**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Основы информатики»

Отчет по лабораторной работе №4

«Нахождение корней нелинейного уравнения»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-13Б |  | преподаватель каф. ИУ5 |
| Поляков Данила |  | Аксёнова М.В. |
| Подпись и дата |  | Подпись и дата: |

Москва, 2021 г.

**Постановка задачи**

1. Найти корень уравнения x – k \* cos(x) = 0 простой итерацией, половинным делением и методом Ньютона с погрешностью eps<0.000001 и для каждого из трех методов определить количество шагов алгоритма.

2. Выполнить п.1 для eps < 0.00000001.

3. Выполнить п.1 для уравнения x – 10cos(x) = 0 и объяснить результаты.

**Разработка алгоритма**

**Описание алгоритма**

Идея итерационных методов состоит в том, чтобы уравнение f(x) = 0 f ( x ) = 0 {\displaystyle f(x)=0} привести к эквивалентному уравнению x = ϕ(x). Выбирается некоторое начальное приближение Х0, с которого начинается вычисление последовательного приближения вида Хj+1 = ϕ(Xj). Число итераций данного цикла ограничено не только до eps, но и по количеству итераций, дабы она не зацикливалась.

**Метод простой итерации:**

В данном методе расчёт каждого следующего приближения происходит по формуле вида Хj+1 = Xj - f(Xj).

**Метод Ньютона:**

В данном методе расчёт каждого следующего приближения происходит по формуле вида Хj+1 = Xj - .

**Метод половинного деления:**

Этот метод для своей работы требует заранее заданных границ [xl, xr], в которых лежит ровно один корень уравнения. Суть его работы заключается в выборе X на середине интервала [xl, xr], и определении f(X). Если значение функции меньше заданной точности, то данный корень считается серединой интервала. Если такое условие не соблюдается, то интервал сдвигается и заменяется интервалом вида [xl, X] или [X, xr]. Выбирается следующий интервал по признаку разных знаков функции на его концах.

**Описание используемых переменных**

Для функций метода Ньютона и метода простых итераций используются следующие переменные:

int k – коэффициент перед cos(x)

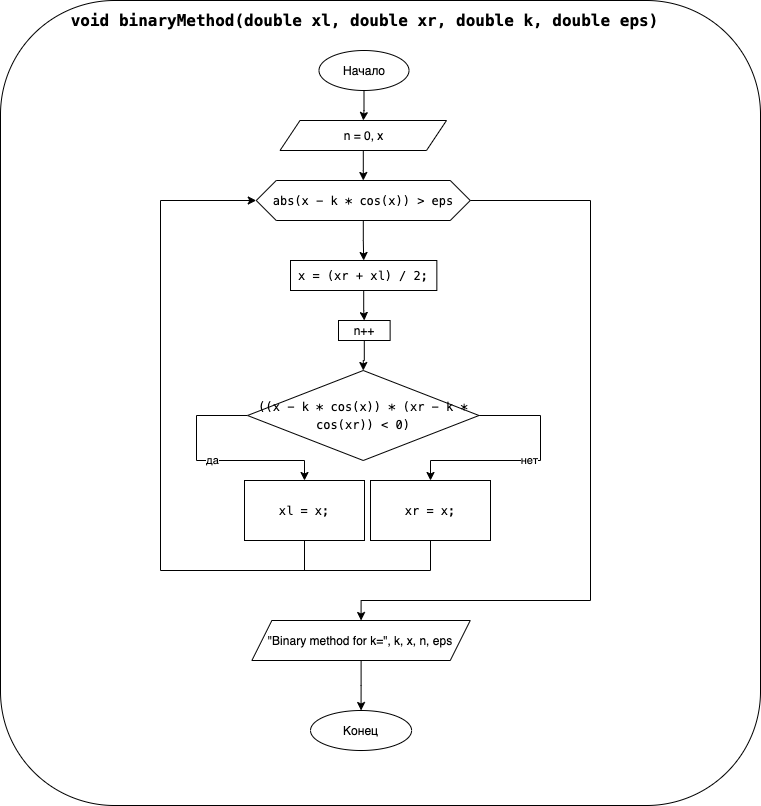
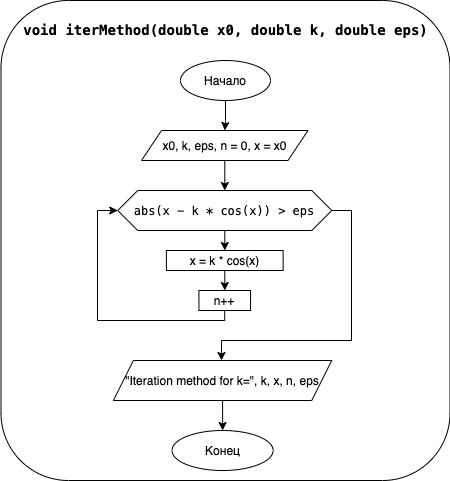
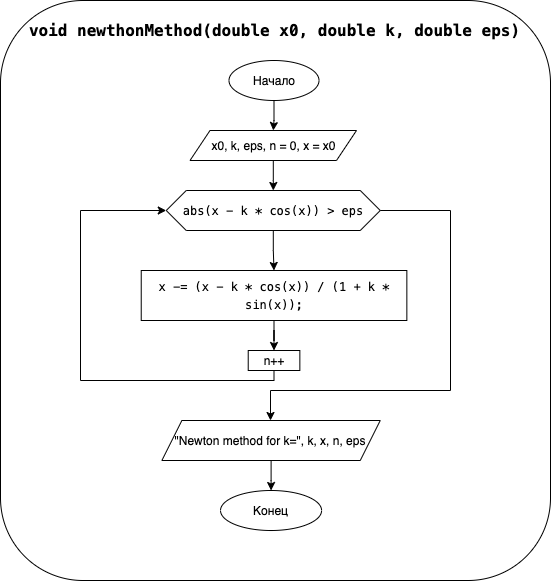
double x0 – начальное значение

double eps – точность

n – количество итераций

Для функции метода двоичного деления вместo x0 используются double xl,xr – начальные границы поиска корня.

**Схема алгоритма**

****

**Текст программы**

**Листинг кода программы:**

**functions.cpp**

#include <cmath>

#include <iostream>

#include <iomanip>

#include "functions.h"

void iterMethod(double x0, double k, double eps)

{

int n = 0;

double x = x0;

do

{

x = k \* cos(x);

n++;

} while (abs(x - k \* cos(x)) > eps);

std::cout << "Iteration method for k = " << k << " | x: " << std::setprecision(10) << x << " n:" << n << " eps = " << eps << std::endl;

}

void binaryMethod(double xl, double xr, double k, double eps)

{

int n = 0;

double x;

do

{

x = (xr + xl) / 2;

n++;

if ((x - k \* cos(x)) \* (xr - k \* cos(xr)) < 0)

{

xl = x;

}

else

{

xr = x;

}

} while (abs(x - k \* cos(x)) > eps);

std::cout << "Binary method for k = " << k << " | x: " << std::setprecision(10) << x << " n:" << n << " eps = " << eps << std::endl;

}

void newthonMethod(double x0, double k, double eps)

{

int n = 0;

double x = x0;

do

{

x -= (x - k \* cos(x)) / (1 + k \* sin(x));

n++;

} while (abs(x - k \* cos(x)) > eps);

std::cout << "Newthon method for k = " << k << " | x: " << std::setprecision(10) << x << " n:" << n << " eps = " << eps << std::endl;

}

void task1()

{

std::cout << "\t\tTASK №1\n";

iterMethod(0, k1, eps6);

binaryMethod(0, 1, k1, eps6);

newthonMethod(0, k1, eps6);

iterMethod(0, k1, eps8);

binaryMethod(0, 1, k1, eps8);

newthonMethod(0, k1, eps8);

}

void task2()

{

std::cout << "\t\tTASK №2\n";

iterMethod(0, k10, eps6);

binaryMethod(-5, 5, k10, eps6);

newthonMethod(0, k10, eps6);

binaryMethod(-5, 5, k10, eps8);

newthonMethod(0, k10, eps8);

}

#include <iostream>

#include <iomanip>

#include <cmath>

#include "functions.h"

**main.cpp**

int main()

{

task1();

task2();

int paragraph = 0;

double x0, k, eps, xr, xl;

std::cout << "\t\tMENU\n1. Simple iterations\n2. Binary method\n3. Newthon method\n";

std::cout << "\t\tEnter number of needed method:\n";

std::cin >> paragraph;

switch (paragraph)

{

case 1:

{

std::cout << "x0 k eps\n";

std::cin >> x0 >> k >> eps;

iterMethod(x0, k, eps);

}

break;

case 2:

{

std::cout << "xr xl k eps\n";

std::cin >> xr >> xl >> k >> eps;

binaryMethod(xr, xl, k, eps);

}

break;

case 3:

{

std::cout << "x0 k eps\n";

std::cin >> x0 >> k >> eps;

newthonMethod(x0, k, eps);

}

break;

default:

break;

}

return 0;

}

**Анализ результатов**