

**Tarea 4 - Métodos numéricos**  
**Giovanni Gamaliel López Padilla**

## Problema 1

**Implement the following algorithms: Bisection, Newton, and Secant methods.**

### Método de bisección

El método de bisección se basa en que si encontramos un intervalo  $[a, b]$ , tal que  $f(a)f(b) < 0$ , entonces por el teorema del valor intermedio, el intervalo  $[a, b]$  contiene al menos una raíz de la función  $f$ . Entonces en este método se ira acotando el intervalo para localizar la raíz.

El criterio para detener la búsqueda es definido en la ecuación 1.

$$\frac{|b - a|}{\max\{1, |b|\}} < \epsilon \quad (1)$$

donde  $\epsilon = 10^{-6}$ . La implementación de este algoritmo se encuentra dentro de las carpetas [Problema\\_2](#) y [Problema\\_3](#) en el archivo [bisection.h](#).

### Método de Newton

El método de Newton supone que existe una  $f \in C^2[a, b]$  y  $x_0 \in [a, b]$ . Se tiene que se puede aproximar a la raíz  $x_r$  tal que  $f(x_0) \neq 0$  y  $|x_r - x_0| \ll 1$ . Entonces  $x_r$  puede aproximarse usando la secuencia de la ecuación 2.

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \quad (2)$$

Los criterios empleados para detener la búsqueda son los siguientes:

- Máximo de intentos:

Se uso el máximo de intentos definido en la ecuación 3.

$$K_{max} = \log_2 \left( \frac{x_0}{\epsilon} \right) \quad (3)$$

donde  $\epsilon = 10^{-6}$ .

- Diferencia relativa:

Se empleo el criterio de la ecuación 1 usando como  $b = x_{k+1}$  y  $a = x_k$ .

La implementación de este algoritmo se encuentra dentro de las carpetas [Problema\\_2](#) y [Problema\\_3](#) en el archivo [newton.h](#).

## Método de la secante

El método de la secante es semejante al de Newton. Su diferencia radica en que el método de Newton necesita el valor de la derivada, en cambio el método de la secante aproxima este valor. La aproximación es calculada en la ecuación 4.

$$f'(x_k) = \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}} \quad (4)$$

Por lo tanto, la secuencia para la aproximación de la raíz es definida en la ecuación 5.

$$x_{k+1} = \frac{x_{k-1}f(x_k) - x_kf(x_{k-1})}{f(x_k) - f(x_{k-1})} \quad (5)$$

El criterio para detener la búsqueda del algoritmo que se emplea es el definido en la ecuación 1. La implementación de este algoritmo se encuentra dentro de las carpetas Problema.2 y Problema.3 en el archivo [secant.h](#).

## Problema 2

Use the previous methods to compute a zero of the following functions:

$$e^x + 2^{-x} + 2\cos(x) - 6 = 0 \quad 1 \leq x \leq 2 \quad (6)$$

$$\ln(x+1) + \cos(x-1) = 0 \quad 1.3 \leq x \leq 2 \quad (7)$$

El parámetro de entrada para el método de Newton es el valor intermedio del intervalo definido como:

$$x_m = \frac{x_1 + x_0}{2}$$

Usando los métodos del problema 1 se obtuvieron los resultados de la tabla 1.

Función	Bisección	Newton	Secante
Función 6	1.829383	1.829384	1.829384
Función 7	1.397748	1.397748	1.397748

**Tabla 1:** Soluciones a las ecuaciones 6 y 7 usando los métodos de bisección, Newton y secante.

El output esperado del programa es el siguiente:

```

1
2 Raices de la funcion 1 en el intervalo
3 [1.000000,2.000000] con el metodo:
4 Biseccion:
5 x = 1.829383
6 Newton:
7 x = 1.829384
8 Secante:
9 x = 1.829384
10
11
12 Raices de la funcion 2 en el intervalo
13 [1.300000,2.000000] con el metodo:
14 Biseccion:
15 x = 1.397748
16 Newton:
17 x = 1.397748
18 Secante:
19 x = 1.397748

```

El programa se encuentra en la carpeta [Problema\\_2](#). Para compilar el programa se debe ingresar la siguiente línea:

```
1 gcc -Wall -Wextra -Werror -pedantic -ansi -o main.out main.c -lm -std=c11
```

## Problema 3

The fourth-degree polynomial:

$$f(x) = 230x^4 + 18x^3 + 9x^2 - 221x - 9 \quad (8)$$

has two real zeros, one in  $[-1, 0]$  and the other in  $[0, 1]$ . Attempt to approximate these zeros to within  $10^6$  using the algorithms implemented in exercise (1). Use the endpoints of each interval as the initial approximations in the Bisection and Secant algorithms and the midpoints as the initial approximation for the Newton Method.

Usando los métodos del problema 1, se obtuvieron los resultados mostrados en la tabla 2. Con estos resultados se comprueba que el método de la secante puede salir del intervalo que se le dio como parámetro y encontrar una solución.

Intervalos	Bisección	Newton	Secante
$[-1, 0]$	-0.040661	-0.040659	-0.040659
$[0, 1]$	0.962399	0.962398	-0.040659

**Tabla 2:** Resultados obtenidos usando la ecuación 8 usando los métodos de bisección, Newton y secante.

El programa se encuentra en la carpeta [Problema\\_3](#). Para compilar el programa se debe ingresar la siguiente línea:

```
1 gcc -Wall -Wextra -Werror -pedantic -ansi -o main.out main.c -lm -std=c11
```

El output esperado del programa es el siguiente:

```
1
2 Raices de la funcion en el intervalo
3 [-1.000000,0.000000] con el metodo:
4 Biseccion:
5 x = -0.040661
6 Newton:
7 x = -0.040659
8 Secante:
9 x = -0.040659
10
11
12 Raices de la funcion en el intervalo
13 [0.000000,1.000000] con el metodo:
14 Biseccion:
15 x = 0.962399
16 Newton:
17 x = 0.962398
18 Secante:
19 x = -0.040659
```