

TÉCNICAS DE PROGRAMACIÓN
SEGUNDO LABORATORIO DIRIGIDO
SEMESTRE ACADÉMICO 2025-1

Horarios: Todos

Duración: 110 minutos

Elaborado por los profesores del curso.

INDICACIONES:

- LA FINALIDAD DE ESTE LABORATORIO ES QUE EL ALUMNOS SE FAMILIARICE CON EL ENTORNO DE DESARROLLO INTEGRADO (IDE) NETBEANS.
- EL ALUMNO CREARÁ UN PROYECTO QUE CONTENGA VARIOS MÓDULOS (archivos . h y . cpp) Y LUEGO COPIARÁ EL PROGRAMA QUE SE LE PROPORCIONA A CONTINUACIÓN. DEBERÁ PROCURAR EN ESTE PUNTO EMPLEAR LAS HERRAMIENTAS QUE LE PROPORCIONA EL EDITOR DEL NETBEANS.
- LUEGO, EL ALUMNO PROCEDERÁ A COMPILAR EL PROGRAMA, CORREGIR LOS ERRORES SINTÁCTICOS QUE APAREZCAN Y FINALMENTE EJECUTARÉ EL PROGRAMA DESDE LA LÍNEA DE COMANDOS, VERIFICANDO SUS RESULTADOS.
- EN ESTE LABORATORIO NO DEBERÁ EMPLEAR EL DEPURADOR DEL ENTORNO NI DEL COMPILADOR.

PRIMERA PARTE: Uso del entorno NetBeans

Utilizando el entorno NetBeans, cree un proyecto con nombre “LabortorioDirigido01_RaizDeEcuacion”. Una vez creado, copie el programa completamente tal cual lo ve en la hoja. El programa tiene errores sintácticos colocados adrede; por esta razón, si usted advierte alguno de ellos, no los corrija ya que lo que se pretende es que, al compilar su programa, sea el compilador el que detecte y muestre los errores, y que el alumno aprenda a interpretarlos y por lo tanto a corregirlos a partir de los mensajes que aparezcan en el entorno.

Debe crear los archivos correspondientes e incluir los comentarios, respete los nombres de las variables y funciones dadas y la simbología empleada.

Descripción del problema:

Se desea elaborar un proyecto que permita encontrar la raíz de una ecuación empleando el método de la posición falsa, la función que se empleará tiene la forma siguiente:

$$f(x) = c_4 * x^4 + c_3 * x^3 + c_2 * x^2 + c_1 * x + c_0$$

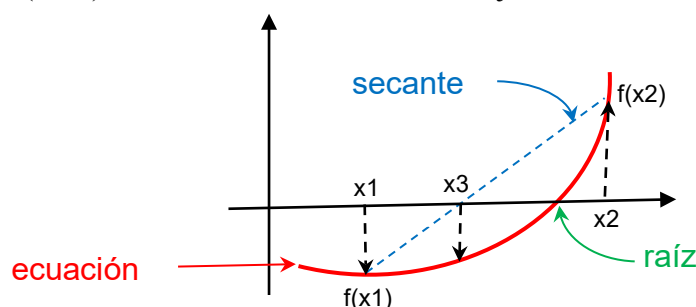
Por ejemplo, para las ecuaciones:

$$f_1(x) = 8.05 * x^4 + 14 * x^3 + 71.3 * x^2 - 66 * x - 100$$

o

$$f_2(x) = 10.01 * x^4 - 14.22 * x^3 - 65.4 * x^2 + 13.3 * x + 1.07$$

Esta técnica empieza con la determinación de los puntos x_1 y x_2 de modo que al evaluar $f(x_1)$ y $f(x_2)$ se obtengan valores con signos diferentes, si se logran encontrar, podemos garantizar que entre esos dos puntos existe una raíz para la ecuación. A partir de ese momento, este método, que busca determinar esa raíz, seguirá el siguiente proceso: se dibuja una línea (llamada secante) entre los puntos $(x_1, f(x_1))$ y $(x_2, f(x_2))$ y se determina el punto $(x_3, 0)$ en donde dicha secante cruza el eje x , tal como se muestra en la siguiente figura:



El nuevo punto x_3 , estará más cerca de la raíz que cualquiera de los puntos originales. Ahora, se determinará el valor de $f(x_3)$ y se verifica si $f(x_1)$ y $f(x_3)$ tiene signos diferentes, si esto es cierto, entonces se descarta el punto x_2 , si no es cierto entonces se descarta el punto x_1 .

A partir de aquí se repite el proceso antes descrito, pero ahora empleando x_3 y el x que no se descartó. En cada ciclo, se genera un nuevo punto que se acercan cada vez más al valor de la raíz. No se obtendrá una respuesta exacta, pero se seguirá acercando a la respuesta y se detendrá cuando se ha llegado a un nivel deseado de precisión.

EL nivel deseado de precisión se dará por satisfecho cuando el valor de la función en el punto más reciente calculado, $f(x_i)$, es muy cercano a cero, esto es: $|f(x_i)| < e$, donde e (épsilon) es un número pequeño positivo como por ejemplo 0.0001.

El programa que resuelve este problema se muestra a continuación:

En el archivo main.cpp tenemos:

```
/*
 * Archivo:    main.cpp
 * Autor:      ***COLOQUE AQUÍ SU CÓDIGO NOMBRE Y APELLIDO***
 * Fecha y Hora: ***COLOQUE AQUÍ LA FECHA Y HORA EN LA QUE EMPEZÓ A ESCRIBIR***
 *
 *=====
 * Programa para encontrar la raíz de una ecuación empleando el método
 * de la Posición falsa, ejercicio adaptado del libro de (Schneider et al.,1986)
 *=====
 */
#include <iostream>
#include <iomanip>
#include <cmath>
using namespace std;
#include "funcionesAuxiliares.h"

int main(int argc, char** argv) {

    double coeficiente4, coeficiente3, coeficiente2, coeficiente1, coeficiente0;
    double x1, x2;
    double epsilon;
    double funcion1, funcion2;
    bool darsePorVencido, datosBuenos;

    datosBuenos = false;
    darsePorVencido = false;

    while(not datosBuenos && not darsePorVencido){
        cout<<"Ingresar los dos puntos de inicio: "<<endl;
        cin>>x1>>x2;
        if((x1 == 0) y (x2 == 0)){
            darsePorVencido = true;
        }
        else{
            cout<<"Ingresar los coeficientes: c4 al c0<<endl;
            cin>>coeficiente4>>coeficiente3>>coeficiente2>>coeficiente1>>coeficiente0;
            funcion1 = calcularFx(x1, coeficiente4, coeficiente3, coeficiente2,
                                coeficiente1, coeficiente0);
            funcion2 = calcularFx(x2, coeficiente4, coeficiente3, coeficiente2,
                                coeficiente1, coeficiente0);
            if((funcion1 <= 0.0 and funcion2 ==> 0.0) o
                (funcion1 >= 0.0 and funcion2 <= 0.0)) {
                datosBuenos = True;
            }
            else {
                cout<<"Lo siento, los puntos dados no estan opuestos"<<endl;
                cout<<"Ingresar 0,0 para terminar (puntos de inicio)"<<endl;
            }
        }
    }
    if (darsePorVencido){
        cout<<"Lo siento, el programa se concluye por una falla"<<endl;
        cout<<"para encontrar un intervalo inicial valido"<<endl;
    }
}
```

```

    else{
        solucionProblema(epsilon, x1, x2, coeficiente4, coeficiente3, coeficiente2,
                        coeficiente1, coeficiente0, funcion1, funcion);
    }
    return 0;
}

```

En el archivo FuncionesAuxiliares.h tenemos:

```

/*
 * Archivo:    funcionesAuxiliares.h
 * Autor:     ***COLOQUE AQUÍ SU CÍDIGOM NOMBRE Y APELLIDO***
 * Fecha y Hora: ***COLOQUE AQUÍ LA FECHA Y HORA EN LA QUE EMPEZÓ A ESCRIBIR***
 */

#ifndef FUNCIONES_AUXILIARES_H
#define FUNCIONES_AUXILIARES_H

double calcularFx(double ,double ,double ,double ,double, double);
double calcularPotencia(double, int);
void solucionarProblema(double, double, double, double, double, double, double, double,
                        double, double, double);

#endif /* FUNCIONES_AUXILIARES_H */

```

En el archivo FuncionesAuxiliares.cpp tenemos:

```

/*
 * Archivo:    FuncionesAuxiliares.cpp
 * Autor:     ***COLOQUE AQUÍ SU CÍDIGOM NOMBRE Y APELLIDO***
 * Fecha y Hora: ***COLOQUE AQUÍ LA FECHA Y HORA EN LA QUE EMPEZÓ A ESCRIBIR***
 */

#include <iostream>
#include <cmath>
using namespace std;
#define CUENTA_MAXIMA 100

double calcularPotencia(double x, int n){
    double potencia = 1.0;
    int i;
    for(i=0; i < n; i++){
        potencia *= x;
    }
    return potencia;
}

double calcularFx(double x, double coef4, double coef3, double coef2,
                 double coef1, double coef0){
    double fx;

    fx = coef4 * calcularPotencia(x,4) + coef3 * calcularPotencia(x,3) +
        coef2 * calcularPotencia(x,2) + coef1 * x + coef0;
    return fx;
}

void solucionarProblema(double epsilon, double x1, double x2, double coef4,
                       double coef3, double coef2, double coef1, double coef0, double funcion1,
                       double funcion2){

    int cuenta;
    bool raizEncontrada;
    double x3, funcion3;

    cout<<"Ingresar la exactitud deseada: "<<endl;
    cin>>epsilon;

```

```

raizEncontrada = false;
cuenta = 0;
while (not raizEncontrada and (cuenta <= CUENTA_MAXIMA)){

    x3 = (x2 * funcion1 - x1 * funcion2)/(funcion1 - funcion2);
    funcion3 = calcularFx(x3, coef4, coef3, coef2, coef1, coef0);
    cout<<"cuenta= "<<cuenta<<"  x1= "<<x1<<"  x3= "<<x3<<endl;
    if (fabs(funcion3)<epsilon)
        raizEncontrada = true;
    if (((funcion1 <= 0.0) and (funcion3 <= 0.0)) or
        ((funcion1 >= 0.0) and (funcion3 >= 0.0))){
        x1 = x3 ;
        funcion1 = funcion3;
    }
    else{
        x2 = x3;
        funcion2 = funcion3;
    }
    cuenta += 1;
    if (fabs(funcion3)<epsilon)
        raizEncontrada = true;
}

if (raizEncontrada)
    cout<<"La raiz es "<< x3 <<" , con una exactitud de "
    <<epsilon<<endl;
else
    cout<<"Lo sentimos, no pudimos encontrar la raiz en "
    <<CUENTA_MAXIMA<<" iteraciones"<<endl;
}

```

SEGUNDA PARTE: Compilación, corrección de errores sintácticos y ejecución

Después de copiar el proyecto en el entorno de NetBeans, compile el proyecto. Deberán aparecer errores en la ventana de salida (Output). Allí ubíquese en el primer error que tenga (nunca debe empezar por el último error), lea el error y trate de entender qué produjo el error. Si lo entiende, presione la línea con el error y el sistema lo llevará a la línea de código donde se cometió el error, entonces corríjalo, vuelva a compilar el programa repitiendo el proceso hasta que ya no haya más errores. Si no lo entiende solicite la asistencia de un Jefe de práctica.

Luego de corregir todos los errores de sintaxis, ejecute el programa. Para realizar esta tarea debe dirigirse a la carpeta donde está su archivo ejecutable en el explorador de archivos de Windows y ejecute el **cmd**.

Para verificar que su programa se ejecuta correctamente, ingrese los coeficientes de esta función:

$$f_1(x) = 8.05 * x^4 + 14 * x^3 + 71.3 * x^2 - 66 * x - 100$$

Tome el intervalo $[-3, 1]$ y una exactitud de 0.0025, compruebe que la raíz es -0.831635 (por la precisión con la que trabaja el computador, el resultado podría variar ligeramente). Si no obtiene el resultado, solicite la asistencia de un Jefe de práctica.

Vuelva a ejecutar el programa, pero ahora ingrese los datos de esta función:

$$f_2(x) = 10.01 * x^4 - 14.22 * x^3 - 65.4 * x^2 + 13.3 * x + 1.07$$

Para el intervalo $[-3, 3]$ y con una exactitud de 0.001, compare el resultado con uno de sus compañeros para ver si coinciden.

Pruebe el programa con otros valores.

San Miguel, 31 de marzo del 2025