

Universidad Nacional del Altiplano
Facultad de Ingeniería Estadística e Informática

Docente:

Ing. Torres Cruz Fred

Alumno:

Ticona Miramira Roberto Angel

Trabajo 6 - Método de Regula Falsi

» DESCRIPCIÓN

El método de Regula Falsi, también conocido como método de la Falsa Posición, es una técnica numérica utilizada para encontrar raíces reales de ecuaciones no lineales de la forma $f(x) = 0$. Se basa en el mismo principio del método de Bisección: la existencia de una raíz dentro de un intervalo $[a, b]$ donde $f(a)$ y $f(b)$ tienen signos opuestos, es decir, $f(a)f(b) < 0$.

A diferencia de la Bisección, el método de Regula Falsi no toma el punto medio del intervalo, sino que aproxima la raíz mediante la intersección de la recta secante que pasa por los puntos $(a, f(a))$ y $(b, f(b))$ con el eje x . El valor aproximado de la raíz en cada iteración se calcula mediante la fórmula:

$$c = b - \frac{f(b)(a - b)}{f(a) - f(b)}$$

El proceso se repite reemplazando uno de los extremos del intervalo según el signo de $f(c)$, de modo que siempre se mantenga la condición de cambio de signo. El método converge más rápido que la Bisección, especialmente cuando la función se comporta de forma aproximadamente lineal cerca de la raíz.

Importancia de la visualización previa

Antes de aplicar el método, es recomendable graficar la función $f(x)$ en el intervalo de estudio. Esta representación permite verificar visualmente la existencia de una raíz (es decir, el cruce con el eje x) y seleccionar adecuadamente los valores iniciales a y b donde la función cambia de signo. Una elección incorrecta del intervalo puede hacer que el método falle o no converja hacia la raíz deseada.

Restricciones y limitaciones

- Requiere que $f(a)$ y $f(b)$ tengan signos opuestos ($f(a)f(b) < 0$).
- Si la función tiene una pendiente muy diferente en los extremos del intervalo, la convergencia puede volverse lenta.
- No garantiza la convergencia si la función no es continua o no cambia de signo en el intervalo.

» APLICACIÓN DEL MÉTODO EN PYTHON

El objetivo fue desarrollar un programa en Python que permita al usuario ingresar una función $f(x)$, graficarla en un intervalo definido y aplicar el método de Regula Falsi si así lo desea. El programa realiza iteraciones hasta que se cumpla un criterio de tolerancia, mostrando en cada paso los valores de a , b , c , $f(c)$ y el error estimado.

» ENTRADA

- Una cadena de texto que representa la función matemática $f(x)$.
- Los valores iniciales a y b que delimitan el intervalo de búsqueda.
- El intervalo para graficar la función.

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

» SALIDA

- Gráfica de la función $f(x)$ en el intervalo ingresado.
- La raíz aproximada encontrada.
- Número de iteraciones realizadas hasta cumplir la tolerancia.

» RESTRICCIONES

- La función debe ser continua en el intervalo $[a, b]$.
- Debe cumplirse la condición de cambio de signo $f(a)f(b) < 0$.

Código

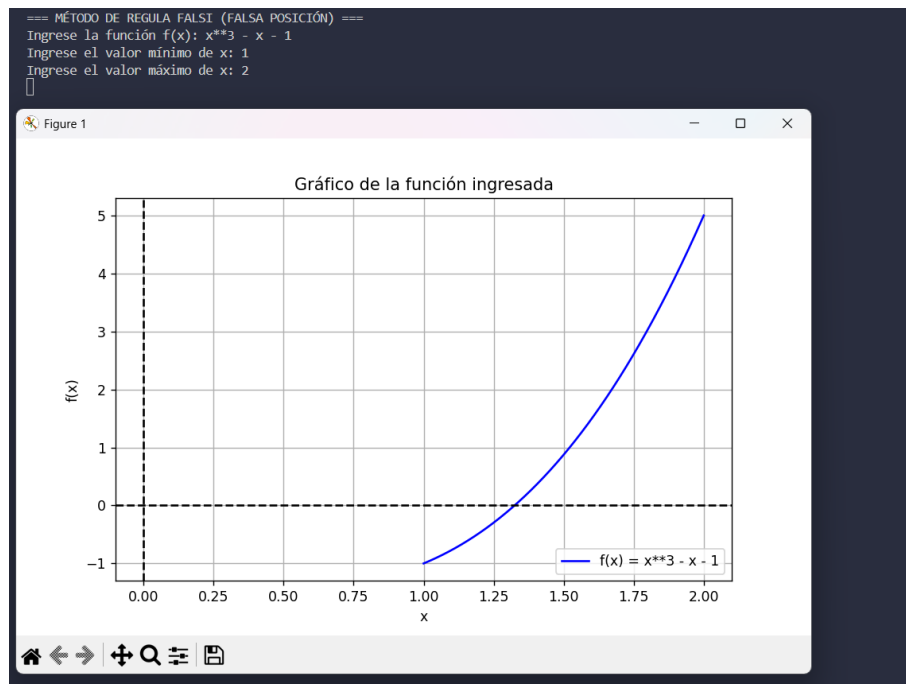
```

1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 # === 1. Ingreso de la función ===
5 print("=== MÉTODO DE REGULA FALSI (FALSA POSICIÓN) ===")
6 func_str = input("Ingrese la función f(x): ") # Ejemplo: x**3 - x - 1
7
8 # Definición de la función
9 def f(x):
10     return eval(func_str, {"np": np, "x": x})
11
12 # === 2. Graficar la función ===
13 xmin = float(input("Ingrese el valor mínimo de x: "))
14 xmax = float(input("Ingrese el valor máximo de x: "))
15
16 x = np.linspace(xmin, xmax, 400)
17 y = f(x)
18
19 plt.figure(figsize=(8, 5))
20 plt.plot(x, y, label=f"f(x) = {func_str}", color='blue')
21 plt.axhline(0, color='black', linestyle='--')
22 plt.axvline(0, color='black', linestyle='--')
23 plt.title("Gráfico de la función ingresada")
24 plt.xlabel("x")
25 plt.ylabel("f(x)")
26 plt.legend()
27 plt.grid(True)
28 plt.show()
29
30 # === 3. Pregunta si desea aplicar el método de Regula Falsi ===
31 op = input("¿Desea aplicar el método de Regula Falsi? (s/n): ").lower()
32
33 if op == "s":
34     # === 4. Ingreso de los extremos del intervalo ===
35     a = float(input("Ingrese el valor de a (extremo izquierdo): "))
36     b = float(input("Ingrese el valor de b (extremo derecho): "))

```

```
37
38 # Comprobación del cambio de signo
39 if f(a) * f(b) > 0:
40     print(" La función no cambia de signo en el intervalo. No se puede aplicar
41           ↪ el método.")
42 else:
43     tol = 1e-6
44     max_iter = 100
45
46     print("\nIteración |      a      |      b      |      c      |      f(c)
47           ↪      Error")
48     print("-----")
49
50     c_old = a
51     for i in range(1, max_iter + 1):
52         # Fórmula de Regula Falsi
53         c = b - (f(b) * (a - b)) / (f(a) - f(b))
54         error = abs(c - c_old)
55         c_old = c
56
57         print(f"{i:9d} | {a:10.6f} | {b:10.6f} | {c:10.6f} | {f(c):10.6f} |
58               ↪ {error:10.6f}")
59
60         if abs(f(c)) < tol or error < tol:
61             print(f"\n Raíz aproximada encontrada: {c:.6f}")
62             print(f"Iteraciones realizadas: {i}")
63             break
64
65         # Actualización de intervalos
66         if f(a) * f(c) < 0:
67             b = c
68         else:
69             a = c
70
71     else:
72         print("\n No se alcanzó la convergencia después de", max_iter,
73               ↪ "iteraciones.")
74 else:
75     print("No se aplicó el método de Regula Falsi.")
```

Ejecución



```

=== MÉTODO DE REGULA FALSI (FALSA POSICIÓN) ===
Ingrese la función f(x): x**3 - x - 1
Ingrese el valor mínimo de x: 1
Ingrese el valor máximo de x: 2
¿Desea aplicar el método de Regula Falsi? (s/n): s
Ingrese el valor de a (extremo izquierdo): 1
Ingrese el valor de b (extremo derecho): 2

```

Iteración	a	b	c	f(c)	Error
1	1.000000	2.000000	1.166667	-0.578704	0.166667
2	1.166667	2.000000	1.253112	-0.285363	0.086445
3	1.253112	2.000000	1.293437	-0.129542	0.040325
4	1.293437	2.000000	1.311281	-0.056588	0.017844
5	1.311281	2.000000	1.318989	-0.024304	0.007707
6	1.318989	2.000000	1.322283	-0.010362	0.003294
7	1.322283	2.000000	1.323684	-0.004404	0.001402
8	1.323684	2.000000	1.324279	-0.001869	0.000595
9	1.324279	2.000000	1.324532	-0.000793	0.000253
10	1.324532	2.000000	1.324639	-0.000336	0.000107
11	1.324639	2.000000	1.324685	-0.000143	0.000045
12	1.324685	2.000000	1.324704	-0.000060	0.000019
13	1.324704	2.000000	1.324712	-0.000026	0.000008
14	1.324712	2.000000	1.324715	-0.000011	0.000003
15	1.324715	2.000000	1.324717	-0.000005	0.000001
16	1.324717	2.000000	1.324717	-0.000002	0.000001

☒ Raíz aproximada encontrada: 1.324717
 Iteraciones realizadas: 16