Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Кафедра теоретичних основ радіотехніки

**ЗВІТ З ЛАБОРАТОРНОЇ РОБОТИ №3**

з дисципліни: «Інформатика »

|  |  |
| --- | --- |
|  | Виконав: Бондарчук Н.О  Група: РЕ-11  Викладачі: доцент Катін П.Ю.  Оцінка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Підпис: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

Київ – 2021

**Мета роботи**: розробка програми розв’язання нелінійних рівнянь

**Ключові моменти**:

1)Для того щоб отримати час затрачений на знаходження кореня я використовую бібліотеку <time.h> та функцію clock() яка повертає значення часу у даний момент.

Пишу на початку обчислень clock\_t T1 = clock();

Функція clock() повертає час, пройдений з моменту старту програми, що визвала функцію clock().

2) За допомогою getch() записую у змінну і значення яке ввів користувач з клавіатури.

i = getch();

3) if((int)i==27) – цим я перевію, чи дорівнює значення І == 27, бо саме 27 означає ESC

4)У кінці знаходження кореня я пишу clock\_t T2 = clock().Так я записую у змінну Т2 значення часу у даний момент, і щоб отримати значення за який час програма знайшла корінь я пишу:

(double)(T2 - T1) / CLOCKS\_PER\_SEC

5) system("cls"); використовується для очищення консолі

**Код:**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

#include <conio.h>

#include <time.h>

#define lip 0.0001

double f(double x)// Функція

{

return 2\*pow(x,3)-0.2\*pow(x,2)+0.5\*x+3;

}

double f1(double x) //Перша директива функції

{

return 6\*pow(x,2)-0.4\*x+0.5 ;

}

double f2(double x) //Друга директива функції

{

return 12\*x-0.4;

}

void dump(double xi,double fault,int n,double Timer)

{

printf("x = %lf\n",xi);

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

printf("Count of iterations = %d\n",n);

printf("Calculation time = %lf\n",Timer);

}

int main()

{

double x1, x2,xi,timer,fault;

unsigned int N, n = 0,var;

char i;

printf("1 - chord method\n2 - half chord method\n3 - method Newton\n");

scanf("%d",&var);

switch(var)

{

case 1://Метод хорд

do

{

printf("Left limit: \n");

scanf("%lf",&x1);

printf("Right limit: \n");

scanf("%lf",&x2);

}

while(f(x1)>0 && f(x2)<0);

printf("Fault: \n");

scanf("%lf",&fault);

printf("Max count of iterations: \n");

clock\_t T1 = clock();

scanf("%d",&N);

do

{

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

if( f(xi) > 0 )

{

x2=xi;

}

else

{

x1=xi;

}

if(n==N)

{

printf("The maximum number of iterations is reached\n");

printf("Press ESC to finish the calculation \nor press any key to continue the calculation\n");

i = getch();

if((int)i==27)

{

system("cls");

break;

}

else

{

system("cls");

}

}

n++;

}

while ( fabs (f(xi)) > fault );

clock\_t T2 = clock();

dump(xi,fabs (f(xi)),n,(double)(T2 - T1) / CLOCKS\_PER\_SEC);

break;

case 2:// Метод половинного ділення

do

{

printf("Left limit: \n");

scanf("%lf",&x1);

printf("Right limit: \n");

scanf("%lf",&x2);

}

while(f(x1)>0 && f(x2)<0);

printf("Fault: \n");

scanf("%lf",&fault);

printf("Max count of iterations: \n");

clock\_t T11 = clock();

scanf("%d",&N);

do

{

xi=(x1+x2)/2;

if( f(xi) > 0 )

{

x2=xi;

}

else

{

x1=xi;

}

if(n==N)

{

printf("The maximum number of iterations is reached\n");

printf("Press ESC to finish the calculation \nor press any key to continue the calculation\n");

i = getch();

if((int)i==27)

{

system("cls");

break;

}

else

{

system("cls");

}

}

n++;

}

while ( fabs (f(xi)) > fault );

clock\_t T22 = clock();

dump(xi,fabs (f(xi)),n,(double)(T22 - T11) / CLOCKS\_PER\_SEC);

break;

case 3://Метод Ньютона

do

{

printf("Left limit: \n");

scanf("%lf",&x1);

printf("Right limit: \n");

scanf("%lf",&x2);

}

while(f(x1)>0 && f(x2)<0);

printf("Fault: \n");

scanf("%lf",&fault);

printf("Max count of iterations: \n");

clock\_t T111 = clock();

scanf("%d",&N);

if(f(x1)\*f2(x1)<0)

xi=x1;

else

xi=x2;

do

{

xi=xi-f(xi)/f1(xi);

n++;

}

while (fabs(f(xi))>=lip);

if(n==N)

{

printf("The maximum number of iterations is reached\n");

printf("Press ESC to finish the calculation \nor press any key to continue the calculation\n");

i = getch();

if((int)i==27)

{

system("cls");

break;

}

else

{

system("cls");

}

}

n++;

while ( fabs (f(xi)) > fault );

clock\_t T222 = clock();

dump(xi,fabs (f(xi)),n,(double)(T222 - T111) / CLOCKS\_PER\_SEC);

break;

}

return 0;

}

**Висновок:** При виконанні лабараторної, ознайомився з використанням бібліотеки <time.h> та функції clock().Дізнався більше про функцію getch та її використання.