Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Кафедра теоретичних основ радіотехніки

**ЗВІТ З ЛАБОРАТОРНОЇ РОБОТИ №3**

з дисципліни: «Інформатика 1»

|  |  |
| --- | --- |
|  | Виконав : Рокунець Андрій Олександрович  Група: РЕ-12  Викладачі: доцент Катін П.Ю.  Оцінка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Підпис: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

Київ – 2021

Мета: Створити програму для обрахунку нелінійних рівнянь за двома алгоритмами пошуку невідомого.

Обидва методи працюють у програмі. Метод хорд на однакових проміжках є точнішим методом, але метод половинного ділення є швидшим

**Код**

#include <stdio.h>

#include <stdlib.h>

#include <conio.h>

#include <math.h>

double f(double x);

double method\_hord(double x1, double x2, unsigned int N, unsigned int debug);

double method\_halfDiv(double x1, double x2, unsigned int N, unsigned int debug);

int main()

{

double x1, x2, root;

double eps = 0.0001;

unsigned int Variant; // 1-method\_hord,2-method\_halfDiv

unsigned int N, debug;

printf("Variant=");

scanf\_s("%u", &Variant);

printf("\nN=");

scanf\_s("%u", &N);

printf("\ndebug=");

scanf\_s("%u", &debug);

do {

printf("x1=");

scanf\_s("%lf", &x1);

printf("x2=");

scanf\_s("%lf", &x2);

} while (f(x1) \* f(x2) > 0);

while (Variant != 1 && Variant != 2){

printf("\nERROR. Choose Method: 1 or 2 ");

scanf\_s("%u", &Variant);

}

if (Variant == 1) {

root = method\_hord(x1, x2, N, debug);

printf("\n\nx=%lf f(x)=%lf", root, f(root));

}

else if (Variant == 2) {

root = method\_halfDiv(x1, x2, N, debug);

printf("\n\nx=%lf f(x)=%lf", root, f(root));

}

return 0;

}

//-----------------------------------------------------------------

double f(double x)

{

return pow(x, 3) + 2 \* x;

}

//------------------------------------------------------------------

double method\_hord(double x1, double x2, unsigned int N, unsigned int debug)

{

double eps = 0.0001;

double xi;

int n = 1;

while (debug != 0 && debug != 1)

{

printf("\nERROR. Choose debug: 0 or 1 ");

scanf\_s("%u", &debug);

}

if (debug == 0) {

do {

xi = (f(x2) \* x1 - f(x1) \* x2) / (f(x2) - f(x1));

if (f(xi) > 0)

x1 = xi;

else

x2 = xi;

} while (fabs(f(xi)) > eps);

}

else if (debug == 1) {

do {

xi = (f(x2) \* x1 - f(x1) \* x2) / (f(x2) - f(x1));

if (f(xi) > 0)

x1 = xi;

else

x2 = xi;

printf("\n%d %lf %lf", n, xi, f(xi));

n++;

} while (fabs(f(xi)) > eps);

}

return xi;

}

double method\_halfDiv(double x1, double x2, unsigned int N, unsigned int debug)

{

double eps = 0.0001;

double xi;

int n = 1;

while (debug != 0 && debug != 1)

{

printf("\nERROR. Choose debug: 0 or 1 ");

scanf\_s("%u", &debug);

}

if (debug == 0) {

do {

xi = (x1 + x2) / 2;

if (f(xi) > 0)

x1 = xi;

else

x2 = xi;

} while ((fabs(f(x1) - f(x2))) > eps);

}

else if (debug == 1) {

do {

xi = (x1 + x2) / 2;

if (f(xi) > 0)

x1 = xi;

else

x2 = xi;

printf("\n%d %lf %lf", n, xi, f(xi));

n++;

} while ((fabs(f(x1) - f(x2))) > eps);

}

return xi;

}

//---------------------------------------------------------------------------

