

MÉTODO DE LA REGLA FALSA (REGULA FALSI)

Reporte Académico

1. Introducción

El método de la regla falsa, también conocido como *Regula Falsi*, es un método numérico para la aproximación de raíces de ecuaciones no lineales. Este método combina la seguridad del método de la bisección con la idea de aproximación lineal del método de la secante.

La Regula Falsi mantiene siempre el intervalo donde ocurre el cambio de signo, garantizando la existencia de la raíz, pero utilizando una aproximación más eficiente que la simple división del intervalo.

2. Objetivos

2.1. Objetivo general

Aplicar el método de la regla falsa para aproximar raíces de ecuaciones no lineales de forma confiable.

2.2. Objetivos específicos

- Analizar el fundamento matemático del método.
- Describir el algoritmo iterativo.
- Evaluar la convergencia del método.
- Resolver un problema práctico.

3. Fundamentación teórica

Sea una función continua $f(x)$ definida en un intervalo $[a, b]$ tal que:

$$f(a) \cdot f(b) < 0 \tag{1}$$

Entonces existe al menos una raíz $c \in (a, b)$. El método de la regla falsa aproxima la raíz utilizando la intersección de la recta secante con el eje x .

4. Fórmula del método

La aproximación de la raíz se obtiene mediante:

$$c = b - f(b) \frac{b - a}{f(b) - f(a)} \quad (2)$$

Este valor se utiliza para redefinir el intervalo conservando el cambio de signo.

5. Algoritmo del método

1. Elegir un intervalo inicial $[a, b]$ tal que $f(a)f(b) < 0$.
2. Calcular el punto de intersección c mediante la fórmula de la regla falsa.
3. Evaluar $f(c)$.
4. Reemplazar a o b manteniendo el cambio de signo.
5. Repetir hasta cumplir el criterio de parada.

6. Criterios de parada

El proceso iterativo se detiene cuando se cumple al menos uno de los siguientes criterios:

- $|f(c)| < \varepsilon$
- $|c_n - c_{n-1}| < \varepsilon$
- Se alcanza el número máximo de iteraciones.

7. Convergencia

El método de la regla falsa presenta convergencia lineal. En algunos casos, uno de los extremos del intervalo puede permanecer fijo durante muchas iteraciones, lo que reduce la velocidad de convergencia.

8. Ventajas y desventajas

8.1. Ventajas

- Garantiza la existencia de la raíz.

- Más rápido que la bisección en muchos casos.
- No requiere derivadas.

8.2. Desventajas

- Convergencia lenta en ciertos casos.
- Puede estancarse en uno de los extremos.
- Menos eficiente que Newton-Raphson.

9. Ejemplo de aplicación

Sea la función:

$$f(x) = x^3 - x - 2 \quad (3)$$

Con intervalo inicial $[1, 2]$, se obtiene la siguiente aproximación iterativa:

Iteración	a	b	c
1	1.0	2.0	1.333
2	1.333	2.0	1.462
3	1.462	2.0	1.521

La raíz aproximada es:

$$x \approx 1,521$$

10. Implementación computacional

A continuación se deja el espacio reservado para la implementación del método de la regla falsa en un lenguaje de programación (por ejemplo, Python):

```

1 def falsa_posicion(f, a, b, tol=1e-6, max_iter=100):
2     """
3     Encuentra la raíz usando el método de falsa posición
4     f: función
5     a, b: intervalo inicial [a, b]
6     tol: tolerancia
7     max_iter: número máximo de iteraciones
8     """
9     print("\n=== M TODO DE LA FALSA POSICI N ===")
10
11

```

```

12     if f(a) * f(b) > 0:
13         print("Error: f(a) y f(b) deben tener signos opuestos"
14             )
15         return None
16
17     print(f"'Iter':<6} {'a':<15} {'b':<15} {'c':<15} {'f(c)
18         ':<15} {'Error':<15}")
19     print("-" * 95)
20
21     c_anterior = a
22
23     for i in range(max_iter):
24         fa = f(a)
25         fb = f(b)
26
27         # F r m u l a   d e   l a   f a l s a   p o s i c i o n
28         c = (a * fb - b * fa) / (fb - fa)
29         fc = f(c)
30
31         if i > 0:
32             error = abs(c - c_anterior)
33         else:
34             error = abs(b - a)
35
36         print(f"{i+1:<6} {a:<15.8f} {b:<15.8f} {c:<15.8f} {fc
37             :<15.8e} {error:<15.8e}")
38
39         if error < tol or abs(fc) < tol:
40             print(f"\nConvergi en {i+1} iteraciones")
41             print(f"Ra z aproximada: x = {c:.8f}")
42             return c
43
44         if fa * fc < 0:
45             b = c
46         else:
47             a = c
48
49         c_anterior = c
50
51     print(f"\nNo convergi en {max_iter} iteraciones")

```

```
50     return c
51
52 # Ejemplo de uso: Encontrar raíz de  $x^2 - 2 = 0$ 
53 def f(x):
54     return x**2 - 2
55
56 # Ejecutar
57 raiz = falsa_posicion(f, 1, 2)
```

11. Aplicaciones

El método de la regla falsa se emplea en la resolución de ecuaciones no lineales en problemas de ingeniería, física, economía y análisis numérico computacional.

12. Conclusiones

El método de la regla falsa es una técnica numérica confiable que combina la seguridad del método de la bisección con la eficiencia del método de la secante. Aunque puede presentar problemas de estancamiento, sigue siendo una herramienta valiosa para la aproximación de raíces de ecuaciones no lineales.