**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Московский Авиационный Институт»**

**(Национальный Исследовательский Университет)**

**Институт №8 «Информационные технологии и прикладная математика»**

Курсовой проект

по курсу «Вычислительные системы»

1 семестр

Задание 4

Процедуры и функции в качестве параметров

**Автор работы:**

студент 1 курса, гр. М8О-105Б-21

Номер по списку 23

Титеев Р.М.

**Руководитель проекта:**

Титов В.К.

**Дата сдачи:**

17.12.2021

Москва, 2021

**Задача**

Составить программу на языке Си с процедурами решения трансцендентных алгебраических уравнений методами итераций, Ньютона и дихотомии. Нелинейные уравнения оформить как параметры-функции, разрешив относительно неизвестной величины в случае необходимости.

**Вариант 23**



**Алгоритм**

Для начала нужно создать функции, которые будут возвращать значение функций и их производных из данных вариантов в точке x. Также необходимо создать функции, которые получают на вход функцию, и границы отрезка, после чего решают уравнение.

После этого результаты выводятся на экран в виде таблицы.

**Описание программы**

Программа написана на языке Си в соответствии со стандартом ISO/IEC 9899:1999. При написании использовался компилятор G++.

В программе использованы следующие функции:

|  |  |  |
| --- | --- | --- |
| Имя функции | Тип | Назначение |
| function\_1 | double | Функция из варианта 23 |
| function\_2 | Свой вариант №1 |
| function\_3 | Свой вариант №2 |
| Function\_1 | Функция из в.23 вида x = f(x) |
| Function\_2 | Функция из своего в. №1 вида x = f(x) |
| Function\_3 | Функция из своего в. №2 вида x = f(x) |
| derivative | Производная функции от точки x |
| dichotomy | Решает уравнение методом дихотомии |
| iteration | Решает уравнение методом итераций |
| newton | Решает уравнение методом Ньютона |
| chord | Решает уравнение методом Хорд |

**Протокол**

(base) ramil@ramil:~/labs and curs/curs 4$ cat Kp.cpp

#include <stdlib.h>

#include <stdio.h>

#include <math.h>

#define eps 0.00000001

double function\_1(double x){

return 3\*x - 4\*log(x) - 5;

}

double Function\_1(double x){

return (4\*log(x) + 5)/3;

}

double function\_2(double x){

return x\*x\*x\*x + 3\*x + sin(x);

}

double Function\_2(double x){

return (x\*x\*x\*x + sin(x))/(-3);

}

double function\_3(double x){

return 3\*log(1/x)-cos(x)+5\*x-13;

}

double Function\_3(double x){

return (cos(x)+13-3\*log(1/x))/5;

}

double derivative(double f\_eps, double f){

return (f\_eps - f)/eps;

}

double dichotomy(double function(double), double left, double right){

double result;

while(fabs(left - right) > eps){

result = (right + left) / 2;

if (function(left) \* function(result) > 0) left = result;

else right = result;

}

return result;

}

double iteration(double Function(double), double left, double right){

double x = (left + right) / 2;

double previousX = 0;

while(fabs(x - previousX) > eps){

previousX = x;

x = Function(x);

}

return x;

}

double newton(double function(double), double left, double right){

double x = (left + right) / 2;

double previousX = 0;

while(fabs(x - previousX) > eps){

previousX = x;

x = x - function(x) / derivative(function(x+eps), function(x));

}

return x;

}

double chord(double f(double), double a, double b) {

double prevX = b, ya = f(a), yb = f(b);

double x = (ya\*b-yb\*a)/(ya-yb);

while(fabs(prevX - x) > eps) {

if(ya\*f(x) > 0) a = x;

else b = x;

ya = f(a), yb = f(b);

prevX = x;

x = (ya\*b-yb\*a)/(ya-yb);

}

return x;

}

int main() {

printf("------------------------------------------------------------------------------\n");

printf("| Уравнение | Отрезок | Метод | Результат |\n");

printf("------------------------------------------------------------------------------\n");

printf("| 3 \* x - 4 \* log(x) - 5 = 0 | [2;4] | Дихотомии | %.16f |\n", dichotomy(function\_1, 2, 4));

printf("| | | Итераций | %.16f |\n", iteration(Function\_1, 2, 4));

printf("| | | Ньютона | %.16f |\n", newton(function\_1, 2, 4));

printf("| | | Хорд | %.16f |\n", chord(function\_1, 2, 4));

printf("------------------------------------------------------------------------------\n");

printf("| x^4 + 3 \* x +sin(x) = 0 | [-1;1] | Дихотомии | %.16f |\n", dichotomy(function\_2, -1, 1));

printf("| | | Итераций | %.16f |\n", iteration(Function\_2, -1, 1));

printf("| | | Ньютона | %.16f |\n", newton(function\_2, -1, 1));

printf("| | | Хорд | %.16f |\n", chord(function\_2, -1, 1));

printf("------------------------------------------------------------------------------\n");

printf("| 3\*log(1/x)-cos(x)+5\*x-13 = 0 | [3;4] | Дихотомии | %.16f |\n", dichotomy(function\_3, 3,4));

printf("| | | Итераций | %.16f |\n", iteration(Function\_3, 3,4));

printf("| | | Ньютона | %.16f |\n", newton(function\_3, 3,4));

printf("| | | Хорд | %.16f |\n", chord(function\_3, 3,4));

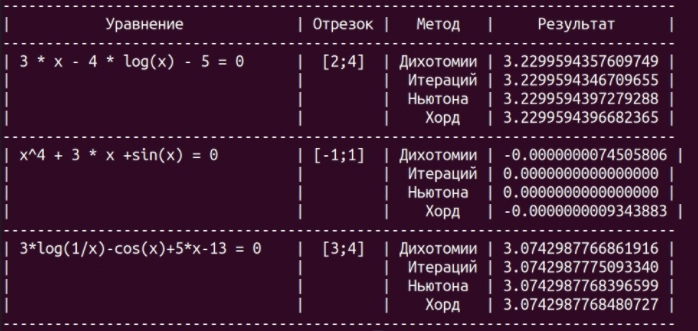
printf("------------------------------------------------------------------------------\n");

return 0;

}

(base) ramil@ramil:~/labs and curs/curs 4$ g++ Kp.cpp

(base) ramil@ramil:~/labs and curs/curs 4$ ./a.out



**Вывод**

Я научился составлять программы на языке СИ, решающие уравнение методом итераций, дихотомии и Ньютона.