

Universidad Nacional del Altiplano

Facultad de Ingeniería Estadística e Informática

Alumna: Leydy Griselda Aguilar Ccopa

Docente: Ing. Torres Cruz Fred

Trabajo N° 6 – Método Bisección

El método de bisección es un método numérico que sirve para encontrar una aproximación de la raíz (es decir, el valor de x donde $f(x)=0$) de una función continua.

Su nombre viene de que divide (“bisecca”) un intervalo en dos partes en cada paso.

El método se basa en el Teorema del Valor Intermedio, el cual dice:

Si $f(x)$ es continua en un intervalo $[a,b]$ y $f(a)$ y $f(b)$ tienen signos opuestos, entonces existe al menos un punto $c \in (a,b)$ tal que $f(c)=0$.

CODIGO:

Método de Bisección

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

"""

Encuentra la raíz usando el método de bisección

f: función

a, b: intervalo inicial $[a, b]$

tol: tolerancia

max_iter: número máximo de iteraciones

"""

```
    print("\n==== MÉTODO DE BISECCIÓN ===")
```

```
# 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"\n{'Iter':<6} {'a':<15} {'b':<15} {'c':<15} {'f(c)':<15} {'Error':<15}")
print("-" * 90)

for i in range(max_iter):
    c = (a + b) / 2
    fc = f(c)
    error = abs(b - a) / 2

    print(f"\n{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("\nConvergió en {} iteraciones".format(i+1))
        print("Raíz aproximada: x = {:.8f}".format(c))
        return c

    # Actualizar intervalo
    if f(a) * fc < 0:
        b = c
    else:
        a = c

print("\nNo convergió en {} iteraciones".format(max_iter))
```

```
return c
```

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

```
def f(x):
```

```
    return x**2 - 2
```

```
# Ejecutar
```

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

==== MÉTODO DE BISECCIÓN ===					
Iter	a	b	c	f(c)	Error
1	1.00000000	2.00000000	1.50000000	2.50000000e-01	5.00000000e-01
2	1.00000000	1.50000000	1.25000000	-4.37500000e-01	2.50000000e-01
3	1.25000000	1.50000000	1.37500000	-1.09375000e-01	1.25000000e-01
4	1.37500000	1.50000000	1.43750000	6.64062500e-02	6.25000000e-02
5	1.37500000	1.43750000	1.40625000	-2.24609375e-02	3.12500000e-02
6	1.40625000	1.43750000	1.42187500	2.17285156e-02	1.56250000e-02
7	1.40625000	1.42187500	1.41406250	-4.27246094e-04	7.81250000e-03
8	1.41406250	1.42187500	1.41796875	1.06353760e-02	3.90625000e-03
9	1.41406250	1.41796875	1.41601562	5.10025024e-03	1.95312500e-03
10	1.41406250	1.41601562	1.41503906	2.33554840e-03	9.76562500e-04
11	1.41406250	1.41503906	1.41455078	9.53912735e-04	4.88281250e-04
12	1.41406250	1.41455078	1.41430664	2.63273716e-04	2.44140625e-04
13	1.41406250	1.41430664	1.41418457	-8.20010900e-05	1.22070312e-04
14	1.41418457	1.41430664	1.41424561	9.06325877e-05	6.10351562e-05
15	1.41418457	1.41424561	1.41421509	4.31481749e-06	3.05175781e-05
16	1.41418457	1.41421509	1.41419983	-3.88433691e-05	1.52587891e-05
17	1.41419983	1.41421509	1.41420746	-1.72643340e-05	7.62939453e-06
18	1.41420746	1.41421509	1.41421127	-6.47477282e-06	3.81469727e-06
19	1.41421127	1.41421509	1.41421318	-1.07998130e-06	1.90734863e-06
20	1.41421318	1.41421509	1.41421413	1.61741718e-06	9.53674316e-07

```
Convergió en 20 iteraciones
```

```
Raíz aproximada: x = 1.41421413
```