МИНЕСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА ПРИКЛАДНОЙ МАТЕМАТИКИ

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| доц., канд. техн. наук |  |  |  | О. М. Косогоров |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ |
| Решение нелинейных уравнений. |
| по курсу: ВЫЧИСЛИТЕЛЬНАЯ МАТЕМАТИКА |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | М023 |  |  |  | Д.А.Трегуб |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2021

**1)Задание:**

Реализовать методы дихотомии, хорд и Ньютона как на C++, так и на MATLAB, решить одно из уравнений вашего варианта (в файле задания это пункт 2) всеми тремя методами, сравнить результаты.

**2)Решение:**

2.1.Код на C++:

#include <iostream>

#include "math.h"

using namespace std;

class nonlinear\_queat {

private:

double a, b, c, i, j, eps, x0;

public:

nonlinear\_queat(){

a = 0; b = 0; c = 0; i = 0, j = 0, eps = 0.1, x0 = 0;

}

void enter(int k = 0) {

do {

if (k != 0) {

cout << "Enter x0: ";

cin >> x0;

}

else {

cout << "Enter interval ( i, j): ";

cin >> i >> j;

}

cout << "Enter coefficients of a non-Linear equations ( a, b, c): ";

cin >> a >> b >> c;

if ((func(i, a, b, c) \* func(j, a, b, c) > 0) && (k==0))

cout << "Wrong interval!\n";

} while (func(i, a, b, c) \* func(j, a, b, c) > 0 && (k == 0));

cout << "epsilon: ";

cin >> eps;

}

double func(double x, double a, double b, double c) {return a \* x \* x + b \* x + c;}

double Dihotomia(){

double q = 0;

enter();

while ((j - i) / 2 > eps) {

q = (i + j) / 2;

if ((func(i, a, b, c) \* func(q, a, b, c)) > 0)

i = q;

else

j = q;

}

return q;

}

double Chords() {

double Fi = 0, Fj = 0, Fq = 0; double q;

enter();

Fi = func(i, a, b, c);

Fj = func(j, a, b, c);

if (Fi \* Fj > 0) {

cout << "error" << endl;

return 0;

}

do {

q = i - (j - i) / (func(j, a, b, c) - func(i, a, b, c)) \* func(i, a, b, c);

Fq = func(q, a, b, c);

if (Fi \* Fq) {

i = q;

Fi = Fq;

}

else {

j = q;

Fj = Fq;

}

} while (abs(j - i) > eps);

return q;

}

double Newton(int k) {

double f0 = 0, f1 = 0, f2 = 0, x1 = 0;

int t = 0;

enter(k);

f0 = func(x0, a, b, c);

f1 = (func((x0 + eps), a, b, c) - func((x0 - eps), a, b, c) / (2 \* eps));

f2 = (func((x0 + eps), a, b, c) - 2 \* func(x0, a, b, c) + func((x0 - eps), a, b, c) / (eps \* eps));

if (abs(f0 \* f2 / pow(f1, 2)) >= 1) {

cout << "Process diverges" << endl;

return 0;

}

do {

if (t > 0) {

x0 = x1;

f0 = func(x0, a, b, c);

}

x1 = x0 - f0 / f1;

t++;

} while (abs(x1 - x0) > eps);

return x1;

}

};

int main() {

int k;

double x = 0;

bool t = true;

nonlinear\_queat q = nonlinear\_queat();

do {

cout << "Choose method:" << endl;

cout << "1)Dichotomy:" << endl;

cout << "2)Chords:" << endl;

cout << "3)Newton:" << endl;

cin >> k;

switch (k) {

case(1):

x = q.Dihotomia();

break;

case(2):

x = q.Chords();

break;

case(3):

x = q.Newton(k);

break;

default:

cout << "You enter wrong value";

}

cout << "x = " << x << endl;

cout << "Proceed calculations?1/0" << endl;

cin >> t;

system("cls");

} while (t == true);

return 0;

}

2.2.Код на Matlab:

Метод Дихотомии:  
f=@(x) exp(-2\*x) + (-2\*x + 1) ;%Заданная функция  
a = 0; % Нижняя граница интервала  
b = 2; % Верхняя граница интервала  
e = 0.0001 ;% Точность решения  
while abs(b-a)>e % Выполняем действия до превышения точности  
c=(b+a)/2. ; % Формула вычислений  
if (f(a))\*(f(c))<=0; % Если функция больше или равна нулю то берем[b,c]иначе[a,c]  
b=c;  
else  
a=c;  
end  
end  
x=(a+b)/2;  
disp(x); %вывод ответа  
disp('Проверка:');  
disp(exp(-2\*x) + (-2\*x + 1));

Метод хорд:

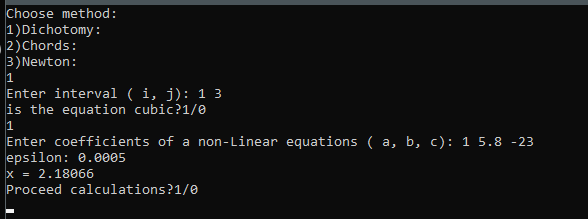
funq=inline('3\*exp(-5\*x)\*sin(7\*x+0.1)-1')  
x=roots([3 -5 7 0.1 -1])  
a=-0.2;  
b=0.2;  
E=0.0001;  
i=0;  
while abs(b-a)>=E  
i=i+1;  
c=a-funq(a)\*(b-a)/(funq(b)-funq(a));  
if funq(a)\*funq(c)<0  
b=c;  
else;  
a=c;  
end;  
end;  
c=a-funq(a)\*(b-a)/(funq(b)-funq(a));  
i

Метод Ньютона:

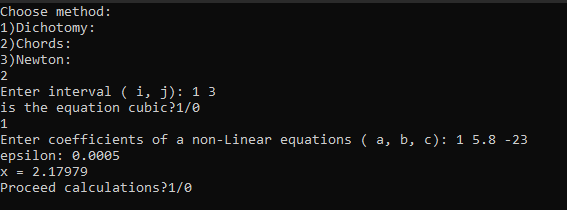
function [Xk k] = newton(f,df,tx)  
x0 = tx(1); % нижняя граница интервала  
x1 = tx(2); % верхняя граница интервала  
F = inline(f); % определяем функцию  
DF = inline(df); % определяем производную  
e = 1e-5; % Погрешность  
% начальные приближения  
Xkm = (x0+x1)/2;   
Yo = F(x0);  
k = 0; % счетчик итераций  
while abs(Yo) > e   
 Xk = Xkm - F(Xkm)/DF(Xkm);  
 Xkm = Xk;  
 Yo = F(Xk);  
 k = k + 1;  
end  
clear, clc  
  
diap = [0, 3]; % диапазон x  
fun = 'x^3 -2\*x - 5'; % функция  
dfun = char(diff(sym(fun))) % производная  
[x0 n] = newton(fun,dfun,diap);  
fprintf('Корень уравнения х = %f, \n', x0);  
fprintf('Количество итераций равно %d. \n',n);

**3.Решение нелинейного уравнения:**

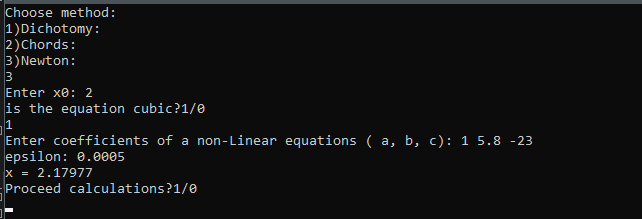
Уравнение:   
Метод Дихотомии:



Метод хорд:



Метод Ньютона:



**4.Вывод:**  
Выполняя данную лабораторную работу я реализовал методы дихотомии, хорд и Ньютона как на C++, так и на Matlab. После решил уравнение всеми тремя методами c заданной по заданию точностью, получив следующие результаты:  
Дихотомия: 2.18066  
Хорд:2.17979  
Ньютона:2.17977  
При увеличении точности вычисления и уменьшение интервала все три метода сходятся к одному числу - 2.17979. При одинаковых входных данных метод Хорд оказался точнее метода Дихотомии. Метод Ньютона при относительной близости приближения показал очень хорошие результаты, дав ошибку всего в 0.00002.