Universidad Nacional del Altiplano

Facultad de Ingeniería Estadística e Informática

Alumna: Leydy Griselda Aguilar Ccopa

Docente: Ing. Torres Cruz Fred

Trabajo N° 6 – Método Falsa Posición

El método de falsa posición es un procedimiento numérico iterativo que se utiliza para encontrar raíces reales de una ecuación no lineal de la forma:

$$f(x)=0$$

Es muy parecido al método de bisección, pero en lugar de tomar el punto medio del intervalo, utiliza una interpolación lineal entre los extremos del intervalo para estimar la raíz.
Esto lo hace más rápido que el método de bisección en muchos casos.

Si una función continua f(x) cambia de signo entre dos puntos a y b:

$$f(a) \cdot f(b) < 0$$

entonces hay una raíz entre ellos.
El método de falsa posición traza una recta (secante) entre los puntos (a, f(a)) y (b, f(b)), y el punto donde esa recta corta el eje X se toma como una nueva aproximación de la raíz.

El nuevo punto c se calcula como:

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

o de forma equivalente:

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

(Esencialmente es la ecuación de la recta secante, aplicada para encontrar la intersección con el eje X.)

CODIGO:

```
# Método de la Falsa Posición (Regla Falsi)

def falsa_posicion(f, a, b, tol=1e-6, max_iter=100):

    """
```

```
Encuentra la raíz usando el método de falsa posición

f: función

a, b: intervalo inicial [a, b]

tol: tolerancia

max_iter: número máximo de iteraciones

"""

print("\n=== MÉTODO DE LA FALSA POSICIÓN ===")


# Verificar que hay cambio de signo

if f(a) * f(b) > 0:

    print("Error: f(a) y f(b) deben tener signos opuestos")

    return None


print(f"{'Iter':<6} {'a':<15} {'b':<15} {'c':<15} {'f(c)':<15} {'Error':<15}")

print("-" * 95)


c_anterior = a


for i in range(max_iter):

    fa = f(a)

    fb = f(b)


    # Fórmula de la falsa posición

    c = (a * fb - b * fa) / (fb - fa)
```

```
    fc = f(c)

    if i > 0:
        error = abs(c - c_anterior)
    else:
        error = abs(b - a)

    print(f"{i+1:<6} {a:<15.8f} {b:<15.8f} {c:<15.8f} {fc:<15.8e} {error:<15.8e}")

    if error < tol or abs(fc) < tol:
        print(f"\nConvergió en {i+1} iteraciones")
        print(f"Raíz aproximada: x = {c:.8f}")
        return c

    # Actualizar intervalo
    if fa * fc < 0:
        b = c
    else:
        a = c

    c_anterior = c

print(f"\nNo convergió en {max_iter} iteraciones")
return c
```

```
# Ejemplo de uso: Encontrar raíz de x^2 - 2 = 0

def f(x):

    return x**2 - 2



# Ejecutar

raiz = falsa_posicion(f, 1, 2)
```

```
=== MÉTODO DE LA FALSA POSICIÓN ===
Iter    a               b               c               f(c)            Error
-------------------------------------------------------------------------------------------------
1       1.00000000      2.00000000      1.33333333      -2.22222222e-01 1.00000000e+00
2       1.33333333      2.00000000      1.40000000      -4.00000000e-02 6.66666667e-02
3       1.40000000      2.00000000      1.41176471      -6.92041522e-03 1.17647059e-02
4       1.41176471      2.00000000      1.41379310      -1.18906064e-03 2.02839757e-03
5       1.41379310      2.00000000      1.41414141      -2.04060810e-04 3.48310693e-04
6       1.41414141      2.00000000      1.41420118      -3.50127797e-05 5.97692905e-05
7       1.41420118      2.00000000      1.41421144      -6.00728684e-06 1.02550429e-05
8       1.41421144      2.00000000      1.41421320      -1.03068876e-06 1.75949467e-06
9       1.41421320      2.00000000      1.41421350      -1.76838272e-07 3.01881780e-07

Convergió en 9 iteraciones
Raíz aproximada: x = 1.41421350
```