**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 ИУ5-14Б |  | преподаватель каф. ИУ5 |
| Байдаков Владислав |  | Аксенова Мария Владимировна |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2023 г.

Постановка задачи

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

Метод итераций:

double x – вещественное число, аргумент уравнения

double eps – вещественное число, погрешность вычислений

int k – целое число, множитель перед cos(x) в уравнении

int maxt – целое число, максимальное количество итераций

int it - целое число, количество итераций

Метод Ньютона:

double x – вещественное число, аргумент уравнения

double eps – вещественное число, погрешность вычислений

int k – целое число, множитель перед cos(x) в уравнении

int maxt – целое число, максимальное количество итераций

int it - целое число, количество итераций

Метод половинного деления:

double x – вещественное число, аргумент уравнения

double eps – вещественное число, погрешность вычислений

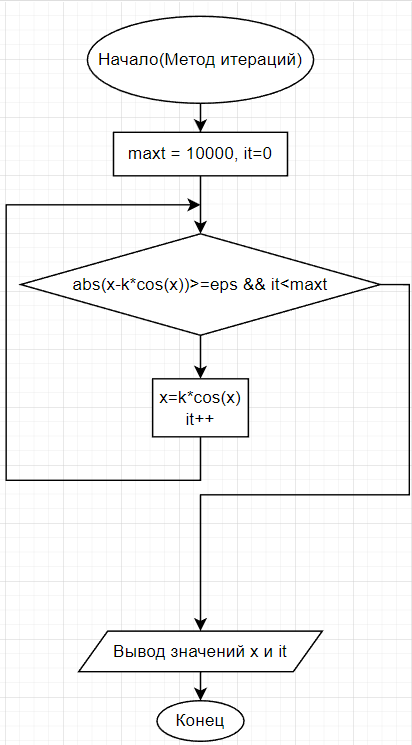
int k – целое число, множитель перед cos(x) в уравнении

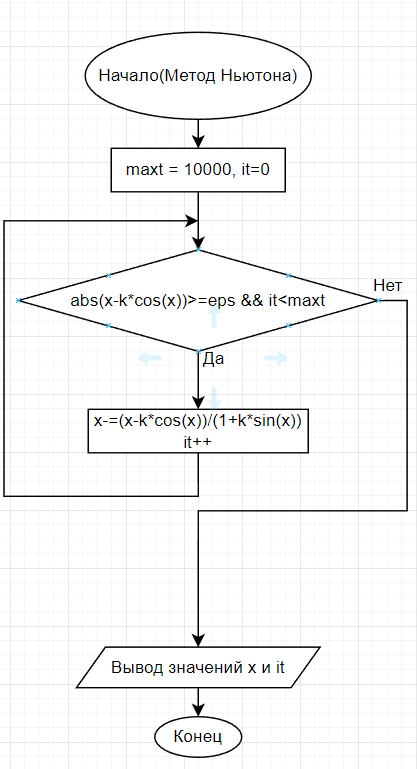
int maxt – целое число, максимальное количество итераций

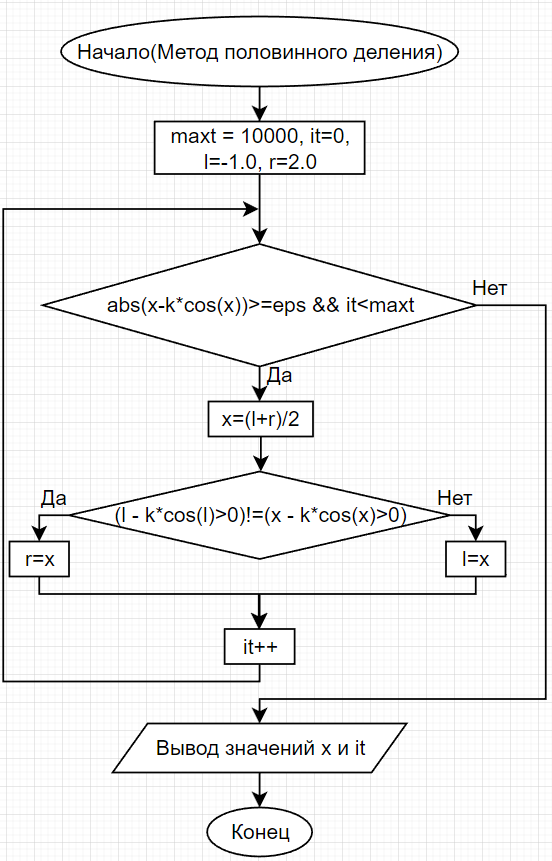
int it - целое число, количество итераций

double l, double r – вещественные числа, начальные значения границ в которых находится корень уравнения

Блок-схемы







Текст программы

void iter(double x,double eps,int k){

cout << "Метод итераций\n";

int maxt=10000, it = 0;

while(abs(x-k\*cos(x))>=eps && it<maxt){

x=k\*cos(x);

it++;

}

cout<<setprecision(7)<<x << " " << it << '\n';

}

void newt(double x,double eps,int k){

cout << "Метод Ньютона\n";

int it=0;

int maxt=10000;

while (( abs(x-k\*cos(x)) >= eps)&& (it<maxt))

{

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

it++;

}

cout<<setprecision(7)<<x << " " << it << '\n';

}

void dele(double x, double eps,int k){

cout << "Метод половинного деления\n";

double l=-1.0,r=2.0;

int maxt=10000;

int it=0;

while (abs(x- k\*cos(x))>eps &&(it<maxt)){

x=(l+r)/2;

if ((l - k\*cos(l)>0)!=(x - k\*cos(x)>0)){

r=x;

}

else

{

l=x;

}

it++;

}

cout<<setprecision(7)<<x << " " << it << '\n';

}

Анализ результатов

