ESTUDIOS
GENERALES
CIENCIAS

PONTIFICIA
UNIVERSIDAD
CATÓLICA
DEL PERÚ

TÉCNICAS DE PROGRAMACIÓN

PRIMERA PRÁCTICA CALIFICADA
SEMESTRE ACADÉMICO 2021-2

Horario: Todos

Duración: 110 minutos

Elaborado por los profesores del curso.

ADVERTENCIAS:

- **SE LES RECUERDA QUE, DE ACUERDO AL REGLAMENTO DISCIPLINARIO DE NUESTRA INSTITUCIÓN, CONSTITUYE UNA FALTA GRAVE COPIAR DEL TRABAJO REALIZADO POR OTRA PERSONA O COMETER PLAGIO. ESTO, Y EL HECHO DE ENCONTRAR CUALQUIER ARCHIVO YA SEA .c O .h CON FECHA U HORA DE CREACIÓN ANTERIOR AL LABORATORIO SERÁ CONSIDERADO UNA FALTA DE PROBIDAD Y POR LO TANTO AMERITARÁ LA ANULACIÓN DE LA PRUEBA.**

INDICACIONES:

- DEBE COLOCAR SU NOMBRE Y CÓDIGO EN CADA UNO DE LOS ARCHIVOS .h Y .c QUE EMPLEE EN SUS PROYECTOS, DE LO CONTRARIO SE LE DESCONTARÁ 0.5 PUNTOS POR CADA OMISIÓN. ***NO SE HARÁN EXCEPCIONES.***

PRIMERA PARTE: Uso del entorno NetBeans

En esta primera parte usted creará un proyecto, escribirá el programa que se le proporcionará, lo compilará, ejecutará, depurará y contestará a las preguntas que se le harán. **Ver indicaciones al final del documento.**

Se desea escribir un programa que permita encontrar la raíz de una ecuación empleando el método de la posición falsa, la función es del tipo:

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

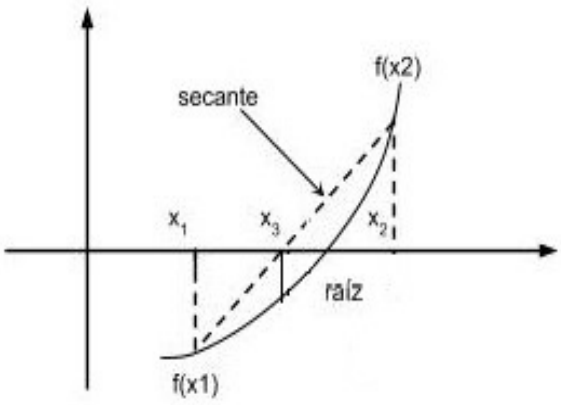
Por ejemplo para las ecuaciones:

$$f_1(x) = x^4 - 3.44 * x^3 c_2 - 18.162 * x^2 + 31.304 * x + 34.496$$

ó

$$f_1(x) = 0.01x^4 - 0.479 * x^3 c_2 + 7.772 * x^2 - 48.682 * x + 94.212$$

Esta técnica empieza con la determinación de dos puntos x1 y x2 de modo que al evaluar f(x1) y f(x2) se obtengan valores con signos diferentes, si se logran encontrar, podemos garantizar que entre esos dos puntos existe una raíz para esa 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 (x1, f(x1)) y (x2, f(x2)) y se determina el punto (x3, 0) en donde dicha secante cruza el eje x, tal como se muestra en la siguiente figura:



El nuevo punto x3, estará más cerca de la raíz que cualquiera de los puntos originales. Ahora se determina el valor de f(x3) y se verifica si f(x1) y f(x3) tienen signos diferentes, si esto es cierto, entonces se descarta el punto x2, si no es cierto entonces se descarta el punto x1

A partir de aquí se repite el proceso antes descrito, pero ahora empleando x3 y el x que no se descartó. Se generan una serie de puntos 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(xi)) es muy cercano a cero, esto es: $|f(xi)| < e$, 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.c TENEMOS:

```
/*
 * Archivo:   main.c
 * 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,
 * G., Weingart, S., Perlman, D.(1986). "Introducción a la programación
 * y solución de problemas con Pascal. México: Editorial LIMUSA.
 *=============================================================================
 */
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "funcionesAuxiliares.h"
int main(int argc, char** argv) {
    const int MAX_CICLOS = 100;
    int veces, muchosCiclos, datosCorrectos, raizEncontrada;
    double c_4, c_3, c_2, c_1, c_0; /*Coeficientes de la ecuación*/
    double x1, x2, x3, exactitud;
    /*x1 y x2 intervalo que contiene la raiz, x3 es el nuevo punto*/
    double f1x, f2x, f3x;
    /*primero encontraremos un intervalo que contenga la raíz */
    datosCorrectos = 0;
    muchosCiclos = 0;
    while(!datosCorrectos && !muchosCiclos){
        printf("Ingresar dos puntos de inicio: \n");
        scanf("%lf %lf",&x_1,&x_2);
        printf("Ingresar los coeficientes: c4 al c0 \n");
        scanf("%lf %lf %lf %lf %lf",&c_4,&c_3,&c_2,&c_1,&c_0);
        f1x = calculaFuncion(c_4,c_3,c_2,c_1,c_0,x1);
        f2x = calculaFuncion(c_4,c_3,c_2,c_1,c_0,x2);
        if((fabs(x1)<exactitud) && (fabs(x2)<exactitud)) {
             muchosCiclos = 1;
        }
        else {
            if(((f1x <= 0.0) && (f2x => 0.0)) ||
                    ((f1x >= 0.0) && (f2x <= 0.0))) {
                datosCorrectos = 1;
            }
            else {
                printf("Lo siento, los puntos dados no estan opuestos\n");
                printf("Ingresar 0,0 para terminar (puntos de inicio)\n");
            }
        }
    }
```

(CONTINUA EN LA SIGUIENTE PÁGINA)

```
    if (muchosCiclos) {
        printf("Lo siento, el programa se concluye por una falla\n");
        printf("para encontrar un intervalo inicial valido\n");
    }
    else {  //solución del problema
        printf("Ingresar la exactitud deseada: \n");
        scanf("%lf",&exactitud);
        raizEncontrada = 0;
        veces = 0;
        while (!raizEncontrada && (veces<=MAX_CICLOS)){
             x3 = (x2 * f1x - x1 * f2x)/(f1x - f2x);
             f3x = calculaFuncion(c_4,c_3,c_2,c_1,c_0,x3);
              if (fabs(f3x)<exactitud) //hemos encontrado la raiz exacta
                raizEncontrada = 1;
             if (((f1x <= 0.0) && (f3x <= 0.0)) ||
                     ((f1x >= 0.0) && (f3x >= 0.0))){
                x1 = x3 ;   f1x = f3x;
             }
             else {
                x2 = x3;    f2x = f3x;
             }
             veces += 1;
             if (fabs(f3x)<exactitud)
                raizEncontrada = 1;
        }
        if (raizEncontrada)
            printf("\nLa raiz es = %lf, con una exactitud de %lf\n",
                   x3, exactitud);
        else
            printf("%s %d %s\n",
                    "\nLo sentimos, no pudimos encontrar la raiz en ",
                    MAX_CICLOS," iteraciones");
    }
    return (EXIT_SUCCESS);
}
```

EN EL ARCHIVO funcionesAuxiliares.h TENEMOS:

```
/*
 * Archivo:   funcionesAuxiliares.h
 * Autor: ***COLOQUE AQUÍ SU CÓDIGO, NOMBRE Y APELLIDO ***/
#ifndef FUNCIONES_H
#define FUNCIONES_H

double calculaFuncion(double ,double ,double ,double ,double ,double);
double potencia(double,int);

#endif /* FUNCIONES_H */
```

EN EL ARCHIVO funciones.c TENEMOS:

```
/*
 * Archivo:   funcionesAuxiliares.c
 * Autor: ***COLOQUE AQUÍ SU CÓDIGO, NOMBRE Y APELLIDO ****/
#include "funciones.h"
double calculaFuncion(double c_4, double c_3,double c_2,
                      double c_1, double c_0, double xx) {
   double fxx;
   fxx =  c_4*potencia(xx,4) + c_3*potencia(xx,3) + c_2*potencia(xx,2) +
          c_1*xx + c_0;
   return fxx;
}
double potencia(double x, int n){
    double p = 1.0;
    for(int i=0; i<n; i++)
        p *= x;
    return p;
}
```

Según lo anterior:

En su computador cree, utilizando el entorno NetBeans, un proyecto con nombre "**Practica01Parte01_2021_2**". Una vez creado copie el programa completamente **tal cual lo ve en la hoja,** debe crear los archivos correspondientes, incluya los comentarios; respete los nombres de variables dados y la simbología empleada.

Ingrese al enlace que se encuentra en Paideia de **Cuestionario** y responda las preguntas.

ADVERTENCIAS:
- Obligatoriamente debe desarrollar su proyecto bajo NetBeans en Windows, no podrá desarrollarlo empleando otro IDE ni otro sistema operativo. - Al finalizar el laboratorio, comprima la carpeta Practica01Parte01_2021_2 en un archivo de tipo .zip (Solo debe usar el programa Zip que viene por defecto en el Windows, **NO** deberá usar Win-RAR, RAR, 7Zip, etc.) y súbalo a la tarea programa en Paideia para este laboratorio. El nombre del archivo compilado será el mismo que el del proyecto.

San Miguel, 13 de septiembre del 2021