Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Кафедра теоретичних основ радіотехніки

**ЗВІТ З ЛАБОРАТОРНОЇ РОБОТИ №3**

з дисципліни: «Інформатика 1

|  |  |
| --- | --- |
|  | Виконав: Демчук В’ячеслав Андрійович  Група: РЕ-11  Викладач: доцент Катін П.Ю.  Оцінка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Підпис: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

Київ – 2021

Мета роботи: навчитися працювати з операторами часу, вказівниками, засвоїти раніше набуті навички.

Код програми:

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

#include <time.h>

#include <locale.h>

double function (double x) {

x = pow(x, 2) - 18 \* x - 83;

return x;

}

void chord\_method (double \*left\_applicant, double \*right\_applicant, double \*memory, double \*sought) {

\*sought = \*right\_applicant - function(\*right\_applicant) \* (\*left\_applicant - \*right\_applicant) / (function(\*left\_applicant) - function(\*right\_applicant));

\*left\_applicant = \*right\_applicant;

\*right\_applicant = \*memory;

}

void half\_division (double \*left\_applicant, double \*right\_applicant, double \*memory, double \*sought) {

\*sought = (\*left\_applicant + \*right\_applicant) / 2;

if (function(\*sought) \* function(\*left\_applicant) < 0)

\*right\_applicant = \*sought; else \*left\_applicant = \*sought;

}

int main()

{

setlocale(LC\_ALL, "Ukr");

double left\_limit, right\_limit, sought, error;

double left\_applicant, right\_applicant, memory, time1, time2, times;

int metod\_calculation, contin, contin3 = 0, mod, number;

int contin2 = 0;

unsigned long int counter = 0;

{

printf("\n\n\t++===================================================++\n");

printf("\t|| Лабораторна робота №3 ||\n");

printf("\t++---------------------------------------------------++\n");

printf("\t|| Розв'язати нелінійне рівняння ||\n");

printf("\t++===================================================++\n\n");

}

do {//введення початкових данних

printf("\t Введіть ліву границю\t"); scanf("%lf", &left\_limit);

left\_applicant = left\_limit;

printf("\t Введіть праву границю\t"); scanf("%lf", &right\_limit);

right\_applicant = right\_limit;

printf("\n\t\t Від 3 до 5 порядка");

printf("\n\t Введіть похибку для обчислення кореня рівняння:\t"); scanf("%lf", &error);

error = 1/pow(10, error);

printf("\t Введіть допустиму кількість ітерацій\t"); scanf("%d", &number);

} while (left\_applicant \* right\_applicant > 0);

if (left\_applicant > right\_applicant) {

memory = right\_applicant;

right\_applicant = left\_applicant;

left\_applicant = memory;

}

{//вибір метода

printf("\n\t\t 1 - Хорд");

printf("\n\t\t 2 - Половинного ділення");

printf("\n\t Оберіть метод обчислення кореня рівняння:\t"); scanf("%d", &metod\_calculation);

}

{//вибір режима

printf("\n\t\t 0 - Ні");

printf("\n\t\t 1 - Так");

printf("\n\t Працювати у режимі корегування ?:\t"); scanf("%d", &mod);

}

memory = 0; sought = 0;

do {memory = sought; time1 = GetTickCount();

if (metod\_calculation == 1) chord\_method (&left\_applicant, &right\_applicant, &memory, &sought);

if (metod\_calculation == 2) half\_division(&left\_applicant, &right\_applicant, &memory, &sought);

counter++;

if (mod) {printf("\n\t Результат обчислень на ітерації %d:\t", counter); printf("%lf", sought);}

if (!contin2) {

if (counter % number == 0) {

printf("\n\n\t Кількість проміжків вийшла за %d\t", number);

printf("\n\t\t 1 - Продовжити ще %d раз", number);

printf("\n\t\t 2 - Продовжити до кінця");

printf("\n\t\t 3 - Завершити виконання програми");

printf("\n\t Які подальші дії?\t");

scanf("%d", &contin); switch (contin) {

case 1: break;

case 2: contin2 = 1; break;

case 3: contin3 = 1; break;

}

}

}

if (contin3) break;

} while (fabs(function(sought)) > error);

time2 = GetTickCount(); times = time2 - time1;

{//результати

printf("\n\n\t Лівий вказівник:\t"); printf("%lf", left\_applicant);

printf("\n\t Правий вказівник:\t"); printf("%lf", right\_applicant);

printf("\n\t Кількість ітерацій:\t"); printf("%lu", counter);

printf("\n\t Похибка обчислень:\t"); printf("%lf", error);

printf("\n\n\t Початкова функція:\t"); printf("f(x) = x^3 - 18x - 83");

printf("\n\t Ліва границя:\t"); printf("%lf", left\_limit);

printf("\n\t Права границя:\t"); printf("%lf", right\_limit);

printf("\n\t Час роботи:\t"); printf("%lu ms", times);

printf("\n\t Результат обчислень:\t"); printf("%lf", sought);

}

printf("\n\n\t\t\tПісля натиснення будь-якої клавіші програма повториться\n\n");

getch(); system("cls"); main();

return 0;

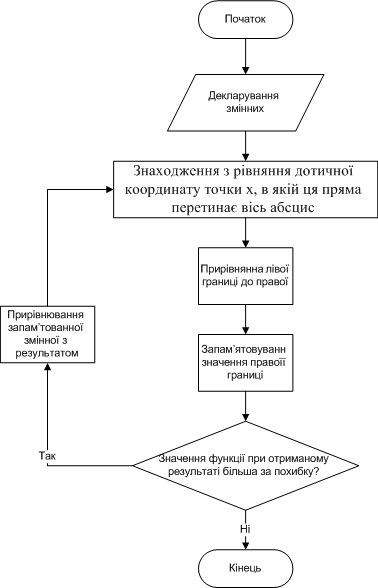
}

Висновки:

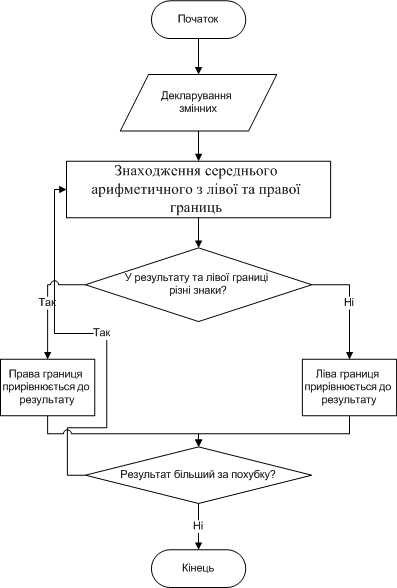
Обидва способи мають досить велику точність та малу кількість ітерацій, час майже неможливо виміряти, так як він дуже малий. Але обидва методи можуть працювати некоректно, якщо на заданому проміжку є 2 або більше коренів.

Блок-схеми:

Метод хорд:



Метод половинного ділення:



Обрана функція та обраховане вручну значення кореня:

f(x)=x2-18x-83

X=5,705