Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Кафедра теоретичних основ радіотехніки

**ЗВІТ З ЛАБОРАТОРНОЇ РОБОТИ №3**

з дисципліни: «Інформатика 1»

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

Київ – 2021

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

**Обрана функція**: 

**Ключові моменти**:

1. Для знаходження часу обчислення використовую <time.h> та clock(). На почтаку clock\_Т1, в кінці clock\_Т2.
2. getch() повертає символ, зчитаний з консолі, але не записує(потрібно для таблиці Аскі)
3. Використовуємо таблиці Аскі аби зчитати "Space"(значення у таблиці 32).
4. Знаходжу час розрахунків :(Т2-Т1)/ CLOCKS\_PER\_SEC (CLOCKS\_PER\_SEC – змінна, зафіксована у бібліотеці <time.h>)

**Код:**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

#include <conio.h>

#include <time.h>

double F(double x)

{

return pow(x,4)-0.2\*pow(x,3)+0.5\*pow(x,2)+1.5;

}

void info(double xi,double mistake,int n,double Timer)

{

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

printf("Fault = %lf\n",mistake);

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

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

}

int main()

{

//My function y=pow(x,3)-0.2\*pow(x,2)+0.5\*x+1.5

double x1, x2,xi,timer,mistake;

unsigned int N, n = 0,var;

char i;

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

scanf("%d",&var);

switch(var)

{

case 1:

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",&mistake);

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 Space to finish the calculation \nor press any key to continue the calculation\n");

i = getch();

if((int)i==32)

{

system("cls");

break;

}

else

{

system("cls");

}

}

n++;

}

while ( fabs (F(xi)) > mistake );

clock\_t T2 = clock();

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

break;

case 2:

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",&mistake);

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 Space to finish the calculation \nor press any key to continue the calculation\n");

i = getch();

if((int)i==32)

{

system("cls");

break;

}

else

{

system("cls");

}

}

n++;

}

while ( fabs (F(xi)) > mistake );

clock\_t T22 = clock();

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

break;

case 3:

do

{

printf("Left limit: \n");

scanf("%lf",&x1);

}

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

printf("Fault: \n");

scanf("%lf",&mistake);

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

clock\_t T111 = clock();

scanf("%d",&N);

do

{

xi = x1 - F(x1);

if( fabs(F(xi))< fabs(F(x1)) )

{

x1 = xi;

}

if(n==N)

{

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

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

i = getch();

if((int)i==32)

{

system("cls");

break;

}

else

{

system("cls");

}

}

n++;

}

while ( fabs (F(xi)) > mistake );

clock\_t T222 = clock();

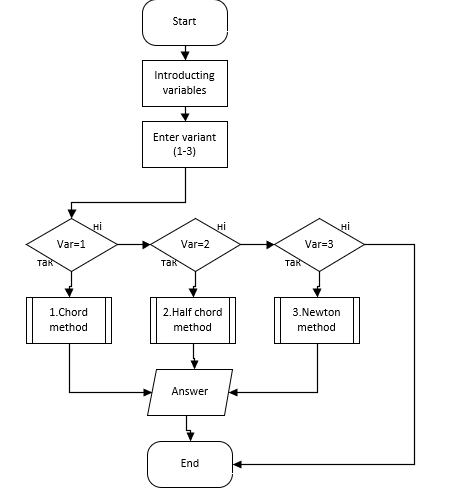
info(xi,fabs (F(xi)),n,(double)(T222 - T111) / CLOCKS\_PER\_SEC);

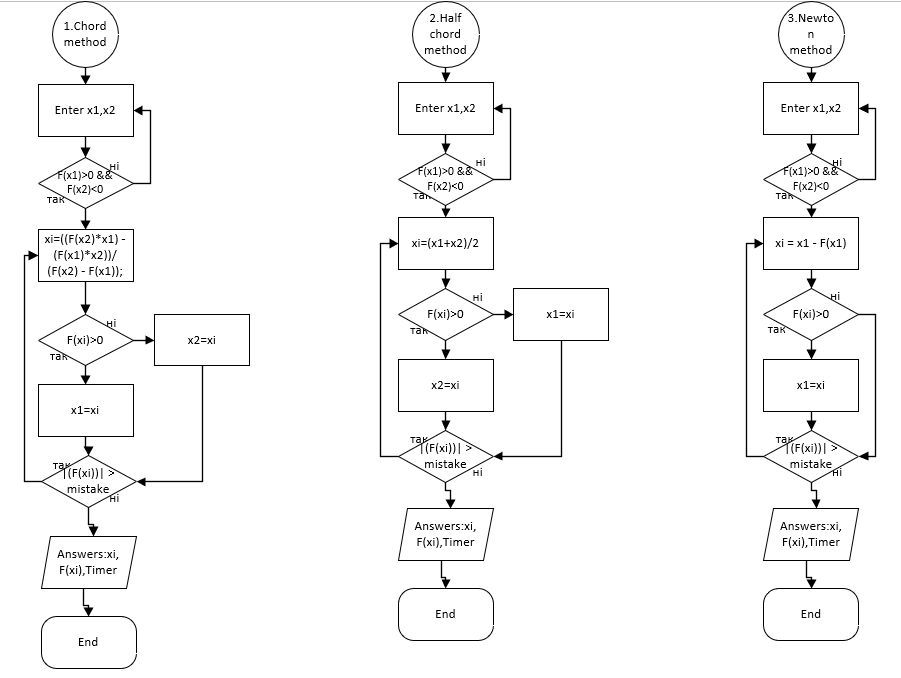
break;

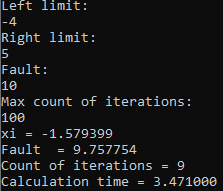
}

return 0;

}





***Висновки:***

Розробили програми розв’язання нелінійних рівняннь (Метод хорд(мал.1), метод половинних хорд(мал.2), метод Ньютона(мал.3,4))

