

Método de Bisección

Resolución de ecuaciones no lineales

Profesor: Diego Valencia Enríquez

Universidad Mariana

17 de septiembre de 2025



¿Qué es el método de bisección?

Concepto básico

El método de bisección es un algoritmo de **búsqueda de raíces** que aplica el **teorema de Bolzano** para encontrar una solución aproximada de una ecuación $f(x) = 0$.

Teorema de Bolzano

Si f es continua en $[a, b]$ y $f(a) \cdot f(b) < 0$, entonces existe al menos un $c \in (a, b)$ tal que $f(c) = 0$.

- Método **robusto** y **convergente**.
- Convergencia **lenta** pero segura.
- Ideal para funciones continuas en un intervalo cerrado.



Algoritmo del método de bisección

Pasos a seguir

Dada $f(x)$ continua en $[a, b]$ con $f(a) \cdot f(b) < 0$:

- 1 Calcular el punto medio: $c = \frac{a+b}{2}$.
- 2 Evaluar $f(c)$.
- 3 Si $f(c) = 0$ (o muy cercano a 0), ¡se encontró la raíz!
- 4 Si $f(a) \cdot f(c) < 0$, la raíz está en $[a, c]$.
- 5 Si $f(c) \cdot f(b) < 0$, la raíz está en $[c, b]$.
- 6 Repetir hasta alcanzar la tolerancia deseada.

El error máximo después de n iteraciones es:

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

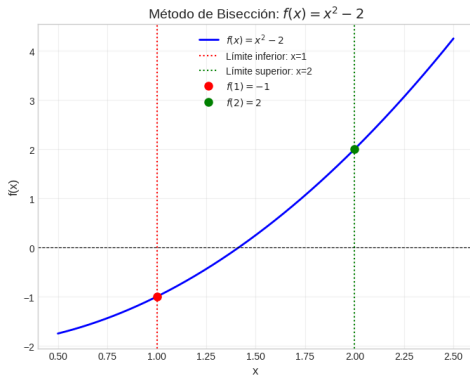


Ejemplo 1: $f(x) = x^2 - 2$

Encontrar $\sqrt{2}$

Queremos resolver $x^2 - 2 = 0$ en $[1, 2]$.

- $f(1) = -1$, $f(2) = 2$
- $f(1) \cdot f(2) < 0$



Iter	a	b	c	$f(c)$
1	1.0	2.0	1.5	0.25
2	1.0	1.5	1.25	-0.4375
3	1.25	1.5	1.375	-0.1094
4	1.375	1.5	1.4375	0.0664
5	1.375	1.4375	1.40625	-0.0225
6	1.40625	1.4375	1.421875	0.0217

Cuadro: Primeras 6 iteraciones del método

Resultado después de 6 iteraciones

$c \approx 1.421875$, Error ≤ 0.03125

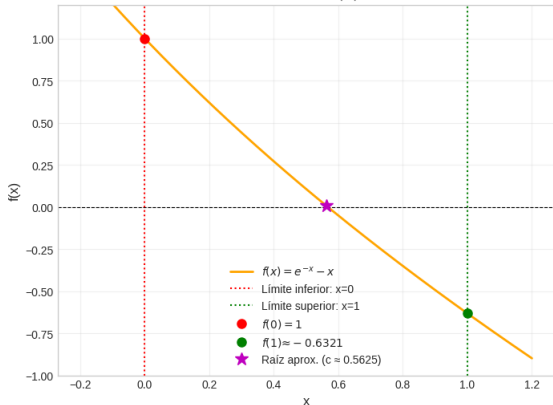


Ejemplo 2: $f(x) = e^{-x} - x$

Raíz en $[0, 1]$

- $f(0) = 1, f(1) = e^{-1} - 1 \approx -0,6321$
- $f(0) \cdot f(1) < 0$

Método de Bisección: $f(x) = e^{-x} - x$



Iter	a	b	c	f(c)
1	0.0	1.0	0.5	0.1065
2	0.5	1.0	0.75	-0.2776
3	0.5	0.75	0.625	-0.0897
4	0.5	0.625	0.5625	0.0073
5	0.5625	0.625	0.59375	-0.0415
6	0.5625	0.59375	0.578125	-0.0173

Cuadro: Iteraciones del método de bisección

Resultado después de 4 iteraciones

$$c \approx 0,5625$$
$$f(c) \approx 0,0073$$

Resultado después de 6 iteraciones

$$c \approx 0,578125$$
$$f(c) \approx -0,0173$$

Ejercicios propuestos

Resuelve las siguientes ecuaciones usando el método de bisección con tolerancia $\epsilon = 0,01$:

- 1 $f(x) = x^3 - x - 1 = 0$ en $[1, 2]$
- 2 $f(x) = \cos(x) - x = 0$ en $[0, 1]$
- 3 $f(x) = \ln(x) - 1 = 0$ en $[2, 3]$

Pregunta adicional: ¿Cuántas iteraciones se necesitan para alcanzar un error menor a 10^{-5} si $a = 0$, $b = 1$?



Ventajas y desventajas

Ventajas:

- Fácil de implementar.
- Siempre converge si se cumplen las condiciones.
- No requiere derivadas.

Desventajas:

- Convergencia lenta.
- Requiere un intervalo inicial válido.
- No aprovecha información adicional de $f(x)$.

Nota

Es útil para obtener una aproximación inicial para métodos más rápidos como Newton-Raphson.



Implementación en Python (opcional)

Código básico y ejemplo de uso

```
def biseccion(f, a, b, tol, max_iter=100):
    if f(a)*f(b) >= 0:
        raise ValueError("No hay cambio de signo")
    for i in range(max_iter):
        c = (a + b) / 2
        if abs(f(c)) < tol:
            return c
        if f(a)*f(c) < 0:
            b = c
        else:
            a = c
    return c
```

Ejemplo: Encontrar $\sqrt{2}$ con $f(x) = x^2 - 2$

```
def funcion_ejemplo(x):
    return x**2 - 2
```



¡Gracias por su atención!
¿Preguntas?

