

UNIVERSIDAD NACIONAL DEL ALTIPLANO

FACULTAD DE INGENIERÍA ESTADÍSTICA E INFORMÁTICA

Curso: Programación Numérica

Docente: Fred Cruz Torres

Estudiante: Nexu Yohan Mamani Yucra

Método de Regula Falsi

1. Definición del Método

El método de **Regula Falsi** (o de la Falsa Posición) es un algoritmo de búsqueda de raíces que combina la estrategia de "encierro" del método de bisección con la eficiencia de la interpolación lineal.

A diferencia de la bisección, que divide el intervalo $[a, b]$ por la mitad, este método traza una línea recta (secante) entre los puntos $(a, f(a))$ y $(b, f(b))$. La intersección de esta recta con el eje x proporciona una aproximación más precisa de la raíz en cada paso.

2. Desarrollo Matemático

La fórmula para encontrar el punto de corte c se deriva de la ecuación de la recta secante:

$$y - f(b) = \frac{f(b) - f(a)}{b - a}(x - b) \quad (1)$$

Haciendo $y = 0$ y despejando x (que llamaremos c), obtenemos:

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

3. Algoritmo y Procedimiento

1. Se verifica que la función sea continua y que $f(a) \cdot f(b) < 0$.
2. Se calcula el punto c mediante la fórmula de la falsa posición.
3. Se evalúa $f(c)$:
 - Si $f(a) \cdot f(c) < 0$, la raíz está en $[a, c]$. El nuevo límite superior es $b = c$.
 - Si $f(a) \cdot f(c) > 0$, la raíz está en $[c, b]$. El nuevo límite inferior es $a = c$.
4. Se repite el proceso hasta que $|f(c)| < \text{Tolerancia}$.

4. Implementación en Python

A continuación, se presenta una versión optimizada del programa que incluye corrección sintáctica de expresiones matemáticas.

```
1 import math
2
3 def corregir_expresion(expr):
4     """Mejora la sintaxis de la expresion ingresada por el
5     usuario."""
6     for char in ['_', '^']:
7         expr = expr.replace(char, '**')
8
9     nueva = ""
10    for i in range(len(expr)):
11        nueva += expr[i]
12        if i < len(expr)-1:
13            if (expr[i].isdigit() or expr[i] in 'x') and (
14                expr[i+1] in 'x('):
15                nueva += '*'
16    return nueva
17
18 def regula_falsi(f, a, b, tol, max_iter=100):
19     if f(a) * f(b) >= 0:
20         raise ValueError("Error: La funcion no cambia de
21         signo en el intervalo.")
22
23     print(f"{'Iter':<5} | {'c':<12} | {'f(c)':<12}")
24     print("-" * 35)
25
26     for i in range(max_iter):
27         fa, fb = f(a), f(b)
28         c = b - (fb * (b - a)) / (fb - fa)
29         fc = f(c)
30
31         print(f"{'i+1':<5} | {'c':<12.6f} | {'fc':<12.6e}")
32
33         if abs(fc) < tol:
34             return c
35
36         if fa * fc < 0:
37             b = c
38         else:
39             a = c
40
41     return c
42
43 # Bloque de ejecucion
44 if __name__ == "__main__":
45     raw_expr = input("Ingresa f(x): ")
46     expr = corregir_expresion(raw_expr)
47     f = lambda x: eval(expr, {"x": x, "math": math, "exp":
```

```

45     math.exp, "sin": math.sin})
46
47     a = float(input("Limite a: "))
48     b = float(input("Limite b: "))
49     t = float(input("Tolerancia: "))
50
51     res = regula_falsi(f, a, b, t)
52     print(f"\nRaiz encontrada: {res}")

```

Listing 1: Script de Regula Falsi en Python

5. Conclusiones

- El método de Regula Falsi garantiza la convergencia (a diferencia del método de la Secante) siempre que el intervalo inicial sea válido.
- Es generalmente más rápido que la bisección porque aprovecha la magnitud de los valores de la función para estimar la posición de la raíz.
- En la ingeniería estadística, este método es vital para resolver ecuaciones de máxima verosimilitud donde la función no es lineal.