**ЗВІТ З ЛАБОРАТОРНОЇ РОБОТИ**

**№3**

з дисципліни:

«ІНФОРМАТИКА. Основи програмування та алгоритми»

|  |  |
| --- | --- |
|  | Виконав: Сорока Михайло  Група: РЕ–11  Викладачі:  доцент Катін П.Ю.  Оцінка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

Київ – 2021

Мета роботи: навчитися створювати програму для розв’язку нелінійного рівняння двома методами: методом половинного ділення та методом Ньютона.

Обрана функція: e^x -5x+1 .

Обраховане вручну значення кореня:

Код:

#include <stdlib.h>

#include <math.h>

#include <stdbool.h>

#include <time.h>

double f(double x)//моя функция

{

return exp(x)-5\*x+1;

}

double diff1(double (\*f)(double x), double x)//первая производная

{

double step = 1e-4;

return (f(x+step)-f(x-step))/(2\*step);

}

double diff2(double (\*f), double x)//вторая производная

{

double step = 1e-4;

return (diff1(f, x+step)-diff1(f, x-step))/(2\*step);

}

double diff1\_max(double (\*f)(double x), double a, double b)//максимальное значение производной = M

{

double max, step, tmp;

step = 0.1;

max = fabs(diff1(f, a));

for(int i =1; a+i\*step;i++ )

{

tmp = fabs(diff1(f, a + i\*step));

if(tmp > max) max = tmp;

}

return max;

}

double diff1\_min(double (\*f)(double x), double a, double b)//миним. знач. производное = m

{

double min, step, tmp;

step = 0.1;

min = fabs(diff1(f, a));

for(int i =1; a+i\*step;i++ )

{

tmp = fabs(diff1(f, a + i\*step));

if(tmp < min) min = tmp;

}

return min;

}

double chord(double (\*f)(double x), double a, double b, double step, int counter\_of\_iterations)//метод хорд

{

int start = clock();//засекаю время

bool True = true;

double x\_new, x\_last, pin, m, M;//x\_new = X(n);x\_last = X(n-1)

//pin - закрепленный конец,M - макс. значение модуля производной

//m - мин.знач. модуля произв.

int count = 1;

int variant1 = 1;

if (f(a)\*diff2(f,a) > 0)//функции выпуклой вниз

{

x\_last = b;

pin = a;

}

else if(f(b)\*diff2(f,b) > 0)

{

x\_last = a;

pin = b;

}

else if(f(a)\*diff2(f,a) < 0)//выпуклая вверх

{

x\_last = a;

pin = b;

}

else if(f(b)\*diff2(f,b) < 0)

{

x\_last = b;

pin = a;

}

else

{

printf("Error");

return 0;

}

m = diff1\_min(f, a, b);

M = diff1\_max(f, a, b);

while(true)

{

if(count%counter\_of\_iterations == 0)

{

printf("\nN = %d\n", count);

printf("\nEnter \"1\" to select a different number of iterations.\nEnter \"2\" to continue with the same number of iterations.\nEnter \"3\" to finish at that number of iterations");

printf("\nYour choise:");

scanf("%d", &variant1);

if(variant1 == 1)

{

printf("Enter the number of iterations before stopping\n");

scanf("%d", &counter\_of\_iterations);

printf("Your choise: %d", counter\_of\_iterations);

x\_new = x\_last - (pin - x\_last)\*(f(x\_last)) / (f(pin)-f(x\_last));

count++;

if(fabs(x\_new - x\_last) < step\*m/(M-m)) break;//условие выхода

x\_last = x\_new;

}

else if(variant1 == 2)

{

printf("Press enter to continue");

count++;

getch();

}

else if(variant1 == 3)break;

}

else

{

x\_new = x\_last - (pin - x\_last)\*(f(x\_last)) / (f(pin)-f(x\_last));

count++;

if(fabs(x\_new - x\_last) < step\*m/(M-m)) break;

x\_last = x\_new;

}

}

printf("\nN= %d", count);

int end = clock();

int time = (end - start) / CLOCKS\_PER\_SEC;//подсчет времени

printf("\n%d second\n", time);

return x\_new;

}

double newton(double (\*f)(double x), double a2, double b2, double step, int counter\_of\_iterations)//метод ньютона

{

int start = clock();//засекаю время

double x1, x2, x3, q;//X1 - X(n-1),X2- X(n),X3 - X(n-1)

int count = 0;

int variant1;

if(f(a2) \* diff2(f, a2) > 0) x1 = a2;// выпуклая вниз

else x1 = b2;//Вверх

while(true)

{

if(count%counter\_of\_iterations == 0)//реализация остановки при определеном количестве итерации

{

printf("\nN = %d\n", count);

printf("\nEnter \"1\" to select a different number of iterations.\nEnter \"2\" to continue with the same number of iterations.\nEnter \"3\" to finish at that number of iterations");

printf("\nYour choise:");

scanf("%d", &variant1);

if(variant1 == 1)

{

printf("Enter the number of iterations before stopping\n");

scanf("%d", &counter\_of\_iterations);

printf("Your choise: %d", counter\_of\_iterations);

if(count == 0)

{

x2 = x1 - f(x1) / diff1(f, x1);

x3 = x2 - f(x2) / diff1(f, x2);

}

q = (x3 - x2)/(x2 - x1);

if((x3 -x2)\*q/(1-q) < step) break;

x1 = x2;

x2 = x1 - f(x1) / diff1(f, x1);

x2 = x3;

x3 = x2 - f(x2) / diff1(f, x2);

count++;

}

else if(variant1 == 2)

{

printf("Press enter to continue");

count++;

getch();

}

else if(variant1 == 3)break;

}

else

{

if(count == 0)

{

x2 = x1 - f(x1) / diff1(f, x1);

x3 = x2 - f(x2) / diff1(f, x2);

}

q = (x3 - x2)/(x2 - x1);

if((x3 -x2)\*q/(1-q) < step) break;

x1 = x2;

x2 = x1 - f(x1) / diff1(f, x1);

x2 = x3;

x3 = x2 - f(x2) / diff1(f, x2);

count++;

}

}

printf("\nN = %d", count);

int end = clock();

int time = (end - start) / CLOCKS\_PER\_SEC;

printf("\n%d second\n", time);

return x3;

}

double half(double(\*f)(double x), double a, double b, double eps, int counter\_of\_iterations)// метод половиного деления

{

int start = clock();//засекаю время

int variant1;

int count = 1;

double c;//середина отрезка

do

{

if(count%counter\_of\_iterations == 0)

{

printf("\nN = %d\n", count);

printf("\nEnter \"1\" to select a different number of iterations.\nEnter \"2\" to continue with the same number of iterations.\nEnter \"3\" to finish at that number of iterations");

printf("\nYour choise:");

scanf("%d", &variant1);

if(variant1 == 1)

{

printf("Enter the number of iterations before stopping\n");

scanf("%d", &counter\_of\_iterations);

printf("Your choise: %d", counter\_of\_iterations);

c = (a + b) / 2;

if(f(a) \* f(c) < 0) b = c;

else if(f(b) \* f(c) < 0 ) a = c;

else

{

printf("The root is not found");

return 0;

}

count++;

}

else if(variant1 == 2)

{

printf("Press enter to continue");

count++;

getch();

}

else if(variant1 == 3) break;

}

else

{

c = (a + b) / 2;

if(f(a) \* f(c) < 0) b = c;

else if(f(b) \* f(c) < 0 ) a = c;

else

{

printf("The root is not found");

return 0;

}

count++;

}

}while (fabs(b-a) > eps);// условие выполнения функции

int end = clock();

int time = (end - start) / CLOCKS\_PER\_SEC;

printf("\n%d second\n", time);

printf("\n\n\nN = %d\n", count);

return c;

}

int main()// основная функция

{

double a = -1;

double b = 15;

double a2 = -1;

double b2 = 25;

double eps = 0.0001;

int variant;

int counter\_of\_iterations;

printf("Hello world\n");

printf("My error is 0.0001\n\n");

printf("If you want to solve the equation by the \"Newton method\", enter the \"1\".\nIf you want to solve by the \"Chord method\", enter the \"2\".\nIf you want to use the \"Half division\" method, enter the \"3\".\nYour choice:");

scanf("%d", &variant);

if(variant == 2)//метод хорд

{

printf("\nYour choose the \"Chord method\"");

printf("\nEnter the number of iterations before stopping\n");

scanf("%d", &counter\_of\_iterations);

printf("Your choise: %d", counter\_of\_iterations);

printf("\nEnter the x1:");

scanf("%lf", &a);

printf("\nx1 = %.2lf", a);

printf("\nEnter the x2:");

scanf("%lf", &b);

printf("\nx2 = %.2lf", b);

printf("\nRoot by chord = %.8lf\n", chord(f, a, b, eps, counter\_of\_iterations));

}

if(variant == 1)//метод ньютона

{

printf("\nYour choose the \"Newton method\"");

printf("\nEnter the number of iterations before stopping\n");

scanf("%d", &counter\_of\_iterations);

printf("Your choise: %d", counter\_of\_iterations);

printf("\nEnter the x1:");

scanf("%lf", &b2);

printf("\nx1 = %.2lf", b2);

printf("\nRoot by newton = %.lf\n", newton(f, a2, b2, eps, counter\_of\_iterations));

}

if(variant == 3)//метод половиного деления

{

printf("\nYour choose the \"Half division\"");

printf("\nEnter the number of iterations before stopping\n");

scanf("%d", &counter\_of\_iterations);

printf("Your choise: %d", counter\_of\_iterations);

printf("\nEnter the x1:");

scanf("%lf", &a);

printf("\nx1 = %.2lf", a);

printf("\nEnter the x2:");

scanf("%lf", &b);

printf("\nx2 = %.2lf", b);

printf("\nRoot by chord = %.8lf\n", half(f, a, b, eps, counter\_of\_iterations));

}

return 0;

}