10\,\text{sen}(x)$

```
p = -100: (np.float64(-1.3793702695076215), 8)
p = -50: (np.float64(-1.3793645982150824), 7)
p = -25: (np.float64(1.379368417659703), 7)
p = 25: (np.float64(-1.379368417659703), 7)
p = 50: (np.float64(1.3793645982150824), 7)
p = 100: (np.float64(1.3793702695076215), 8)
```

2.18 La función $f(x) = \ln(x^2+1) - e^{0,4x}\cos \pi x$ tiene un número infinito de ceros.

a) Determine, con precisión de $10^{-6}$, el único cero negativo.

b) Determine, con precisión de $10^{-6}$, los cuatro ceros positivos más pequeños.

c) Determine una aproximación inicial razonable para encontrar el n-ésimo cero positivo más pequeño.

d) Utilice el resultado del inciso anterior para encontrar el 25avo cero positivo más pequeño.

$$f(x) = \ln(x^2+1) - e^{\frac{2}{5}x} \cdot \cos(\pi x)$$

$$f'(x) = \frac{2x}{x^2+1} + \left[-\frac{2}{5}e^{\frac{2}{5}x} \cdot \cos(\pi x) + e^{\frac{2}{5}x}\pi \operatorname{sen}(\pi x)\right]$$

a)

ARRANQUE EN -2

```
(np.float64(-0.43414304735877096), 6)
```

b)

```
Solution 1: 0.4506567715410421
Solution 2: 1.7447380525331295
Solution 3: 2.2383197951552014
Solution 4: 3.709041201369113
```

c y d) NOSE QUE ESTIMACION HACEN, ACA TODO FUERZA BRUTA:

```
Solution 1: 0.4506567715410421
Solution 2: 1.7447376141846922
Solution 3: 2.2383198922259675
Solution 4: 3.709041201257064
Solution 5: 4.322649939810808
Solution 6: 5.619935330386819
Solution 7: 6.40693361506305
Solution 8: 7.563210545449977
Solution 9: 8.453481053968527
Solution 10: 9.531834402763371
Solution 11: 10.4773058872284
Solution 12: 11.515571386935454
Solution 13: 12.48910683607646
Solution 14: 13.50747153200588
Solution 15: 14.494831051000123
Solution 16: 15.50353998571139
Solution 17: 16.49756833755037
Solution 18: 18.498863627834528
Solution 19: 20.499471563815813
Solution 20: 21.500359925088713
Solution 21: 22.499755466520227
Solution 22: 23.50016633648808
Solution 23: 28.499976120376346
Solution 24: 30.49998906117503
Solution 25: 32.49999550436371
```

25vo CERO POSITIVO MAS CHICO.

2.19 Utilice el método de Newton para resolver la ecuación

$$0 = \frac{1}{2} + \frac{1}{4}x^2 - x \sin x - \frac{1}{2}\cos 2x,$$

con $p_0 = \pi/2$. ¿Algo le resultado llamativo? Repita con $p_0 = 5\pi$ y $p_0 = 10\pi$.

$f(x) = \frac{1}{2} + \frac{1}{4}x^2 - x\,\text{sen}(x) - \frac{1}{2}\cos(2x)$

$f'(x) = \frac{1}{2}x - \text{sen}(x) - x\cos(x) + \text{sen}(2x)$

```
p0 = pi/2: (np.float64(1.895492805077969), 18)
p0 = pi*5: (np.float64(1.8954929506373934), 22)
p0 = pi*10: (np.float64(-1.512438727979111e-06), 629)
```

RARO ESTO, ME MOVI MAS A LA DERECHA PERO ENCONTRE UN CERO MÁS A LA IZQUIERDA.

2.20 La ecuación de anualidades ordinaria

$$A = \frac{P}{i}[1 - (1+i)^{-n}],$$

se utiliza para calcular cuanto se debe pagar de una hipoteca en un período fijo de tiempo. $A$ es el monto de la hipoteca, $P$ es el monto de cada pago, e $i$ es la tasa de interés por período por los $n$ períodos de pago. Suponga que es necesaria una hipoteca a 30 años con un valor de \$135000 y que el prestatante puede pagar como máximo \$1000 por mes. ¿Cuál es el interés máximo que puede pagar el prestatante?

$30 \cdot 12 = 360$

$$A = \frac{P}{i}\left[1-(1+i)^{-n}\right] \overset{\text{TENGO}}{\Longrightarrow} 135000 = \frac{1000}{i}\left[1-(1+i)^{-360}\right]$$

$$\rightsquigarrow f(i) = \frac{1000}{i}\left[1-(1+i)^{-360}\right] - 135000$$

$$\rightsquigarrow f'(i) = -\frac{1000}{i^2}\left[1-(1+i)^{-360}\right] + \frac{1000}{i}\left[360(1+i)^{-361}\right]$$

USO METODO DE NEWTON.

```
p0 = 1e-6: (np.float64(0.006749429094715076), 5)
```

0,67% INTERES MENSUAL.

2.21 La ecuación de iteración para el método de la secante se puede escribir de la siguiente forma

$$p_n = \frac{f(p_{n-1})p_{n-2} - f(p_{n-2})p_{n-1}}{f(p_{n-1}) - f(p_{n-2})}.$$

Explique por qué, en general, es probable que esta ecuación sea menos precisa que la vista en clase.

DE LA CLASE SABEMOS QUE:

$$P_n = P_{n-1} - \frac{f(P_{n-1})}{f'(P_{n-1})}$$

TAMBIÉN: $f'(P_{n-1}) = \dfrac{f(P_{n-1}) - f(P_{n-2})}{P_{n-1} - P_{n-2}}$

REEMPLAZO:

$$\Rightarrow P_n = P_{n-1} - f(P_{n-1})\left[\frac{P_{n-1} - P_{n-2}}{f(P_{n-1}) - f(P_{n-2})}\right]$$

$a = P_{n-1}$
$b = P_{n-2}$

$= \cancel{af(a)} - af(b) - \cancel{af(a)} + bf(a)$

$$\Rightarrow P_n = \frac{P_{n-1}\left(f(P_{n-1}) - f(P_{n-2})\right) - \left(P_{n-1} - P_{n-2}\right)f(P_{n-1})}{f(P_{n-1}) - f(P_{n-2})}$$

$$\Rightarrow P_n = \frac{f(P_{n-1})P_{n-2} - f(P_{n-2})P_{n-1}}{f(P_{n-1}) - f(P_{n-2})}$$

EQUIVALENTE PERO MENOS PRECISA POR MAYOR CANTIDAD DE OPERACIONES, CANCELACION CATASTROFICA EN EL NUMERADOR.

2.22 Derive la siguiente formula de error para el método de Newton

$$|p - p_{n+1}| \leq \frac{M}{2|f'(p_n)|}|p - p_n|^2,$$

asumiendo que las hipótesis del teorema visto en clase se cumplen, que $|f'(p_n)| \neq 0$ y que $M = \max |f''(x)|$. Ayuda: utilice la derivación del método de Newton a partir de la serie de Taylor de $f(p)$ vista en clase.

HAGO EXPANSION DE TAYLOR DE $f$ EN $P$.

$$f(P) = f(P_n) + (P-P_n) f'(P_n) + \frac{f''(c)(P-P_n)^2}{2}$$

COMO $P$ ES RAIZ $\rightarrow f(P) = 0$

$$f'(P_n)(P-P_n) = -f(P_n) - \frac{1}{2} f''(c)(P-P_n)^2$$

$$P - P_n = -\frac{f(P_n)}{f'(P_n)} - \frac{1}{2}\frac{f''(c)}{f'(P_n)}(P-P_n)^2$$

$$P = \underbrace{\boxed{P_n - \frac{f(P_n)}{f'(P_n)}}}_{\text{PASO DE NEWTON RAPHSON } (P_{n+1})} - \frac{1}{2}\frac{f''(c)}{f'(P_n)}(P-P_n)^2$$

$$\Rightarrow \quad P - P_{n+1} = -\frac{f''(c)}{2 f'(P_n)}(P-P_n)^2$$

APLICO MODULO

$$\Rightarrow \quad |P - P_{n+1}| \leq \frac{\overset{\max(|f''(c)|)}{M}}{2|f'(P_n)|}|P-P_n|^2$$

SALE LA INECUACION CUANDO PASO DE USAR $f''(c)$ a $\max(|f''(c)|)$

2.23 Dada $F : \mathbb{R}^n \to \mathbb{R}^n$ el método Newton-Raphson generalizado consiste en realizar la iteración vectorial

$$x_{k+1} = x_k - (DF|_{x_k})^{-1} F(x_k),$$

donde $x_k \in \mathbb{R}^n$ y $(DF|_{x_k})^{-1}$ es la inversa de la matriz diferencial de $F$ evaluada en $x_k$ . Usar Newton-Raphson generalizado para hallar una raíz de la función

$$F(x_1, x_2) = \begin{bmatrix} x_1 + x_2 - 3 \\ x_1^2 + x_2^2 - 9 \end{bmatrix}.$$

$$F(x) = \begin{pmatrix} x_1 + x_2 - 3 \\ x_1^2 + x_2^2 - 9 \end{pmatrix}$$

$$\rightsquigarrow F'(x) = \begin{pmatrix} 1 & 1 \\ 2x_1 & 2x_2 \end{pmatrix}$$

```
(array([-1.77021204e-16,  3.00000000e+00]), 7)
```

CORTA

2.24 Muestre que las siguientes secuencias convergen sublinealmente a $p = 0$. ¿Qué tan grande debe ser $n$ para que $|p_n - p| \leq 5 \times 10^{-2}$?

a) $p_n = \frac{1}{n}$, $n \geq 1$,

b) $p_n = \frac{1}{n^2}$, $n \geq 1$.

SEGUN GPT PARA QUE CONVERJA SUBLINEALMENTE TIENE QUE PASAR:

$$\Rightarrow \lim_{n\to\infty} : \frac{|p_{n+1} - \overset{=0}{p}|}{|p_n - \underset{=0}{p}|} = 1$$

a) $\rightarrow \lim_{n\to\infty} : \frac{1}{n+1} \cdot n \overset{\text{L'HOPITAL}}{=} \lim_{n\to\infty} : 1 = 1$ ✓

b) $\sim \lim_{n\to\infty} : \frac{1}{(n+1)^2} \cdot n^2 \overset{\text{L'HOPITAL}\times 2}{=} \lim_{n\to\infty} : \frac{2}{2} = 1$ ✓

AHORA BUSCAMOS $n$. $|p_n - \overset{=0}{p}| \leq 5 \cdot 10^{-2}$

a) $\left|\frac{1}{n}\right| = \frac{1}{n} \leq \frac{5}{100} \Rightarrow 100 \leq 5n \Rightarrow n \geq 20$

b) $\left|\frac{1}{n^2}\right| = \frac{1}{n^2} \leq \frac{5}{100} \Rightarrow 100 \leq 5n^2 \Rightarrow n^2 \geq 20 \Rightarrow n \geq \overset{4{,}47}{\sqrt{20}} \rightsquigarrow n \geq 5$

2.25 Se puede demostrar que si $\{p_n\}_{n=1}^{\infty}$ es una secuencia dada por el método de la secante que converge a $p$, la solución de $f(x)=0$, entonces existe una constante $C$ tal que

$$|p_{n+1}-p| \approx C|p_n-p||p_{n-1}-p|.$$

Asuma que $\{p_n\}$ converge con orden $\alpha$, es decir, que

$$\lim_{n\to\infty} \frac{|p_{n+1}-p|}{|p_n-p|^{\alpha}} = \lambda,$$

con $\lambda > 0$ una constante, utilice esta información para mostrar que $\alpha = \frac{1+\sqrt{5}}{2}$. *Ayuda: la suposición sobre el orden de convergencia quiere decir que puede tomar* $|p_{n+1}-p| \approx \lambda|p_n-p|^{\alpha}$.

$\{P_n\}$ CONVERGE CON ORDEN $\alpha$:

$$\lim_{n\to\infty} : \frac{|P_{n+1}-P|}{|P_n-P|^{\alpha}} = \lambda$$

PUEDO TOMAR A $|P_{n+1}-P| = e_{n+1}$

AYUDA

$\rightsquigarrow$ $e_{n+1} = \lambda e_n^{\alpha}$ y $e_n = \lambda e_{n-1}^{\alpha}$ $\Rightarrow e$

PERO TAMBIEN: $e_{n+1} \approx C e_n e_{n-1}$

$\Downarrow$ $e_{n+1} \approx C\lambda e_{n-1}^{\alpha} e_{n-1}^{1}$

$\Rightarrow \lambda e_n^{\alpha} \approx C\lambda e_{n-1}^{\alpha+1}$

$= \lambda[\lambda e_{n-1}^{\alpha}]^{\alpha} = \lambda\lambda^{\alpha}(e_{n-1}^{\alpha})^{\alpha} = \lambda^{\alpha+1} e_{n-1}^{\alpha^2}$

$\Rightarrow \lambda^{\alpha+1} e_{n-1}^{\alpha^2} \approx C\lambda e_{n-1}^{\alpha+1}$

PROPONGO $C = \lambda^{\alpha}$ CTE

$\Rightarrow \cancel{\lambda^{\alpha+1}} e_{n-1}^{\alpha^2} \approx \cancel{\lambda^{\alpha}\lambda} e_{n-1}^{\alpha+1}$

$\Rightarrow \alpha^2 \approx \alpha + 1 \Rightarrow \alpha^2 - \alpha - 1 \approx 0$

$\alpha_1 = \frac{1+\sqrt{5}}{2}$

$\cancel{\alpha_2 = \frac{1-\sqrt{5}}{2}}$ NEGATIVO

3.1 Sea $x_0 = 0$, $x_1 = 0{,}6$ y $x_2 = 0{,}9$. Construya interpolaciones de primer o segundo grado para aproximar $f(0{,}45)$ dadas las siguientes funciones

a) $f(x) = \cos x$,

b) $f(x) = \sqrt{1+x}$,

c) $f(x) = \ln(x+1)$,

Encuentre cotas para el error de las aproximaciones.

$$P_n(x) = \sum_{i=0}^{n} f(x_i) L_i(x) \quad \text{con} \quad L_i(x) = \prod_{\substack{j=0 \\ i \neq j}}^{n} \frac{x - x_j}{x_i - x_j}$$

Poli de Lagrange

$$f(x) - P_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{i=0}^{n} (x - x_i) \Rightarrow |f(x) - P_n(x)| \leq \frac{\mathrm{MAX}(|f^{(n+1)}(t)|)}{(n+1)!} \prod_{i=0}^{n} |x - x_i|, \quad t \in [a,b]$$

Cotas de error

---

a) $\Rightarrow$

$$P_2 = \sum_{i=0}^{2} \cos(x_i) \prod_{\substack{j=0 \\ i \neq j}}^{2} \frac{x - x_j}{x_i - x_j} = \underbrace{\cos(0) \frac{x - 0{,}6}{-0{,}6} \frac{x - 0{,}9}{-0{,}9}}_{i=0} + \underbrace{\cos(0{,}6) \frac{x}{0{,}6} \frac{x - 0{,}9}{0{,}6 - 0{,}9}}_{i=1} + \underbrace{\cos(0{,}9) \frac{x}{0{,}9} \frac{x - 0{,}6}{0{,}9 - 0{,}6}}_{i=2} = P_3(x)$$

Python

$\sim P_2(0{,}45) \approx 0{,}8981$ Con poli de grado 1 $\Rightarrow \approx 0{,}869$

Cota de error (Poli de 2do orden) $\Rightarrow$

$$|f(x) - P_2(x)| \leq \frac{M_{n+1}}{6} |x (x - 0{,}6) \underbrace{(x - 0{,}9)}|, \quad M_{n+1} = \mathrm{MAX}(|f^{(n+1)}(t)|)$$

Esta cota es igual para a) b) y c), solo cambia $M_{n+1}$

Esto en el error de primer orden no está

b) $P_2(0{,}45) \approx 1{,}203$

c) $P_3(0{,}45) \approx 0{,}368$

Fue voy a calcular errores.

```python
# Ej 3.1

def productorio(X, i, x):
    n = len(X)
    res = [(x - X[j])/(X[i] - X[j]) for j in range(n) if j != i]
    return np.prod(res)

def poli_lagrange(f, X, x):
    n = len(X)
    return sum([f(X[i]) * productorio(X, i, x) for i in range(n)])

X = [0, 0.6, 0.9]

def f1(x):
    return np.cos(x)

def f2(x):
    return np.sqrt(1 + x)

def f3(x):
    return np.log(1 + x)

print(poli_lagrange(f1, X, 0.45))
print(poli_lagrange(f2, X, 0.45))
print(poli_lagrange(f3, X, 0.45))
```

✓ 0.0s

```
0.898100074705722
1.2034237282735152
0.36829061135835384
```

$$(\cos(x))''' = \mathrm{Sen}(x) \Rightarrow \mathrm{MAX}(|\mathrm{Sen}(x)|) = \mathrm{Sen}(0{,}9) \approx 0{,}7833$$

a) $|f(x) - P_2(x)| \leq \frac{1}{6} |0{,}45(0{,}45-0{,}6)(0{,}45-0{,}9)| 0{,}7833 \approx 0{,}00397$

b) $(\sqrt{1+x})''' = \frac{3}{8}(x+1)^{-1} \Rightarrow \mathrm{MAX}(|\dots|) = \frac{3}{8}$

$\Rightarrow |f(x) - P_2(x)| \leq \frac{1}{6} \cdot \frac{3}{8} |0{,}45(0{,}45-0{,}6)(0{,}45-0{,}9)| \approx 0{,}001875$

c) $(\ln(1+x))''' = \frac{2}{(1+x)^3} \Rightarrow \mathrm{MAX}(|\dots|) = 2$

$\Rightarrow |f(x) - P_2(x)| \leq \frac{1}{6} \cdot 2 |0{,}45(0{,}45-0{,}6)(0{,}45-0{,}9)| \approx 0{,}01$

3.2 Use los interpolinomios de Lagrange apropiados de grado 1, 2 y 3 para aproximar lo siguiente

a) $f(8{,}4)$ con $f(8{,}1) = 16{,}94410$, $f(8{,}3) = 17{,}56492$, $f(8{,}6) = 18{,}50515$, $f(8{,}7) = 18{,}82091$.

b) $f(-1/3)$ con $f(-0{,}75) = -0{,}0718125$, $f(-0{,}5) = -0{,}02475$, $f(-0{,}25) = 0{,}33493750$, $f(0) = 1{,}101$.

c) $f(0{,}25)$ con $f(0{,}1) = 0{,}62049958$, $f(0{,}2) = -0{,}28398668$, $f(03) = 0{,}00660095$, $f(0{,}4) = 0{,}2484244$.

```
Polinomios de Lagrange grado 1,2,3
a)
Grado 1: 17.8771425
Grado 2: 17.877129999999998
Grado 3: 17.875329999999998
b)
Grado 1: -0.19184722222222228
Grado 2: 0.006625000000000006
Grado 3: -0.0718125
c)
Grado 1: -0.21033722187499995
Grado 2: -0.2880771012499999
Grado 3: -0.7362298099999999
```

3.3 Los datos del ejercicio anterior fueron generados con las siguientes funciones. Use la formula del error para encontrar una cota a éste y compárelo con el error obtenido en los casos $n = 1$ y $n = 2$.

a) $f(x) = x \ln x$,

b) $f(x) = x^3 + 4{,}001x^2 + 4{,}002x + 1{,}101$,

c) $f(x) = x \cos x - 2x^2 + 3x - 1$.

a)

$f(x) = x\ln(x) \qquad f'(x) = \ln(x) + 1 \qquad \overset{0{,}12345}{f''(x) = \frac{1}{x}} \qquad \overset{MAX \approx 0{,}001881}{f'''(x) = -\frac{1}{x^2}} \qquad \overset{MAX \approx 0{,}00376}{f^{IV}(x) = \frac{2}{x^3}}$

COTA $n=3 \Rightarrow \leq \frac{0{,}00376}{4!}\left|(8{,}4-8{,}1)(8{,}4-8{,}3)(8{,}4-8{,}6)(8{,}4-8{,}7)\right| \approx 2{,}82 \times 10^{-7}$

COTA $n=2 \Rightarrow \leq \frac{0{,}001881}{3!}\left|(8{,}4-8{,}1)(8{,}4-8{,}3)(8{,}4-8{,}6)\right| \approx 1{,}881 \times 10^{-6}$

COTA $n=1 \Rightarrow \leq \frac{0{,}12345}{2!}\left|(8{,}4-8{,}1)(8{,}4-8{,}3)\right| \approx 1{,}85185 \times 10^{-3}$

UNA PORONGA EL DE $n=1$.

b)

$f(x) = \ldots \qquad f'(x) = 3x^2 + 8{,}002x + 4{,}002 \qquad \overset{MAX = 12{,}502}{f''(x) = 6x + 8{,}002} \qquad \overset{MAX = 6}{f'''(x) = 6} \qquad \overset{MAX = 0}{f^{IV}(x) = 0}$

COTA $n=3 \Rightarrow \leq 0$ EXACTO

COTA $n=2 \Rightarrow \leq \frac{6}{3.2}\left|\left(-\frac{1}{3}+0{,}5\right)\left(-\frac{1}{3}+0{,}25\right)\left(-\frac{1}{3}\right)\right| \approx 0{,}004629$

COTA $n=1 \Rightarrow \leq \frac{12{,}502}{2}\left|\left(-\frac{1}{3}+0{,}5\right)\left(-\frac{1}{3}+0{,}25\right)\right| \approx 0{,}0868194$

c) $f(x) = \ldots \qquad f'(x) = \cos(x) - x\operatorname{sen}(x) - 4x + 3 \qquad \overset{MAX \approx 5{,}14726108 22 \; f''(0{,}4)}{f''(x) = -2\operatorname{sen}(x) - x\cos(x) - 4} \qquad \overset{MAX \approx 2{,}97502915 42 \; f'''(0{,}1)}{f'''(x) = -3\cos(x) + x\operatorname{sen}(x)} \qquad \overset{MAX \approx 1{,}9260977668 \; (f^{IV}(0{,}4))}{f^{IV}(x) = 4\operatorname{sen}(x) + x\cos(x)}$

COTA $n=3 \Rightarrow \leq \frac{1{,}926097\ldots}{4!}\left|(0{,}25-0{,}1)(0{,}25-0{,}2)(0{,}25-0{,}3)(0{,}25-0{,}4)\right| \approx 4{,}51429 \times 10^{-6}$

COTA $n=2 \Rightarrow \leq \frac{2{,}975\ldots}{3!}\left|(0{,}25-0{,}1)(0{,}25-0{,}2)(0{,}25-0{,}3)\right| \approx 0{,}000185938$

COTA $n=1 \Rightarrow \leq \frac{5{,}147\ldots}{2!}\left|(0{,}25-0{,}2)(0{,}25-0{,}3)\right| \approx 0{,}00643375$

3.4 Escriba una función en *python* para realizar interpolación de Lagrange. La misma debería tomar como argumento los nodos (puntos de evaluación), y la función evaluada en los nodos.

Aplique a las siguientes funciones, evaluando en una grilla equiespaciada entre $x_0$ y $x_2$, y grafique el *ground truth* junto con la aproximación, así como el error.

a) $f(x) = e^{-2x}\sin 3x$, $x_0 = 0$, $x_1 = \pi/6$, $x_2 = \pi/4$, $n = 2$,

b) $f(x) = \log_{10} x$, $x_0 = 3{,}0$, $x_1 = 3{,}2$, $x_2 = 3{,}5$, $n = 2$,

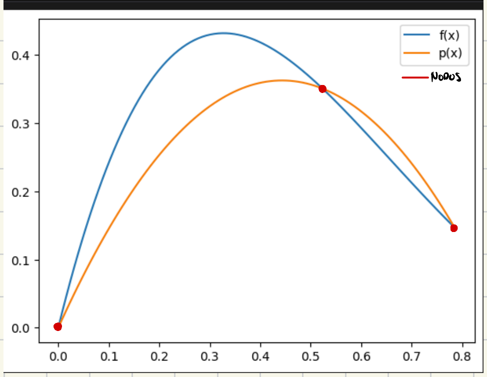
c) $f(x) = e^{x} + e^{-x}$, $x_0 = -0{,}3$, $x_1 = 0$, $x_2 = 0{,}3$, $n = 2$,

Encuentre una cota para el error absoluto en el intervalo $[x_0, x_n]$. Compare esta cota con el error obtenido.

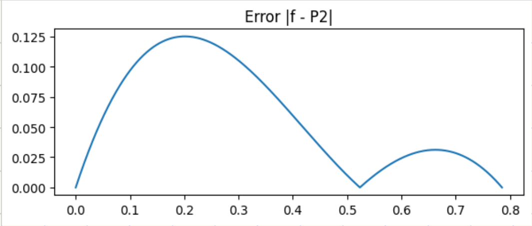
CLARO CLARO RECIÉN EN EL 3.4 PELOTUDO.

c)

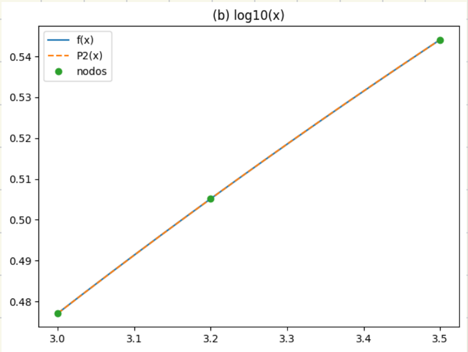
```python
def f1(x):
    return np.exp(-2*x)*np.sin(3*x)

vals1 = np.linspace(0, np.pi/4, 100)

res1_f = [f1(x) for x in vals1]
res1_p = [poli_lagrange(f1, [0, np.pi/6, np.pi/4], x) for x in vals1]

plt.plot(vals1, res1_f, label="f(x)")
plt.plot(vals1, res1_p, label="p(x)")
plt.legend()
plt.show()
```
✓ 0.0s



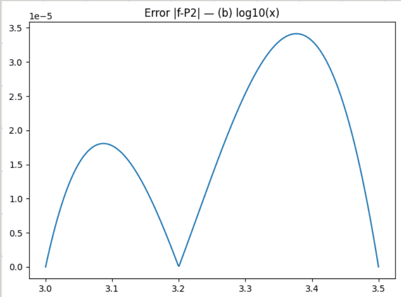

$$\mathcal{E}_{ABS} = |f(x) - P(x)| \le \frac{MAX(|f'''(t)|)}{6}\left|x(x-\pi/6)(x-\pi/4)\right|$$



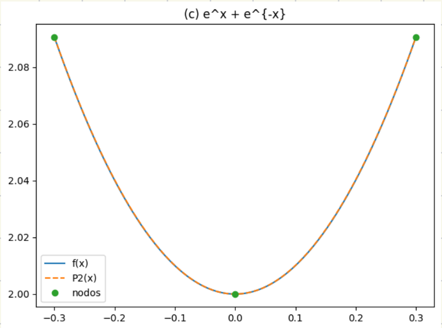

b)



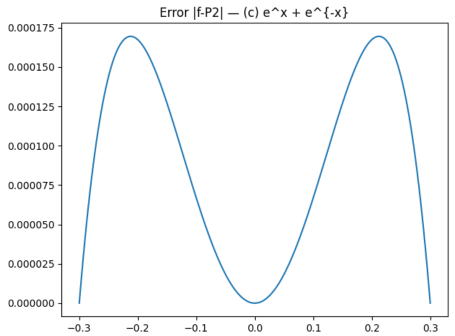

$$\Rightarrow \mathcal{E}_{ABS} \le \frac{MAX(|f'''(t)|)}{6}\left|(x-3)(x-3{,}2)(x-3{,}5)\right|$$




c)




$$\Rightarrow \mathcal{E}_{ABS} \le \frac{MAX(|f'''(t)|)}{6}\left|(x-0{,}3)x(x+0{,}3)\right|$$



3.5 Sea $f(x) = e^x$ con $x \in [0, 2]$.

a) Aproxime $f(0{,}25)$ utilizando interpolación lineal con $x_0 = 0$ y $x_1 = 0{,}5$.

b) Aproxime $f(0{,}75)$ utilizando interpolación lineal con $x_0 = 0{,}5$ y $x_1 = 1$.

c) Aproxime $f(0{,}25)$ y $f(0{,}75)$ utilizando el segundo polinomio interpolador con $x_0 = 0$, $x_1 = 1$, $x_2 = 2$.

d) ¿Cuál aproximación es mejor y por qué?

$e^{0,25} \approx 1{,}28403 \qquad e^{0,75} \approx 2{,}117$

```
f(0.25) Lineal
1.324360635350064
```

a)

```
f(0.75) Lineal
2.183501549579587
```

b)

```
f(0.25) Cuadratico
1.1527742906760838

f(0.75) Cuadratico
2.0119152049056064
```

c)

SE PUEDE VER QUE a) y b) SON MEJORES APROXIMACIONES YA QUE ESTAN EN UN INTERVALO MAS ACOTADO DE LO QUE SE BUSCA APROXIMAR.

EJ CON $f(0{,}25)$

$$\varepsilon_1 \leq \frac{M_2}{2} \cdot \left| (0{,}25)(0{,}25-0{,}5) \right| \approx \frac{1}{32} M_2$$

$$\varepsilon_2 \leq \frac{M_3}{6} \cdot \left| (0{,}25)(0{,}25-1)(0{,}25-2) \right| \approx \frac{21}{64} \cdot \frac{1}{6} M_3$$

ACA $f''(x) = f'''(x) = e^x$

$M_2 = e^{0,5}$

$M_3 = e^2$

$\Rightarrow$ $\varepsilon_1 \leq 0{,}051$

$\varepsilon_2 \leq 0{,}404$

$\varepsilon_1$ ES MEJOR.

3.6 Suponé que necesitás construir tablas con ocho cifras decimales para la función logaritmo común, o en base 10, desde $x = 1$ hasta $x = 10$, de manera que la interpolación lineal sea precisa dentro de un margen de error de $10^{-6}$. Determina un límite para el tamaño del paso en esta tabla. ¿Qué elección de tamaño de paso harías para asegurar que $x = 10$ esté incluido en la tabla?

ARRANCO CON EL ERROR DEL INTERPOLADOR LINEAL

$$|f(x) - P_1(x)| \leq \underbrace{\frac{M_2}{2} \overbrace{|(x-a)(x-b)|}^{\text{MÁXIMO DE ESTO}}}$$

USO $h = b - a$ ANCHO DEL INTERVALO

$m = \frac{b+a}{2}$ PUNTO MEDIO

$$x^2 - x(a+b) + ab \Rightarrow x^2 - 2mx + ab + m^2 - m^2 \Rightarrow (x^2 - 2mx + m^2) + (ab - m^2)$$

$$ab - m^2 = ab - \frac{a^2 + 2ab + b^2}{4} = -\frac{(b-a)^2}{4} = -\frac{h^2}{4}$$

$$\Rightarrow (x^2 - 2mx + m^2) - \frac{1}{4}h^2$$

$$\Rightarrow \underbrace{(x-m)^2} - \frac{1}{4}h^2$$

$$|x - m| \leq \frac{h}{2}$$

LA DISTANCIA AL PTO. MEDIO ES SIOSI $\leq$ A LA MITAD DE TAMAÑO DEL INTERVALO

$$\Rightarrow \left|(x-m)^2 - \frac{1}{4}h^2\right|$$

ESTO TOMA MÁXIMO EN $x = m$. $\Rightarrow \frac{1}{4}h^2$

$$\Rightarrow |f(x) - P_1(x)| \leq \frac{M^2}{8} \cdot h^2$$

$$M_2 = \mathrm{MAX}(|f''(\xi)|) \Rightarrow |\log_{10}(x)''| = \frac{1}{x^2 \ln(10)} \Rightarrow \underset{[1,10]}{\mathrm{MAX}} = \overset{x=1}{1/\ln(10)}$$

ESTE ERROR ES EL MAYOR DE TODOS LOS POSIBLES $P_1(x)$ ENTONCES VALE PARA $P_1(x)$, YA QUE ESTE TENDRÍA UN ERROR MENOR

$$\Rightarrow |f(x) - P_1(x)| \leq \frac{h^2}{8\ln(10)} < 10^{-6}$$

$$\Rightarrow h < \sqrt{8 \cdot 10^{-6} \cdot \ln(10)} \approx 4{,}29 \cdot 10^{-3}$$

PARA QUE $x=10$ ESTE INCLUIDO EN MI TABLA DE NODOS UNIFORMES $x_k = 1 + kh$:

$$1 + kh = 10 \Rightarrow h = \frac{9}{k} \Rightarrow k = \frac{9}{h \approx 4{,}29 \cdot 10^{-3}} \Rightarrow k = 2098$$

$\Rightarrow$ EL $h$ MÍNIMO PARA QUE $x=10$ ESTE INCLUIDO ES $h = 9/2098 \approx 0{,}004289$

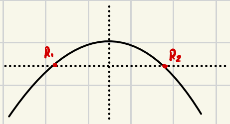
3.7 Muestre que $\max_{x_j \le x \le x_{j+1}} |g(x)| = h^2/4$ con $g(x) = (x - jh)(x - (j+1)h)$.

ASUMO $h$ y $J > 0$

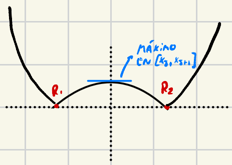
$g(x) = (x - Jh)(x - (J+1)h)$

$g(0) = (\cancel{-}Jh)(\cancel{-}(J+1)h) > 0$

CUADRATICA CON RAICES $Jh$ y $(J+1)h$




$\Rightarrow |g(x)|$




COMO ES SIMETRICA → EL MAXIMO ESTA EN EL PUNTO MEDIO: $\frac{x_J + x_{J+1}}{2} = \frac{Jh + (J+1)h}{2} = \frac{h(2J+1)}{2} = x_{medio}$

HAGO $g(x_{medio}) = \left(\frac{h(2J+1)}{2} - Jh\right)\left(\frac{h(2J+1)}{2} - h(J+1)\right)$

$= h\left[\frac{2J+1}{2} - J\right] h \left[\frac{2J+1}{2} - J - 1\right]$

$= \frac{h}{2} \cdot -\frac{h}{2} = -\frac{h^2}{4} \Rightarrow |\max(g(x))| = \frac{h^2}{4}$

3.8 Muestre que el polinomio que interpola los siguientes datos tiene grado 3: $f(-2) = 1$, $f(-1) = 4$, $f(0) = 11$, $f(1) = 16$, $f(2) = 13$, $f(3) = -4$.

TENGO $P(x) = ax^3 + bx^2 + cx + d$

$\rightarrow f(0) = d = 11$

$\Rightarrow f(x) = ax^3 + bx^2 + cx + 11$

$$\left.\begin{array}{l} P(1) = a+b+c+11 = 16 \\ P(-1) = -a+b-c+11 = 4 \\ P(-2) = -8a+4b-2c+11 = 1 \end{array}\right\}$$

$a = 5-b-c$

$\rightarrow -5+b+\cancel{c}+b-\cancel{c}+11 = 4$

$\Rightarrow 2b = -2 \Rightarrow b = -1$

$$\left.\begin{array}{l} a = 6-c \\ -8a-2c+7 = 1 \end{array}\right\}$$

$\rightarrow -48+6c+7 = 1$

$6c = 42 \Rightarrow c = 7$

$a = -1$

$\Rightarrow P(x) = -x^3 - x^2 + 7x + 11$

$P(2) = -8-4+14+11 = 13$ ✓

$P(3) = -27-9+21+11 = -4$ ✓

$P(x)$ PASA POR TODOS LOS PUNTOS Y ES DE GRADO 3.

3.9 Sea la función $f(x) = \dfrac{1}{1 + 25x^2}$ en el intervalo $[-1, 1]$ (función de Runge).

*a)* En *python* (empleando funciones propias o de librerías), construya el polinomio interpolador de Lagrange para $f(x)$ de grado $n$ usando varios valores de $n$ empleando:

a) $n + 1$ nodos equiespaciados.

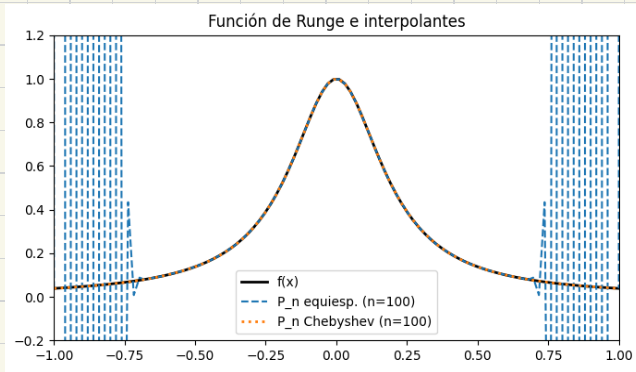
b) $n + 1$ nodos de Chebyshev de primer tipo dados por:

$$x_k = \cos\left(\frac{2k + 1}{2(n + 1)}\pi\right), \quad k = 0, 1, \ldots, n.$$

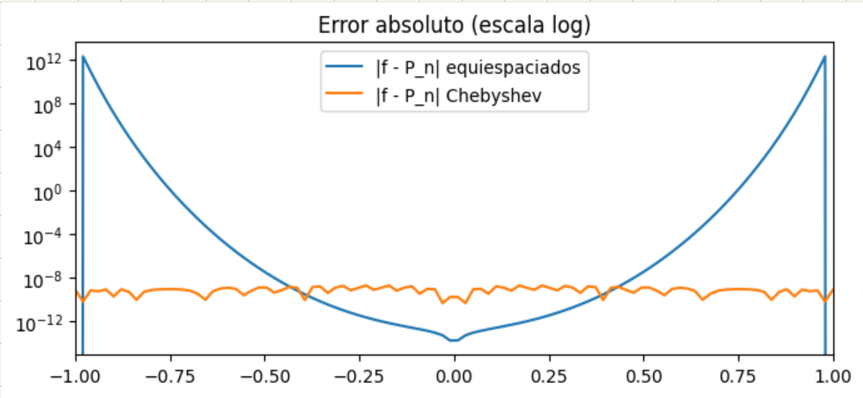
*b)* Graficar en cada caso la función $f(x)$ y el polinomio interpolador en el intervalo $[-1, 1]$.

*c)* Calcular y graficar el error absoluto $|f(x) - P_n(x)|$ en una malla fina de puntos y comparar ambas formas de colocación de nodos.

```python
def f(x):
    return 1 / (1 + (25*x*x))

def chevy_nodes(n):
    k = np.arange(n+1)
    x = np.cos((2*k + 1) * np.pi / (2*(n+1)))
    return np.sort(x)

n = 100

dom_a = np.linspace(-1, 1, n + 1)
dom_b = chevy_nodes(n)

inputs = np.linspace(-1, 1, n)

vals_real = [f(x) for x in inputs]

vals_equis_lagrange = [poli_lagrange(f, dom_a, x) for x in inputs]
vals_chevy_lagrange = [poli_lagrange(f, dom_b, x) for x in inputs]

# curvas a graficar
fx = [f(x) for x in inputs]

# curvas en la grilla común
fx = np.array(vals_real)
pe = np.array(vals_equis_lagrange)
pc = np.array(vals_chevy_lagrange)
```




c)



3.10 Utilice splines cúbicos para aproximar los siguientes valores y calcule el error asociado. Compare con el obtenido usando polinomio de Lagrange.

*a)* $f(x) = x \ln x$, con condiciones de borde çlampedüsando los valores de $f$ en $x_0 = 1$ y $x_1 = 2$, aproxime $f(1{,}5)$ y $f'(1{,}5)$.

*b)* $f(x) = \sin(e^x - 2)$, con condiciones de borde naturales usando los valores de $f$ en $x_0 = 0{,}7$, $x_1 = 0{,}8$ y $x_2 = 1{,}0$, aproxime $f(0{,}9)$ y $f'(0{,}9)$.

a)

$$S_1 = a_1 + b_1(x-1) + c_1(x-1)^2 + d_1(x-1)^3$$

$$S_1'(x) = b_1 + 2c_1(x-1) + 3d_1(x-1)^2$$

$$f(x) = x\ln(x)$$

$$f'(x) = \ln(x) + 1$$

$$S_1(1) = f(1) \Rightarrow a_1 = 0$$

$$S_1'(1) = f'(1) \Rightarrow b_1 = 1$$

$$S_1(2) = f(2) \Rightarrow 1 + c_1 + d_1 = 2\ln(2)$$

$$S_1'(2) = f'(2) \Rightarrow 1 + 2c_1 + 3d_1 = \ln(2) + 1$$

$$\begin{cases} c_1 + d_1 = 2\ln(2) - 1 & (1) \\ 2c_1 + 3d_1 = \ln(2) & (2) \end{cases}$$

$$(2) - 2(1) \Rightarrow d_1 = \ln(2) - 4\ln(2) + 2$$

$$\Rightarrow d_1 = -3\ln(2) + 2$$

$$\Rightarrow c_1 = 5\ln(2) - 3$$

$$\Rightarrow S_1(x) = (x-1) + [5\ln(2) - 3](x-1)^2 + [-3\ln(2) + 2](x-1)^3$$

$$\Rightarrow S_1'(x) = 1 + 2[5\ln(2) - 3] + 3[-3\ln(2) + 2](x-1)^2$$

LAGRANGE

$$P(x) = \frac{x-2}{1-2} + (2\ln(2))\frac{x-1}{2-1} = 2\ln(2)(x-1)$$

$$P'(x) = 2\ln(2)$$

ENTONCES:

$f(1{,}5) \approx 0{,}6081$

$f'(1{,}5) \approx 1{,}40546$

$P(1{,}5) \approx 0{,}69314$

$P'(1{,}5) \approx 1{,}38629$

$S(1{,}5) \approx 0{,}6065$

$S'(1{,}5) \approx 1{,}4061$

MUCHO MEJOR CUBIC SPLINES

b)

$$S_1 = a_1 + b_1(x-1) + c_1(x-1)^2 + d_1(x-1)^3$$

$$S_1'(x) = b_1 + 2c_1(x-1) + 3d_1(x-1)^2$$

$$f(x) = \operatorname{sen}(e^x - 2)$$

$$f'(x) = e^x\cos(e^x - 2)$$

$$S_0(x) = a_0 + b_0(x-0{,}7) + c_0(x-0{,}7)^2 + d_0(x-0{,}7)^3$$

$$S_0'(x) = b_0 + 2c_0(x-0{,}7) + 3d_0(x-0{,}7)^2$$

$$S_1(x) = a_1 + b_1(x-0{,}8) + c_1(x-0{,}8)^2 + d_1(x-0{,}8)^3$$

$$S_1'(x) = b_1 + 2c_1(x-0{,}8) + 3d_1(x-0{,}8)^2$$

... PASA DESPEJAR

```python
# Ej 3.10

def f(x):
    return np.sin(np.exp(x) - 2)

x = np.array([0.7, 0.8, 1.0])

S = scipy.interpolate.CubicSpline(x, f(x), bc_type='natural')

xq = 0.9

print(S(xq))
print(S(xq, 1))
```

✓ 0.0s

```
0.4390258400041164
2.1784149094794123
```

DESPEJADO !

3.11 Un auto viaja en línea recta y su posición y velocidad son registradas a distintos tiempos. Estos valores son:

| Tiempo (s) | 0 | 3 | 5 | 8 | 13 |
|---|---|---|---|---|---|
| Posición (m) | 0 | 68 | 116 | 190 | 303 |
| Velocidad (m/s) | 22,86 | 23,47 | 24,38 | 22,56 | 21,95 |

a) Utilice splines cúbicos con *scipy.interpolate.CubicSpline* con condiciones de borde *not-a-knot* para predecir la posición y velocidad del auto cuando $t = 10\,\mathrm{s}$.

b) ¿Cuál es la velocidad máxima que alcanza el auto? ¿En qué momento la alcanza?

```python
# Ej 3.11

t = np.array([0.0, 3.0, 5.0, 8.0, 13.0])
x = np.array([0.0, 68.0, 116.0, 190.0, 303.0])
v = np.array([22.86, 23.47, 24.38, 22.56, 21.95])

Sx = scipy.interpolate.CubicSpline(t, x, bc_type='not-a-knot')
Sv = scipy.interpolate.CubicSpline(t, v, bc_type='not-a-knot')

tq = 10.0
x10 = Sx(tq)
v10 = Sv(tq)
print(f"x(10 s) ≈ {x10:.6f} m")
print(f"v(10 s) ≈ {v10:.6f} m/s")
```

✓ 0.0s

```
x(10 s) ≈ 238.002698 m
v(10 s) ≈ 20.926124 m/s
```

```python
t_fine = np.arange(0, 13.1, 0.1)
max_vel = 0
max_vel_time = 0

for time in t_fine:
    est_vel = Sv(time)
    if est_vel > max_vel:
        max_vel = est_vel
        max_vel_time = time

print(f"Velocidad máxima ≈ {max_vel:.6f} m/s en t = {max_vel_time:.1f} s")
```

✓ 0.0s

```
Velocidad máxima ≈ 24.380145 m/s en t = 5.1 s
```

3.12 Suponga que $f(x)$ es un polinomio de grado 3. Muestre que $f(x)$ es su propio spline cúbico engrampado pero que no puede ser su propio spline cúbico natural.

CONDICIONES DEL SPLINE CUBICO CLAMPED:

$\Rightarrow$ $\forall x_i$ EN NODOS, S ES UN POLINOMIO DE GRADO 3
$\Rightarrow$ $S$, $S'$ y $S''$ CONTINUAS EN TODOS LOS NODOS
$\Rightarrow$ $S(x_i) = f(x_i)$ $\forall x_i$ EN NODOS
$\Rightarrow$ $S'(x_i) = f'(x_i)$ $\forall x_i$ EN NODOS

SI PASA TODO ESTO EL SPLINE EXISTE Y ES UNICO.

- SI MI $S(x)$ ES $f(x)$ PODEMOS TOMAR A LOS NODOS COMO CUALQUIER CONJUNTO DE VALORES Y:

$\Rightarrow$ $\forall x_i$ EN NODOS, $f(x)$ ES UN POLINOMIO DE GRADO 3 ($S_0 \dots S_i \dots$) ✓
$\Rightarrow$ $f$, $f'$ y $f''$ CONTINUAS EN TODOS LOS NODOS ($f$ ES $C^2$) ✓
$\Rightarrow$ $f(x_i) = f(x_i)$ $\forall x_i$ EN NODOS
$\Rightarrow$ $f'(x_i) = f'(x_i)$ $\forall x_i$ EN NODOS
} (TRIVIAL) ✓

OSEA QUE $f$ ES VALIDA Y LA ÚNICA SOLUCIÓN. $f(x)$ ES SU PROPIO CUBIC SPLINE.

CONDICIONES DEL SPLINE CUBICO NATURAL:

$\Rightarrow$ $\forall x_i$ EN NODOS, S ES UN POLINOMIO DE GRADO 3
$\Rightarrow$ $S$, $S'$ y $S''$ CONTINUAS EN TODOS LOS NODOS
$\Rightarrow$ $S(x_i) = f(x_i)$ $\forall x_i$ EN NODOS
$\Rightarrow$ $S''(x_0) = 0$ y $S''(x_n) = 0$

- $f$ CUMPLE LAS PRIMERAS 3, PERO AL HACER $f''(x)$ ME QUEDA ALGO DE LA PINTA:

$f''(x) = ax + c$

RESUELVO:

$$\begin{cases} a x_0 + c = 0 & (1) \\ a x_n + c = 0 & (2) \end{cases}$$

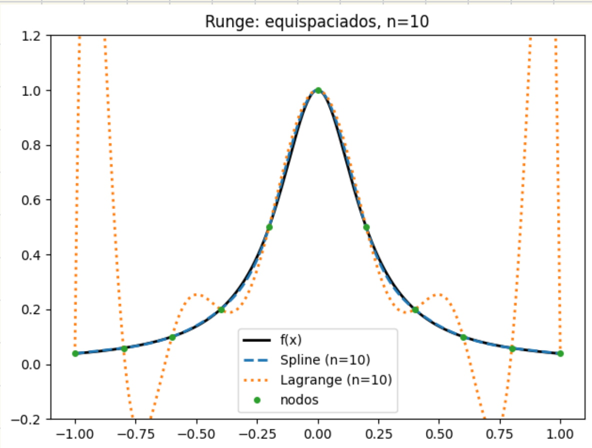
$(1) - (2) \Rightarrow a(x_0 - x_n) = 0$

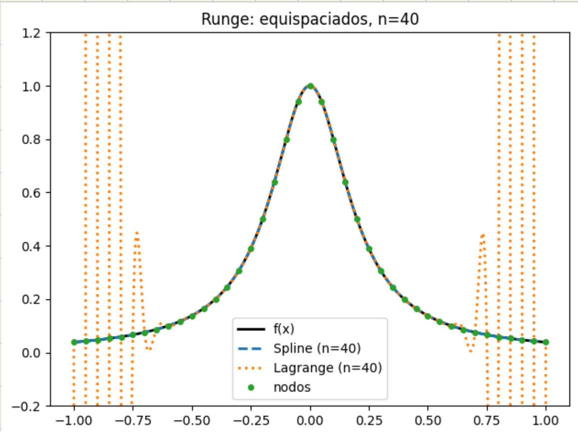
$\cancel{x_0 = x_n}$ ABS!

$f$ ES CLAMPED PERO NO PUEDE SER NATURAL

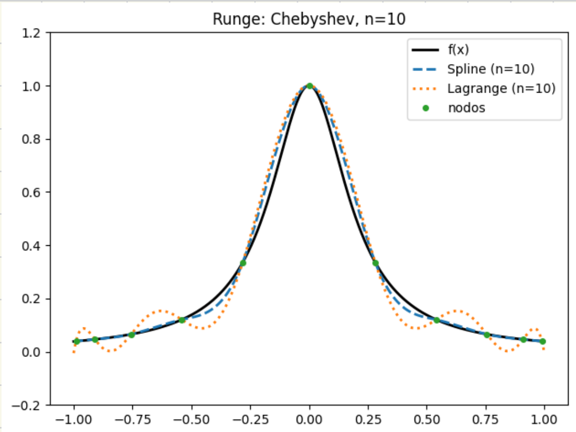
3.13 Aplique *CubicSplines* a la función de Runge (Ej. 3.9) y compare con lo obtenido con polinomios de Lagrange.

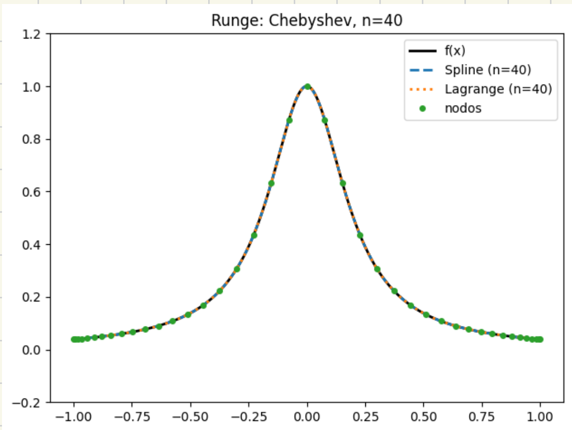
```python
def f(x):
    return 1.0 / (1.0 + 25.0*x*x)

def nodes_equispaced(n):
    return np.linspace(-1.0, 1.0, n+1)

def nodes_chebyshev(n):
    k = np.arange(n+1)
    return np.cos((2*k + 1) * np.pi / (2*(n+1)))  # tipo I, dentro de [-1,1]

xf = np.linspace(-1.0, 1.0, 2001)
yf = f(xf)

def compare_for(n, nodes_fun, title_tag):
    x = np.sort(nodes_fun(n))
    y = f(x)

    # Spline cúbico (not-a-knot por defecto; podés cambiar bc_type='natural' si querés)
    S = scipy.interpolate.CubicSpline(x, y)           # piecewise cúbico
    ys = S(xf)

    # Lagrange (forma baricéntrica estable)
    P = scipy.interpolate.BarycentricInterpolator(x, y)   # polinomio de grado n
    yp = P(xf)

    # errores máximos en la malla fina
    err_spline   = np.max(np.abs(ys - yf))
    err_lagrange = np.max(np.abs(yp - yf))

    print(f"n={n:2d}  {title_tag:12s}  max|f - S| = {err_spline:.3e}   max|f - P| = {err_lagrange:.3e}")

    # gráfico
    plt.plot(xf, yf, 'k', lw=2, label='f(x)')
    plt.plot(xf, ys, '--', lw=2, label=f'Spline (n={n})')
    plt.plot(xf, yp, ':', lw=2, label=f'Lagrange (n={n})')
    plt.plot(x, y, 'o', ms=4, label='nodos')
    plt.ylim(-0.2, 1.2)
    plt.title(f'Runge: {title_tag}, n={n}')
    plt.legend()
    plt.tight_layout()
    plt.show()

# --- corré comparaciones ---
n_list = [10, 20, 40]

print("Errores máximos en [-1,1] (malla de 2001 pts):")
for n in n_list:
    compare_for(n, nodes_equispaced, "equispaciados")
for n in n_list:
    compare_for(n, nodes_chebyshev, "Chebyshev")
```




Cubic splines GOAT.



**Métodos Numéricos y Optimización - Segundo semestre 2024**
Universidad de San Andrés

**Guía 1:** *Punto flotante. Resolución de ecuaciones no lineales. Interpolación*

# 1. Representaciones numéricas

## Aproximaciones y errores

1.1 Calcular el error absoluto y relativo de las siguientes aproximaciones a $p$ por $p^*$:

a) $p = e^{10}$, $p^* = 22000$,

b) $p = 10^{\pi}$, $p^* = 1400$,

c) $p = 8!$, $p^* = 39900$,

d) $p = 9!$, $p^* = \sqrt{18\pi}(9/e)^9$.

1.2 Encuentre el intervalo más grande en donde $p^*$ puede estar de forma que el error relativo al aproximar los siguientes $p$ no sea mayor a $10^{-4}$.

a) $\pi$,

b) $e$,

c) $\sqrt{2}$,

d) $7^{1/3}$.

1.3 El número $\pi$ se puede obtener a partir de las siguientes igualdades.

a) $\pi = 4\left[\arctan\left(\frac{1}{2}\right) + \arctan\left(\frac{1}{3}\right)\right]$,

b) $\pi = 16\arctan\left(\frac{1}{5}\right) - 4\arctan\left(\frac{1}{239}\right)$.

Para cada uno de los casos, calcule una aproximación a $\pi$ considerando los primeros 3 términos no nulos de la serie de Taylor de la arcotangente

$$\arctan(x) \approx x - (1/3)x^3 + (1/5)x^5,$$

Usando todos los dígitos disponibles en la calculadora. ¿Cuál fórmula llevó a un menor error? Argumente la razón de la diferencia.

1.4 Demostrar que:

a) $\cos(x) = 1 - \frac{x^2}{2} + O(x^4)$

b) $\sin(x) = x - \frac{x^3}{6} + O(x^5)$

1.5 Encuentre la tasas de convergencia de las siguientes secuencias cuando $n \to \infty$.

a) $\lim_{n\to\infty} \sin\frac{1}{n} = 0$,

b) $\lim_{n\to\infty} \sin \frac{1}{n^2} = 0$,

c) $\lim_{n\to\infty} \left(\sin \frac{1}{n}\right)^2 = 0$,

d) $\lim_{n\to\infty} [\ln(n+1) - \ln(n)] = 0$.

## Aritmética de punto flotante

1.6 Encuentre numéricamente en *python* el valor del epsilon de máquina en *doble precision*, *simple precision* y *half precision*.

1.7 Utilice aritmética de punto flotante redondeando a tres dígitos para realizar los siguientes cálculos. Calcule el error absoluto y relativo, con el valor exacto determinado a al menos cinco dígitos.

a) $133 + 0{,}921$,

b) $133 - 0{,}499$,

c) $(121 - 0{,}327) - 119$,

d) $(121 - 119) - 0{,}327$.

1.8 Suponga que $fl(y)$ es una aproximación de redondeo de $k$-dígitos a $y$. Muestre que

$$\left|\frac{y - fl(y)}{y}\right| \leq 0{,}5 \times 10^{-k+1}.$$

Ayuda: si $d_{k+1} < 5$, entonces $fl(y) = 0.d_1 d_2 \cdots d_k \times 10^n$, si $d_{k+1} \geq 5$ entonces $fl(y) = 0.d_1 d_2 \cdots d_k \times 10^n + 10^{n-k}$.

1.9 El desarrollo de Taylor de la función $e^x$ proporciona una forma muy inestable de calcular este valor cuando $x$ es negativo. Evalúe numéricamente el desarrollo de Taylor hasta grado $n$ de la función $e^x$ en $x = -12$, para $n = 1, .., 100$. Comparar con el valor exacto: $0{,}000006144212353328210...$ ¿Cuáles son las principales fuentes de error? Proponer un método alternativo para estimar $e^{-12}$. Verificar si la aproximación obtenida es mejor.

1.10 Considere la función $f(x) = \frac{1-\cos(x)}{x^2}$ y analice su comportamiento numérico para valores de $x$ cercanos a cero, $x = 10^{-k}$, $k = 1, \ldots$ Evalúe la función en doble precisión para una serie de valores decrecientes de $x$ y compare los resultados con el valor límite $\lim_{x\to 0} f(x) = \frac{1}{2}$ (no olvide usar escala logarítmica para visualizar). Justifique la pérdida de precisión observada y reformule la expresión de $f(x)$ de manera algebraicamente equivalente que mejore la estabilidad numérica. Compare los resultados obtenidos con ambas expresiones.

1.11 ¿Cuántas multiplicaciones y cuántas sumas se necesitan para realizar la siguiente operación en la forma en la que está dada?

$n=5$

$$\sum_{i=1}^{5} \sum_{j=1}^{i} a_i b_j$$

$$\sum_{i=1}^{n} \sum_{j=1}^{i} a_i b_j.$$

Productos: $1+2+3+4+5$ $\quad \frac{(n+1)n}{2}$

Sumas: $1+1+2+1+3+1+4+1+5$ $\quad \frac{(n+1)n}{2} + n - 1$

Modifique la suma de modo tal que reduzca el número de operaciones necesarias. Compare la *performance* de ambos algoritmos en *python* usando un vector aleatorio de *numpy*.

1.12 **Método de Horner**: Dado un polinomio $p(x) = a_n x^n + a_{n-1}x^{n-1} + ... + a_1 x + a_0$. ¿Cuántos productos y cuántas sumas se realizan al evaluar el poliomio en un cierto $x_0$ ? Horner propone como alternativa escribir a $p$ como $p(x) = a_0 + x(a_1 + x(a_2 + ... + x(a_{n-1} + xa_n)))$. ¿Cuántos productos y cuántas sumas se realizan al evaluar $p$ bajo esta forma?

# 2. Ecuaciones no-lineales

## Método de la bisección

2.1 Realice 3 iteraciones del método de bisección en papel sobre las siguientes funciones. Luego, cree una función en *python* que tome como entrada $f$ (cualquier función $f(x)$), $a$ y $b$ (intervalo inicial), $N$ (número máximo de iteraciones), y $\mathrm{tol}$ (tolerancia, que será la cota superior del error). Encuentre aproximaciones a las raíces tomando la tolerancia $\mathrm{tol} = 10^{-5}$ y $N = 1000$. Puede elegir tomar el criterio de terminación en la iteración $i$ como

$$|p_i - p_{i-1}| < \mathrm{tol}, \quad \frac{|p_i - p_{i-1}|}{|p_i|} < \mathrm{tol}, \text{ o } \quad |f(p_i)| < \mathrm{tol}. \tag{1}$$

a) $3x - e^x = 0$ con $x \in [1, 2]$,

b) $2x + 3\cos x - e^x = 0$ con $x \in [1, 2]$,

c) $x^2 - 4x + 4 - \ln x = 0$ con $x \in [1, 2]$ y $x \in [2, 4]$,

d) $x + 1 - 2\sin \pi x = 0$ con $x \in [0, 0{,}5]$ y $x \in [0{,}5, 1]$.

2.2 Sea $f(x) = (x + 2)(x + 1)^2 x(x - 1)^3(x - 2)$. ¿A qué cero de $f$ converge el método de la bisección cuando se aplica en los siguientes intervalos?

a) $[-1{,}5, 2{,}5]$,

b) $[-0{,}5, 2{,}4]$,

c) $[-0{,}5, 3]$,

d) $[-3, -0{,}5]$.

2.3 Una partícula cae desde el reposo por un plano inclinado cuyo ángulo $\theta$ cambia a una tasa constante

$$\frac{d\theta}{dt} = \omega < 0.$$

La distancia desde el punto de partida de la partícula a tiempo $t$ está dado por la siguiente fórmula

$$x(t) = -\frac{g}{2\omega^2}\left(\frac{e^{\omega t} - e^{-\omega t}}{2} - \sin \omega t\right).$$

Suponga que la partícula se movió $0{,}5\,\mathrm{m}$ en $1\,\mathrm{s}$ y asuma que $g = 9{,}8\,\mathrm{m\,s^{-2}}$. Encuentre una aproximación de $\omega$ con error menor a $10^{-5}$ (elegir uno de los 3 criterios).

2.4 Sea $f \in C[a,b]$ y $f(a)f(b) < 0$. Muestre que el método de la bisección genera una secuencia $\{p\}_{n=1}^{\infty}$ que aproxima a $p$ tal que

$$|p_n - p| \leq \frac{b-a}{2^n},$$

cuando $n \geq 1$. Muestre que esta cota de error converge linealmente a $0$.

2.5 Encuentre una cota al número de iteraciones necesarias para lograr una aproximación con error menor a $10^{-3}$ a la solución de $x^3 = -x + 4$ en el intervalo $[1, 4]$. Encuentre la aproximación.

2.6 Sea $f(x) = (x-1)^{10}$, $p = 1$ y $p_n = 1 + 1/n$. Muestre que $|f(p_n)| < 10^{-3}$ para cualquier $n > 1$ pero que para tener $|p - p_n| < 10^{-3}$ se necesita $n > 1000$.

## Método de punto fijo

2.7 Sea $f(x) = x^3 - 2x + 1$. Para resolver $f(x) = 0$ se proponen los siguientes cuatro problemas de punto fijo. Verifique que los puntos fijos de las funciones $g(x)$ se corresponden con raíces de $f(x)$. Escriba una función de *python* que tome como entrada $g$ (cualquier función $g(x)$), $p_0$ (valor inicial), $N$ (número máximo de iteraciones), y tol (tolerancia). Encuentre aproximaciones a las raíces tomando la tolerancia $\mathrm{tol} = 10^{-8}$ y $N = 1000$.

a) $g(x) = \frac{1}{2}(x^3 + 1)$, $p_0 = \frac{1}{2}$,

b) $g(x) = \frac{2}{x} - \frac{1}{x^2}$, $p_0 = \frac{1}{2}$,

c) $g(x) = \sqrt{2 - \frac{1}{x}}$, $p_0 = \frac{1}{2}$,

d) $g(x) = -\sqrt[3]{1-2x}$, $p_0 = \frac{1}{2}$,

2.8 Para las funciones de punto fijo del ejercicio anterior, encuentre cuáles cumplen las condiciones de convergencia de punto fijo y en qué intervalo.

2.9 a) Demuestre el teorema de punto fijo (*Burden Faires 2.4*).

b) Demuestre que si $g$ satisface las hipótesis del teorema de punto fijo, entonces las cotas de error de la iteración de punto fijo son

$$|p_n - p| \leq k^n \max\{p_0 - a,\ b - p_0\},$$

y

$$|p_n - p| \leq \frac{k^n}{1-k}|p_1 - p_0|, \quad \text{para todo } n \geq 1.$$

2.10 Los siguientes cuatro métodos son propuestos para computar $21^{1/3}$. Asumiendo $p_0 = 1$, ordénelos de mayor a menor con respecto a su velocidad aparente de convergencia.

a) $p_n = \dfrac{20p_{n-1} + 21/p_{n-1}^2}{21}$

b) $p_n = p_{n-1} - \dfrac{p_{n-1}^3 - 21}{3p_{n-1}^2}$

c) $p_n = p_{n-1} - \frac{p_{n-1}^4 - 21p_{n-1}}{p_{n-1}^2 - 21}$

d) $p_n = \left(\frac{21}{p_{n-1}}\right)^{1/2}$

2.11 Utilice un método de iteración de punto fijo para encontrar la solución de las siguientes ecuaciones con una precisión de $10^{-2}$.

a) $x^3 - x - 1 = 0$, intervalo de interés: $[1, 2]$ y $p_0 = 1$.

b) $e^{-x} - x = 0$, intervalo de interés: $[0, 1]$ y $p_0 = 1$.

c) $\sin(\ln(x)) - (x^3 - x^2) = 0$, intervalo de interés: $[0{,}1,\ 1]$ y $p_0 = 0{,}1$.

2.12 Muestre que $g(x) = 2^{-x}$ tiene un único punto fijo en $[1/3, 1]$. Utilice iteración de punto fijo para encontrarlo con una precisión de $10^{-4}$. Encuentre una cota al número de iteraciones necesarias para llegar a esta tolerancia partiendo de $p_0 = 1$. y compárela con el que obtuvo empíricamente.

2.13 Encuentre un valor aproximado iterando 3 veces para $\sqrt[3]{25}$ utilizando el método de la bisección y el método de iteración de punto fijo. Compare los resultados. Busque el número de iteraciones necesarias para tener una precisión de $10^{-4}$.

2.14 Sea $g \in C^1[a, b]$ y $p \in (a, b)$ con $g(p) = p$ y $|g'(p)| > 1$. Muestre que existe un $\delta > 0$ tal que si $0 < |p_0 - p| < \delta$, entonces $|p_0 - p| < |p_1 - p|$. Es decir, no importa que tan cerca esté $p_0$ de $p$, la siguiente iteración $p_1$ siempre estará más lejos, por lo que el método de iteración de punto fijo no converge si $p_0 \neq p$.

2.15 Encuentre una función $g$ definida en $[0, 1]$ que no cumple ninguna de las hipótesis del teorema de punto fijo pero igual tiene un único punto fijo en $[0, 1]$.

## Método de Newton

2.16 Utilice el método de Newton para encontrar soluciones con de los siguientes problemas

a) $e^x + 2^{-x} + 2\cos x - 6 = 0$, con $x \in [1, 2]$,

b) $\ln(x - 1) + \cos(x - 1) = 0$, con $x \in [1{,}3, 2]$,

c) $(x - 2)^2 - \ln x = 0$, con $x \in [1, 2]$.

Primero resuelva un par de iteraciones a mano, partiendo de algún $p_0$ dentro de los intervalos, y luego implemente una función de *python* para converger con una tolerancia de $10^{-5}$.

2.17 La ecuación $x^2 - 10\cos x = 0$ tiene dos soluciones $\pm 1{,}3793646$. Utilice el método de Newton para aproximar las soluciones con los siguientes $p_0$

a) $p_0 = -100$,

b) $p_0 = -50$,

c) $p_0 = -25$,

d) $p_0 = 25$,

e) $p_0 = 50$,

f) $p_0 = 100$.

2.18 La función $f(x) = \ln(x^2+1) - e^{0,4x}\cos \pi x$ tiene un número infinito de ceros.

a) Determine, con precisión de $10^{-6}$, el único cero negativo.

b) Determine, con precisión de $10^{-6}$, los cuatro ceros positivos más pequeños.

c) Determine una aproximación inicial razonable para encontrar el n-ésimo cero positivo más pequeño.

d) Utilice el resultado del inciso anterior para encontrar el 25avo cero positivo más pequeño.

2.19 Utilice el método de Newton para resolver la ecuación

$$0 = \frac{1}{2} + \frac{1}{4}x^2 - x\sin x - \frac{1}{2}\cos 2x,$$

con $p_0 = \pi/2$. ¿Algo le resultado llamativo? Repita con $p_0 = 5\pi$ y $p_0 = 10\pi$.

2.20 La ecuación de anualidades ordinaria

$$A = \frac{P}{i}[1-(1+i)^{-n}],$$

se utiliza para calcular cuanto se debe pagar de una hipoteca en un período fijo de tiempo. $A$ es el monto de la hipoteca, $P$ es el monto de cada pago, e $i$ es la tasa de interés por período por los $n$ períodos de pago. Suponga que es necesaria una hipoteca a 30 años con un valor de \$135000 y que el prestatante puede pagar como máximo \$1000 por mes. ¿Cuál es el interés máximo que puede pagar el prestatante?

2.21 La ecuación de iteración para el método de la secante se puede escribir de la siguiente forma

$$p_n = \frac{f(p_{n-1})p_{n-2} - f(p_{n-2})p_{n-1}}{f(p_{n-1}) - f(p_{n-2})}.$$

Explique por qué, en general, es probable que esta ecuación sea menos precisa que la vista en clase.

2.22 Derive la siguiente formula de error para el método de Newton

$$|p - p_{n+1}| \leq \frac{M}{2|f'(p_n)|}|p-p_n|^2,$$

asumiendo que las hipótesis del teorema visto en clase se cumplen, que $|f'(p_n)| \neq 0$ y que $M = \max|f''(x)|$. Ayuda: utilice la derivación del método de Newton a partir de la serie de Taylor de $f(p)$ vista en clase.

2.23 Dada $F : \mathbb{R}^n \to \mathbb{R}^n$ el método Newton-Raphson generalizado consiste en realizar la iteración vectorial

$$x_{k+1} = x_k - (DF|_{x_k})^{-1} F(x_k),$$

donde $x_k \in \mathbb{R}^n$ y $(DF|_{x_k})^{-1}$ es la inversa de la matriz diferencial de $F$ evaluada en $x_k$ . Usar Newton-Raphson generalizado para hallar una raíz de la función

$$F(x_1, x_2) = \begin{bmatrix} x_1 + x_2 - 3 \\ x_1^2 + x_2^2 - 9 \end{bmatrix}.$$

2.24 Muestre que las siguientes secuencias convergen sublinealmente a $p = 0$. ¿Qué tan grande debe ser $n$ para que $|p_n - p| \leq 5 \times 10^{-2}$?

a) $p_n = \frac{1}{n}$, $n \geq 1$,
b) $p_n = \frac{1}{n^2}$, $n \geq 1$.

2.25 Se puede demostrar que si $\{p_n\}_{n=1}^{\infty}$ es una secuencia dada por el método de la secante que converge a $p$, la solución de $f(x) = 0$, entonces existe una constante $C$ tal que

$$|p_{n+1} - p| \approx C|p_n - p||p_{n-1} - p|.$$

Asuma que $\{p_n\}$ converge con orden $\alpha$, es decir, que

$$\lim_{n\to\infty} \frac{|p_{n+1} - p|}{|p_n - p|^{\alpha}} = \lambda,$$

con $\lambda > 0$ una constante, utilice esta información para mostrar que $\alpha = \frac{1+\sqrt{5}}{2}$. *Ayuda: la suposición sobre el orden de convergencia quiere decir que puede tomar* $|p_{n+1} - p| \approx \lambda |p_n - p|^{\alpha}$.

# 3. Interpolación

## Interpolación de Lagrange

3.1 Sea $x_0 = 0$, $x_1 = 0{,}6$ y $x_2 = 0{,}9$. Construya interpolaciones de primer o segundo grado para aproximar $f(0{,}45)$ dadas las siguientes funciones

a) $f(x) = \cos x$,
b) $f(x) = \sqrt{1 + x}$,
c) $f(x) = \ln(x + 1)$,

Encuentre cotas para el error de las aproximaciones.

3.2 Use los interpolinomios de Lagrange apropiados de grado 1, 2 y 3 para aproximar lo siguiente

a) $f(8{,}4)$ con $f(8{,}1) = 16{,}94410$, $f(8{,}3) = 17{,}56492$, $f(8{,}6) = 18{,}50515$, $f(8{,}7) = 18{,}82091$.
b) $f(-1/3)$ con $f(-0{,}75) = -0{,}0718125$, $f(-0{,}5) = -0{,}02475$, $f(-0{,}25) = 0{,}33493750$, $f(0) = 1{,}101$.

c) $f(0{,}25)$ con $f(0{,}1) = 0{,}62049958$, $f(0{,}2) = -0{,}28398668$, $f(03) = 0{,}00660095$, $f(0{,}4) = 0{,}2484244$.

3.3 Los datos del ejercicio anterior fueron generados con las siguientes funciones. Use la formula del error para encontrar una cota a éste y compárelo con el error obtenido en los casos $n = 1$ y $n = 2$.

a) $f(x) = x \ln x$,

b) $f(x) = x^3 + 4{,}001x^2 + 4{,}002x + 1{,}101$,

c) $f(x) = x \cos x - 2x^2 + 3x - 1$.

3.4 Escriba una función en *python* para realizar interpolación de Lagrange. La misma debería tomar como argumento los nodos (puntos de evaluación), y la función evaluada en los nodos.

Aplique a las siguientes funciones, evaluando en una grilla equiespaciada entre $x_0$ y $x_2$, y grafique el *ground truth* junto con la aproximación, así como el error.

a) $f(x) = e^{-2x} \sin 3x$, $x_0 = 0$, $x_1 = \pi/6$, $x_2 = \pi/4$, $n = 2$,

b) $f(x) = \log_{10} x$, $x_0 = 3{,}0$, $x_1 = 3{,}2$, $x_2 = 3{,}5$, $n = 2$,

c) $f(x) = e^x + e^{-x}$, $x_0 = -0{,}3$, $x_1 = 0$, $x_2 = 0{,}3$, $n = 2$,

Encuentre una cota para el error absoluto en el intervalo $[x_0, x_n]$. Compare esta cota con el error obtenido.

3.5 Sea $f(x) = e^x$ con $x \in [0, 2]$.

a) Aproxime $f(0{,}25)$ utilizando interpolación lineal con $x_0 = 0$ y $x_1 = 0{,}5$.

b) Aproxime $f(0{,}75)$ utilizando interpolación lineal con $x_0 = 0{,}5$ y $x_1 = 1$.

c) Aproxime $f(0{,}25)$ y $f(0{,}75)$ utilizando el segundo polinomio interpolador con $x_0 = 0$, $x_1 = 1$, $x_2 = 2$.

d) ¿Cuál aproximación es mejor y por qué?

3.6 Suponé que necesitás construir tablas con ocho cifras decimales para la función logaritmo común, o en base 10, desde $x = 1$ hasta $x = 10$, de manera que la interpolación lineal sea precisa dentro de un margen de error de $10^{-6}$. Determina un límite para el tamaño del paso en esta tabla. ¿Qué elección de tamaño de paso harías para asegurar que $x = 10$ esté incluido en la tabla?

3.7 Muestre que $\max_{x_j \leq x \leq x_{j+1}} |g(x)| = h^2/4$ con $g(x) = (x - jh)(x - (j + 1)h)$.

3.8 Muestre que el polinomio que interpola los siguientes datos tiene grado 3: $f(-2) = 1$, $f(-1) = 4$, $f(0) = 11$, $f(1) = 16$, $f(2) = 13$, $f(3) = -4$.

3.9 Sea la función $f(x) = \dfrac{1}{1 + 25x^2}$ en el intervalo $[-1, 1]$ (función de Runge).

a) En *python* (empleando funciones propias o de librerías), construya el polinomio interpolador de Lagrange para $f(x)$ de grado $n$ usando varios valores de $n$ empleando:

a) $n+1$ nodos equiespaciados.

b) $n+1$ nodos de Chebyshev de primer tipo dados por:

$$x_k = \cos\left(\frac{2k+1}{2(n+1)}\pi\right), \quad k = 0, 1, \ldots, n.$$

b) Graficar en cada caso la función $f(x)$ y el polinomio interpolador en el intervalo $[-1, 1]$.

c) Calcular y graficar el error absoluto $|f(x) - P_n(x)|$ en una malla fina de puntos y comparar ambas formas de colocación de nodos.

### Splines Cúbicos

3.10 Utilice splines cúbicos para aproximar los siguientes valores y calcule el error asociado. Compare con el obtenido usando polinomio de Lagrange.

a) $f(x) = x \ln x$, con condiciones de borde çlampedüsando los valores de $f$ en $x_0 = 1$ y $x_1 = 2$, aproxime $f(1{,}5)$ y $f'(1{,}5)$.

b) $f(x) = \sin(e^x - 2)$, con condiciones de borde naturales usando los valores de $f$ en $x_0 = 0{,}7$, $x_1 = 0{,}8$ y $x_2 = 1{,}0$, aproxime $f(0{,}9)$ y $f'(0{,}9)$.

3.11 Un auto viaja en línea recta y su posición y velocidad son registradas a distintos tiempos. Estos valores son:

| Tiempo (s) | 0 | 3 | 5 | 8 | 13 |
|---|---|---|---|---|---|
| Posición (m) | 0 | 68 | 116 | 190 | 303 |
| Velocidad (m/s) | 22,86 | 23,47 | 24,38 | 22,56 | 21,95 |

a) Utilice splines cúbicos con *scipy.interpolate.CubicSpline* con condiciones de borde *not-a-knot* para predecir la posición y velocidad del auto cuando $t = 10$ s.

b) ¿Cuál es la velocidad máxima que alcanza el auto? ¿En qué momento la alcanza?

3.12 Suponga que $f(x)$ es un polinomio de grado 3. Muestre que $f(x)$ es su propio spline cúbico engrampado pero que no puede ser su propio spline cúbico natural.

3.13 Aplique *CubicSplines* a la función de Runge (Ej. 3.9) y compare con lo obtenido con polinomios de Lagrange.

## 4. Ejercicios integradores

4.1 El proceso de trilateración es una forma de determinar la posición de un objeto conocida solamente su posición relativo a un conjunto de sensores y es una de las bases de como funcionan los sistemas de GPS. Arme un sistema de trilateración en tres dimensiones asumiendo conocimiento perfecto de las mediciones. La posición de la partícula puede recuperarse resolviendo ecuaciones del tipo

$$\sqrt{(x-x_1)^2+(y-y_1)^2+(z-z_1)^2} = d_1,$$
$$\sqrt{(x-x_2)^2+(y-y_2)^2+(z-z_2)^2} = d_2.$$
$$\sqrt{(x-x_3)^2+(y-y_3)^2+(z-z_3)^2} = d_3,$$

donde $(x, y, z)$ es la posición de la partícula en el momento dado, $(x_i, y_i, z_i)$ con $i = 1, 2, 3$ son las posiciones (conocidas) de los 3 sensores y $d_i$ son las mediciones realizadas.

Utilizando los datos provistos en los archivos *4.1.measurements.txt* y *4.1.sensor_positions.txt*, recupere la trayectoria de la partícula $\mathbf{r}(t_i) = [x(t_i), y(t_i), z(t_i)]$ resolviendo el sistema de ecuaciones mediante el método de Newton-Raphson generalizado para los tiempos medidos $t_i$.

Interpole las posiciones obtenidas para generar un trayectoria en función del tiempo $\mathbf{r}(t) = [x(t), y(t), z(t)]$ y evalúe las funciones en los tiempos de la trayectoria ground truth (*4.1.trajectory.txt*) para así obtener un error de la interpolación.