Métodos Numéricos

Grupo: 4F21 **Semestre**: 4to

Actividad:

Problemario

Integrantes del Equipo:

Diego Alonso Coronel Vargas

Priscila Cortés Ramírez

Hernández Cruz Luis Manuel

Angel Dario Vargas Bautista

Leonel Montalvo Vigil

Índice

Introducción

En el estudio de ecuaciones no lineales, encontrar soluciones exactas puede ser complicado o incluso imposible mediante métodos algebraicos tradicionales. Por ello, los métodos numéricos juegan un papel fundamental, proporcionando aproximaciones eficientes a las raíces de estas ecuaciones. En este trabajo, exploramos cinco métodos numéricos ampliamente utilizados: Bisección, Regla Falsa, Punto Fijo, Newton y Secante, analizando su funcionamiento, ventajas, desventajas y aplicaciones. Además, implementamos cada uno en código y Excel para visualizar su comportamiento y compararlos en diferentes escenarios. Finalmente, integramos todos los métodos en un Excel para evaluar su precisión y eficiencia en la resolución de ecuaciones.

Problemario de Métodos de Búsqueda de una Raíz

Métodos Cerrados

Método de Bisección

Seleccionar un intervalo inicial [a,b] tal que $f(a) \cdot f(b) < 0$ (esto garantiza la existencia de una raíz por el **teorema del valor intermedio**).

Calcular el punto medio del intervalo:

$$c = \frac{a+b}{2}$$

Evaluar f(c):

- Si $f(c)=0$ entonces c es la raíz exacta.
- Si $f(c) \neq 0$ reducir el intervalo:
    - Si $f(a) \cdot f(c) < 0$ la raíz está en [a,c], por lo que reemplazamos b=c
    - Si $f(b) \cdot f(c) < 0$ la raíz está en [c,b], por lo que reemplazamos a=c.
    - **Repetir el proceso** hasta que el error sea menor que una tolerancia predefinida.

$$\text{Error} = \left| c_{\text{nuevo}} - c_{\text{anterior}} \right|$$

Ejercicio 1

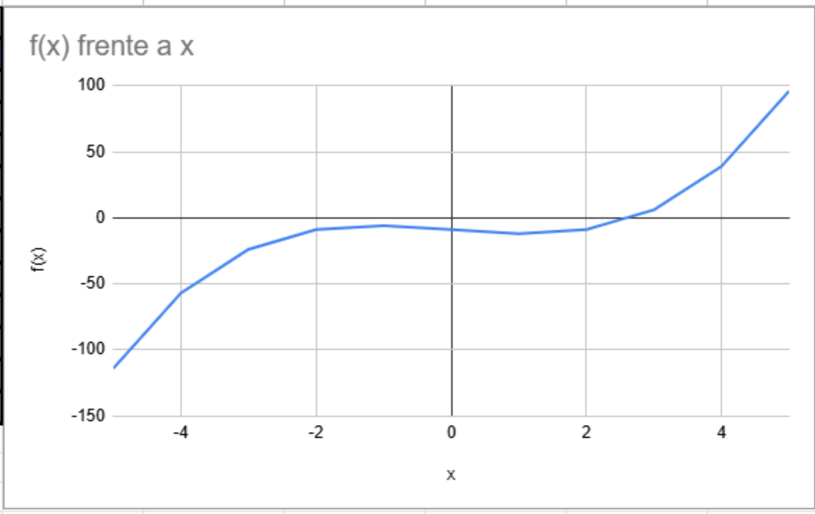
$$f(x) = x^3 - 4x - 9 \text{ en } [2, 3]$$

B	C	D	E	F	G	H	I	J
Ecuación	x^3-4x-9							
Iteraciones	14		Intervalos	2	3			

Iteración	a	b	mitad	f(a)	f(b)	f(mitad)	Error
1	2,000	3,000	2,500	-9,000	6,000	-3,375	
2	2,500	3,000	2,750	-3,375	6,000	0,797	0,250
3	2,500	2,750	2,625	-3,375	0,797	-1,412	0,125
4	2,625	2,750	2,688	-1,412	0,797	-0,339	0,063
5	2,688	2,750	2,719	-0,339	0,797	0,221	0,031
6	2,688	2,719	2,703	-0,339	0,221	-0,061	0,016
7	2,703	2,719	2,711	-0,061	0,221	0,079	0,008
8	2,703	2,711	2,707	-0,061	0,079	0,009	0,004
9	2,703	2,707	2,705	-0,061	0,009	-0,026	0,002
10	2,705	2,707	2,706	-0,026	0,009	-0,009	0,001
11	2,706	2,707	2,707	-0,009	0,009	0,000	0,000
12	2,706	2,707	2,706	-0,009	0,000	-0,004	0,000
13	2,706	2,707	2,706	-0,004	0,000	-0,002	0,000
14	2,706	2,707	2,706	-0,002	0,000	-0,001	0,000

Tabla de valores

x	f(x)
-5	-114
-4	-57
-3	-24
-2	-9
-1	-6
0	-9
1	-12
2	-9
3	6
4	39
5	96



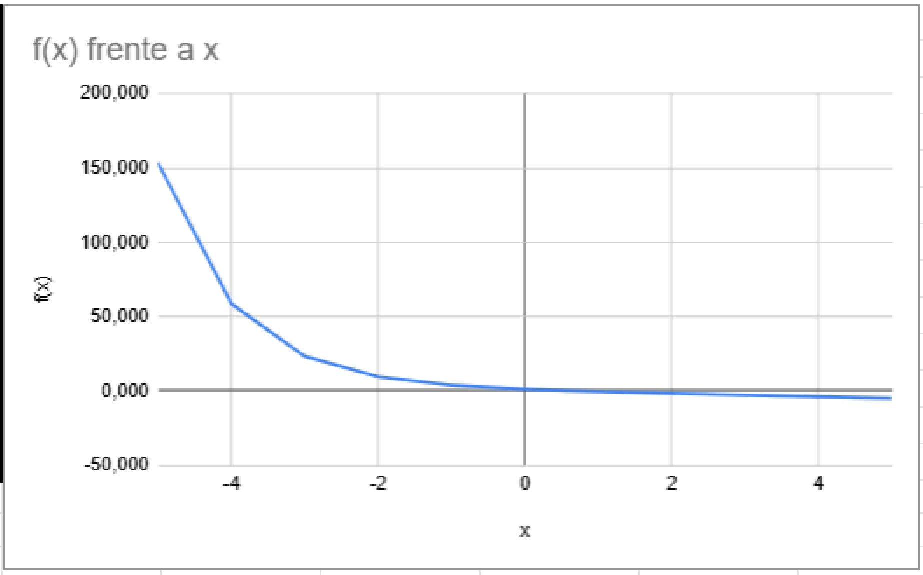
Ejercicio 2

$$f(x) = e^{-x} - x \text{ en } [0, 1]$$

Ecuación	e^-x -x						
Iteraciones	14	Intervalos	0	1			

Iteración	a	b	mitad	f(a)	f(b)	f(mitad)	Error
1	0,000	1,000	0,500	1,000	-0,632	0,107	
2	0,500	1,000	0,750	0,107	-0,632	-0,278	0,250
3	0,500	0,750	0,625	0,107	-0,528	-0,090	0,125
4	0,500	0,625	0,563	0,107	-0,465	0,007	0,063
5	0,563	0,625	0,594	0,007	-0,465	-0,041	0,031
6	0,563	0,594	0,578	0,007	-0,448	-0,017	0,016
7	0,563	0,578	0,570	0,007	-0,439	-0,005	0,008
8	0,563	0,570	0,566	0,007	-0,435	0,001	0,004
9	0,566	0,570	0,568	0,001	-0,435	-0,002	0,002
10	0,566	0,568	0,567	0,001	-0,434	0,000	0,001
11	0,566	0,567	0,567	0,001	-0,433	0,000	0,000

Tabla de valores	
x	f(x)
-5	153,413
-4	58,598
-3	23,086
-2	9,389
-1	3,718
0	1,000
1	-0,632
2	-1,865
3	-2,950
4	-3,982
5	-4,993

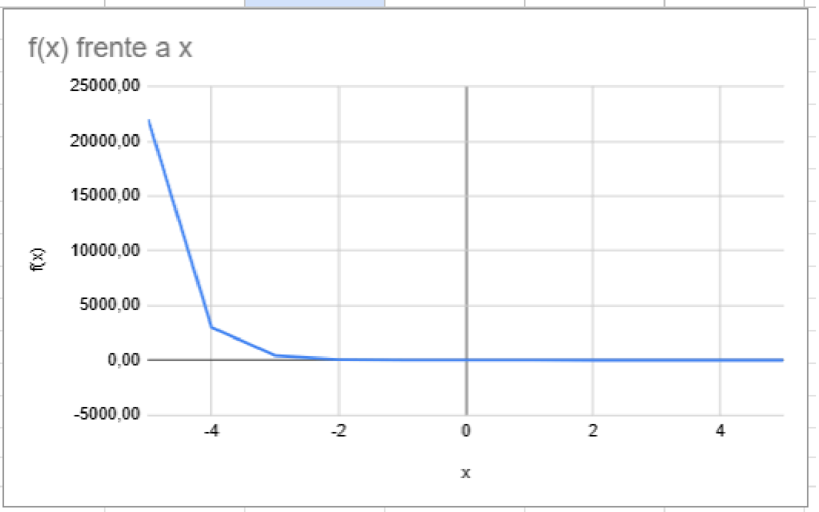


Ejercicio 3

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

B	C	D	E	F	G	H	I
Ecuación	e^-2x -2x						
Iteraciones	14		Intervalos	0	1		
Iteración	a	b	mitad	f(a)	f(b)	f(mitad)	Error
1	0,000	1,000	0,500	1,000	-1,865	-0,632	
2	0,000	0,500	0,250	1,000	-0,632	0,107	0,250
3	0,250	0,500	0,375	0,107	-0,632	-0,278	0,125
4	0,250	0,375	0,313	0,107	-0,278	-0,090	0,063
5	0,250	0,313	0,281	0,107	-0,090	0,007	0,031
6	0,281	0,313	0,297	0,007	-0,090	-0,041	0,016
7	0,281	0,297	0,289	0,007	-0,041	-0,017	0,008
8	0,281	0,289	0,285	0,007	-0,017	-0,005	0,004
9	0,281	0,285	0,283	0,007	-0,005	0,001	0,002
10	0,283	0,285	0,284	0,001	-0,005	-0,002	0,001
11	0,283	0,284	0,284	0,001	-0,002	0,000	0,000

Tabla de valores	
x	f(x)
-5	22036,47
-4	2988,96
-3	409,43
-2	58,60
-1	9,39
0	1,00
1	-1,86
2	-3,98
3	-6,00
4	-8,00
5	-10,00

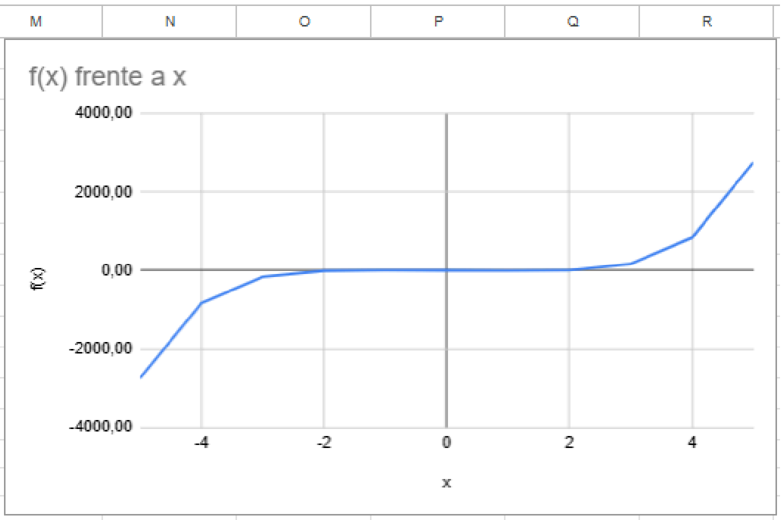


Ejercicio 4

$$f(x) = x^5 - 3x^3 + 2 \text{ en } [-2, 0]$$

B	C	D	E	F	G	H	I
Ecuación	$h(x) = x^5 - 3x^3 + 2$						
Iteraciones	14		Intervalos	-2	0		
Iteración	**a**	**b**	**mitad**	**f(a)**	**f(b)**	**f(mitad)**	**Error**
1	-2,000	0,000	-1,000	-6,000	2,000	4,000	
2	-2,000	-1,000	-1,500	-6,000	4,000	4,531	0,500
3	-2,000	-1,500	-1,750	-6,000	4,531	1,665	0,250
4	-2,000	-1,750	-1,875	-6,000	1,665	-1,399	0,125
5	-1,875	-1,750	-1,813	-1,399	1,665	0,302	0,063
6	-1,875	-1,813	-1,844	-1,399	0,302	-0,503	0,031
7	-1,844	-1,813	-1,828	-0,503	0,302	-0,090	0,016
8	-1,828	-1,813	-1,820	-0,090	0,302	0,109	0,008
9	-1,828	-1,820	-1,824	-0,090	0,109	0,010	0,004
10	-1,828	-1,824	-1,826	-0,090	0,010	-0,040	0,002
11	-1,826	-1,824	-1,825	-0,040	0,010	-0,015	0,001
12	-1,825	-1,824	-1,825	-0,015	0,010	-0,002	0,000

K	L
Tabla de valores	
x	**f(x)**
-5	-2748,00
-4	-830,00
-3	-160,00
-2	-6,00
-1	4,00
0	2,00
1	0,00
2	10,00
3	164,00
4	834,00
5	2752,00



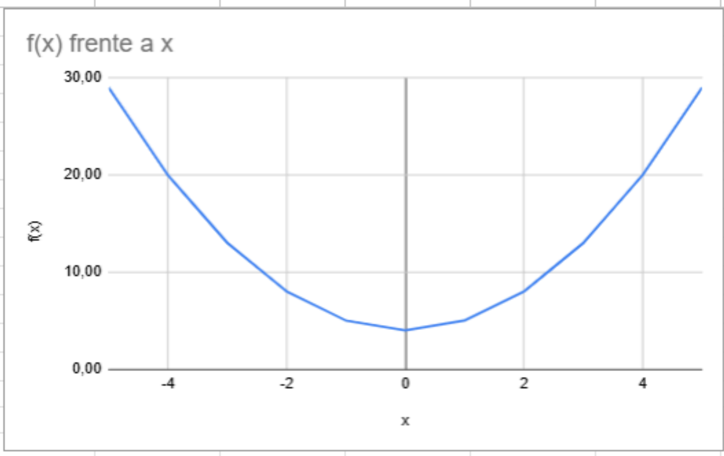
Ejercicio 5 (Caso en el que el método falla)

$$5.\quad f(x) = x^2 + 4 \text{ en } [-2, 2]$$

B	C	D	E	F	G	H	I
Ecuación	$p(x) = x^2 + 4$						
Iteraciones	14		Intervalos	-2	2		

Iteración	a	b	mitad	f(a)	f(b)	f(mitad)	Error
1	-2,000	2,000	0,000	8,000	8,000	4,000	
2	0,000	0,000	0,000	4,000	4,000	4,000	0,000
3	0,000	0,000	0,000	4,000	4,000	4,000	0,000
4	0,000	0,000	0,000	4,000	4,000	4,000	0,000
5	0,000	0,000	0,000	4,000	4,000	4,000	0,000
6	0,000	0,000	0,000	4,000	4,000	4,000	0,000
7	0,000	0,000	0,000	4,000	4,000	4,000	0,000
8	0,000	0,000	0,000	4,000	4,000	4,000	0,000
9	0,000	0,000	0,000	4,000	4,000	4,000	0,000
10	0,000	0,000	0,000	4,000	4,000	4,000	0,000
11	0,000	0,000	0,000	4,000	4,000	4,000	0,000
12	0,000	0,000	0,000	4,000	4,000	4,000	0,000
13	0,000	0,000	0,000	4,000	4,000	4,000	0,000
14	0,000	0,000	0,000	4,000	4,000	4,000	0,000

Tabla de valores	
x	f(x)
-5	29,00
-4	20,00
-3	13,00
-2	8,00
-1	5,00
0	4,00
1	5,00
2	8,00
3	13,00
4	20,00
5	29,00



El **método de bisección** no puede encontrar la raíz en un intervalo si **no existe un cambio de signo** en los valores de la función en los extremos del intervalo. Esto puede ocurrir si la función no tiene una raíz en el intervalo o si el intervalo seleccionado es incorrecto.

Ejercicio extra con ID distinto

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

Intervalo $[0, 1]$

Código python:

```
def bisection_method(f, a, b, tol=1e-6, max_iter=100):
    """
    Método de bisección para encontrar la raíz de la ecuación f(x) = 0
    :param f: Función para la cual se busca la raíz
    :param a: Límite inferior del intervalo
    :param b: Límite superior del intervalo
    :param tol: Tolerancia para el error
    :param max_iter: Número máximo de iteraciones
    :return: Raíz aproximada
    """
    # Verificar si el cambio de signo es válido en el intervalo
    if f(a) * f(b) > 0:
        print("No hay un cambio de signo en el intervalo, no se puede
aplicar el método de bisección.")
        return None

    iter_count = 0
    while (b - a) / 2 > tol:  # Convergencia basada en el tamaño del
intervalo
        c = (a + b) / 2  # Punto medio
        if f(c) == 0:  # Si encontramos la raíz exacta
            break
        elif f(a) * f(c) < 0:  # Si el cambio de signo es entre a y c
            b = c
        else:  # Si el cambio de signo es entre c y b
            a = c

        iter_count += 1
        if iter_count >= max_iter:  # Control de iteraciones
            print("Se alcanzó el número máximo de iteraciones.")
            break

    # Raíz aproximada
    c = (a + b) / 2
    print(f"Raíz aproximada: {c}")
    print(f"Error relativo: {abs(f(c))}")
```

```python
    return c

# Ejemplo de uso: encontrar la raíz de f(x) = e^(-x) - x en el intervalo
[0, 1]
import math

def f(x):
    return math.exp(-x) - x

# Llamada al método de bisección
raiz = bisection_method(f, 0, 1)
```

```
.debugpy-2025.0.1-win32-x64\bundled\libs\debugpy\launcher' '55045'
Raíz aproximada: 0.567
```

Método de Regla Falsa

El método de la regla falsa es una técnica numérica utilizada para encontrar raíces de ecuaciones no lineales de la forma:

$$f(x) = 0$$

Este método pertenece a la categoría de los métodos de búsqueda de raíces por intervalos, lo que significa que requiere de un intervalo [a,b] en el cual la función cambia de signo, es decir, $f(a)\cdot f(b) < 0$. Esto garantiza que existe al menos una raíz en el intervalo, según el teorema de Bolzano.

El método de la regla falsa es similar al método de bisección, pero en lugar de dividir el intervalo en dos partes iguales, utiliza una interpolación lineal entre los puntos $(a, f(a))$ y $(b, f(b))$ para encontrar una mejor aproximación a la raíz.

La idea detrás del método de la regla falsa es encontrar el punto donde la recta secante entre $(a, f(a))$ y $(b, f(b))$ cruza el eje x. La ecuación de la recta secante es:

$$x_r = \frac{af(b) - bf(a)}{f(b) - f(a)}$$

Tras calcular x_r, verificamos si la raíz se encuentra en el subintervalo $[a, x_r]$ o en $[x_r, b]$, aplicando nuevamente el teorema de Bolzano.

- Si $f(a)\cdot f(x_r) < 0$, entonces la raíz se encuentra en $[a, x_r]$ y el nuevo intervalo será $[a, x_r]$.
- Si $f(x_r)\cdot f(b) < 0$, entonces la raíz está en y el nuevo intervalo será $[x_r, b]$.
- Si $f(x_r) = 0$, entonces hemos encontrado la raíz exacta y el proceso finaliza.

Ejercicio 1

Función: $f(x) = x^3 - 4x + 1$

Intervalos: $[a, b] = [0, 1]$

Ejercicio en Excel:

Método de la regla falsa		
Función	x^3 - 4x + 1	
Intervalos	0	1

a	b	f(a)	f(b)	Xr	f(Xr)	f(a)*f(Xr)	f(b)*f(Xr)	E. A.	E. R.
0.000	1.000	1.000	-2.000	0.333	-0.296	-0.296	0.593		
0.000	0.333	1.000	-0.296	0.257	-0.012	-0.012	0.003	0.076	23%
0.000	0.257	1.000	-0.012	0.254	0.000	0.000	0.000	0.003	1%
0.000	0.254	1.000	0.000	0.254	0.000	0.000	0.000	0.000	0%

Ejercicio 2

Función: $f(x) = e^x - 3x$

Intervalos: $[a, b] = [0, 2]$

Ejercicio en Excel:

Método de la regla falsa		
Función	e^x - 3x	
Intervalos	0	1

a	b	f(a)	f(b)	Xr	f(Xr)	f(a)*f(Xr)	f(b)*f(Xr)	E. A.	E. R.
0.000	1.000	1.000	-0.282	0.780	-0.159	-0.159	0.045		
0.000	0.780	1.000	-0.159	0.673	-0.059	-0.059	0.009	0.107	14%
0.000	0.673	1.000	-0.059	0.636	-0.019	-0.019	0.001	0.038	6%
0.000	0.636	1.000	-0.019	0.624	-0.006	-0.006	0.000	0.012	2%
0.000	0.624	1.000	-0.006	0.621	-0.002	-0.002	0.000	0.003	1%
0.000	0.621	1.000	-0.002	0.619	0.000	0.000	0.000	0.001	0%

Ejercicio 3

Función: $f(x)\ =\ cos(x)\ -\ x$

Intervalos: $[a, b]\ =\ [0, 1]$

Ejercicio en Excel:

Método de la regla falsa									
Función	cos(x) - x								
Intervalos	0	1							
a	b	f(a)	f(b)	Xr	f(Xr)	f(a)*f(Xr)	f(b)*f(Xr)	E. A.	E. R.
0.000	1.000	1.000	-0.460	0.685	0.089	0.089	-0.041		
0.685	1.000	0.089	-0.460	0.736	0.005	0.000	-0.002	0.051	7%
0.736	1.000	0.005	-0.460	0.739	0.000	0.000	0.000	0.003	0%

Ejercicio 4

Función: $f(x)\ = x^2\ -\ 5$

Intervalos: $[a, b]\ =\ [2, 3]$

Ejercicio en Excel:

Método de la regla falsa									
Función	cos(x) - x								
Intervalos	0	1							
a	b	f(a)	f(b)	Xr	f(Xr)	f(a)*f(Xr)	f(b)*f(Xr)	E. A.	E. R.
0.000	1.000	1.000	-0.460	0.685	0.089	0.089	-0.041		
0.685	1.000	0.089	-0.460	0.736	0.005	0.000	-0.002	0.051	7%
0.736	1.000	0.005	-0.460	0.739	0.000	0.000	0.000	0.003	0%

Ejercicio 5 (Caso en el que el método falla)

Función: $f(x) = x^2 + 4$

Intervalos: $[a, b] = [-2, 2]$

Ejercicio en Excel:

Método de la regla falsa		
Función	x^2+4	
Intervalos	-2	2

a	b	f(a)	f(b)	Xr	f(Xr)	f(a)*f(Xr)	f(b)*f(Xr)	E. A.	E. R.
-2.000	2.000	8.000	8.000	#¡DIV/0!	68.000	544.000	544.000		
#¡DIV/0!	#¡DIV/0!	#¡DIV/0!	#¡DIV/0!	#¡DIV/0!	#¡DIV/0!	#¡DIV/0!	#¡DIV/0!	#¡DIV/0!	#¡DIV/0!

Aquí el método falla porque la función no cambia de signo en el intervalo.

Ejercicio extra con ID distinto

Función: $f(x) = e^{-x} - x$

Intervalos: $[a, b] = [0, 1]$

Código en Python:

```
import numpy as np

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

def regla_falsa(f, a, b, tol=1e-6, max_iter=100):
    if f(a) * f(b) >= 0:
        raise ValueError(f"El intervalo [{a}, {b}] no cambia de signo.
f(a)={f(a)}, f(b)={f(b)}")

    for _ in range(max_iter):
        xr = (a * f(b) - b * f(a)) / (f(b) - f(a))
```

```python
        if abs(f(xr)) < tol:
            return xr

        if f(a) * f(xr) < 0:
            b = xr
        else:
            a = xr

    return xr

raiz = regla_falsa(f, 0, 1)

print(f"La raíz aproximada es: {raiz:.3g}")
```

Ejecución del código:

```
sions\ms-python.debugpy-2025.0.1-win32-x64\bundled\libs\debugpy\launcher' '58570' '--' 'C:\Us
rs\diego\OneDrive\Documentos\ITESA\CUARTO_SEMESTRE\4F21_Metodos_Numericos\Tema_2\Problemario\
xcel.py'
La raíz aproximada es: 0.567
PS C:\Users\diego\OneDrive\Documentos\ITESA\CUARTO_SEMESTRE\4F21_Metodos_Numericos\Tema_2\Pro
lemario>
```

Ejercicio en Excel:

Método de la regla falsa		
Función	e^(-x) - x	
Intervalos	-2	2

a	b	f(a)	f(b)	Xr	f(Xr)	f(a)*f(Xr)	f(b)*f(Xr)	E. A.	E. R.
-2.000	2.000	9.389	-1.865	1.337	-1.075	-10.090	2.004		
-2.000	1.337	9.389	-1.075	0.994	-0.625	-5.864	0.671	0.343	26%
-2.000	0.994	9.389	-0.625	0.808	-0.362	-3.397	0.226	0.187	19%
-2.000	0.808	9.389	-0.362	0.704	-0.209	-1.959	0.076	0.104	13%
-2.000	0.704	9.389	-0.209	0.645	-0.120	-1.126	0.025	0.059	8%
-2.000	0.645	9.389	-0.120	0.611	-0.069	-0.646	0.008	0.033	5%
-2.000	0.611	9.389	-0.069	0.592	-0.039	-0.370	0.003	0.019	3%
-2.000	0.592	9.389	-0.039	0.582	-0.023	-0.212	0.001	0.011	2%
-2.000	0.582	9.389	-0.023	0.575	-0.013	-0.121	0.000	0.006	1%
-2.000	0.575	9.389	-0.013	0.572	-0.007	-0.069	0.000	0.004	1%
-2.000	0.572	9.389	-0.007	0.570	-0.004	-0.040	0.000	0.002	0%

Métodos Abiertos

Método de Punto Fijo

Como tal el punto fijo, también llamado método de sustitución sucesiva. Este procedimiento se basa en escribir una ecuación f(x)=0 en la forma equivalente:

$$x = g(x)$$

Esta transformación se logra mediante manipulaciones algebraicas, reorganizando los términos de la ecuación original. Una vez que la ecuación ha sido transformada a la forma x=g(x), se puede utilizar un valor inicial x0x_0x0 para generar nuevas aproximaciones sucesivas de la solución mediante la fórmula iterativa:

$$xi + 1 = g(xi)$$

El proceso continúa hasta que la diferencia entre iteraciones consecutivas sea lo suficientemente pequeña, indicando que la solución se ha aproximado con la precisión deseada. En términos de error, se emplea el **error relativo normalizado** para evaluar la convergencia, definido como:

$$e_a = \left| \frac{x_{i+1} - x_i}{x_{i+1}} \right| \times 100\%$$

Para garantizar que el método converja a la raíz deseada, la función g(x) debe cumplir la condición de contracción, lo que significa que su derivada absoluta en el intervalo de interés debe ser menor que 1:

$$|g'(x)| < 1$$

Si esta condición no se cumple, el método podría no converger o hacerlo de manera muy lenta.

(Chapra & Canale, 2015, 143)

Ejercicio 1

f(x)=x^2−4

g(x)= x^2 +4/ 4

x0=2.5

Tolerancia: 0.01

Tabla de Iteraciones:

DATOS DE ENTRADA	
f(x)=	x^2−4
g(x)=	x^2 + 4/ 4
x0=	2.5
Tolerancia: 0.01	

Metodo de Punto Fijo				
Iteración	xi	g(xi)	Error	Resultado
3	2.125	2.3613	-0.055	Continúa
4	1.9375	2.3857	-0.12	Termina

Raíz aproximada: x≈2.34x \approx 2.34x≈2.34

Ejercicio 2

$f(x) = x - cosx$

$g(x) = cosx$

$x0 = 0.5$

Tolerancia: 0.01

Tabla de Iteraciones:

DATOS DE ENTRADA	
f(x)=	x−cosx
g(x)=	cosx
x0=	0.5
Tolerancia: 0.01	

Metodo de Punto Fijo				
Iteración	xi	g(xi)	Error	Resultado
1	0.5	0.8776	0.3776	Continúa
2	0.8776	0.639	0.2386	Continúa
3	0.639	0.8027	0.1637	Continúa
4	0.8027	0.6948	0.1079	Continúa
5	0.6948	0.7682	0.0734	Continúa
6	0.7682	0.7221	0.0461	Continúa
7	0.7221	0.7501	0.028	Continúa
8	0.7501	0.7314	0.0187	Continúa
9	0.7314	0.7442	0.0128	Continúa
10	0.7442	0.7356	0.0086	Termina

Raíz aproximada: x≈0.7356x \approx 0.7356x≈0.7356

Ejercicio 3

$f(x) = e - x - x$

$g(x) = e - x$

$x0 = 0.5$

Tolerancia: 0.01

DATOS DE ENTRADA	
f(x)=	e^-x -x
g(x)=	e^-x
x0=	0.5
Tolerancia: 0.01	

Metodo de Punto Fijo				
Iteración	xi	g(xi)	Error	Resultado
1	0.5	0.8776	0.3776	Continúa
2	0.6065	0.639	0.2386	Continúa
3	0.5452	0.8027	0.1637	Continúa
4	0.5796	0.6948	0.1079	Continúa
5	0.5592	0.7682	0.0734	Continúa
6	0.5712	0.7221	0.0461	Continúa

Raíz aproximada: x≈0.5649

Ejercicio 4

$f(x) = x^2 - 4x + 2$

$g(x) = \frac{x^2+2}{4}$

$x0 = 0.1$

Tolerancia: 0.01

Método de Punto Fijo					
Función f(x)	f(x) =x^2 - 4x + 2				
Función g(x)	x = (x^2 + 2)/4				
i	0				
x0	0.1				
iteraciones	i	xi	x_(i+1)	E. A.	E. R.
1	0	0	0.503		
2	1	0.503	0.563	0.061	11%
3	2	0.563	0.579	0.016	3%
4	3	0.579	0.584	0.005	1%
5	4	0.584	0.585	0.001	0%

Ejercicio 5 (ejercicio que falla)

$f(x) = 3x + cos(x) - 1$

$g(x) = 1 - cos(x)$

$x0 = 3$

Tolerancia: 0.01

Método de Punto Fijo	
Función f(x)	f(x) =3x + cos(x) - 1
Función g(x)	x = 1 - cos(x)
i	0
x0	3

iteraciones	i	xi	x_(i+1)	E. A.	E. R.
1	0	3	1.990		
2	4	1.990	3.352	1.362	41%
3	5	3.352	15.268	11.916	78%
4	6	15.268	1423.819	1408.551	99%
5	7	1423.819	##########	##########	100%
6	8	##########	##########	##########	100%

Método de Newton Raphson

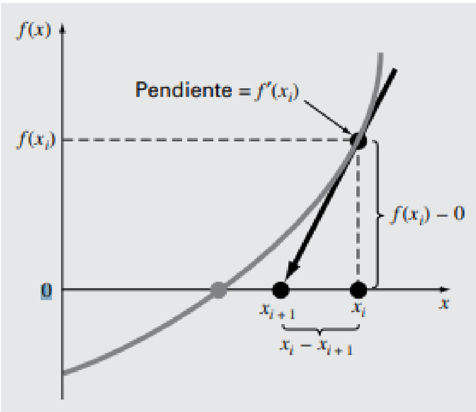
Si el valor inicial para la raíz es x, entonces se puede trazar una tangente desde el punto $[x_i, f(x_i)]$. Por lo común, el punto donde esta tangente cruza al eje x representa una aproximación mejorada de la raíz.

El método de Newton-Raphson se deduce a partir de esta interpretación geométrica (un método alternativo basado en la serie de Taylor).

La serie de Taylor descompone una función en una suma de términos más simples basados en sus derivadas.

$$f(x) \approx f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2 + \frac{f''(x_0)}{3!}(x - x_0)^3 + \cdots$$

Mediante la representación gráfica del método de Newton- Raphson se obtiene una tangente a la función en x_i $[Esto\ es,\ f(x_i)]$ hasta el eje x para obtener una estimación de la raíz en x_{i+1}.



De tal manera que, se sabe que la primera derivada en x es equivalente a la pendiente:

$$f'(x_i) = \frac{f(x_i) - 0}{x_i - x_{i+1}}$$

Esto se acomoda para obtener la fórmula de Newton Raphson.

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

(Chapra & Canale, 2015, 117)

Ejercicio 1

Datos de entrada:

$f(x) = x^3 - 27$

$f'(x) = 3x^2$

$x_i = 3.5$

Tolerancia: 0.001

DATOS DE ENTRADA	
x	**3.5**
f(x)	**x^3-27**
f'(x)	**3x^2**
Toleracncia	**0.001**

Método de Newton Raphson

Iteración	xi	f(xi)	f'(xi)	x	Error A	Error R	Resultado
1	3.500	15.875	36.750	3.068	0.432	14%	Continua
2	3.068	1.879	28.238	3.001	0.067	2%	Continua
3	3.001	0.040	27.027	3.000	0.001	0%	Continua
4	3.000	0.000	27.000	3.000	0.000	0%	Continua
5	3.000	0.000	27.000	3.000	0.000	0%	Continua
6	3.000	0.000	27.000	3.000	0.000	0%	3.000

Solución en Excel.

Ejercicio 2

Datos de entrada:

$f(x) = x^2 - 144$

$f'(x) = 2x$

$x_i = 18$

Tolerancia: 0.001

DATOS DE ENTRADA	
x	**18**
f(x)	**x^2-144**
f'(x)	**2x**
Toleracncia	**0.001**

Método de Newton Raphson

Iteración	xi	f(xi)	f'(xi)	x	Error A	Error R	Resultado
1	18.000	180.000	36.000	13.000	5.000	38%	Continua
2	13.000	25.000	26.000	12.038	0.962	8%	Continua
3	12.038	0.925	24.077	12.000	0.038	0%	Continua
4	12.000	0.001	24.000	12.000	0.000	0%	Continua
5	12.000	0.000	24.000	12.000	0.000	0%	Continua
6	12.000	0.000	24.000	12.000	0.000	0%	12.000

Solución en Excel.

Ejercicio 3

Datos de entrada:

$f(x) = 16x^4 - 256$

$f'(x) = 64x^3$

$x_i = 5$

Tolerancia: 0.001

DATOS DE ENTRADA	
x	5
f(x)	16x^4+256
f'(x)	64x^3
Toleracncia	0.001

Método de Newton Raphson

Iteración	xi	f(xi)	f'(xi)	x	Error A	Error R	Resultado
1	5.000	9744.000	8000.000	3.782	1.218	32%	Continua
2	3.782	3017.453	3462.139	2.910	0.872	30%	Continua
3	2.910	892.038	1577.819	2.345	0.565	24%	Continua
4	2.345	227.896	825.380	2.069	0.276	13%	Continua
5	2.069	37.182	566.818	2.003	0.066	3%	Continua
6	2.003	1.731	514.595	2.000	0.003	0%	Continua
7	2.000	0.004	512.007	2.000	0.000	0%	Continua
8	2.000	0.000	512.000	2.000	0.000	0%	Continua
9	2.000	0.000	512.000	2.000	0.000	0%	2.000

Solución en Excel.

Ejercicio 4 (Caso en el que el método falla)

Datos de entrada:

$f(x) = x^2 + 64$

$f'(x) = 2x$

$x_i = 10$

Tolerancia: 0.001

DATOS DE ENTRADA	
x	8
f(x)	x^2+64
f'(x)	2x
Toleracncia	0.001

Método de Newton Raphson

Iteración	xi	f(xi)	f'(xi)	x	Error A	Error R	Resultado
1	8.000	128.000	16.000	0.000	8.000	#¡DIV/0!	Continua
2	0.000	64.000	0.000	#¡DIV/0!	#¡DIV/0!	#¡DIV/0!	#¡DIV/0!

Solución en Excel.

El método falla, pues en la segunda iteración al sustituir en la derivada de la función se obtiene 0, este valor al ser utilizado en la fórmula de Newton Raphson provoca un número indeterminado.

Ejercicio 5

Datos de entrada:

$$f(x) = x^2 + x - 4$$

$$f'(x) = 2x + 1$$

$x_i = 3$

Tolerancia: 0.001

DATOS DE ENTRADA	
x	**3**
f(x)	**x^2+x-4**
f'(x)	**2x+1**
Toleracncia	**0.001**

Método de Newton Raphson

Iteración	xi	f(xi)	f'(xi)	x	Error A	Error R	Resultado
1	3.000	8.000	7.000	1.857	1.143	62%	Continua
2	1.857	1.306	4.714	1.580	0.277	18%	Continua
3	1.580	0.077	4.160	1.562	0.018	1%	Continua
4	1.562	0.000	4.123	1.562	0.000	0%	Continua
5	1.562	0.000	4.123	1.562	0.000	0%	Continua
6	1.562	0.000	4.123	1.562	0.000	0%	1.562

Solución en Excel.

Ejercicio extra con ID distinto

Datos de entrada:

$$f(x) = x^3 + 6x - 27$$

$$f'(x) = 3x^2 + 6$$

$x_i = 3$

Tolerancia: 0.001

Solución en Python:

```python
import math

# Definir la función f(x) y su derivada f'(x)
def f(x):
    return x**3 + 6*x - 27

def df(x):
    return 3*x**2 + 6

# Método de Newton-Raphson
def newton_raphson(x0, tol=0.001, max_iter=10):
    x = x0
    for i in range(max_iter):
```

```python
        x_new = x - f(x) / df(x)  # Fórmula de Newton-Raphson
        if abs(x_new - x) < tol:
            return x_new, i + 1
        x = x_new
    return None, max_iter

# Parámetros iniciales
x0 = 4 # Punto de inicio
raiz, iteraciones = newton_raphson(x0)

# Mostrar resultado
if raiz:
    print(f"La raíz aproximada es: {raiz:.3f}")
    print(f"Número de iteraciones: {iteraciones}")
else:
    print("El método no funcionó")
```

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

PS C:\Users\prisc> & C:/Users/prisc/AppData/Local/Programs/Python/Python313/python.exe
La raíz aproximada es: 2.347
Número de iteraciones: 5
PS C:\Users\prisc>
```

Ejecución del código.

Método de Secante

El método de la secante es un método numérico para encontrar raíces de una función , es decir, valores de x tales que f(x)=0. Se considera una variación del método de Newton-Raphson, pero sin necesidad de calcular derivadas.

Este método pertenece a la categoría de los **métodos iterativos de aproximación**, lo que significa que se basa en una secuencia de valores que convergen hacia la raíz en cada iteración. A diferencia de los métodos que requieren un intervalo con cambio de signo, como la **bisección** o la **regla falsa**, el método de la secante solo necesita dos aproximaciones iniciales cercanas a la raíz, sin exigir que la función cambie de signo entre ellas.

El método de la secante es una variación del método de Newton-Raphson, pero en lugar de utilizar la derivada de la función, usa una aproximación de la pendiente basada en dos puntos. Esto lo hace útil cuando la derivada es difícil de calcular o no está disponible.

La idea detrás del método de la secante es encontrar el punto donde la **recta secante** entre $(xn - 1, f(xn - 1))$ y $(xn, f(xn))$ cruza el eje x. La ecuación que define esta aproximación es:

$$x_{n+1} = x_n - \frac{f(x_n)(x_n - x_{n-1})}{f(x_n) - f(x_{n-1})}$$

Tras calcular $xn + 1$, se verifica la convergencia del método:

- Si $|Xn + 1 - Xn|$ es menor que un umbral de tolerancia ε, el proceso finaliza y se considera que $Xn + 1$ es una buena aproximación de la raíz.

- En caso contrario, se continúa iterando reemplazando $Xn - 1$ por Xn y Xn por $Xn + 1$, repitiendo el cálculo hasta alcanzar la precisión deseada o hasta un número máximo de iteraciones.

El método de la secante es generalmente más rápido que los métodos basados en intervalos, pero no garantiza la convergencia si los valores iniciales no son adecuados. Además, puede fallar si $f(xn) = f(xn - 1)$, ya que generaría una división por cero.

Ejercicio 1

Función: $f(x) = x^3 - 4x + 1$

Valores iniciales: $x_0 = 0,\ x_1 = 1$

Ejercicio en Excel:

Iteración	x_n	x_(n-1)	f(x_n)	f(x_(n-1))	x_(n+1)	Error
1	1	0	-2	1	0.3333	0.6667
2	0.3333	1	-0.2963	-2	0.2173	0.1159
3	0.2173	0.3333	0.1407	-0.2963	0.2547	0.0373
4	0.2547	0.2173	-0.0023	0.1407	0.2541	0.0006
5	0.2541	0.2547	-0.0001	-0.0023	0.2541	4.32E+00

Ejercicio 2

Función: $f(x) = ex - 3x$

Valores iniciales: $x_0 = 0.5,\ x_1 = 1.5$

Ejercicio en Excel:

Iteración	x_n	x_(n-1)	f(x_n)	f(x_(n-1))	x_(n+1)	Error
1	3	2	4	-1	2.2	0.8
2	2.2	3	-0.16	4	2.2307	0.0307
3	2.2307	2.2	-0.0236	-0.16	2.2361	0.0053
4	2.2361	2.2307	0.0002	-0.0236	2.236	0.0004

Ejercicio 3

Función: $f(x) = x2 - 5x + 6$

Valores iniciales: $x_0 = 1,\ x_1 = 2$

Ejercicio en Excel:

Iteración	x_n	x_(n-1)	f(x_n)	f(x_(n-1))	x_(n+1)	Error
1	2	1	5	-1	1.1667	0.8333
2	1.1667	2	-0.5787	5	1.2531	0.0864
3	1.2531	1.1667	-0.2853	-0.5787	1.3372	0.084
4	1.3372	1.2531	0.0538	-0.2853	1.3238	0.0133
5	1.3238	1.3372	-0.0037	0.0538	1.3245	0.0008
6	1.3247	1.3238	-0.0004	-0.0037	1.3247	0.0001

Ejercicio 4

Función: $f(x) = cos(x) - x$

Valores iniciales: $x_0 = 0,\ x_1 = 1$

Ejercicio en Excel:

Iteración	x_n	x_(n-1)	f(x_n)	f(x_(n-1))	x_(n+1)	Error
0	0	-	1	-	1	-
1	1	0	-0.4597	1	0.75	0.25
2	0.75	1	0.0183	-0.4597	0.739	0.011
3	0.739	0.75	0.0001	0.0183	0.7391	0.0001
4	0.7391	0.739	0	0.0001	0.7391	0

Ejercicio 5 (caso en el que el método falla)

Función: $f(x) = x^{1/3}$

Valores iniciales: $x_0 = 0.01,\ x_1 = 0.01$

Ejercicio en Excel:

Iteración	x_n	x_(n-1)	f(x_n)	f(x_(n-1))	x_(n+1)	Error
1.0000	(0.01+0j)	(-0.01+0j)	(0.215443	(0.107721	(1.734723	0.0200
2.0000	(1.734723	(0.01+0j)	(0.224070	(0.215443	(-0.017611	0.0399
3.0000	(-0.017611	(1.734723	(0.209924	(0.224070	(-0.030534	0.0350

El método falla en la segunda iteración porque genera un nuevo valor $xn + 1 = 0$, lo que hace que **f(xn)=0**, y la siguiente iteración intentaría dividir por cero, deteniendo el cálculo abruptamente.

Ejercicio extra con ID distinto:

```
def secante(f, x0, x1, tol=1e-5, max_iter=100):
    for i in range(max_iter):
        # Calcular el valor de la función en los puntos x0 y x1
        f_x0 = f(x0)
        f_x1 = f(x1)

        # Calcular la siguiente aproximación usando la fórmula de la secante
        x2 = x1 - f_x1 * (x1 - x0) / (f_x1 - f_x0)

        # Verificar si el cambio es menor que la tolerancia
        if abs(x2 - x1) < tol:
            return x2

        # Actualizar los valores de x0 y x1
        x0 = x1
        x1 = x2

    # Si se alcanza el número máximo de iteraciones, devolver None
    print("No se encontró la raíz en el número máximo de iteraciones")
    return None

# Definir la función para la cual queremos encontrar la raíz
def funcion(x):
    return x**2 - 4

# Ejemplo de uso del método de la secante
x0 = 2.5
x1 = 3.0
raiz = secante(funcion, x0, x1)

print("La raíz encontrada es:", raiz)
```

```
La raíz encontrada es: 2.0000000001734155
PS C:\Users\vigil\OneDrive\Documentos\problemarioE
```

Conclusión

En este trabajo, analizamos los métodos numéricos para la resolución de raíces Bisección, Regla Falsa, Punto Fijo, Newton y Secante, evaluando sus aplicaciones, ventajas y desventajas. El método de Bisección es seguro y siempre converge si se cumple el teorema de Bolzano, pero su convergencia es lenta. El método de Regla Falsa mejora la velocidad de convergencia mediante interpolación lineal, aunque puede presentar problemas en algunos casos donde la raíz no se aproxima de manera eficiente. El método de Punto Fijo es simple pero depende de que la función iterativa cumpla condiciones específicas para converger. El método de Newton es uno de los más rápidos, pero requiere el cálculo de la derivada, lo que puede ser una limitación. Finalmente, el método de la Secante ofrece rapidez similar a Newton sin calcular la derivada, aunque puede ser inestable si los puntos iniciales no son adecuados. A lo largo del desarrollo del problemario, implementamos cada método en código y Excel, permitiendo visualizar su comportamiento en distintos problemas. No hay un método único que sea el mejor para todos los casos; la elección depende de la función a resolver y de las condiciones del problema. Gracias a esta investigación y trabajo en equipo, logramos comprender a fondo cada método y su utilidad en la resolución numérica de ecuaciones.

Referencias

Chapra, S. C., & Canale, R. P. (2015). *Métodos numéricos para ingenieros* (S. M. Sarmiento Ortega, Trans.). McGraw-Hill.

Extras

Distribución del trabajo

Gráfico Gantt

Actividades	6:00 pm	7:00 pm	8:00 pm	9:00 pm	10:00 pm	11:00 pm	12:00 am
Método de Bisección				Luis Manuel	Luis Manuel	Luis Manuel	
Método de Regla Falsa	Diego Alonso	Diego Alonso					
Método de Punto Fijo							Angel Dario
Método de Newton Raphson		Priscila			Priscila	Priscila	Priscila
Método de Secante					Leonel	Leonel	Leonel
Ejercicio extra en Excel						Diego Alonso	Diego Alonso

Tabla scrum

Tarea	Responsable	Responsable	Fecha de finalización
Investigación - Método de Bisección	Hernández Cruz Luis Manuel	✅ Completado	26-02-25
Ejercicios (Código y Excel) - Método de Bisección	Hernández Cruz Luis Manuel	✅ Completado	26-02-25
Investigación - Método de Regla Falsa	Diego Alonso Coronel Vargas	✅ Completado	26-02-25
Ejercicios (Código	Diego Alonso	✅ Completado	26-02-25

y Excel) - Método de Regla Falsa	Coronel Vargas		
Investigación - Método de Punto Fijo	Angel Dario Vargas Bautista	✅ Completado	26-02-25
Ejercicios (Código y Excel) - Método de Punto Fijo	Angel Dario Vargas Bautista Diego Alonso Coronel Vargas	✅ Completado	27-02-25
Investigación - Método de Newton	Priscila Cortés Ramírez	✅ Completado	26-02-25
Ejercicios (Código y Excel) - Método de Newton	Priscila Cortés Ramírez	✅ Completado	26-02-25
Investigación - Método de Secante	Leonel Montalvo Vigil	✅ Completado	26-02-25
Ejercicios (Código y Excel) - Método de Secante	Leonel Montalvo Vigil	✅ Completado	26-02-25
Ejercicio Extra en Excel	Diego Alonso Coronel Vargas	✅ Completado	26-02-25