**Universidad Nacional del Altiplano**
**Facultad de Ingeniería Estadística e Informática**

**Docente:** Ing. Fred Torres Cruz
**Estudiante:** Luis Angel Quenaya Loza
**Código:** 241411

# Actividad N°06

**Método de Bisección**

## Resumen del Método de Bisección

El **método de bisección** es un procedimiento iterativo para aproximar las raíces de una función continua $f(x)$. Se basa en el teorema de Bolzano, que indica que si $f(a) \cdot f(b) < 0$ en un intervalo $[a, b]$, existe al menos una raíz dentro del intervalo.

El método consiste en dividir el intervalo a la mitad y seleccionar el subintervalo donde la función cambia de signo, repitiendo este proceso hasta alcanzar la tolerancia deseada:

$$x_m = \frac{a+b}{2}$$

### Importancia de graficar la función

Antes de iniciar, es recomendable graficar $f(x)$ para ubicar aproximadamente el intervalo donde la función cambia de signo y elegir valores iniciales $a$ y $b$.

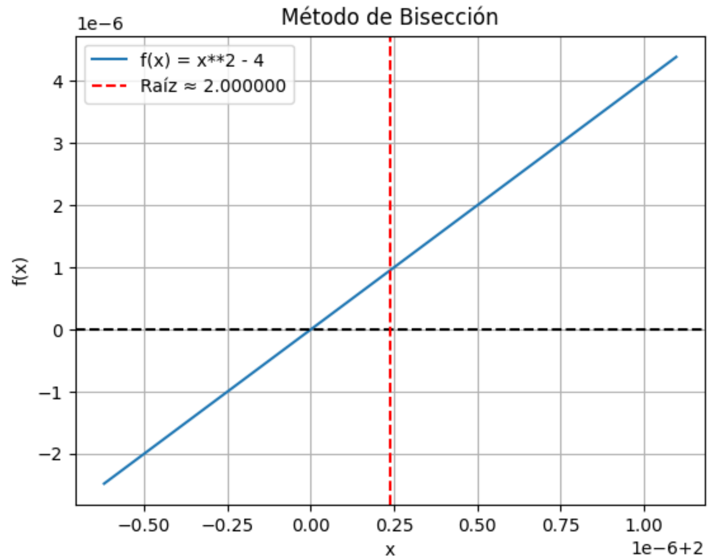
### Ventajas y limitaciones

- Siempre converge si la función es continua y $f(a) \cdot f(b) < 0$.
- Simple de implementar.
- Convergencia lineal (más lenta que secante o Newton-Raphson).
- Requiere que la función cambie de signo en el intervalo.

### Código en Python

```
import numpy as np
import matplotlib.pyplot as plt

func_str = input("Ingrese la funci n f(x): ")

```

```
def f(x):
    return eval(func_str, {"np": np, "x": x})

xmin = float(input("Ingrese el valor m nimo de x: "))
xmax = float(input("Ingrese el valor m ximo de x: "))

x = np.linspace(xmin, xmax, 400)
y = f(x)

plt.plot(x, y, color='blue', label=f"f(x)={func_str}")
plt.axhline(0, color='black', linestyle='--')
plt.axvline(0, color='black', linestyle='--')
plt.title("Gr fico de la funci n")
plt.xlabel("x")
plt.ylabel("f(x)")
plt.legend()
plt.grid(True)
plt.show()

x0 = float(input("Ingrese x0: "))
x1 = float(input("Ingrese x1: "))

tol = 1e-6
max_iter = 100

print("Iter | x0         | x1         | f(x0)      | f(x1)      | x2         | Error")
print("----------------------------------------------------------------------------")

for i in range(1, max_iter+1):
    f0 = f(x0)
    f1 = f(x1)
    if f1 - f0 == 0:
        print(f"Divisi n por cero en iteraci n {i}")
        break
    x2 = x1 - f1 * (x1 - x0) / (f1 - f0)
    error = abs(x2 - x1)
    print(f"{i:4d} | {x0:10.6f} | {x1:10.6f} | {f0:10.6f} | {f1:10.6f} | {x2:10.6f} | {error:10.6f}")
    if error < tol:
        print(f"Ra z aproximada: {x2:.6f} en {i} iteraciones")
        break
    x0, x1 = x1, x2
else:
    print("No se alcanz  convergencia"))
```




```
======== RESTART: C:/Users/LENOVO/AppData/Local/Programs/Python/Python312/biseccion.py =======
Ingrese la función f(x): x**2 - 4
Ingrese el valor inicial a: 0
Ingrese el valor inicial b: 3

Iter | a        | b        | xm       | f(xm)     | Error
------------------------------------------------------------
   1 | 0.000000 | 3.000000 | 1.500000 | -1.750000 | 1.500000
   2 | 1.500000 | 3.000000 | 2.250000 | 1.062500 | 0.750000
   3 | 1.500000 | 2.250000 | 1.875000 | -0.484375 | 0.375000
   4 | 1.875000 | 2.250000 | 2.062500 | 0.253906 | 0.187500
   5 | 1.875000 | 2.062500 | 1.968750 | -0.124023 | 0.093750
   6 | 1.968750 | 2.062500 | 2.015625 | 0.062744 | 0.046875
   7 | 1.968750 | 2.015625 | 1.992188 | -0.031189 | 0.023438
   8 | 1.992188 | 2.015625 | 2.003906 | 0.015640 | 0.011719
   9 | 1.992188 | 2.003906 | 1.998047 | -0.007809 | 0.005859
  10 | 1.998047 | 2.003906 | 2.000977 | 0.003907 | 0.002930
  11 | 1.998047 | 2.000977 | 1.999512 | -0.001953 | 0.001465
  12 | 1.999512 | 2.000977 | 2.000244 | 0.000977 | 0.000732
  13 | 1.999512 | 2.000244 | 1.999878 | -0.000488 | 0.000366
  14 | 1.999878 | 2.000244 | 2.000061 | 0.000244 | 0.000183
  15 | 1.999878 | 2.000061 | 1.999969 | -0.000122 | 0.000092
  16 | 1.999969 | 2.000061 | 2.000015 | 0.000061 | 0.000046
  17 | 1.999969 | 2.000015 | 1.999992 | -0.000031 | 0.000023
  18 | 1.999992 | 2.000015 | 2.000004 | 0.000015 | 0.000011
  19 | 1.999992 | 2.000004 | 1.999998 | -0.000008 | 0.000006
  20 | 1.999998 | 2.000004 | 2.000001 | 0.000004 | 0.000003
  21 | 1.999998 | 2.000001 | 2.000000 | -0.000002 | 0.000001
  22 | 2.000000 | 2.000001 | 2.000000 | 0.000001 | 0.000001

Raíz aproximada: 2.000000 encontrada en 22 iteraciones
```

Figura 1: Compilador y gráfica del método de Bisección.