**Московский государственный технический**

**университет им. Н.Э. Баумана.**

Факультет «Информатика и управление»

Кафедра ИУ5. Курс «Основы программирования»

Отчет по лабораторной работе №4

«Нахождение корней нелинейного уравнения»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-12Б  Казицин Алексей |  | преподаватель каф. ИУ5  Козлов А.Д. |
|  |  |  |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2021 г.

Постановка задачи

**Цель работы**:

- программирование численных методов решения нелинейных уравнений;  
- сравнительный анализ методов простой итерации, половинного деления и метода Ньютона.

**Задание:**

1. Найти корень уравнения

x - 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

Заданное уравнение f(x) = 0 приводим к виду

x = ϕ(x).

Выбрав некоторое начальное приближение Х0, вычислим последовательные приближения

Хj+1 = ϕ(Xj), (j=0, 1, 2, …)

Где для метода итераций соответствует формула:

Хj+1 = Xj - f(Xj).

А для метода Ньютона:

Хj+1 = Xj - f(Xj)/ f ′(Xj).

**Метод половинного деления**

Для использования этого метода нужно задать границы интервала на оси абсцисс, содержащего ровно один корень [a, b] и требуемую точность вычислений.

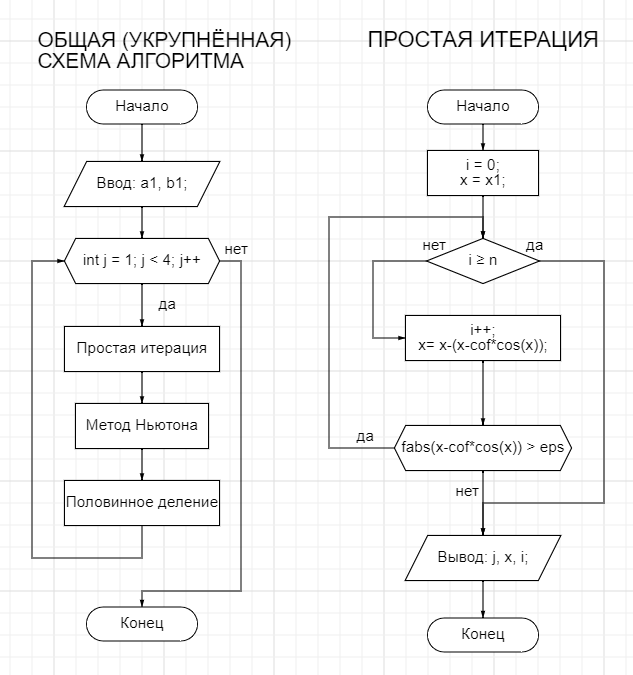
Выбираем Х на середине интервала [a, b] и определяем f(X). Если f(X) < eps, то середина интервала считается корнем функции, иначе корень ищется на том интервале из двух полученных, для которого значения функции на концах имеют разные знаки.

**Описание используемых переменных с указанием наименования, типа (int, float, и т.п.) и назначения в программе:**

double eps – точность; int i - число шагов; int cof - коэф. перед иксом; int n - макс число шагов; double x (x1) – корень уравнения; double a (a1), b (b1), c – начало, конец и середина интервала; int j – счетчик;

Блок-схема алгоритма

(Общая схема алгоритма и уточненная схема блока №1 «Простая итерация»)



Текст программы

#include <iostream>

#include <iomanip>

#include <cmath>

#include <windows.h>

using namespace std;

int main(){

SetConsoleCP(65001);

SetConsoleOutputCP(65001);

double eps = 0.000001; //точность

int i, cof = 1, n = 100000; // i - число шагов, cof - коэф. перед иксом, n - макс число шагов

double x, c, a, b, x1, a1, b1;

cout << "Введите начальный x для метода простых итераций. ";

cin >> x1;

cout <<"Введите границы интервала для метода половинного деления. ";

cin >> a1 >> b1;

while(a1 == b1){

cout <<"Ошибка (a должно быть не равно b). Введите границы интервала для метода половинного деления. ";

cin >> a1 >> b1;

}

for (int j = 1; j < 4; j++){

if (j == 2){ //изменение изначальных данных под нужные задачи (j - номер задачи)

eps = 0.00000001;

} else if (j == 3){

cof = 10;

eps = 0.000001;

}

cout <<setprecision(16) <<"Нахождение корней нелинейного уравнения y=x-" << cof << "\*cos(x)\n";

cout <<"Точность вычисления = " <<eps <<"\n";

if(b1 < a1){

b1= b1+a1;

a1= b1-a1;

b1= b1-a1;

}

a = a1;

b = b1;

cout << j <<") Простая итерация.\n";

i= 0;

x= x1;

do{

if (i >= n){

break;

}

i++;

x= x-(x-cof\*cos(x));

}while (fabs(x-cof\*cos(x)) > eps);

cout << j << ") Корень уравнения= " << x <<" вычислен за " << i <<" шагов.\n\n";

cout << j << ") Метод Ньютона.\n";

i= 0;

x= x1;

do{

if (i >= n){

break;

}

i++;

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

}while(fabs(x-cof\*cos(x)) > eps);

cout << j << ") Корень уравнения= " << x <<" вычислен за " << i << " шагов.\n\n";

cout << j << ") Половинное деление.\n";

i= 0;

do{

i++;

c= (a+b)/2;

if ((a-cof\*cos(a))\*(c-cof\*cos(c)) > 0) a= c;

else b=c;

}while(fabs(b-a) > eps);

x= (a+b)/2;

cout << j << ") Корень уравнения= " << x <<" вычислен за " << i <<" шагов.\n\n";

}

system("Pause");

return 0;

}

Анализ результатов

