Anthony Fabian Ramirez Orellana

Carne: 9490-22-958

sección: “A”

catedrático: Jorge Pérez

**Tarea No. 2**

17/02/24

**TAREA No 2**:

1. Codificar el algoritmo de **Bisección** en **C++**. Debe incluir el código escrito.

La salida debe considerar las siguientes columnas:

* Número de iteración
* El valor de **a** en cada iteración
* El valor de **b** en cada iteración
* El valor **pn** calculado en cada iteración
* El valor de **f(pn)** para cada iteración
* El error cometido en cada iteración: **(b-a)/2**

Utilizando la solución anterior (**C++**), resuelva lo siguiente:

Aplicar el método de **Bisección** para encontrar soluciones aproximadas dentro de **10-4** y **10-8** para las siguientes funciones en el intervalo dados:

1. **f(x) = x – 2-x = 0**; intervalo: **0 ≤ x ≤ 1**
2. **f(x) = ex – x2 + 3x – 2 = 0**; intervalo: **[0, 1]**
3. **f(x) = 2x cos(x) – (x + 1)2 = 0**; intervalo: **[-1, 0]**
4. **f(x) = x cos(x) – 2x2 + 3x – 1 = 0**; intervalos: **[0.2, 0.3]** y **[1.2, 1.3]**

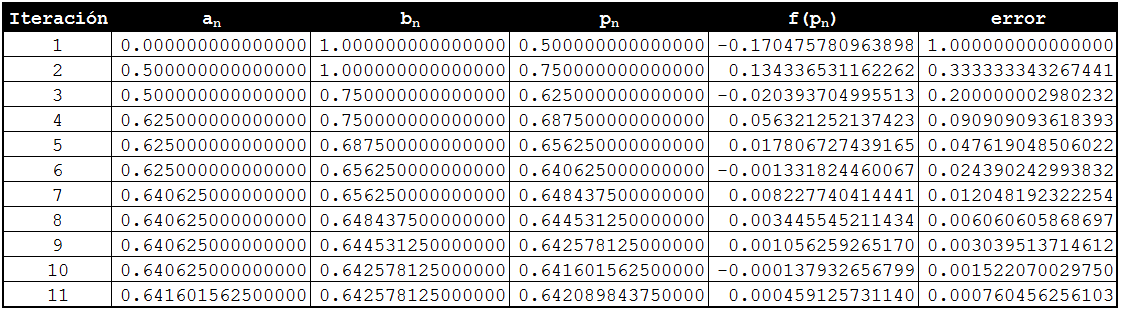
Utilizar un máximo de **100** iteraciones en cada caso. La salida debe considerar **15 decimales**.

**Ejemplo**: Aplicar el método de **Bisección** para obtener una aproximación de la solución a la siguiente función:



Utilizar una tolerancia de **10-3** y un máximo de iteraciones de **100**. El intervalo a considerar es **[0, 1]**.

La salida del algoritmo es una tabla como la que se muestra a continuación:



La solución aproximada se obtiene en la décima primera iteración, con **p11 = 0.64208984375** y **f(p11) = 0.00045912573114**.

Programa:

#include <iostream>//biblioteca estandar para operaciones de entrada/salida (input/output stream)

#include <math.h>//archivo de cabecera que contiene funciones matematicas basicas, es una libreria de terminos para uso de operaciones matematicas

#include <iomanip>//biblioteca se usa para ajustar decimales, esta libreria la utilice unicamente para mostrar los 15 decimales de respuesta.

using namespace std;//importa todo el espacio de nombres std al codigo actual, se utiliza mayormente para utilizar el prefijo std::

/\*Autor: Anthony Fabian Ramirez Orellana. Nombre del programa: Metodo de biseccion\*/

void **cabezaTabla**(){//creo un proceso de tipo vacio, ya que no necesito que me de una respuesta como una funcion, sino que realize una porcion de codigo, enviandole a la propia funcion ningun parametro, ya que solo pintara la consola.

**system**("cls");//codigo del cmd para limpiar la consola

    cout**<<**"-----------------------------------------------------------------------------------------------------------------------"**<<** **endl**;//cout se utiliza para imprimir en consola el contenido dentro de los parentesis.

    cout**<<**"Iteracion      Numero A             Numero B              Numero P              f(p)                   Error            "**<<** **endl**;//<< se puede usar para concatenar texto a desplegar, endl es una intruccion de salto de linea.

    cout**<<**"-----------------------------------------------------------------------------------------------------------------------"**<<** **endl**;

}

void **contTabla**(int ite, float na, float nb, float np, float fp, float ne){//en este proceso si envio parametros para la correcta colocacion de los datos.

    if(ite<=9){//un condicional if, que indica que si se cumple ite menor o igual a 9 realizara el contenido dentro del mismo, solo corre un espacio el codigo dependiendo si las iteraciones son de 1 o 2 digitos, ya que si no se veran desfazadas.

        cout**<<**"------------------------------------------------------------------------------------------------------------------------"**<<** **endl**;

        cout**<<**" "**<<**ite**<<**"            "**<<fixed<<setprecision**(15)**<<**na**<<**"     "**<<fixed<<setprecision**(15)**<<**nb**<<**"     "**<<fixed<<setprecision**(15)**<<**np**<<**"     "**<<fixed<<setprecision**(15)**<<**fp**<<**"     "**<<fixed<<setprecision**(15)**<<**ne**<<endl**;

    }

    if(ite>9){//Este condicional es para todos los datos despues de la iteracion 9 ya que todo el contenido de la fila despues de la iteracion se debe correr un espacio a la izquierda para que no se vea desfazado.

        cout**<<**"------------------------------------------------------------------------------------------------------------------------"**<<** **endl**;

        cout**<<**" "**<<**ite**<<**"           "**<<fixed<<setprecision**(15)**<<**na**<<**"     "**<<fixed<<setprecision**(15)**<<**nb**<<**"     "**<<fixed<<setprecision**(15)**<<**np**<<**"     "**<<fixed<<setprecision**(15)**<<**fp**<<**"     "**<<fixed<<setprecision**(15)**<<**ne**<<endl**;

    }

}

int **main**() {//funcion principal de tipo int, devuelve un valor entero, me da igual si es int o void ya que es simplemente la funcion principal de mi programa, y por el modo que plantee la programacion puedo hacerlo un vucle.

    int IT = 100, i = 1, opcion = 0;//declaro 3 variables de tipo entero, la primera para las iteraciones que bien podria ser una constante, la segunda para el bucle for aunque despues reafirmo su valor por si acaso, la tercera para las opciones del menu.

    float  numeroA = 0, numeroB = 0, numeroP = 0, FA = 0, FP = 0, prueba = 0, TOL = 0, numeroE = 0;//declaro 8 variables de tipo flotante, para tener en cuenta los decimales, todos para uso de los calculos necesarios

    menu://etiqueta para usar el goto, esta etiqueta marca todo el codigo despues de ella como un marcador para regresar u omitir codigo

    cout**<<** "" **<<** **endl**;//imprimo en la consola todo el menu, con saltos de linea para que se vea bonito.

    cout**<<**"Ingrese la opcion que desea ejecutar:" **<<** **endl**;

    cout**<<**"0) Regresar a este menu." **<<** **endl**;

    cout**<<**"1) f(x) = x-(2^-x) = 0 con intervalo: 0 <= x <= 1   tolerancia: 10^-4" **<<** **endl**;

    cout**<<**"2) f(x) = x-(2^-x) = 0 con intervalo: 0 <= x <= 1   tolerancia: 10^-8" **<<** **endl**;

    cout**<<**"3) f(x) = (e^x)-(x^2)+3x-2 = 0 con intervalo: [0, 1]   tolerancia: 10^-4" **<<** **endl**;

    cout**<<**"4) f(x) = (e^x)-(x^2)+3x-2 = 0 con intervalo: [0, 1]   tolerancia: 10^-8" **<<** **endl**;

    cout**<<**"5) f(x) = 2xcos(x)-(x+1)^2 = 0 con intervalo: [-1, 0]   tolerancia: 10^-4" **<<** **endl**;

    cout**<<**"6) f(x) = 2xcos(x)-(x+1)^2 = 0 con intervalo: [-1, 0]   tolerancia; 10^-8" **<<** **endl**;

    cout**<<**"7) f(x) = xcos(x)-2(x^2)+3x-1 = 0 con intervalos: [0.2, 0.3] y [1.2, 1.3]   tolerancia: 10^-4" **<<** **endl**;

    cout**<<**"8) f(x) = xcos(x)-2(x^2)+3x-1 = 0 con intervalos: [0.2, 0.3] y [1.2, 1.3]   tolerancia: 10^-8" **<<** **endl**;

    cout**<<**"9) Salir.";

    //espero una respuesta del usuario con cin para guardarla en la variable opcion.

    if( (cin**>>**opcion).**fail**() ){//compurebo si la entrada de datos falla, ya que el usuario puede ingresar un dato no valido, como una letra.

**system**("cls");//de ser asi, limpio la consola

        cin.**clear**();//reseteo los flags(uno o mas bits que almacenan valor binario o codigo)

**fflush**(**stdin**);//limpio el buffer(espacio de memoria para almacenar datos antes de procesarlos) de entrada

        opcion = 99;//igual la variable opcion a 99 para que se vaya al caso default(por defecto) de mi condicional switch

    }

    switch(opcion){//creo un condicional switch que utiliza la variable opcion para cada caso.

        case 0:// esta etiqueta determina el caso de que el usuario ponga un 0

**system**("cls");//limpio la consola

            goto menu;//realizo el salto a la etiqueta menu, para crear un semi-bucle

        break;//termina el caso, en este caso esto esta de mas, pero por si algun fallo yo lo deje, ya que esto solo indica al codigo que ya no siga prestando atencion a los demas casos.

        case 1://en caso de que la entrada sea 1

**system**("cls");//limpio la consola

**cabezaTabla**();//llamo al proceso cabezaTabla(), mandando ningun valor dentro de los parentesis ya que no lo necesito, que solo pinta la cabezera de la tabla en la consola.

            TOL = **pow**(10, -4);//igualo la variable TOL a la verdadera tolerancia usada en el problema a 10^-4

            numeroA = 0;//igualo la variable numeroA a cero ya que es el primer digito del intervalo, para usarlo en la funcion

            numeroB = 1;//Igualo la variable numeroB a 1 que es el segundo valor del intervalo

            FA = (numeroA - (**pow**(2, -numeroA)));//igualo la variable FA a la funcion, cambiando la x por la variable numeroA, utilizando asi mismo la funcion pow de math.h para elevar un numero a cierta potencia.

            for(i = 1; i < IT; i++){//creo un bucle for, con la variable i = 1 para que comienze en dicho numero, luego declaro que i debe llegar hasta IT-1 ya que se utiliza el menor que, y declaro que i avance de 1 en 1

                numeroP = ((numeroA + numeroB)/2);//siguiendo con el algoritmo del calculo igualo la variable numeroP al calculo de la variable numeroA + numeroB luego el resultado lo divido en 2

                FP = (numeroP - (**pow**(2, -numeroP)));//luego igualo la variable FP al calculo de la funcion cambiando la x por la variable p

                prueba = ((numeroB-numeroA)/2);//luego igualo la variable prueba al calculo de la variable numeroB - numeroA y el resultado lo divido entre 2

                numeroE = **abs**((numeroB-numeroA)/2);//luego a la variable numeroE la igualo a la instruccion de math.h abs que es para usar el valor absoluto, calculando numeroB - numeroA y el resultado dividiendolo en 2

**contTabla**(i, numeroA, numeroB, numeroP, FP, numeroE);//llamo al metodo conTabla mandando como datos las variables anteriormente usadas, para desplegar en orden los datos en consola dentro del metodo.

                if(FP == 0 || prueba < TOL){//creo un condicional para validar el exito del metodo numerico, igualando FP a 0 o si prueba es menor a TOL ejecutara lo que se encuantra dentro del mismo.

                    cout**<<**"Proceso finalizado exitosamente en la iteracion: "**<<** i **<<** **endl**;

                    cout**<<**"La solucion aproximada es p: "**<<fixed<<setprecision**(15)**<<** numeroP **<<** **endl**;//muestro el resultado usando fixed<<setprecision(15) para que el resultado me lo mustre con 15 decimales de presicion.

                    cout**<<**"Con f(p): "**<<fixed<<setprecision**(15)**<<** FP **<<** **endl**;

                    goto parar;//utilizo el goto para ir a la etiqueta parar, para asi ignorar las siguientes  instrucciones

                }

                if(FA \* FP > 0){//de no cumplirse el condicional anterior valido si FA \* FP es mayor a cero, de ser cierto

                    numeroA = numeroP;// igualo la variable numeroA a la variable numeroP

                    FA = FP;//Igualo FA a Fp para realizar la siguiente iteracion

                }else{//de no cumplirse el condicional anterior entonces

                    numeroB = numeroP;//igual la variable numeroB a numeroP

                }

            }

            cout**<<** "El metodo fracaso despues de " **<<** IT **<<** " iteraciones." **<<** **endl**;//despligo en consola que el metodo fracaso en IT que contienen el nuemro maximo de iteraciones

        break;

        case 2:

**system**("cls");

**cabezaTabla**();

            TOL = **pow**(10, -8);

            numeroA = 0;

            numeroB = 1;

            FA = (numeroA - (**pow**(2, -numeroA)));

            for(i = 1; i < IT; i++){

                numeroP = ((numeroA + numeroB)/2);

                FP = (numeroP - (**pow**(2, -numeroP)));

                prueba = ((numeroB-numeroA)/2);

                numeroE = **abs**((numeroB-numeroA)/2);

**contTabla**(i, numeroA, numeroB, numeroP, FP, numeroE);

                if(FP == 0 || prueba < TOL){

                    cout**<<**"Proceso finalizado exitosamente en la iteracion: "**<<** i **<<** **endl**;

                    cout**<<**"La solucion aproximada es p: "**<<fixed<<setprecision**(15)**<<** numeroP **<<** **endl**;

                    cout**<<**"Con f(p): "**<<fixed<<setprecision**(15)**<<** FP **<<** **endl**;

                    goto parar;

                }

                if(FA \* FP > 0){

                    numeroA = numeroP;

                    FA = FP;

                }else{

                    numeroB = numeroP;

                }

            }

            cout**<<** "El metodo fracaso despues de " **<<** IT **<<** " iteraciones." **<<** **endl**;

        break;

        case 3:

**system**("cls");

**cabezaTabla**();

            TOL = **pow**(10, -4);

            numeroA = 0;

            numeroB = 1;

            FA = (**exp**(numeroA)-(**pow**(numeroA, 2))+(3\*numeroA)-2);//utilizo exp de la libreria math.h para utilizar el numero e y elevarlo a cierto exponente

            for(i = 1; i < IT; i++){

                numeroP = ((numeroA + numeroB)/2);

                FP = (**exp**(numeroP)-(**pow**(numeroP, 2))+(3\*numeroP)-2);

                prueba = ((numeroB-numeroA)/2);

                numeroE = **abs**((numeroB-numeroA)/2);

**contTabla**(i, numeroA, numeroB, numeroP, FP, numeroE);

                if(FP == 0 || prueba < TOL){

                    cout**<<**"Proceso finalizado exitosamente en la iteracion: "**<<** i **<<** **endl**;

                    cout**<<**"La solucion aproximada es p: "**<<fixed<<setprecision**(15)**<<** numeroP **<<** **endl**;

                    cout**<<**"Con f(p): "**<<fixed<<setprecision**(15)**<<** FP **<<** **endl**;

                    goto parar;

                }

                if(FA \* FP > 0){

                    numeroA = numeroP;

                    FA = FP;

                }else{

                    numeroB = numeroP;

                }

            }

            cout**<<** "El metodo fracaso despues de " **<<** IT **<<** " iteraciones." **<<** **endl**;

        break;

        case 4:

**system**("cls");

**cabezaTabla**();

            TOL = **pow**(10, -8);

            numeroA = 0;

            numeroB = 1;

            FA = (**exp**(numeroA)-(**pow**(numeroA, 2))+(3\*numeroA)-2);

            for(i = 1; i < IT; i++){

                numeroP = ((numeroA + numeroB)/2);

                FP = (**exp**(numeroP)-(**pow**(numeroP, 2))+(3\*numeroP)-2);

                prueba = ((numeroB-numeroA)/2);

                numeroE = **abs**((numeroB-numeroA)/2);

**contTabla**(i, numeroA, numeroB, numeroP, FP, numeroE);

                if(FP == 0 || prueba < TOL){

                    cout**<<**"Proceso finalizado exitosamente en la iteracion: "**<<** i **<<** **endl**;

                    cout**<<**"La solucion aproximada es p: "**<<fixed<<setprecision**(15)**<<** numeroP **<<** **endl**;

                    cout**<<**"Con f(p): "**<<fixed<<setprecision**(15)**<<** FP **<<** **endl**;

                    goto parar;

                }

                if(FA \* FP > 0){

                    numeroA = numeroP;

                    FA = FP;

                }else{

                    numeroB = numeroP;

                }

            }

            cout**<<** "El metodo fracaso despues de " **<<** IT **<<** " iteraciones." **<<** **endl**;

        break;

        case 5:

**system**("cls");

**cabezaTabla**();

            TOL = **pow**(10, -4);

            numeroA = 0;

            numeroB = 1;

            FA = (((2\*numeroA)\***cos**(numeroA))-**pow**((numeroA+1), 2));//utilizo la palabra cos de math.h para sacar el coseno del numero dentro de los parentesis siguientes

            for(i = 1; i < IT; i++){

                numeroP = ((numeroA + numeroB)/2);

                FP = (((2\*numeroP)\***cos**(numeroP))-**pow**((numeroP+1), 2));

                prueba = ((numeroB-numeroA)/2);

                numeroE = **abs**((numeroB-numeroA)/2);

**contTabla**(i, numeroA, numeroB, numeroP, FP, numeroE);

                if(FP == 0 || prueba < TOL){

                    cout**<<**"Proceso finalizado exitosamente en la iteracion: "**<<** i **<<** **endl**;

                    cout**<<**"La solucion aproximada es p: "**<<fixed<<setprecision**(15)**<<** numeroP **<<** **endl**;

                    cout**<<**"Con f(p): "**<<fixed<<setprecision**(15)**<<** FP **<<** **endl**;

                    goto parar;

                }

                if(FA \* FP > 0){

                    numeroA = numeroP;

                    FA = FP;

                }else{

                    numeroB = numeroP;

                }

            }

            cout**<<** "El metodo fracaso despues de " **<<** IT **<<** " iteraciones." **<<** **endl**;

        break;

        case 6:

**system**("cls");

**cabezaTabla**();

            TOL = **pow**(10, -8);

            numeroA = 0;

            numeroB = 1;

            FA = (((2\*numeroA)\***cos**(numeroA))-**pow**((numeroA+1), 2));

            for(i = 1; i < IT; i++){

                numeroP = ((numeroA + numeroB)/2);

                FP = (((2\*numeroP)\***cos**(numeroP))-**pow**((numeroP+1), 2));

                prueba = ((numeroB-numeroA)/2);

                numeroE = **abs**((numeroB-numeroA)/2);

**contTabla**(i, numeroA, numeroB, numeroP, FP, numeroE);

                if(FP == 0 || prueba < TOL){

                    cout**<<**"Proceso finalizado exitosamente en la iteracion: "**<<** i **<<** **endl**;

                    cout**<<**"La solucion aproximada es p: "**<<fixed<<setprecision**(15)**<<** numeroP **<<** **endl**;

                    cout**<<**"Con f(p): "**<<fixed<<setprecision**(15)**<<** FP **<<** **endl**;

                    goto parar;

                }

                if(FA \* FP > 0){

                    numeroA = numeroP;

                    FA = FP;

                }else{

                    numeroB = numeroP;

                }

            }

            cout**<<** "El metodo fracaso despues de " **<<** IT **<<** " iteraciones." **<<** **endl**;

        break;

        case 7:

**system**("cls");

**cabezaTabla**();

            TOL = **pow**(10, -4);

            numeroA = 0;

            numeroB = 1;

            FA = ((numeroA\***cos**(numeroA))-(2\*(**pow**(numeroA, 2)))+(3\*numeroA)-1);

            for(i = 1; i < IT; i++){

                numeroP = ((numeroA + numeroB)/2);

                FP = ((numeroP\***cos**(numeroP))-(2\*(**pow**(numeroP, 2)))+(3\*numeroP)-1);

                prueba = ((numeroB-numeroA)/2);

                numeroE = **abs**((numeroB-numeroA)/2);

**contTabla**(i, numeroA, numeroB, numeroP, FP, numeroE);

                if(FP == 0 || prueba < TOL){

                    cout**<<**"Proceso finalizado exitosamente en la iteracion: "**<<** i **<<** **endl**;

                    cout**<<**"La solucion aproximada es p: "**<<fixed<<setprecision**(15)**<<** numeroP **<<** **endl**;

                    cout**<<**"Con f(p): "**<<fixed<<setprecision**(15)**<<** FP **<<** **endl**;

                    goto parar;

                }

                if(FA \* FP > 0){

                    numeroA = numeroP;

                    FA = FP;

                }else{

                    numeroB = numeroP;

                }

            }

            cout**<<** "El metodo fracaso despues de " **<<** IT **<<** " iteraciones." **<<** **endl**;

        break;

        case 8:

**system**("cls");

**cabezaTabla**();

            TOL = **pow**(10, -8);

            numeroA = 0;

            numeroB = 1;

            FA = ((numeroA\***cos**(numeroA))-(2\*(**pow**(numeroA, 2)))+(3\*numeroA)-1);

            for(i = 1; i < IT; i++){

                numeroP = ((numeroA + numeroB)/2);

                FP = ((numeroP\***cos**(numeroP))-(2\*(**pow**(numeroP, 2)))+(3\*numeroP)-1);

                prueba = ((numeroB-numeroA)/2);

                numeroE = **abs**((numeroB-numeroA)/2);

**contTabla**(i, numeroA, numeroB, numeroP, FP, numeroE);

                if(FP == 0 || prueba < TOL){

                    cout**<<**"Proceso finalizado exitosamente en la iteracion: "**<<** i **<<** **endl**;

                    cout**<<**"La solucion aproximada es p: "**<<fixed<<setprecision**(15)**<<** numeroP **<<** **endl**;

                    cout**<<**"Con f(p): "**<<fixed<<setprecision**(15)**<<** FP **<<** **endl**;

                    goto parar;

                }

                if(FA \* FP > 0){

                    numeroA = numeroP;

                    FA = FP;

                }else{

                    numeroB = numeroP;

                }

            }

            cout**<<** "El metodo fracaso despues de " **<<** IT **<<** " iteraciones." **<<** **endl**;

        break;

        case 9://por si la cantidad del usuario es 9 ejecuta lo siguiente

            goto final;//nos manda con goto a la etiqueta final, que finaliza el programa

        break;

        default://si escribe cualquier otra cantidad no descrita en los casos anteriores ejecuta lo siguiente

            cout**<<** "Porfavor ingrese una opcion valida del menu."**<<** **endl**;//muestra el mensaje de que debe ingresar una opcion valida

        break;

    }

    parar://etiqueta parar para regresar al menu

    goto menu;//este goto nos devuelve al menu de inicio

    final://etiqueta de final del programa, para finalmente salir de la ejecucion del mismo

    return 0;//como es una funcion de tipo int debe en el final del mismo retornar un valor, en este caso lo deje default en 0

}

Video de funcionamiento:

<https://drive.google.com/file/d/1wnaBMc2UJM8yMmmXYEos9uc6h2Vr1XpX/view?usp=drive_link>