**Московский государственный технический**

**университет им. Н.Э. Баумана.**

Факультет «Информатика и управление»

Кафедра «Системы обработки информации и управления»

Курс «Основы информатики»

Отчет по лабораторной работе №4

«Численные методы решения нелинейных уравнений»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-15Б |  | преподаватель каф. ИУ5 |
| Коновалов Илья |  | Папшев И.С. |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2021 г.

**Задача:**

Разработать программу для вычисления корней уравнения x – k\*cos(x) = 0 простой итерацией, половинным делением и методом Ньютона с погрешностью eps < 0.000001 и eps < 0.00000001. Для каждого из трех методов определить (и вывести на экран) количество шагов алгоритма, использованных для получения результата.

Разработать «универсальную» функцию для вычисления корней уравнений вида *y(x)=0* методом половинного деления, используя в качестве параметра указатель на функцию. Применить эту функцию для нахождения всех корней уравнения

*y=x-5\*cos(x)* и *y=x-10\*cos(x),*  добавив следующие параметры:

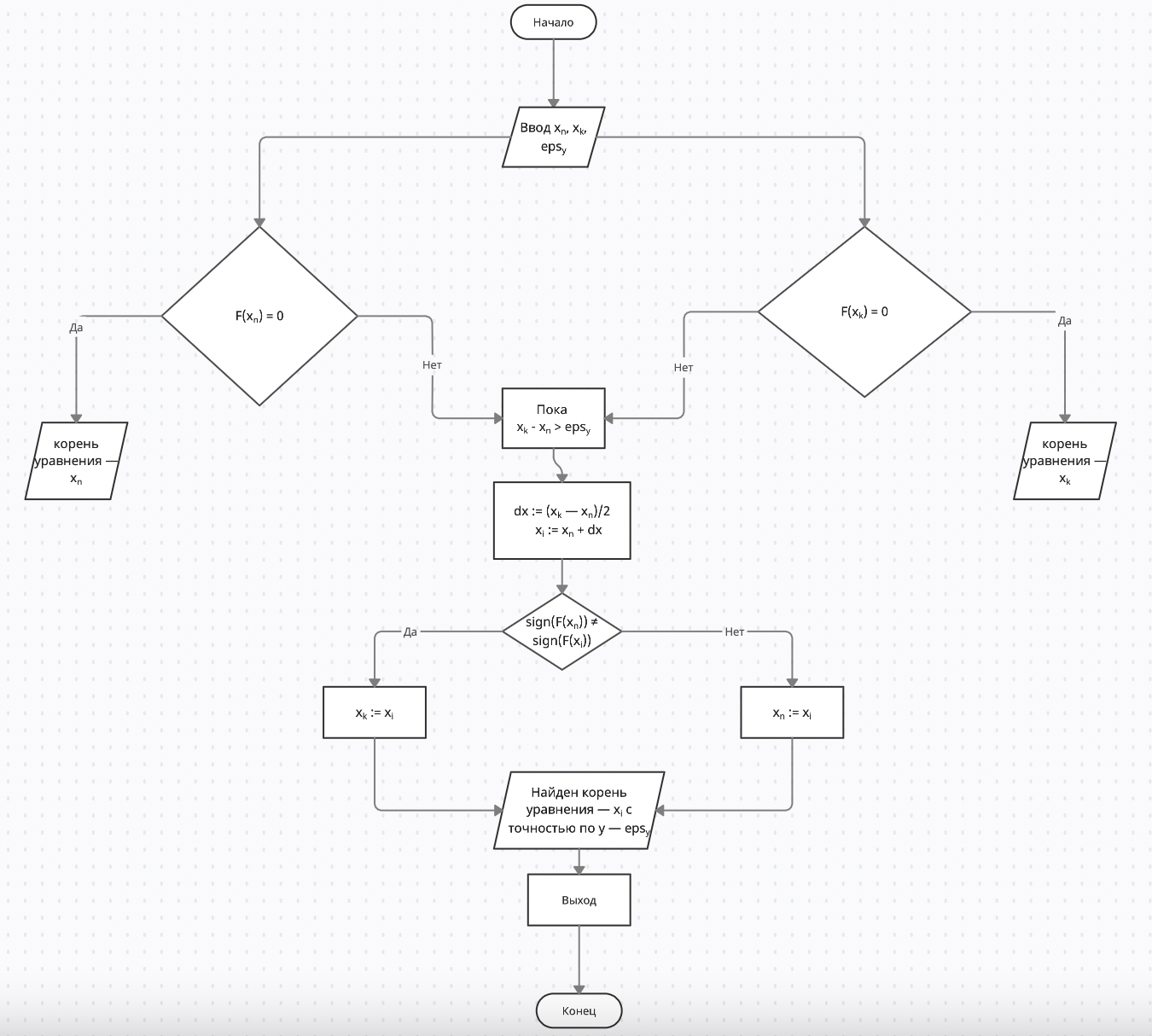
eps - требуемая точность вычислений

xl и xr - левая и правая границы интервала, содержащего ровно один корень.

**Алгоритм:**

Выбирают Х на середине интервала [xl, xr] и определяют f(X). Если |f(X)| < eps, то середина интервала считается корнем уравнения, иначе корень ищется на том интервале из двух полученных, для которого значения функции на концах имеют разные знаки. Действия повторяются до тех пор, пока интервал > eps. Перед входом в цикл необходимо

проверить, не являются ли границы интервала корнями уравнения f(X)=0.



**Код программы:**

#define \_MATH\_DEFINES  
#include **<iostream>**#include **<iomanip>**#include **<cmath>**#include **<vector>  
  
using namespace** std;  
**double** eps = 1e-8;  
  
**double** AnyFunc(**double** x) {  
 **return** (sin(x) - cos(x));  
}  
  
**double** cos5(**double** x) {  
 **return** (x - 5 \* cos(x));  
}  
  
**double** cos10(**double** x) {  
 **return** (x - 10 \* cos(x));  
}  
  
**int** menu() {  
 cout << **"\n================ЛР №4===================\n"**;  
 cout << **"\t1. y = x - 5cos(x)\n"**;  
 cout << **"\t2. y = x - 10cos(x)\n"**;  
 cout << **"\t3. y = f(x)\n"**;  
 cout << **"\t4. Выход\n"**;  
 cout << **"========================================\n"**;  
 **int** choice;  
 cout << **"Выберите действие: "**;  
 cin >> choice;  
 **while** (cin.fail()) {  
 cout << **"Ошибка ввода. Повторите ввод\n"**;  
 cin.clear();  
 cin.ignore(10, **'\n'**);  
 cin >> choice;  
 }  
 **return** choice;  
}  
  
vector < vector <**double**> > HalfDiv(**double** (\*fPtr)(**double**), **double** xl, **double** xr,  
 **double** step, vector<**double**>\* roots, vector<**double**>\* count) {  
 **double** xi = 0;  
 vector<**double**>rts = \*roots;  
 vector<**double**>vCount = \*count;  
 **int** cnt, RootCnter = 0;  
  
 cout << **"\nВведите значения xn, xk"** << endl;  
 cout << **"\nxn = "**; cin >> xl; cout << **"xk = "**; cin >> xr;  
  
 **for** (**double** i = xl; i <= xr; i += step) {  
 **double** xl = i;  
 **double** xr = i + step;  
 **if** ((fPtr(xl) \* fPtr(xr)) <= 0) {  
 cnt = 0;  
 **while** ((xr - xl) / 2 > eps) {  
 cnt++;  
 xi = (xl + xr) / 2;  
 **if** ((fPtr(xl) \* fPtr(xi)) > 0) xl = xi;  
 **else** xr = xi;  
 }  
 **if** (abs(xi - eps) < eps) { xi = 0; }  
 rts.push\_back(xi);  
 vCount.push\_back(cnt);  
 RootCnter++;  
 }  
 }  
 vector < vector <**double**> > cmb(RootCnter, vector <**double**>(2));  
  
 **for** (**int** i = 0; i < RootCnter; i++) {cmb[i][0] = rts[i];}  
 **for** (**int** j = 0; j < RootCnter; j++) {cmb[j][1] = vCount[j];}  
  
 **return** cmb;  
}  
  
**void** print(**double** a, **double** b, vector < vector <**double**> > d) {  
 **for** (**int** i = 1; i <= a; i++) {  
 cout << **'\n'** << fixed << setprecision(0) << i << **") "** << setprecision(b) << d[i - 1][0] << **'\n'**;  
 }  
 cout << endl << endl;  
}  
  
**int** main() {  
 **double** xl, xr, c, step, size, x;  
  
 xl = xr = 0;  
  
 vector <**double**> roots;  
 vector <**double**> count;  
 vector < vector <**double**> > finals;  
  
 **double**(\*F)(**double**);  
  
 c = abs(log10(eps)) + 1;  
 step = 0.1;  
  
 **while** (**true**) {  
 **switch** (menu()) {  
 **case** 1: *// y = 5\*cos(x)* F = cos5;  
 finals = HalfDiv(F, xl, xr, step, &roots, &count);  
 size = finals.size();  
 print(size, c, finals);  
 *//print(HalfDiv(F, k, xl, xr, step, &roots, &count).size(), c, HalfDiv(F, k, xl, xr, step, &roots, &count));* **break**;  
  
 **case** 2: *// y = 10\*cos(x)* F = cos10;  
 finals = HalfDiv(F, xl, xr, step, &roots, &count);  
 size = finals.size();  
 print(size, c, finals);  
 **break**;  
  
 **case** 3: *// y = f(x)* F = &AnyFunc;  
 finals = HalfDiv (cos, xl, xr, step, &roots, &count);  
 size = finals.size();  
 print(size, c, finals);  
 **break**;  
  
 **case** 4:  
 **return** 0;  
 **break**;  
  
 **default**:  
 cout << **"\nНедопустимое действие. Повторите выбор.\n"**;  
 **break**;  
 }  
 }  
 **return** 0;  
}

**Результаты:**

|  |  |  |  |
| --- | --- | --- | --- |
| Функция | Левая граница | Правая граница | Результат |
| y = x - 5cos(x) | 1 | 3 | 1.306440008 |
| y = x - 5cos(x) | -5 | 5 | -3.837467110  -1.977383029  1.306440008 |
| y = x - 10cos(x) | -5 | 3 | -4.271095335  -1.746329272  1.427551782 |
| y = x - 10cos(x) | 0 | 7 | 1.427551782  5.267116439  7.068891227 |
| y = sin(x) – cos(x) | 0 | 4 | 0.785398161  3.926990807 |
| y = sin(x) – cos(x) | -10 | -5 | -8.639379799  -5.497787154 |
| y = tan(x) + cos(x)/2 | 0 | 7 | 1.570796335  3.568671238  4.712388980  5.856106722 |
| y = tan(x) + cos(x)/2 | -2 | 4 | -1.570796335  -0.427078593  1.570796335  3.568671238 |

**Вывод:**

В ходе выполнение данной лабораторной работы научился программировать численные методы решения нелинейных уравнений, приобрел навыки сравнительного анализа методов простой итерации, половинного деления и метода Ньютона, научился использовать функции и указатели на функцию.