Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

**ЗВІТ З ЛАБОРАТОРНОЇ РОБОТИ №3**

з дисципліни:« Інформатика1. Основи програмування та алгоритми»

|  |  |
| --- | --- |
|  | Виконала: Петруненко Олександр  Група: РЕ-12  Викладач: Доцент Катін Ю.П |

Київ – 2021

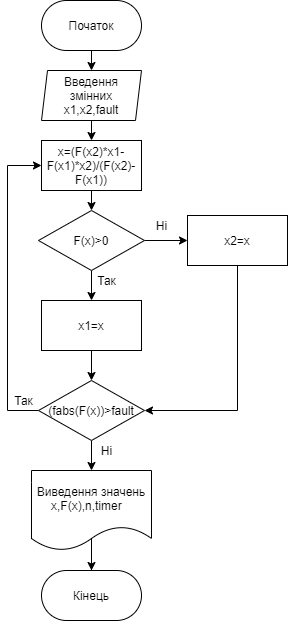
Мета роботи: навчитися створювати програму для розв’язку не лінійного рівняння трьома методами(методом хорд, методом половинного ділення та методом Ньютона).

Обрана функція:

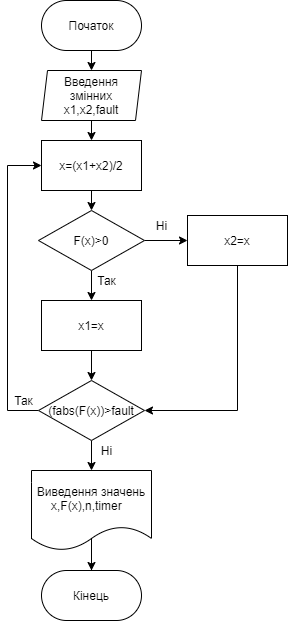
Похідна від функції:

Обраховане вручну значення кореня:

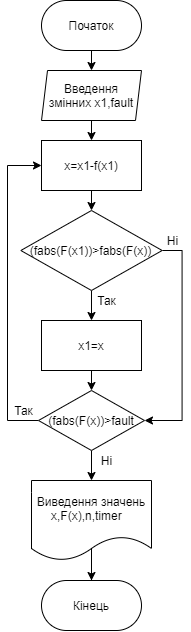
Метод хорд:



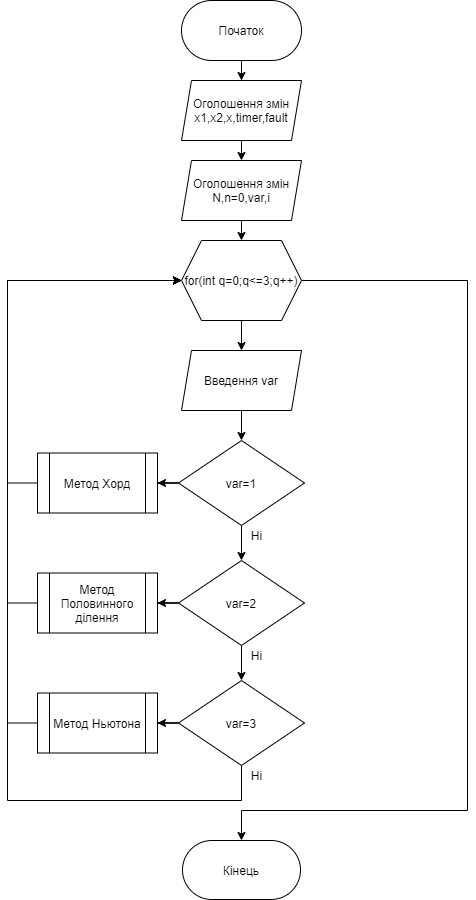
Метод половинного ділення:



Метод Ньютона (дотична):



Основна блок-схема:



Код:

**#include <*stdio.h*>**

**#include <*stdlib.h*>**

**#include <*math.h*>**

**#include <*conio.h*>**

**#include <*time.h*>**

**double F(double x)//function**

**{**

**return pow(2\*x-3,2)-4\*x+7;**

**}**

**double f(double x)//**

**{**

**return (x-2)/2;**

**}**

**void info(double xi,double fault,int n,double Timer)**

**{**

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

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

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

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

**}**

**int main()**

**{**

**//My function**

**double x1, x2,xi,timer,fault;**

**unsigned int N, n = 0,var;**

**char i;**

**for(int q=0;q<=3;q++)**

**{**

**printf("Choose your method:\n");**

**printf("Chord method -- 1\nHalf-Chord method -- 2\nNewton`s method(tangent method) -- 3\nChoice - ");**

**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",&fault);**

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

**scanf("%d",&N);**

**clock\_t T1 = clock();**

**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();**

**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",&fault);**

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

**scanf("%d",&N);**

**clock\_t T11 = clock();**

**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();**

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

**break;**

**case 3:**

**printf("X: \n");**

**scanf("%lf",&x1);**

**printf("Fault: \n");**

**scanf("%lf",&fault);**

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

**scanf("%d",&N);**

**clock\_t T111 = clock();**

**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 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();**

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

**break;**

**default:**

**printf("\nYou`ve entered wrong variant, please choose one from the list below\n\n");**

**}**

**}**

**system("cls");**

**return 0;**

**}**

**Час буде дорівнювати 0, оскільки машина обчислює значення надзвичайно швидко**