

Ensenada, Baja California a 05 de noviembre del 2020



Practica No.5

Métodos cerrados: de la bisección y de la falsa posición.

Profesor: Roilhi Frajo Ibarra Hernández

Alumno: Fabian Díaz Fajardo

Grupo: 021

Desarrollo de la practica.

Tomar como ejemplo las funciones: $f(x)=2x \cos(2x)-(x+1)^2$ en un intervalo $[-3, -2]$ y

$f(x)=\ln(\text{abs}(x))+x^2-4$ en un intervalo $[1, 2]$

-Realizar un programa en lenguaje C donde se realice lo siguiente:

- Codificar las funciones, es decir, que sean definidas por el usuario (la trigonométrica podría llamarse por ejemplo "funcion1" y la logarítmica "funcion2").

- Codificar como funciones los métodos de bisección y de la falsa posición. Se debe recordar que para el primer caso se necesita definir también la derivada de la función.

- Codificar una función de error absoluto.

Función 1.

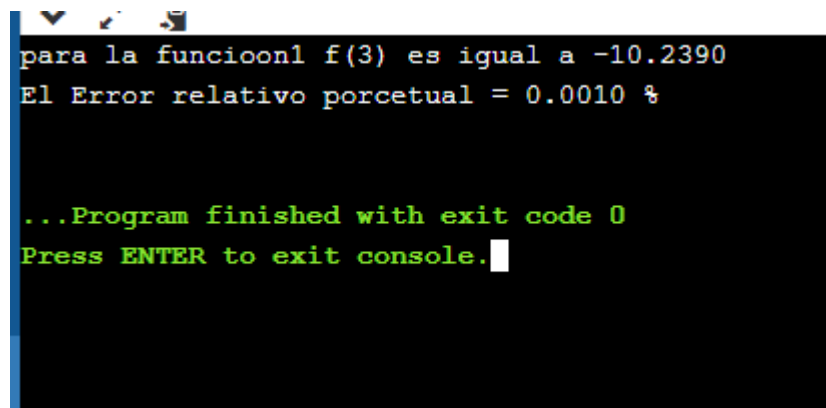
Error absoluto y función.

```
#include <stdio.h>
#include <math.h>

float funcion1(float x_entrada);
float ErrorRelativo(float ValorAnalitico, float ValorNumerico);
float ValorNumerico, ValorAnalitico, x_ent;
int main()
{
    x_ent=3;
    ValorAnalitico= -10.2389;
    ValorNumerico= -10.2390;
    printf ("para la funcioon1 f(3) es igual a %2.4f\n", funcion1(x_ent));
    printf ("El Error relativo porcetual = %2.4f %%\n", ErrorRelativo(ValorAnalitico, ValorNumerico));
}

float funcion1(float x_entrada){
    //  $2x \cdot \cos(2x) - (x+1)^2$ 
    float fx;
    fx=(2*(x_entrada)*cos(2*x_entrada))-pow((x_entrada+1),2);
    return fx;
}

float ErrorRelativo(float ValorAnalitico, float ValorNumerico){
    //epsilon= abs (ValorAnalitico-Valornumerico/ValorNumerico)*100
    float epsilon;
    epsilon= fabs((ValorAnalitico - ValorNumerico)/ValorNumerico)*100;
    return epsilon;
}
```



```
para la funcioon1 f(3) es igual a -10.2390
El Error relativo porcetual = 0.0010 %

...Program finished with exit code 0
Press ENTER to exit console.
```

Método de bisección.

```
#include <stdio.h>
#include <math.h>
#define RAIZ(x) (2*(x)*cos(2*x))-pow((x+1),2)
int main() {
    float a, b, xi, xa, Ea;
    int iteraciones;
    printf("Dame el valor de b: ");
    scanf("%f", &b);
    printf("Dame el valor de a: ");
    scanf("%f", &a);
    iteraciones = 0;
    xa = 0.0;
    Ea = 0;
    printf("%12s %10s %10s %10s %10s\n", "Iteraciones", "b", "a", "Xi", "Error Aproximado");
    do{
        iteraciones++;
        xi = (b + a) / 2;
        if(iteraciones >= 0) {
            Ea = ((xi - xa) / xi) * 100; // Error aproximado
        }
        Ea = fabs(Ea); // valor absoluto del número
        printf("%12d %10f %10f %10f %10f\n", iteraciones, b, a, xi, fabs(Ea));
        xa = xi;
        if(RAIZ(xi) > 0) {
            b = xi;
        } else{
            a = xi;
        }
    } while(Ea > 0.5 || (Ea == 0.0 && iteraciones == 1));
    printf("\nLa raiz es de f(x) = x + 2; es: %f\n", xi);
    return 0;
}
```

```
Dame el valor de b: -2
Dame el valor de a: -3
Iteraciones      b          a          Xi Error Aproximado
1 -2.000000 -3.000000 -2.500000 100.000000
2 -2.000000 -2.500000 -2.250000 11.111112
3 -2.000000 -2.250000 -2.125000 5.882353
4 -2.125000 -2.250000 -2.187500 2.857143
5 -2.187500 -2.250000 -2.218750 1.408451
6 -2.187500 -2.218750 -2.203125 0.709220
7 -2.187500 -2.203125 -2.195312 0.355872

La raiz es de f(x) = x + 2; es: -2.195312
```

Método de falsa posición.

```
#include <stdio.h>
#include <math.h>
float
funcion1 (float x)
{
    float y = 2 * (x) * cos (2 * x) - pow ((x + 1), 2);
    return y;
}
float
falsaposicion (float a, float b, int iter)
{
    float xi, xi_anterior = 0.0, errorRP;
    int i = 1;
    printf ("i \t a \t b \t xi \t Erp\n");
    do
    {
        xi = b - ((funcion1 (b) * (a - b)) / (funcion1 (a) - funcion1 (b)));
        errorRP = ((xi - xi_anterior) / xi) * 100;
        xi_anterior = xi;
        printf ("%d \t %.4f \t %.4f \t %.4f \t %.4f \n", i, a, b, xi,
                fabs (errorRP));
        if (funcion1 (a) * funcion1 (xi) < 0)
        {
            b = xi;
        }
        else
        {
            a = xi;
        }
        i++;
    }
    while (funcion1 (a) * funcion1 (xi) != 0 && i <= iter);
    return xi;
}
int
main ()
{
    float a = -3, b = -2, iteraciones = 5;
    float raiz;
    raiz = falsaposicion (a, b, iteraciones);
    printf ("\n La Raiz aproximada es %.4f", raiz);
    printf ("\n\n");
    return 0;
}
```

i	a	b	xi	Erp
1	-3.0000	-2.0000	-2.1419	100.0000
2	-3.0000	-2.1419	-2.1818	1.8288
3	-3.0000	-2.1818	-2.1897	0.3575
4	-3.0000	-2.1897	-2.1910	0.0623
5	-3.0000	-2.1910	-2.1913	0.0106

La Raiz aproximada es -2.1913

Función 2

Error absoluto y función.

```
#include <stdio.h>
#include <math.h>
float funcion2(float x_entrada);
float ErrorRelativo(float ValorAnalitico, float ValorNumerico);
float ValorNumerico, ValorAnalitico, x_ent;
int main()
{
    x_ent= 5;
    ValorAnalitico= 22.6094;
    ValorNumerico= 22.6666; // Este numero es solo para saber si funciona el error relativo
    printf("para la funcion2 f(5) es igual a %2.4f\n", funcion2(x_ent));
    printf ("El Error relativo porcentual = %2.4f %%\n", ErrorRelativo(ValorAnalitico, ValorNumerico));
}

float funcion2(float x_entrada){
    //  $\ln|x| + x^2 - 4$ 
    float fx;
    fx= (log(x_entrada)) + pow(x_entrada,2) -4;
    return fx;
}

float ErrorRelativo(float ValorAnalitico, float ValorNumerico){
    //epsilon= abs (ValorAnalitico-Valornumerico/ValorNumerico)*100
    float epsilon;
    epsilon= fabs((ValorAnalitico - ValorNumerico)/ValorNumerico)*100;
    return epsilon;
}
```

```
para la funcion2 f(5) es igual a 22.6094
El Error relativo porcentual = 0.2524 %
```

Método de bisección.

```
include <stdio.h>
#include <math.h>
#define RAIZ(x) (log(x)) + pow(x,2) -4 // Definimos la funcion de la que se busca la raiz
int main() {
float a, b, xi, xa, Ea;
int iteraciones;
printf("Dame el valor de b: ");
scanf("%f", &b);
printf("Dame el valor de a: ");
scanf("%f", &a);
iteraciones = 0;
xa = 0.0;
Ea = 0;
printf("%12s %10s %10s %10s %10s\n","Iteraciones", "b", "a", "Xi", "Error Aproximado");
do{
iteraciones++;
xi = (b + a) / 2;
if(iteraciones >= 0) {
Ea = ((xi - xa) / xi) * 100; // Error aproximado
}
Ea = fabs(Ea); // valor absoluto del número
printf("%12d %10f %10f %10f %10f\n",iteraciones, b, a, xi, fabs(Ea));
xa = xi;
if(RAIZ(xi) > 0) {
b = xi;
} else{
a = xi;
}
} while(Ea > 0.5 || (Ea == 0.0 && iteraciones == 1));
printf("\nLa raiz es de f(x) = x + 2; es: %f\n", xi);
return 0;
}
```

Dame el valor de b: 2


Dame el valor de a: 1

Iteraciones	b	a	Xi	Error Aproximado
1	2.000000	1.000000	1.500000	100.000000
2	2.000000	1.500000	1.750000	14.285715
3	2.000000	1.750000	1.875000	6.666667
4	1.875000	1.750000	1.812500	3.448276
5	1.875000	1.812500	1.843750	1.694915
6	1.843750	1.812500	1.828125	0.854701
7	1.843750	1.828125	1.835938	0.425532

La raiz es de f(x) = x + 2; es: 1.835938

Método de falsa posición.

```
#include <stdio.h>
#include <math.h>
float
funcion2 (float x)
{
    float y;
    y = (log (x)) + pow (x, 2) - 4;
    return y;
}
float
falsaposicion (float a, float b, int iter)
{
    float xi, xi_anterior = 0.0, errorRP;
    int i = 1;
    printf ("i \t a \t b \t xi \t Erp\n");
    do
    {
        xi = b - ((funcion2 (b) * (a - b)) / (funcion2 (a) - funcion2 (b)));
        errorRP = ((xi - xi_anterior) / xi) * 100;
        xi_anterior = xi;
        printf ("%d \t %.6f \t %.6f \t %.6f \t %.6f \n", i, a, b, xi,
                fabs (errorRP));
        if (funcion2 (a) * funcion2 (xi) < 0)
        {
            b = xi;
        }
        else
        {
            a = xi;
        }
        i++;
    }
    while (funcion2 (a) * funcion2 (xi) != 0 && i <= iter);
    return xi;
}
int
main ()
{
    float a = 1, b = 2, iteraciones = 5;
    float raiz;
    raiz = falsaposicion (a, b, iteraciones);
    printf ("\n Raiz aproximada es %.6f", raiz);
    printf ("\n\n");
    return 0;
}
```



i	a	b	xi	Erp
1	1.000000	2.000000	1.812315	100.000000
2	1.812315	2.000000	1.840191	1.514837
3	1.840191	2.000000	1.841069	0.047656
4	1.841069	2.000000	1.841096	0.001496
5	1.841096	2.000000	1.841097	0.000045

Raiz aproximada es 1.841097