

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 $|f(c)| <$ 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 expresión ingresada por el usuario."""
5     for char in ['^', '**']:
6         expr = expr.replace('^', '**')
7
8     nueva = ""
9     for i in range(len(expr)):
10        nueva += expr[i]
11        if i < len(expr)-1:
12            if (expr[i].isdigit() or expr[i] in 'x') and (
13                expr[i+1] in 'x('):
14                nueva += '*'
15
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 función no cambia de signo en el intervalo.")
21
22    print(f"{'Iter':<5} {'c':<12} {'f(c)':<12}")
23    print("-" * 35)
24
25    for i in range(max_iter):
26        fa, fb = f(a), f(b)
27        c = b - (fb * (b - a)) / (fb - fa)
28        fc = f(c)
29
30        print(f"{i+1:<5} {c:<12.6f} {fc:<12.6e}")
31
32        if abs(fc) < tol:
33            return c
34
35        if fa * fc < 0:
36            b = c
37        else:
38            a = c
39
40 # Bloque de ejecución
41 if __name__ == "__main__":
42     raw_expr = input("Ingresa la función: ")
43     expr = corregir_expresion(raw_expr)
44     f = lambda x: eval(expr, {"x": x, "math": math, "exp":
```

```

        math.exp , "sin": math.sin})
45
46 a = float(input("Limite\u00e1a:"))
47 b = float(input("Limite\u00e1b:"))
48 t = float(input("Tolerancia:"))
49
50 res = regula_falsi(f, a, b, t)
51 print(f"\nRaiz\u00e1encontrada:{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.