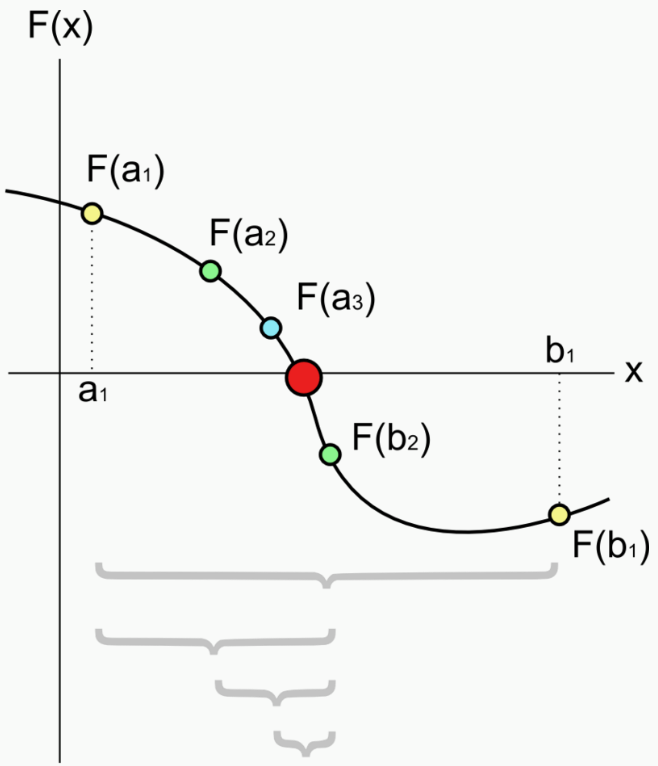
MÉTODOS NUMÉRICOS

Mtro. Sergio Castillo Carrizales

:U-ERRE

MÉTODO DE BISECCIÓN O INTERVALO MEDIO



Jesús Leonardo Jiménez González – ITC – 735817

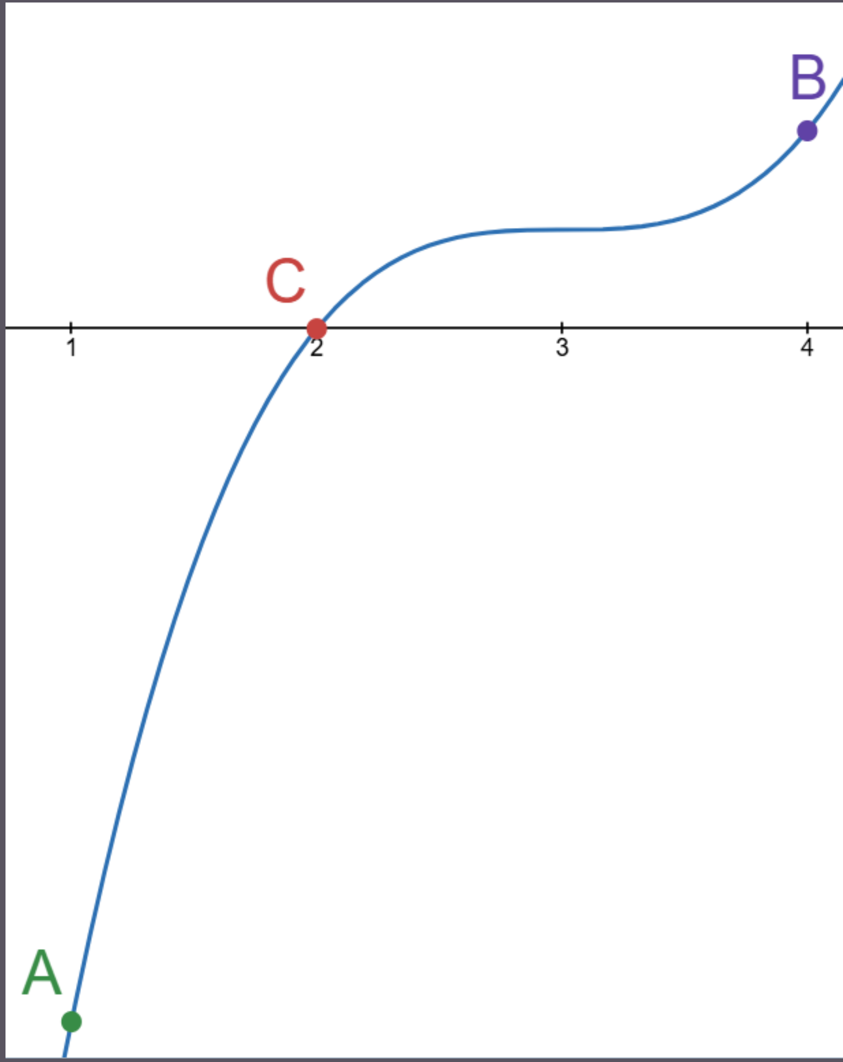
Definición del método

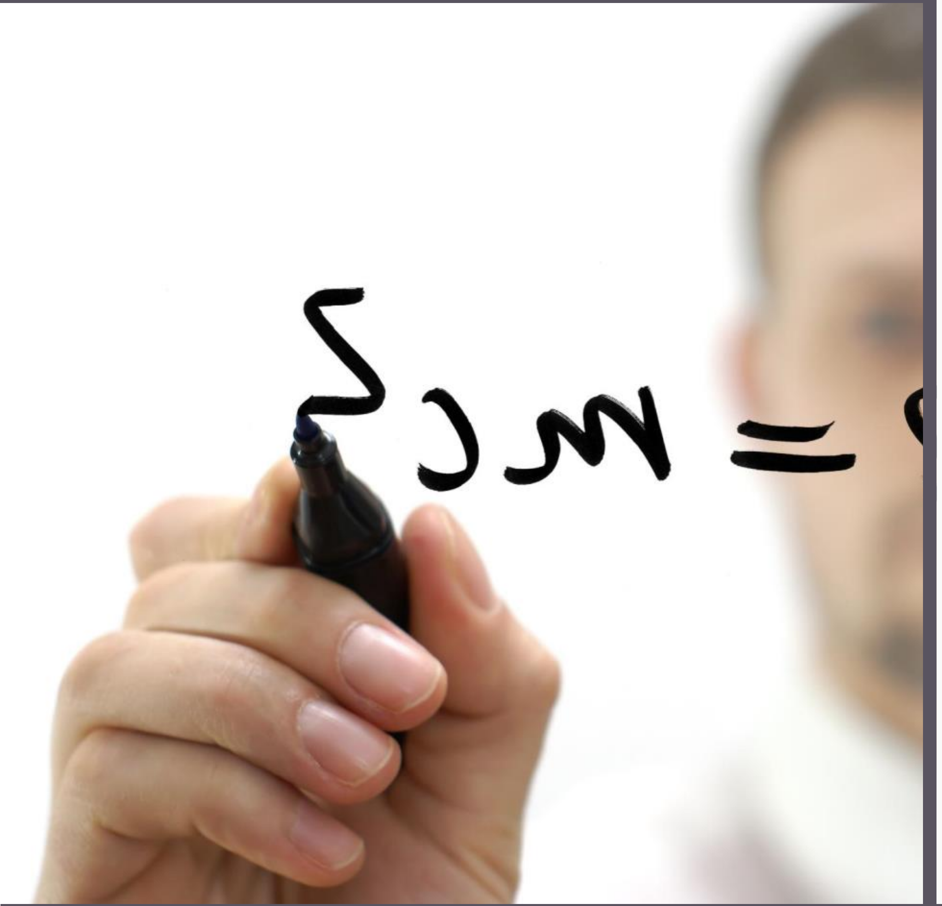
El método de bisección es un procedimiento numérico para encontrar raíces de **funciones continuas** en un **intervalo cerrado** [a,b].

Se evalúa el signo de la función en los extremos del intervalo y se reduce el intervalo, repitiendo el proceso hasta encontrar la raíz con la precisión deseada.

Antecedentes del método

La base de este método es el **Teorema de Bolzano o del valor intermedio**, el cual establece que si una función f(x) es **continua en un intervalo** [a,b] y el **signo** de la función evaluada es diferente en los extremos del intervalo entonces existe al menos un valor c de dicho intervalo en el que la función corta al eje X.





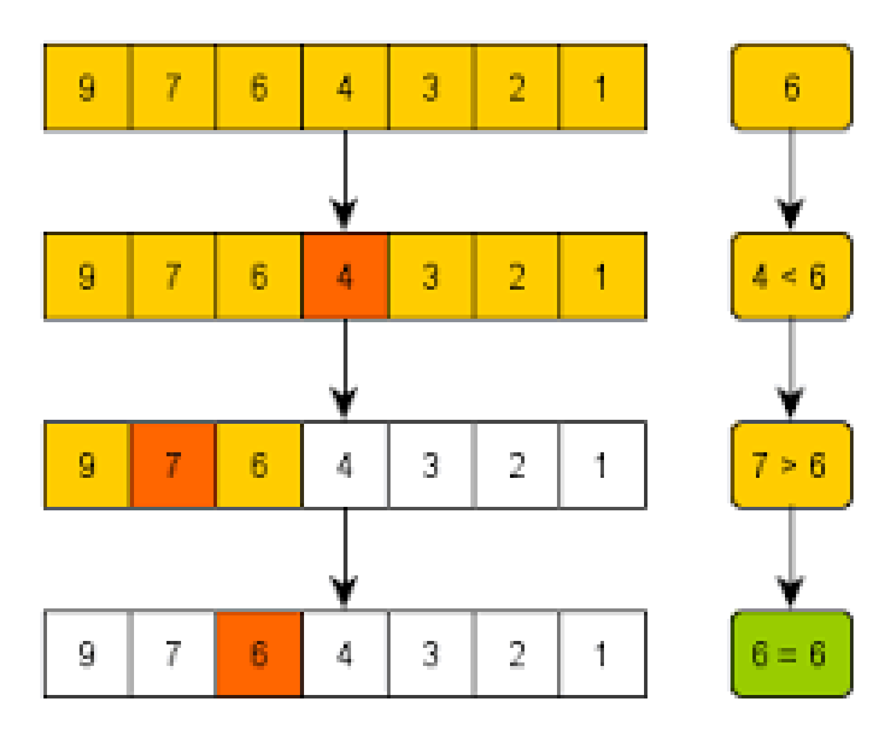
Relación con otros métodos numéricos

Método de Newton-Raphson

Este método utiliza derivadas para aproximar raíces y converge más rápido que el Método de Bisección en muchas situaciones.

Método de la Secante

El Método de la Secante es similar al de Newton-Raphson, pero no requiere derivadas, utiliza puntos secantes para aproximar raíces.

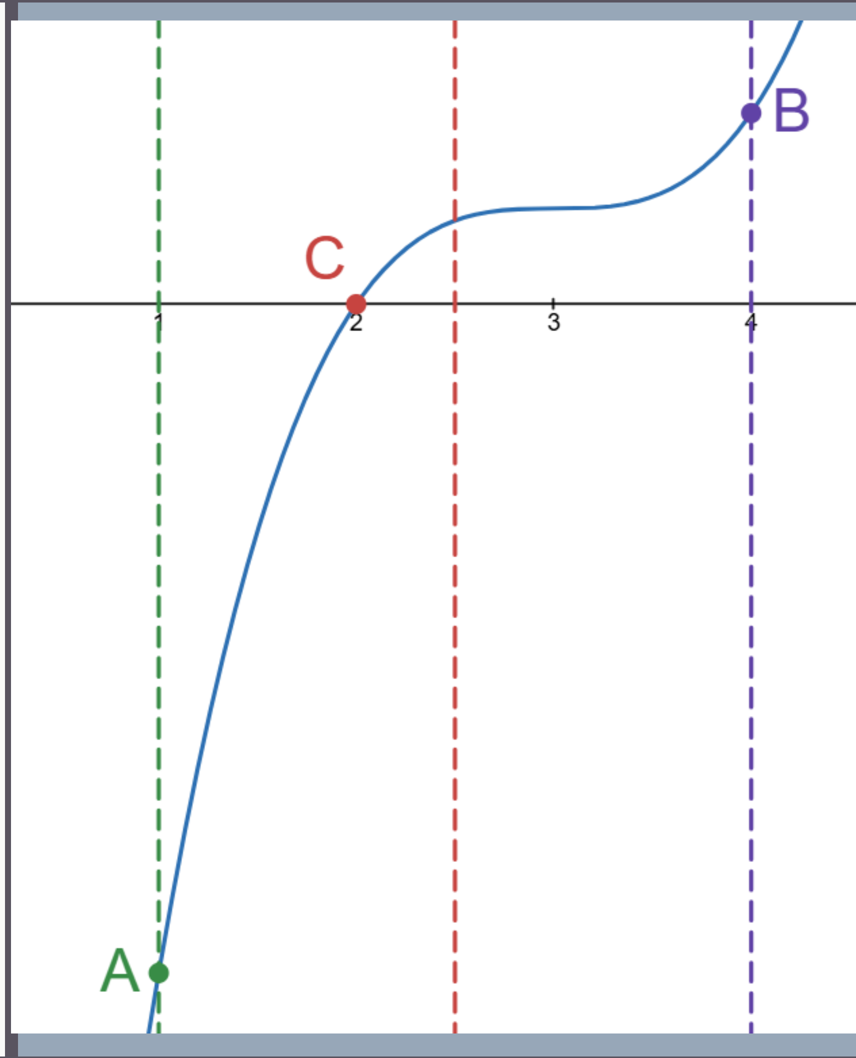


Método de Búsqueda Binaria

El método de búsqueda binaria es un algoritmo eficiente utilizado en programación para encontrar la posición de un elemento en una lista o arreglo ordenado. Su funcionamiento se basa en dividir el espacio de búsqueda en dos mitades en cada iteración, descartando la mitad donde no se encuentra el elemento, hasta localizarlo o determinar que no está presente.

Formula Matemática del Método

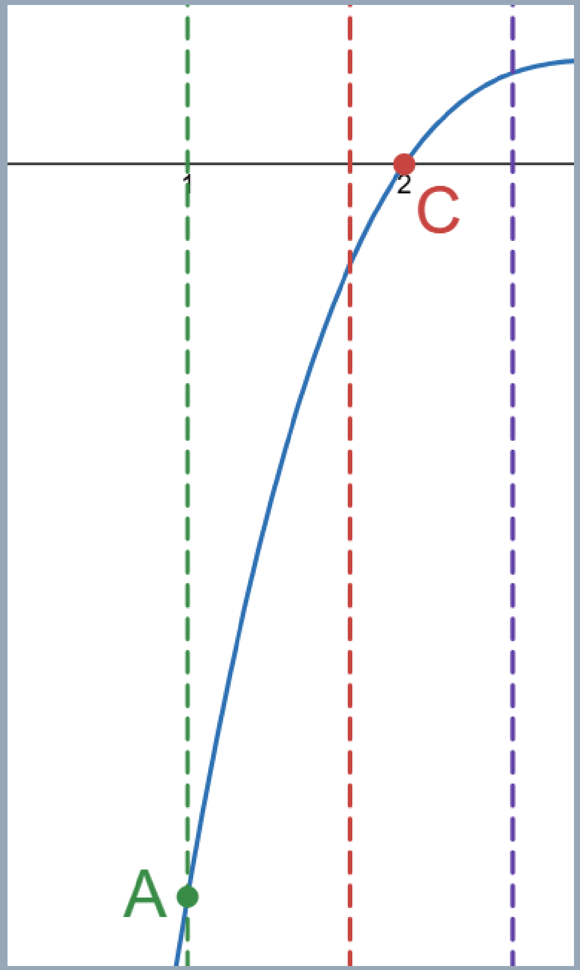
1. A partir de una función continua, se elige un intervalo [a,b] tal que f(a) y f(b) tengan signos opuestos.



Formula Matemática del Método

1. A partir de una función continua, se elige un intervalo [a,b] tal que f(a) y f(b) tengan signos opuestos.
2. Se calcula el punto medio c del intervalo:

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



Formula Matemática del Método

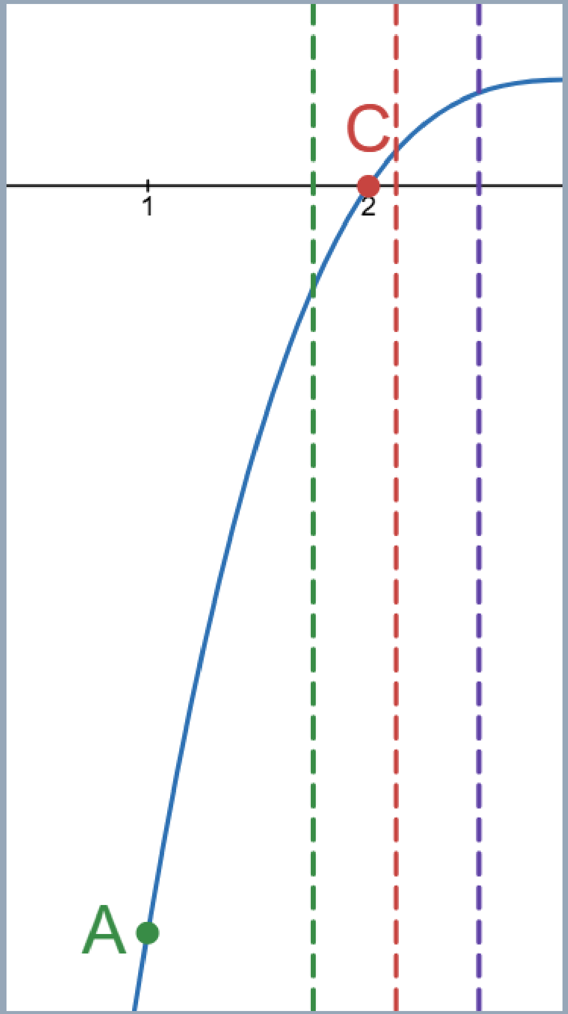
1. A partir de una función continua, se elige un intervalo [a,b] tal que f(a) y f(b) tengan signos opuestos.

2. Se calcula el punto medio c del intervalo:

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

3. Se evalúa f(c):

a)Si f(c)=0, entonces c es una raíz exacta.



Formula Matemática del Método

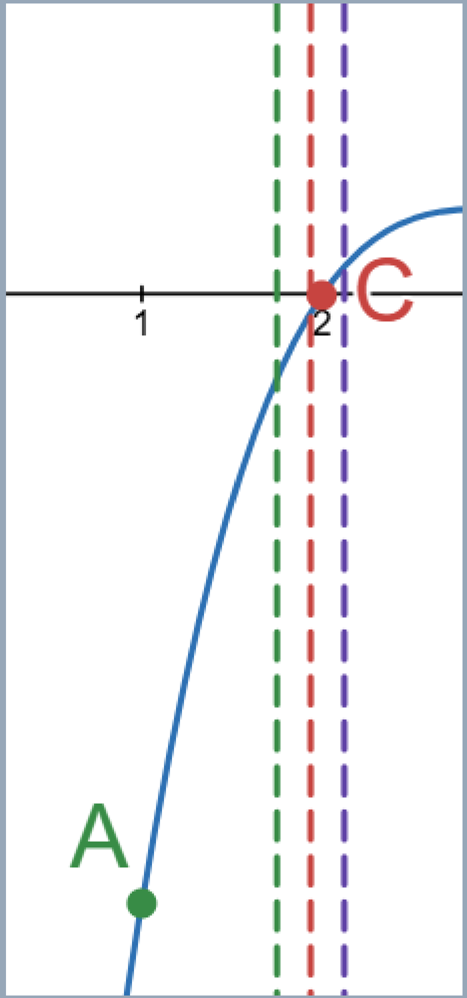
1. **A partir de una función continua, se elige un intervalo [a,b] tal que f(a) y f(b) tengan signos opuestos.**
2. **Se calcula el punto medio c del intervalo:**

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

3. **Se evalúa f(c):**

a)Si f(c)=0, entonces c es una raíz exacta.

b)Si f(c)*f(a) < 0, b=c, y se repite el proceso hasta alcanzar el margen de error planteado.



Formula Matemática del Método

1. **A partir de una función continua, se elige un intervalo [a,b] tal que f(a) y f(b) tengan signos opuestos.**

2. **Se calcula el punto medio c del intervalo:**

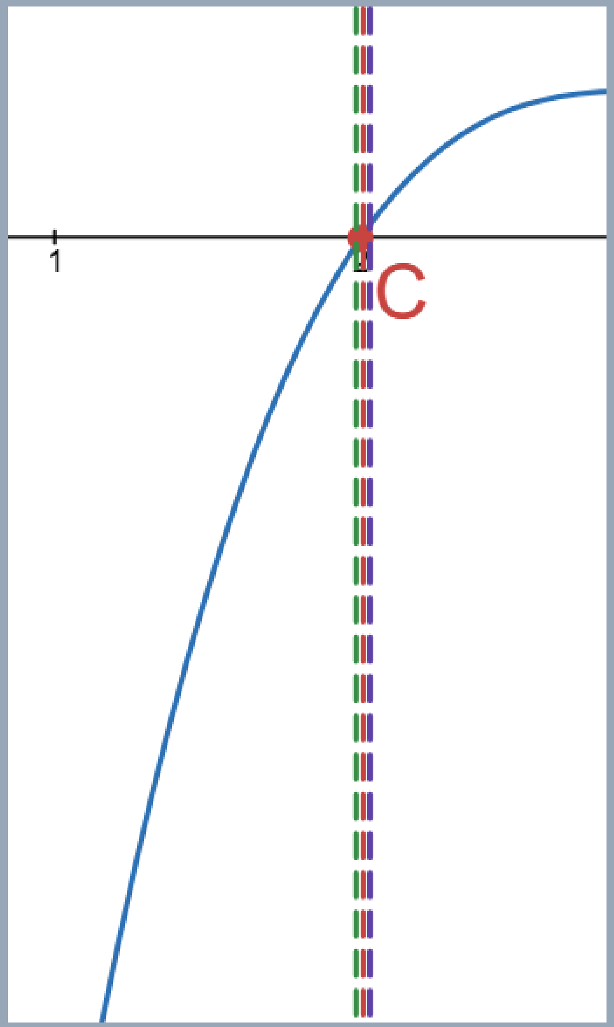
$$c = \frac{a+b}{2}$$

3. **Se evalúa f(c):**

a)Si f(c)=0, entonces c es una raíz exacta.

b)Si f(c)*f(a) < 0, b=c, y se repite el proceso hasta alcanzar el margen de error planteado.

c)Si f(c)*f(b) < 0, a=c, y se repite el proceso hasta alcanzar el margen de error planteado.



Algoritmo (En Python)

```python
import math

def funcion(x):
    return math.exp(x)+math.cos(x)-x**3

def biseccion(a, b, tolerancia):
    if funcion(a) * funcion(b) >= 0:
        print("El método de bisección no es aplicable.")
        return float('nan')

```

Definir una función

```python
import math

def funcion(x):
    return math.exp(x)+math.cos(x)-x**3

def biseccion(a, b, tolerancia):
    if funcion(a) * funcion(b) >= 0:
        print("El método de bisección no es aplicable.")
        return float('nan')

```

Verificar que tengan signos opuestos

```python
    while (b - a) >= tolerancia:
        c = (a + b) / 2
        if funcion(c) == 0.0:
            break
        elif funcion(c) * funcion(a) < 0:
            b = c
        else:
            a = c
        print(f"Aproximacion Actual es: {c:.5f}")
    return c
```

Repetir hasta alcanzar la tolerancia

```python
    while (b - a) >= tolerancia:
        c = (a + b) / 2
        if funcion(c) == 0.0:
            break
        elif funcion(c) * funcion(a) < 0:
            b = c
        else:
            a = c
        print(f"Aproximacion Actual es: {c:.5f}")
    return c
```

Calcular el
punto medio

```python
    while (b - a) >= tolerancia:
        c = (a + b) / 2
        if funcion(c) == 0.0:
            break
        elif funcion(c) * funcion(a) < 0:
            b = c
        else:
            a = c
        print(f"Aproximacion Actual es: {c:.5f}")
    return c
```

Terminar el ciclo si se encuentra la raíz

```python
    while (b - a) >= tolerancia:
        c = (a + b) / 2
        if funcion(c) == 0.0:
            break
        elif funcion(c) * funcion(a) < 0:
            b = c
        else:
            a = c
        print(f"Aproximacion Actual es: {c:.5f}")
    return c
```

Actualizar el intervalo

```python
    while (b - a) >= tolerancia:
        c = (a + b) / 2
        if funcion(c) == 0.0:
            break
        elif funcion(c) * funcion(a) < 0:
            b = c
        else:
            a = c
        print(f"Aproximacion Actual es: {c:.5f}")
    return c
```

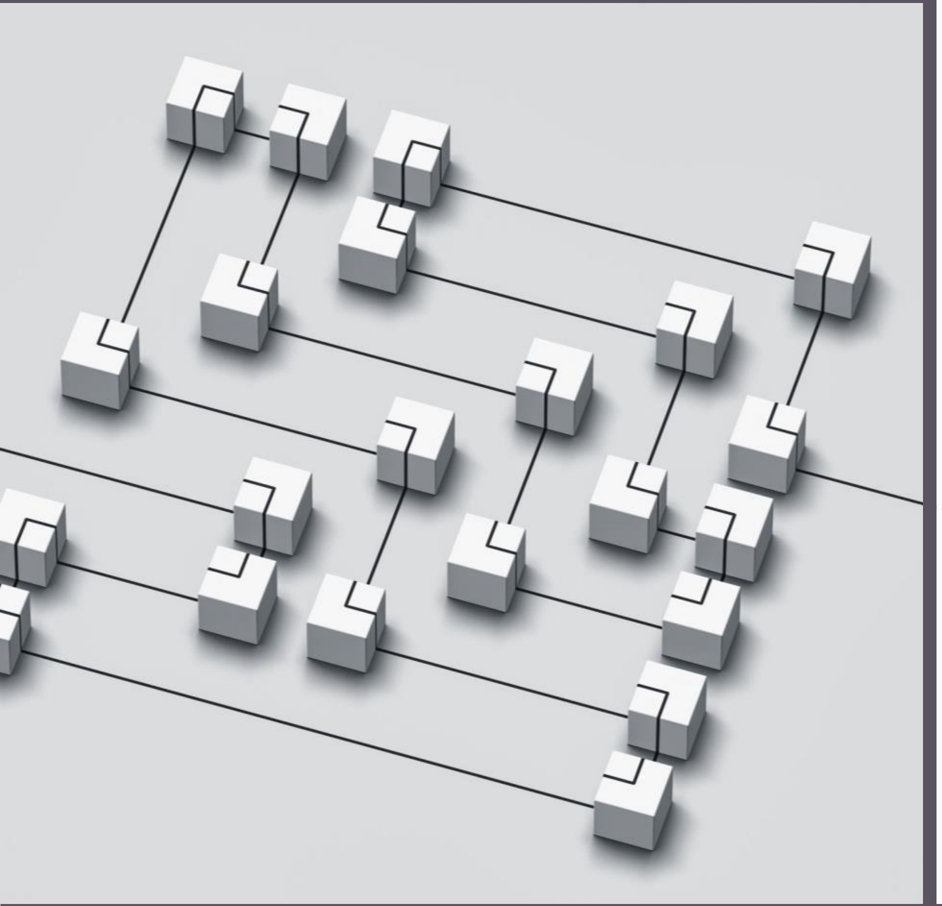
Mostrar el resultado de cada iteración

Solicitar extremos del Intervalo y ejecutar el algoritmo con los datos

```python
def main():
    a = float(input("Introduce el valor de a: "))
    b = float(input("Introduce el valor de b: "))
    tolerancia = float(input("Introduce la tolerancia: "))

    raiz = biseccion(a, b, tolerancia)
    print(f"La raíz es: {raiz:.5f}")
```

¿Qué aplicación tiene en la vida cotidiana?

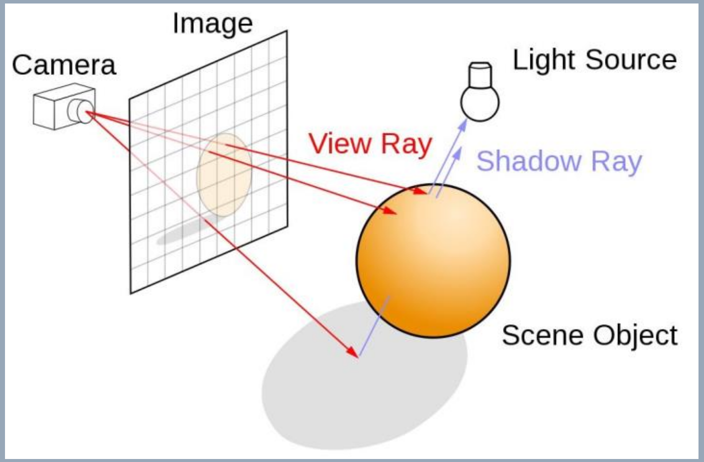


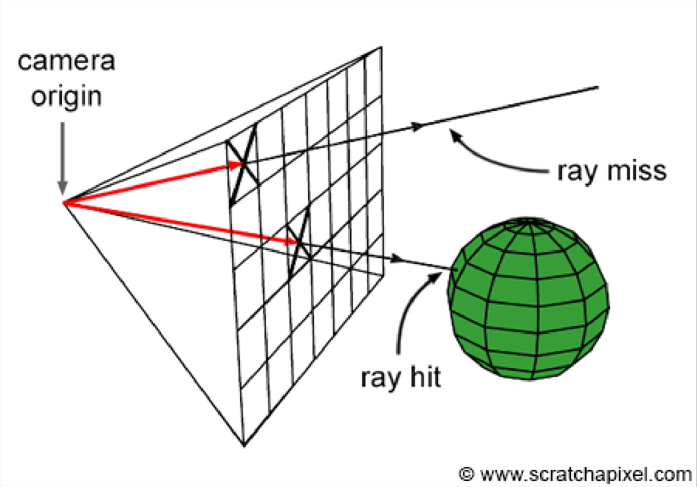
Una aplicación del método es en la conversión entre coordenadas cartesianas y polares, útil en áreas como el procesamiento de señales, robótica y cálculo vectorial. Al tener ecuaciones no lineales como $\boldsymbol{x - r cos\theta = 0}$ o $\mathbf{y} - \boldsymbol{r sin\theta = 0}$, donde se conoce una sola variable, el método ayuda a encontrar las incógnitas.

También se aplica en el trazado de rayos, que es una técnica utilizada en gráficos por computadora para simular cómo los rayos de luz interactúan con los objetos en un entorno.

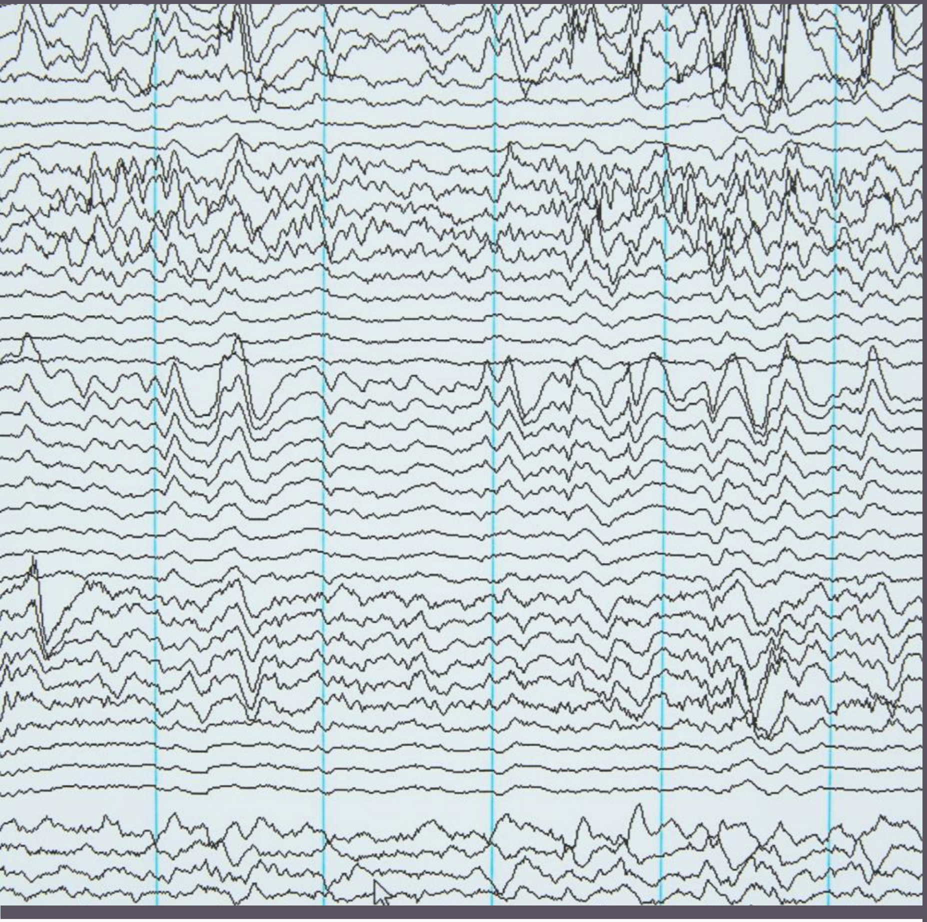
El método de bisección se puede utilizar para encontrar el ángulo de incidencia óptimo para un rayo.

Se estima un intervalo que se espera contenga el valor óptimo del ángulo de incidencia. A partir de un valor inicial aproximado (generalmente el valor medio del intervalo), el método de bisección ajusta el intervalo hasta encontrar el ángulo óptimo.





EJEMPLO PRÁCTICO



Planteamiento del problema

Encontrar el ángulo óptimo **θ** en el intervalo $[0, \pi/2]$ donde la reflexión de un rayo en una superficie es mínima ($f(\theta) = 0$).

La función que describe el ángulo de incidencia está dada por: $\boldsymbol{f(\theta) = sin(\theta) - 0.5}$

Teniendo en cuenta una tolerancia de intervalo de 0.1

1. Revisar que exista cambio de signo en el intervalo:

$$\boldsymbol{f(0) = sin(0) - 0.5 = -0.5}$$

$$\boldsymbol{f(\pi/2) = sin(\pi/2) - 0.5 = 0.5}$$

2. Aplicar el Método

Iteración 1:

$$c = \frac{0 + \boldsymbol{\pi}/2}{2} = \boldsymbol{\pi}/4$$

$$\boldsymbol{f(\pi/}4\boldsymbol{) = sin(\pi/}4\boldsymbol{) - 0.5 \approx 0.2071}$$

Nuevo intervalo [0, $\boldsymbol{\pi}/4$], $\boldsymbol{\pi}/4$>0.1

Iteración 2:

$$c = \frac{0 + \boldsymbol{\pi}/4}{2} = \boldsymbol{\pi}/8$$

$$\boldsymbol{f(\pi/}8\boldsymbol{) = sin(\pi/}8\boldsymbol{) - 0.5 \approx -0.1173}$$

Nuevo intervalo [$\boldsymbol{\pi}/8$, $\boldsymbol{\pi}/4$], $\boldsymbol{\pi}/4$- $\boldsymbol{\pi}/8$ >0.1

Iteración 3:

$$c = \frac{\boldsymbol{\pi}/8 + \boldsymbol{\pi}/4}{2} = 3\boldsymbol{\pi}/16$$

$$\boldsymbol{f}(3\boldsymbol{\pi}/16) = \boldsymbol{sin}(3\boldsymbol{\pi}/16) - \mathbf{0.5} \approx \mathbf{0.0556}$$

<u>Nuevo intervalo $[\boldsymbol{\pi}/8, 3\boldsymbol{\pi}/16]$, $\boldsymbol{\pi}/4$>0.1</u>

Iteración 4:

$$c = \frac{\boldsymbol{\pi}/8 + 3\boldsymbol{\pi}/16}{2} = \mathbf{5}\boldsymbol{\pi}/\mathbf{32}$$

$$\boldsymbol{f}(5\boldsymbol{\pi}/32) = \boldsymbol{sin}(5\boldsymbol{\pi}/32) - \mathbf{0.5} \approx -\mathbf{0.0286}$$

<u>Nuevo intervalo $[5\boldsymbol{\pi}/32, 3\boldsymbol{\pi}/16]$, $3\boldsymbol{\pi}/16$-$5\boldsymbol{\pi}/32$<0.1</u>

Entonces el ángulo óptimo de incidencia θ es aproximadamente: 5π/32 radianes

Referencias

4.1 Método de la bisección. (n.d.). https://www.uv.es/~diaz/mn/node18.html

Admin. (2021, August 4). Bisection Method - definition, procedure, and example. BYJUS. https://byjus.com/maths/bisection-method/

Testbook. (2023, June 6). Bisection Method: Numerical Method for finding Roots with Examples. Testbook. https://testbook.com/maths/bisection-method

Repaso sobre el teorema del valor intermedio (artículo) | Khan Academy. (n.d.). Khan Academy. https://es.khanacademy.org/math/ap-calculus-ab/ab-limits-new/ab-1-16/a/intermediate-value-theorem-review

Mathful. (n.d.). What is Bisection Method: A Comprehensive Guide | Mathful. https://mathful.com/hub/bisection-method

Mineo, C., Cerniglia, D., & Mohseni, E. (2022). Solving ultrasonic ray tracing in parts with multiple material layers through Root-Finding methods. Ultrasonics, 124, 106747. https://doi.org/10.1016/j.ultras.2022.106747

Fernández, J. L. (n.d.). Teorema de Bolzano. Fisicalab. https://www.fisicalab.com/apartado/teorema_bolzano

https://en.wikipedia.org/wiki/Ray_tracing_(graphics)