**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 ИУ5-11Б |  | преподаватель каф. ИУ5 |
| Курзанова Анастасия |  | Козлов А.Д. |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2021 г.

**Постановка задачи:**

1. Найдите корень уравнения

x - cos(x) = 0

простой итерацией, половинным делением и методом Ньютона с погрешностью eps <0.000001 и для каждого из трех методов определите количество шагов алгоритма.

2. Выполните п.1 для eps < 0.00000001.

3. Выполните п.1 для уравнения

x – 10cos(x) = 0

и объясните результаты.

**Разработка алгоритма:**

В программе 9 переменных:

int e - для удобства выбора точности;

double eps – точность значений;

int k – коэффициент перед косинусом;

char a – для повторного ввода значений с консоли;

double x1, x2 – переменные для нахождения корня, например, в методе половинного деления – это отрезок в котором нужно искать корень;

enter – середина отрезка, для метода половинного деления;

int n – значение, до которого нужно идти циклу;

int it – количество итераций;

int i – переменная счетчик в цикле.

Алгоритм получает на вход значение переменной k. Если оно равно 1, то алгоритм запросит пользователя ввести еще значение e, для нужной точности (1 для eps = 1e-6 и 2 для точности eps = 1e-8), а если k != 1, то точность всех значений будет 1е-6.

Далее значение двух переменных k, eps передаются трем функциям:

- Simple\_Iterations(eps, k);

- Newton(eps, k);

- Bisection(eps, k);

Каждая из них считает корень уравнения с определенной точностью и выводит результат на экран, а также количество итераций. В зависимости от значений переменных можно получить следующие результаты.

Пример:

Введите коэффициент перед косинусом

1

Выберите точность: 1) 1e-6 2) 1e-8

1

Метод простых итераций:

X = 0.739085

Итерации: 34

Метод Ньютона:

X = 0.739085

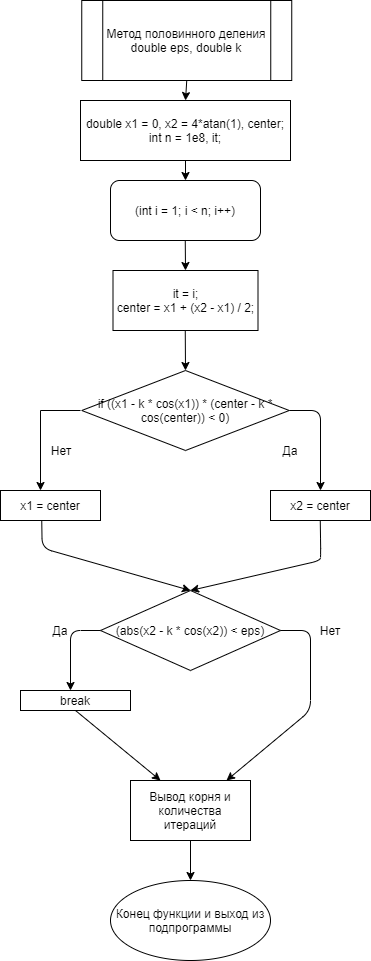
Итерации: 5

Метод половинного деления:

X = 0.739085

Итерации: 23

**Схема алгоритма функции метода половинного деления:**

****

**Текст программы:**

*Файл header.h*

#pragma once

#include <iostream>

#include <math.h>

#include <iomanip>

void Simple\_Iterations(double, double);

void Newton(double eps, double k);

void Bisection(double eps, double k);

*Файл source.cpp (Функции)*

#include "Header.h"

using namespace std;

void Simple\_Iterations (double eps, double k)

{

double x1 = 0, x2 = 0; int n = 1e8, it;

for (int i = 1; i < n; i++)

{

it = i;

x2 = k \* cos(x1);

if (abs(x2 - k \* cos(x2)) < eps) break;

x1 = x2;

}

cout << "Метод простых итераций:" << endl;

cout << "X = " << setprecision(log10(10 / eps)) << x2 << endl << "Итерации: " << it << endl << endl;

}

void Newton(double eps, double k)

{

double x1 = 0, x2 = 0; int n = 1e8, it;

for (int i = 1; i < n; i++)

{

it = i;

x2 = x1 - (x1 - k \* cos(x1)) / (1 + k \* sin(x1));

if (abs(x2 - x1) < eps)

break;

x1 = x2;

}

cout << "Метод Ньютона:" << endl;

cout << "X = " << setprecision(log10(1 / eps)) << x2 << endl << "Итерации: " << it << endl << endl;

}

void Bisection(double eps, double k)

{

double x1 = 0, x2 = 4\*atan(1), center; int n = 1e8, it;

for (int i = 1; i < n; i++)

{

it = i;

center = x1 + (x2 - x1) / 2;

if ((x1 - k \* cos(x1)) \* (center - k \* cos(center)) < 0)

{

x2 = center;

}

else

{

x1 = center;

}

if (abs(x2 - k \* cos(x2)) < eps) break;

}

cout << "Метод половинного деления:" << endl;

cout << "X = " << setprecision(log10(1 / eps)) << center << endl << "Итерации: " << it << endl << endl << endl;

}

*Файл main.cpp (Функция main)*

#include "Header.h"

using namespace std;

int main()

{

setlocale(LC\_ALL, "Russian");

int e; double eps = 1e-6, k;

char a = 0;

do

{

cout << "Введите коэффициент перед косинусом " << endl;

cin >> k;

if (k == 1)

{

cout << "Выберите точность: 1) 1e-6 2) 1e-8" << endl;

cin >> e;

if (e == 1) {

eps = 1e-6;

}

else {

if (e == 2) {

eps = 1e-8;

}

else {

cout << "Введенно некорректное значение";

}

}

}

else {

eps = 1e-6;

}

Simple\_Iterations(eps, k);

Newton(eps, k);

Bisection(eps, k);

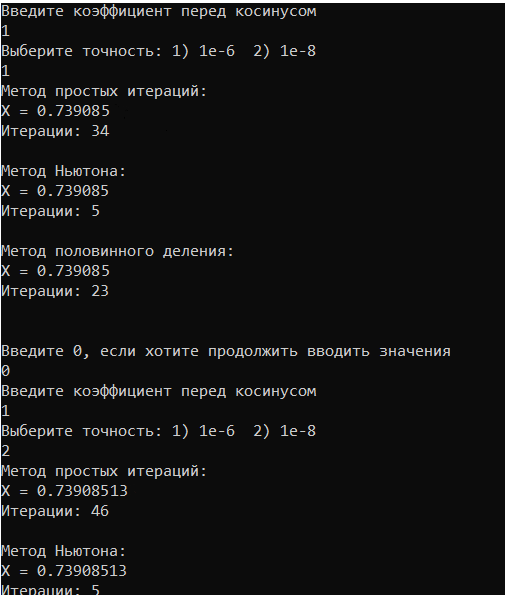
cout << "Введите 0, если хотите продолжить вводить значения" << endl;

cin >> a;

} while (a == '0');

}

**Результаты:**

****

