Anthony Fabian Ramirez Orellana

Carne: 9490-22-958

Sección: “A”

Catedrático: Jorge Pérez

Tarea 03.

24/02/2022

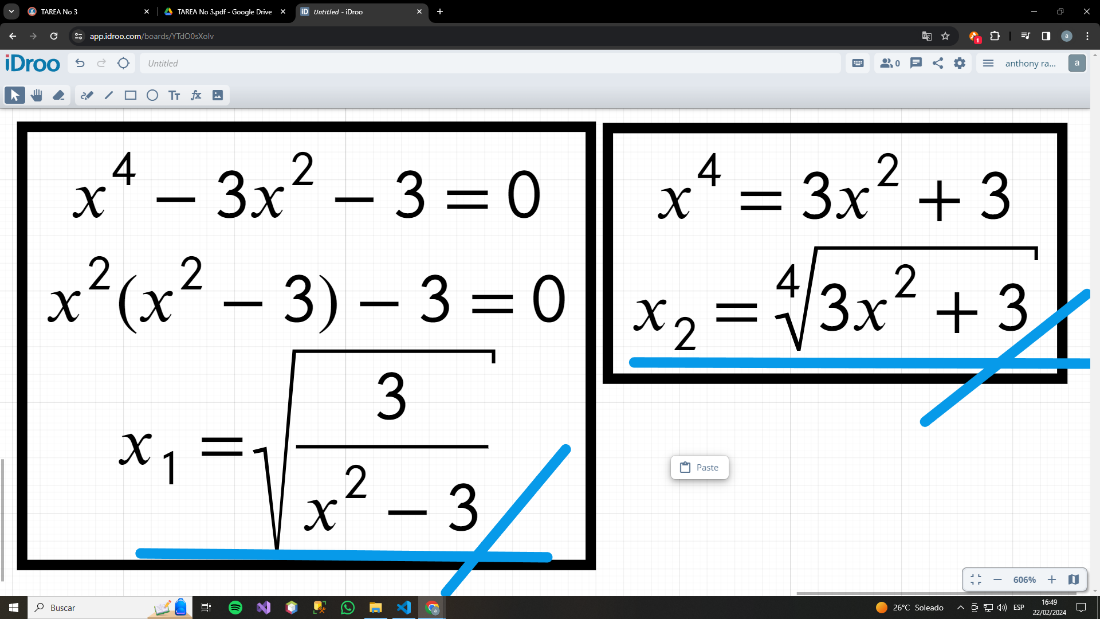
TAREA No 3

Codificar el algoritmo de Punto Fijo en C++. Debe incluir el código escrito al final de su archivo. La salida de su programa debe considerar las siguientes columnas:

* Número de iteración
* El valor de p0 en cada iteración
* El valor de p [g(p0)] en cada iteración
* El valor de f(p) para cada iteración
* Error absoluto en cada iteración (| (P – P0) /P|)
* Error relativo en cada iteración (| (P – P0) |)

Utilice su programa codificado en C++ para resolver lo siguiente:

1. La función f(x) = x˄4 – 3x˄2 – 3 = 0 tiene una raíz en el intervalo [1, 2]

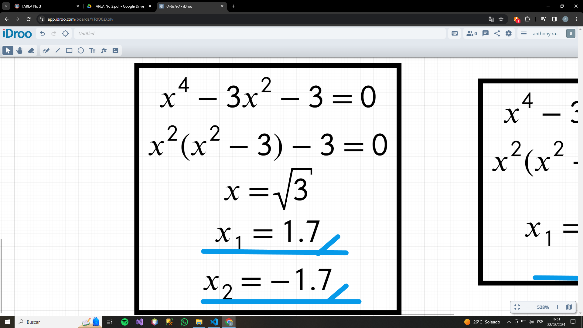


Utilice los siguientes puntos para aproximar una solución a la raíz en el intervalo dado, con una tolerancia de 10˄-5:

* p0 = 1.01
* p0 = 1.50
* p0 = 1.99

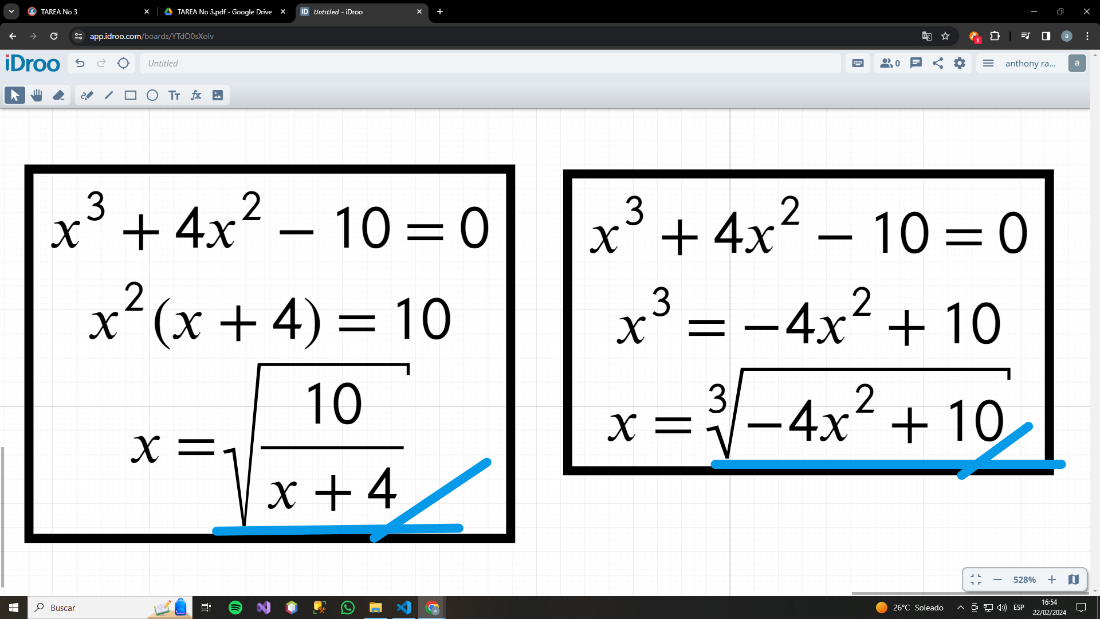
Repita el proceso anterior con una tolerancia de 10-10

¿En qué otro intervalo tiene la función f(x) otra raíz real?

Tiene Raiz en (-2, -1), (1, 2) y Raiz de Y en –3.

1. Utilice su programa codificado en C++ para resolver lo siguiente:

Encuentre una aproximación a una raíz de la función f(x) = x^3 + 4x^2 - 10 = 0, utilizando los siguientes parámetros:



p0 = 0.5; TOLERANCIA: TOL = 10-10; Cantidad máxima de iteraciones: IT = 100

Utilizar la misma tabla del primer problema.

En ambos casos debe utilizar, al menos, 2 transformaciones y dejar evidencia de cada una de ellas.

#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 de punto fijo\*/

void **menu**(){//creo una funcion vacia o proceso, vacia ya que no quiero que devuelva nada solo se ejecute, el proceso imprimira el menu para que se vea bonito.

    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**;//endl genera al final de lo impreso anterior un salto de linea como un enter.

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

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

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

    cout**<<**"1) f(x) = x^4 -3x^2 -3 = 0 con transfomacion a: x = ({3/[(x^2)-3]}^1/2)  tolerancia: 10^-5   P0 = 1.01" **<<** **endl**;

    cout**<<**"2) f(x) = x^4 -3x^2 -3 = 0 con transfomacion a: x = ({3/[(x^2)-3]}^1/2)  tolerancia: 10^-5   P0 = 1.50" **<<** **endl**;

    cout**<<**"3) f(x) = x^4 -3x^2 -3 = 0 con transfomacion a: x = ({3/[(x^2)-3]}^1/2)  tolerancia: 10^-5   P0 = 1.99" **<<** **endl**;

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

    cout**<<**"4) f(x) = x^4 -3x^2 -3 = 0 con transfomacion a: x = ({3/[(x^2)-3]}^1/2)  tolerancia: 10^-10   P0 = 1.01" **<<** **endl**;

    cout**<<**"5) f(x) = x^4 -3x^2 -3 = 0 con transfomacion a: x = ({3/[(x^2)-3]}^1/2)  tolerancia: 10^-10   P0 = 1.50" **<<** **endl**;

    cout**<<**"6) f(x) = x^4 -3x^2 -3 = 0 con transfomacion a: x = ({3/[(x^2)-3]}^1/2)  tolerancia: 10^-10   P0 = 1.99" **<<** **endl**;

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

    cout**<<**"7) f(x) = x^4 -3x^2 -3 = 0 con transfomacion a: x = {[(3x^2)+3]^1/4}  tolerancia: 10^-5   P0 = 1.01" **<<** **endl**;

    cout**<<**"8) f(x) = x^4 -3x^2 -3 = 0 con transfomacion a: x = {[(3x^2)+3]^1/4}  tolerancia: 10^-5   P0 = 1.50" **<<** **endl**;

    cout**<<**"9) f(x) = x^4 -3x^2 -3 = 0 con transfomacion a: x = {[(3x^2)+3]^1/4}  tolerancia: 10^-5   P0 = 1.99" **<<** **endl**;

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

    cout**<<**"10) f(x) = x^4 -3x^2 -3 = 0 con transfomacion a: x = {[(3x^2)+3]^1/4}  tolerancia: 10^-10   P0 = 1.01" **<<** **endl**;

    cout**<<**"11) f(x) = x^4 -3x^2 -3 = 0 con transfomacion a: x = {[(3x^2)+3]^1/4}  tolerancia: 10^-10   P0 = 1.50" **<<** **endl**;

    cout**<<**"12) f(x) = x^4 -3x^2 -3 = 0 con transfomacion a: x = {[(3x^2)+3]^1/4}  tolerancia: 10^-10   P0 = 1.99" **<<** **endl**;

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

    cout**<<**"13) f(x) = x^4 +4x^2 -10 = 0 con transfomacion a: x = {[10/(x+4)]^1/2}  tolerancia: 10^-10   P0 = 0.5" **<<** **endl**;

    cout**<<**"14) f(x) = x^4 +4x^2 -10 = 0 con transfomacion a: x = {[(-4x^2)+10]^1/3}  tolerancia: 10^-10   P0 = 0.5" **<<** **endl**;

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

    cout**<<**"15) Salir.";

}

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

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

}

void **contTabla**(int ite, float nP0, float nP, float Fp, float erA, float erR){//en este proceso si envio parametros para la correcta colocacion de los datos, pero igualmente es vacio para que se ejecute nada mas.

    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**(11)**<<**nP0**<<**"   "**<<fixed<<setprecision**(11)**<<**nP**<<**"   "**<<fixed<<setprecision**(11)**<<**Fp**<<**"   "**<<fixed<<setprecision**(11)**<<**erA**<<**"     "**<<fixed<<setprecision**(11)**<<**erR**<<endl**;//fixed<<setprecision(11) lo utilizo para dejar los parametros con 11 decimales

    }

    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**(11)**<<**nP0**<<**"   "**<<fixed<<setprecision**(11)**<<**nP**<<**"   "**<<fixed<<setprecision**(11)**<<**Fp**<<**"   "**<<fixed<<setprecision**(11)**<<**erA**<<**"     "**<<fixed<<setprecision**(11)**<<**erR**<<endl**;

    }

}

bool **pruebasOk**(float prueba1, float prueba2, float prueba3, float prueba4, float prueba5){//creo una funcion de tipo bool que me devolvera false cuando se detecte que abra un error matematico que puede trabar el programa

    if (prueba1 == 0)//verifico si prueba 1 es igual a cero, ya que seria una divicion entre cero

    {

        cout**<<**"Error, se realizo una divicion en 0 en x = {3/[(x^2)-3]}, se aborto el proceso"**<<endl**;

        return false;//de ser cierto regreso false para no realizar el error matematico.

    }

    if (prueba2 < 0)//asi con cada prueba necesaria por las raices y las diviciones

    {

        cout**<<**"Error, se quiso sacar raiz par a un numero negativo en x= ({3/[(x^2)-3]}^1/2), se aborto el proceso"**<<endl**;

        return false;

    }

    if (prueba3 < 0)

    {

        cout**<<**"Erro, se trato de sacar raiz par de un numero negativo en x = {[(3x^2)+3]^1/4}, se aborto el proceso"**<<endl**;

        return false;

    }

    if (prueba4 == 0)

    {

        cout**<<**"Error, se realizo una divicion en 0 en x = {[10/(x+4)]^1/2}, se aborto el proceso"**<<endl**;

        return false;

    }

    if (prueba5 < 0)

    {

        cout**<<**"Error, se quiso sacar raiz par a un numero negativo en x = {[10/(x+4)]^1/2}, se aborto el proceso"**<<endl**;

        return false;

    }

    return true;

}

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 i = 1, IT = 100, opcion;//creo 4 variables de tipo entero, una para las iteraciones, una para el numero limite de iteraciones y una ultima para el menu

    float numeroP0, numeroP, TOL, Fp, errAbs, errRel, prueba1, prueba2, prueba3, prueba4, prueba5;//creo 11 variables tipo flotante para diversos calculos y para la tolerancia

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

**menu**();//mando a llamar al proceso menu, para que se imprima en consola

    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){//condicional que me permite hacer un caso por cada valor que puede tomar la variable

        case 0://Caso cero solamente para limpriar la consola y que vuelva a imprimir el menu

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

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

            i=1;//igualo la variable i a 1 para que siempre inicie en 1 en los bucles while

        break;

        case 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, -5);//igualo la variable TOL a la verdadera tolerancia usada en el problema a 10^-4

            numeroP0 = 1.01;//igualo la variable al parametro que se utilizara en esta opcion del menu

            while (i <= IT)//declaro un bucle while que este activo mientras que i(1) sea menor o igual a IT(100)

            {

                prueba1 = (**pow**(numeroP0, 2)-3);//calculo los datos para hacer la prueba de fallo

                prueba2 = (3/( **pow**(numeroP0, 2)-3 )  );//como esta transformacion tiene 2 posibles fallos, debo hacer 2 pruebas

                prueba3 = 1;//envio las demas pruebas que no utilizare como uno para que no me den conflicto

                prueba4 = 1;

                prueba5 = 1;

                if(**pruebasOk**(prueba1, prueba2, prueba3, prueba4, prueba5) != true){//mando a llamar mi funcion dentro de un condicional que me evalue si las pruebas estan mal, imprime los datos y aborta

**contTabla**(i, numeroP0, numeroP, Fp, errAbs, errRel);//mando a llamar el proceso conTabla para que me imprima los valores iniciales en la tabla

                    goto abortar1;//voy a la etiqueta abortar 1 para terminar esta opcion y regresar al menu

                }

                numeroP = (**pow**(  (3/(  (  **pow**(numeroP0, 2)  )-3)  ), 1/2)  );//si todas las pruebas estan bien, calculo el numero P, utilizando pow con una fraccion como exponente para sacar raices

                errRel = (  **abs**(numeroP - numeroP0)  );//calculo el error relativo

                errAbs = **abs**((numeroP-numeroP0)/numeroP);//calculo el error absoluto

                Fp = (**pow**(numeroP, 4) - 3\***pow**(numeroP, 2) -3);//calculo el valor de P en la funcion principal

**contTabla**(i, numeroP0, numeroP, Fp, errAbs, errRel);//mando a llamar el proceso conTabla para que me imprima los valores iniciales en la tabla

                if(errRel < TOL){//Si el error relativo es menor a la tolerancia ejecutara lo de dentro del condicional

                    cout**<<**"Proceso finalizado exitosamente en la iteracion: "**<<** i **<<** **endl**;//imprimo un mensaje para mostrar en que iteracion se llego a la respuesta

                    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)**<<** numeroP **<<** **endl**;//imprimo el valor aproximado de fP que deveria dar aproximadamente cero

                    goto parar;//voy a la etiqueta parar, para que finalize esta opcion

                }

                i = i+1;//sumo 1 a la variable i para que el while pare en algun momento

                numeroP0 = numeroP;//Igualo P0 a P para calcular la siguente iteracion

            }

            abortar1://etiqueta para abortar el proceso antes de cometer un error matematico

            cout**<<**"El metodo fracaso despues de: "**<<**i**<<**" iteraciones.";//imprimo que el metodo fracaso en dado caso que se llegue a las 100 iteraciones

            goto parar;//voy a la etiqueta parar

        break;

        case 2:

**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, -5);//igualo la variable TOL a la verdadera tolerancia usada en el problema a 10^-4

            numeroP0 = 1.50;

            while (i <= IT)

            {

                prueba1 = (**pow**(numeroP0, 2)-3);

                prueba2 = (3/( **pow**(numeroP0, 2)-3 )  );

                prueba3 = 1;

                prueba4 = 1;

                prueba5 = 1;

                if(**pruebasOk**(prueba1, prueba2, prueba3, prueba4, prueba5) != true){

**contTabla**(i, numeroP0, numeroP, Fp, errAbs, errRel);//mando a llamar el proceso conTabla para que me imprima los valores iniciales en la tabla

                    goto abortar2;

                }

                numeroP = (**pow**(  (3/(  (  **pow**(numeroP0, 2)  )-3)  ), 1/2)  );

                errRel = (  **abs**(numeroP - numeroP0)  );

                errAbs = **abs**((numeroP-numeroP0)/numeroP);

                Fp = (**pow**(numeroP, 4) - 3\***pow**(numeroP, 2) -3);

**contTabla**(i, numeroP0, numeroP, Fp, errAbs, errRel);//mando a llamar el proceso conTabla para que me imprima los valores iniciales en la tabla

                if(errRel < TOL){

                    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)**<<** numeroP **<<** **endl**;

                    goto parar;

                }

                i = i+1;

                numeroP0 = numeroP;

            }

            abortar2:

            cout**<<**"El metodo fracaso despues de: "**<<**i**<<**" iteraciones.";

            goto parar;

        break;

        case 3:

**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, -5);//igualo la variable TOL a la verdadera tolerancia usada en el problema a 10^-4

            numeroP0 = 1.99;

            while (i <= IT)

            {

                prueba1 = (**pow**(numeroP0, 2)-3);

                prueba2 = (3/( **pow**(numeroP0, 2)-3 )  );

                prueba3 = 1;

                prueba4 = 1;

                prueba5 = 1;

                if(**pruebasOk**(prueba1, prueba2, prueba3, prueba4, prueba5) != true){

**contTabla**(i, numeroP0, numeroP, Fp, errAbs, errRel);//mando a llamar el proceso conTabla para que me imprima los valores iniciales en la tabla

                    goto abortar3;

                }

                numeroP = (**pow**(  (3/(  (  **pow**(numeroP0, 2)  )-3)  ), 1/2)  );

                errRel = (  **abs**(numeroP - numeroP0)  );

                errAbs = **abs**((numeroP-numeroP0)/numeroP);

                Fp = (**pow**(numeroP, 4) - 3\***pow**(numeroP, 2) -3);

**contTabla**(i, numeroP0, numeroP, Fp, errAbs, errRel);//mando a llamar el proceso conTabla para que me imprima los valores iniciales en la tabla

                if(errRel < TOL){

                    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)**<<** numeroP **<<** **endl**;

                    goto parar;

                }

                i = i+1;

                numeroP0 = numeroP;

            }

            abortar3:

            cout**<<**"El metodo fracaso despues de: "**<<**i**<<**" iteraciones.";

            goto parar;

        break;

        case 4:

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

            numeroP0 = 1.01;

            while (i <= IT)

            {

                prueba1 = (**pow**(numeroP0, 2)-3);

                prueba2 = (3/( **pow**(numeroP0, 2)-3 )  );

                prueba3 = 1;

                prueba4 = 1;

                prueba5 = 1;

                if(**pruebasOk**(prueba1, prueba2, prueba3, prueba4, prueba5) != true){

**contTabla**(i, numeroP0, numeroP, Fp, errAbs, errRel);//mando a llamar el proceso conTabla para que me imprima los valores iniciales en la tabla

                    goto abortar4;

                }

                numeroP = (**pow**(  (3/(  (  **pow**(numeroP0, 2)  )-3)  ), 1/2)  );

                errRel = (  **abs**(numeroP - numeroP0)  );

                errAbs = **abs**((numeroP-numeroP0)/numeroP);

                Fp = (**pow**(numeroP, 4) - 3\***pow**(numeroP, 2) -3);

**contTabla**(i, numeroP0, numeroP, Fp, errAbs, errRel);//mando a llamar el proceso conTabla para que me imprima los valores iniciales en la tabla

                if(errRel < TOL){

                    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)**<<** numeroP **<<** **endl**;

                    goto parar;

                }

                i = i+1;

                numeroP0 = numeroP;

            }

            abortar4:

            cout**<<**"El metodo fracaso despues de: "**<<**i**<<**" iteraciones.";

            goto parar;

        break;

        case 5:

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

            numeroP0 = 1.50;

            while (i <= IT)

            {

                prueba1 = (**pow**(numeroP0, 2)-3);

                prueba2 = (3/( **pow**(numeroP0, 2)-3 )  );

                prueba3 = 1;

                prueba4 = 1;

                prueba5 = 1;

                if(**pruebasOk**(prueba1, prueba2, prueba3, prueba4, prueba5) != true){

**contTabla**(i, numeroP0, numeroP, Fp, errAbs, errRel);//mando a llamar el proceso conTabla para que me imprima los valores iniciales en la tabla

                    goto abortar5;

                }

                numeroP = (**pow**(  (3/(  (  **pow**(numeroP0, 2)  )-3)  ), 1/2)  );

                errRel = (  **abs**(numeroP - numeroP0)  );

                errAbs = **abs**((numeroP-numeroP0)/numeroP);

                Fp = (**pow**(numeroP, 4) - 3\***pow**(numeroP, 2) -3);

**contTabla**(i, numeroP0, numeroP, Fp, errAbs, errRel);//mando a llamar el proceso conTabla para que me imprima los valores iniciales en la tabla

                if(errRel < TOL){

                    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)**<<** numeroP **<<** **endl**;

                    goto parar;

                }

                i = i+1;

                numeroP0 = numeroP;

            }

            abortar5:

            cout**<<**"El metodo fracaso despues de: "**<<**i**<<**" iteraciones.";

            goto parar;

        break;

        case 6:

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

            numeroP0 = 1.99;

            while (i <= IT)

            {

                prueba1 = (**pow**(numeroP0, 2)-3);

                prueba2 = (3/( **pow**(numeroP0, 2)-3 )  );

                prueba3 = 1;

                prueba4 = 1;

                prueba5 = 1;

                if(**pruebasOk**(prueba1, prueba2, prueba3, prueba4, prueba5) != true){

**contTabla**(i, numeroP0, numeroP, Fp, errAbs, errRel);//mando a llamar el proceso conTabla para que me imprima los valores iniciales en la tabla

                    goto abortar6;

                }

                numeroP = (**pow**(  (3/(  (  **pow**(numeroP0, 2)  )-3)  ), 1/2)  );

                errRel = (  **abs**(numeroP - numeroP0)  );

                errAbs = **abs**((numeroP-numeroP0)/numeroP);

                Fp = (**pow**(numeroP, 4) - 3\***pow**(numeroP, 2) -3);

**contTabla**(i, numeroP0, numeroP, Fp, errAbs, errRel);//mando a llamar el proceso conTabla para que me imprima los valores iniciales en la tabla

                if(errRel < TOL){

                    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)**<<** numeroP **<<** **endl**;

                    goto parar;

                }

                i = i+1;

                numeroP0 = numeroP;

            }

            abortar6:

            cout**<<**"El metodo fracaso despues de: "**<<**i**<<**" iteraciones.";

            goto parar;

        break;

        case 7:

**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, -5);//igualo la variable TOL a la verdadera tolerancia usada en el problema a 10^-4

            numeroP0 = 1.01;

            while (i <= IT)

            {

                prueba1 = 1;

                prueba2 = 1;

                prueba3 = ( 3\***pow**(numeroP0, 2) +3 );

                prueba4 = 1;

                prueba5 = 1;

                if(**pruebasOk**(prueba1, prueba2, prueba3, prueba4, prueba5) != true){

**contTabla**(i, numeroP0, numeroP, Fp, errAbs, errRel);//mando a llamar el proceso conTabla para que me imprima los valores iniciales en la tabla

                    goto abortar7;

                }

                numeroP = (**pow**(  ( 3\***pow**(numeroP0, 2) +3 ), 1/4)  );

                errRel = (  **abs**(numeroP - numeroP0)  );

                errAbs = **abs**((numeroP-numeroP0)/numeroP);

                Fp = (**pow**(numeroP, 4) - 3\***pow**(numeroP, 2) -3);

**contTabla**(i, numeroP0, numeroP, Fp, errAbs, errRel);//mando a llamar el proceso conTabla para que me imprima los valores iniciales en la tabla

                if(errRel < TOL){

                    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)**<<** numeroP **<<** **endl**;

                    goto parar;

                }

                i = i+1;

                numeroP0 = numeroP;

            }

            abortar7:

            cout**<<**"El metodo fracaso despues de: "**<<**i**<<**" iteraciones.";

            goto parar;

        break;

        case 8:

**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, -5);//igualo la variable TOL a la verdadera tolerancia usada en el problema a 10^-4

            numeroP0 = 1.50;

            while (i <= IT)

            {

                prueba1 = 1;

                prueba2 = 1;

                prueba3 = ( 3\***pow**(numeroP0, 2) +3 );

                prueba4 = 1;

                prueba5 = 1;

                if(**pruebasOk**(prueba1, prueba2, prueba3, prueba4, prueba5) != true){

**contTabla**(i, numeroP0, numeroP, Fp, errAbs, errRel);//mando a llamar el proceso conTabla para que me imprima los valores iniciales en la tabla

                    goto abortar8;

                }

                numeroP = (**pow**(  ( 3\***pow**(numeroP0, 2) +3 ), 1/4)  );

                errRel = (  **abs**(numeroP - numeroP0)  );

                errAbs = **abs**((numeroP-numeroP0)/numeroP);

                Fp = (**pow**(numeroP, 4) - 3\***pow**(numeroP, 2) -3);

**contTabla**(i, numeroP0, numeroP, Fp, errAbs, errRel);//mando a llamar el proceso conTabla para que me imprima los valores iniciales en la tabla

                if(errRel < TOL){

                    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)**<<** numeroP **<<** **endl**;

                    goto parar;

                }

                i = i+1;

                numeroP0 = numeroP;

            }

            abortar8:

            cout**<<**"El metodo fracaso despues de: "**<<**i**<<**" iteraciones.";

            goto parar;

        break;

        case 9:

**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, -5);//igualo la variable TOL a la verdadera tolerancia usada en el problema a 10^-4

            numeroP0 = 1.99;

            while (i <= IT)

            {

                prueba1 = 1;

                prueba2 = 1;

                prueba3 = ( 3\***pow**(numeroP0, 2) +3 );

                prueba4 = 1;

                prueba5 = 1;

                if(**pruebasOk**(prueba1, prueba2, prueba3, prueba4, prueba5) != true){

**contTabla**(i, numeroP0, numeroP, Fp, errAbs, errRel);//mando a llamar el proceso conTabla para que me imprima los valores iniciales en la tabla

                    goto abortar9;

                }

                numeroP = (**pow**(  ( 3\***pow**(numeroP0, 2) +3 ), 1/4)  );

                errRel = (  **abs**(numeroP - numeroP0)  );

                errAbs = **abs**((numeroP-numeroP0)/numeroP);

                Fp = (**pow**(numeroP, 4) - 3\***pow**(numeroP, 2) -3);

**contTabla**(i, numeroP0, numeroP, Fp, errAbs, errRel);//mando a llamar el proceso conTabla para que me imprima los valores iniciales en la tabla

                if(errRel < TOL){

                    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)**<<** numeroP **<<** **endl**;

                    goto parar;

                }

                i = i+1;

                numeroP0 = numeroP;

            }

            abortar9:

            cout**<<**"El metodo fracaso despues de: "**<<**i**<<**" iteraciones.";

            goto parar;

        break;

        case 10:

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

            numeroP0 = 1.01;

            while (i <= IT)

            {

                prueba1 = 1;

                prueba2 = 1;

                prueba3 = ( 3\***pow**(numeroP0, 2) +3 );

                prueba4 = 1;

                prueba5 = 1;

                if(**pruebasOk**(prueba1, prueba2, prueba3, prueba4, prueba5) != true){

**contTabla**(i, numeroP0, numeroP, Fp, errAbs, errRel);//mando a llamar el proceso conTabla para que me imprima los valores iniciales en la tabla

                    goto abortar10;

                }

                numeroP = (**pow**(  ( 3\***pow**(numeroP0, 2) +3 ), 1/4)  );

                errRel = (  **abs**(numeroP - numeroP0)  );

                errAbs = **abs**((numeroP-numeroP0)/numeroP);

                Fp = (**pow**(numeroP, 4) - 3\***pow**(numeroP, 2) -3);

**contTabla**(i, numeroP0, numeroP, Fp, errAbs, errRel);//mando a llamar el proceso conTabla para que me imprima los valores iniciales en la tabla

                if(errRel < TOL){

                    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)**<<** numeroP **<<** **endl**;

                    goto parar;

                }

                i = i+1;

                numeroP0 = numeroP;

            }

            abortar10:

            cout**<<**"El metodo fracaso despues de: "**<<**i**<<**" iteraciones.";

            goto parar;

        break;

        case 11:

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

            numeroP0 = 1.50;

            while (i <= IT)

            {

                prueba1 = 1;

                prueba2 = 1;

                prueba3 = ( 3\***pow**(numeroP0, 2) +3 );

                prueba4 = 1;

                prueba5 = 1;

                if(**pruebasOk**(prueba1, prueba2, prueba3, prueba4, prueba5) != true){

**contTabla**(i, numeroP0, numeroP, Fp, errAbs, errRel);//mando a llamar el proceso conTabla para que me imprima los valores iniciales en la tabla

                    goto abortar11;

                }

                numeroP = (**pow**(  ( 3\***pow**(numeroP0, 2) +3 ), 1/4)  );

                errRel = (  **abs**(numeroP - numeroP0)  );

                errAbs = **abs**((numeroP-numeroP0)/numeroP);

                Fp = (**pow**(numeroP, 4) - 3\***pow**(numeroP, 2) -3);

**contTabla**(i, numeroP0, numeroP, Fp, errAbs, errRel);//mando a llamar el proceso conTabla para que me imprima los valores iniciales en la tabla

                if(errRel < TOL){

                    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)**<<** numeroP **<<** **endl**;

                    goto parar;

                }

                i = i+1;

                numeroP0 = numeroP;

            }

            abortar11:

            cout**<<**"El metodo fracaso despues de: "**<<**i**<<**" iteraciones.";

            goto parar;

        break;

        case 12:

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

            numeroP0 = 1.99;

            while (i <= IT)

            {

                prueba1 = 1;

                prueba2 = 1;

                prueba3 = ( 3\***pow**(numeroP0, 2) +3 );

                prueba4 = 1;

                prueba5 = 1;

                if(**pruebasOk**(prueba1, prueba2, prueba3, prueba4, prueba5) != true){

**contTabla**(i, numeroP0, numeroP, Fp, errAbs, errRel);//mando a llamar el proceso conTabla para que me imprima los valores iniciales en la tabla

                    goto abortar12;

                }

                numeroP = (**pow**(  ( 3\***pow**(numeroP0, 2) +3 ), 1/4)  );

                errRel = (  **abs**(numeroP - numeroP0)  );

                errAbs = **abs**((numeroP-numeroP0)/numeroP);

                Fp = (**pow**(numeroP, 4) - 3\***pow**(numeroP, 2) -3);

**contTabla**(i, numeroP0, numeroP, Fp, errAbs, errRel);//mando a llamar el proceso conTabla para que me imprima los valores iniciales en la tabla

                if(errRel < TOL){

                    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)**<<** numeroP **<<** **endl**;

                    goto parar;

                }

                i = i+1;

                numeroP0 = numeroP;

            }

            abortar12:

            cout**<<**"El metodo fracaso despues de: "**<<**i**<<**" iteraciones.";

            goto parar;

        break;

        case 13:

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

            numeroP0 = 0.5;

            while (i <= IT)

            {

                prueba1 = 1;

                prueba2 = 1;

                prueba3 = 1;

                prueba4 = (numeroP0 +4);

                prueba5 = (**pow**( ( 10/(numeroP0+4) ) , 2));

                if(**pruebasOk**(prueba1, prueba2, prueba3, prueba4, prueba5) != true){

**contTabla**(i, numeroP0, numeroP, Fp, errAbs, errRel);//mando a llamar el proceso conTabla para que me imprima los valores iniciales en la tabla

                    goto abortar13;

                }

                numeroP = (**pow**(  ( 10/(numeroP0+4) ), 1/2)  );

                errRel = (  **abs**(numeroP - numeroP0)  );

                errAbs = **abs**((numeroP-numeroP0)/numeroP);

                Fp = (**pow**(numeroP, 3) + 4\***pow**(numeroP, 2) -10);

**contTabla**(i, numeroP0, numeroP, Fp, errAbs, errRel);//mando a llamar el proceso conTabla para que me imprima los valores iniciales en la tabla

                if(errRel < TOL){

                    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)**<<** numeroP **<<** **endl**;

                    goto parar;

                }

                i = i+1;

                numeroP0 = numeroP;

            }

            abortar13:

            cout**<<**"El metodo fracaso despues de: "**<<**i**<<**" iteraciones.";

            goto parar;

        break;

        case 14:

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

            numeroP0 = 0.5;

            while (i <= IT)

            {

                numeroP = (**pow**(  ((-4\***pow**(numeroP0, 2))+10), 1/3)  );

                errRel = (  **abs**(numeroP - numeroP0)  );

                errAbs = **abs**((numeroP-numeroP0)/numeroP);

                Fp = (**pow**(numeroP, 3) + 4\***pow**(numeroP, 2) -10);

**contTabla**(i, numeroP0, numeroP, Fp, errAbs, errRel);//mando a llamar el proceso conTabla para que me imprima los valores iniciales en la tabla

                if(errRel < TOL){

                    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)**<<** numeroP **<<** **endl**;

                    goto parar;

                }

                i = i+1;

                numeroP0 = numeroP;

            }

            cout**<<**"El metodo fracaso despues de: "**<<**i**<<**" iteraciones.";

            goto parar;

        break;

        case 15://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

**system**("cls");

            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

    TOL=0;//igualo la tolerancia a 0 por si queda un residuo de valor de alguna opcion

    i=1;//igualo la variable i a 1 para que siempre comience en 1 cuando se entra a un while

    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 del funcionamiento:

<https://drive.google.com/file/d/1qAKqZAOjKO80m6w4wbdxG4tfGIVvUemz/view?usp=drive_link>