UNIVERSIDAD NACIONAL DEL ALTIPLANO

Facultad de Ingeniería Estadística e Informática

Método de Bisección

Estudiante: Yohan Mamani Yucra
Docente: Fred Cruz Torres

1. Introducción y Teoría

El método de bisección es un algoritmo de búsqueda de raíces que se basa en el **Teorema de Bolzano**. Este postula que si una función continua $f(x)$ tiene signos opuestos en los extremos de un intervalo $[a, b]$, entonces existe al menos una raíz en dicho intervalo.

Matemáticamente, si $f(a) \cdot f(b) < 0$, el método divide el intervalo a la mitad calculando:

$$c_n = \frac{a_n + b_n}{2} \tag{1}$$

A diferencia de los métodos de punto abierto, la bisección es un método cerrado que garantiza convergencia. La cota de error después de n iteraciones se define como:

$$|E_n| \leq \frac{b - a}{2^n} \tag{2}$$

2. Algoritmo en Python

Se presenta una implementación robusta que incluye el control de iteraciones y la validación del intervalo inicial.

```python
import numpy as np

def biseccion(f, a, b, tol=1e-6):
    # Verificacion de Bolzano
    if f(a) * f(b) >= 0:
        print("El intervalo no contiene una raiz garantizada.")
        return None

    iteracion = 1
```

```python
    while (b - a) / 2 > tol:
        c = (a + b) / 2
        if f(c) == 0:
            return c
        elif f(a) * f(c) < 0:
            b = c
        else:
            a = c
        print(f"Iteracion {iteracion}: raiz aprox = {c}")
        iteracion += 1
    return (a + b) / 2

# Ejemplo: f(x) = x^3 - x - 2
f = lambda x: x**3 - x - 2
raiz = biseccion(f, 1, 2)
```

3. Conclusión

El método de bisección destaca por su estabilidad numérica y su facilidad de implementación. En el campo de la estadística e informática, es fundamental cuando se requiere asegurar la convergencia en funciones que no son suaves o donde el cálculo de la derivada es costoso. Su predictibilidad lo convierte en una herramienta esencial para el análisis numérico académico y profesional.