

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Кафедра теоретичних основ радіотехніки

ЗВІТ З ЛАБОРАТОРНОЇ РОБОТИ №3

з дисципліни: «Інформатика 1»

		Виконав : Луцкевич Віктор Андрійович	
		Група: РЕ-12	
		Викладачі: доцент Катін П.Ю.	
		Оцінка:	

		Підпис:	
--	--	---------	--

Київ – 2021

Мета: розробка програми розв'язання нелінійних рівнянь.

Код

```
#define _CRT_SECURE_NO_WARNINGS

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <conio.h>

#define eps 0.0001 //наближення до кореня в межах певної eps

double f(double x); //прототип функції для якої знаходимо корінь

double metod_hord(double x1, double x2, unsigned int N, unsigned int debug); //N -
кількість ітерацій, debug- режим налагодження відповідає за те чи потрібно виводити кожну
ітерацію

//алгоритм реалізовується в межах функції method_hord

double metod_polowynnego_dilenja(double x1, double x2, unsigned int N, unsigned int
debug);

double metod_Newton(double x1, unsigned int N, unsigned int debug);
```

```

int main()
{
    system("chcp 1251");

    double x1, x2, root, m = 0.001; // x1 x2 початок та кінець діапазону пошуку
    кореня, root - зберігає корінь

    unsigned int v, N = 100, debug = 1; // debug = 0 no debug. 1 - print result at
    each iteration

    printf("1)Метод хорд\n2)Метод половинного ділення\n 3)Метод Ньютона (метод
    дотичних) ");

    do
    {
        printf("\nМетод - ");
        scanf("\n%u, " ,&v);

    } while (v != 1 && v != 2 && v != 3 );
    system("cls");
    if (v== 1)
    {
        //do {

            system("cls");
            printf("x1=");
            scanf("%lf", &x1);
            printf("x2=");
            scanf("%lf", &x2);

            do
            {

```

```

        x1 +=m;

        x2 -= m;

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

    printf("x1=%lf", x1);
    printf("x2=%lf", x2);

    root = metod_hord(x1, x2, N, debug); // після проходження перевірки
    початкових умов, викликається функція metod_hord і передаються параметри

    printf("\n\nx=%lf f(x)=%lf", root, f(root)); //вивести значення x та
функції
}

else if (v == 2)
{
    do {

        system("cls");
        printf("x1=");
        scanf("%lf", &x1);
        printf("x2=");
        scanf("%lf", &x2);

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

    root = metod_polowynnogo_dilenja(x1, x2, N, debug);
    printf("\n\nx=%lf f(x)=%lf", root, f(root));

```

```

    }
    else if (v == 3)
    {

        printf("x1=");
        scanf("%lf", &x1);

        root = metod_Newton(x1, N, debug);
        printf("\n\nx=%lf  f(x)=%lf", root, f(root));
    }

    return 0;
}

double f(double x)
{
    return 2 * pow(x - 4, 3) + 2 * x; // функція pow для x у кубі  $2(x-4)^3+2x=0$ 
}

double metod_hord(double x1, double x2, unsigned int N, unsigned int debug) // алгоритм
виконується в межах функції metod_hord
{
    double xi; // координата точки в якій хорда перетинає вісь абсцис
    int n = 1;

```

```

do
{
    xi = ((f(x2) * x1 - f(x1) * x2) / (f(x2) - f(x1))); // згідно з формулою

    if (f(xi) * f(x1) > 0) // якщо f(xi) и f(x1) знаходяться в одній півплощині
    {
        x1 = xi;
    }
    else
    {
        x2 = xi; //в іншому випадку, якщо у нас f(xi) и f(x1) знаходяться
по різних півплощинах це значить що f(xi) знаходиться в одній півплощині з f(x2)
    }

    if (debug == 1)
        printf("\n %d %lf %lf", n, xi, f(xi)); // n - номер

    n++;
} while (fabs(f(xi) > eps)); // поки модуль f(xi) буде більше eps

return xi;
}

double metod_polowynnego_dilenja(double x1, double x2, unsigned int N, unsigned int
debug) {

    double xj; // нова точка на середині відрізка

    int n = 1;

    do
    {
        xj = (x1 + x2) / 2;

```

```

        if (f(xj) < 0 /*x1 <= f(xj) <= xj*/)
        {
            x1 = xj;
        }
        else if (f(xj) > 0 /*xj <= f(xj) <= x2*/)
        {
            x2 = xj;
        }
        if (debug == 1)
            printf("\n %d %lf %lf", n, xj, f(xj));
        n++;

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

    return xj;

}

double metod_Newton(double x1, unsigned int N, unsigned int debug)
{
    double xJ; // значення рівняння
    int n = 1, h= 0.1; // h - крок

    double x = 0;
    double fr;
    do
    {

```

```

fr = (f(x1+0.1) - f(x1)) / h; // похідна

/*double x = fr * x1;*/

xJ = f(x1) + (fr * (x - x1));

xJ = x1 - (f(x1) / fr);

if (debug == 1)
    printf("\n %d %lf %lf", n, xJ, f(xJ));
n++;

} while (fabs(f(xJ))<=eps);
return xJ;

}

```

Приклад роботи програми

C:\Users\User\source\repos\Lab3v1\Debug\Lab3v1.exe

Active code page: 1251

1)Метод хорд

2)Метод половинного ділення

3)Метод Ньютона (метод дотичних)

Метод -

Microsoft Visual Studio Debug Console

x1=3

x2=2

1	2.500000	-1.750000
2	2.250000	-6.218750
3	2.125000	-8.933594
4	2.062500	-10.421387
5	2.031250	-11.199158
6	2.015625	-11.596672
7	2.007813	-11.797606
8	2.003906	-11.898620
9	2.001953	-11.949265
10	2.000977	-11.974621
11	2.000488	-11.987308
12	2.000244	-11.993653
13	2.000122	-11.996826
14	2.000061	-11.998413

x=2.000061 f(x)=-11.998413

C:\Users\User\source\repos\Lab3v1\Debug\Lab3v1.exe (process 47852) exited

To automatically close the console when debugging stops, enable Tools->Op

Press any key to close this window . . .

