**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-101Б-21

Филимонов Н.Н.

**Руководитель проекта:**

Титов В. К.

**Дата сдачи:**

Москва, 2021

**СОДЕРЖАНИЕ**

1. Задача………………………….……………………………………………3
2. Вариант……………………………………………………………………..3
3. Алгоритм…………………………………………………………...………3
4. Описание программы……………………………………………………...3
5. Протокол……………………………………………………………………3
6. Вывод……………………………………………………………………….5

**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**

****

sqrt(2-x)-tan(x)=0 [0, 1] 0.825554051914

ln(2x)-cosx=0 [0,1] 0.917904899

**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 GCC.

В программе использованы следующие функции:

|  |  |  |
| --- | --- | --- |
| Имя функции | Тип | Назначение |
| function\_1 | double | Функция из варианта 5 |
| function\_2 | Функция из варианта 6 |
| derivative\_1 | Производная функции из в. 5 |
| derivative\_2 | Производная функции из в. 6 |
| dichotomy | Решает уравнение методом дихотомии |
| iteration | Решает уравнение методом итераций ( нужна проверка на сходимость ) |
| chord | Решает уравнение методом хорд |
| newton | Решает уравнение методом Ньютона |

**Протокол**

#include <stdio.h>

#include <math.h>

#include <locale.h>

const double eps = 0.00001;

double f1(double x) { return 3\*x - 4\*log(x) - 5; }

double F1(double x) { return (5+4\*log(x))/3; }

double Fp1(double x) { return 3 - 4./x; }

double f2(double x) { return sqrtf(2-x) - tan(x); }

double F2(double x) { return atan(sqrt(2-x)); }

double Fp2(double x){ return -1. / cos(x) / cos(x) - 0.5/sqrt(2-x); }

double f3(double x) { return log(2\*x)-cos(x); }

double F3(double x) { return acos(log(2\*x));}

double Fp3(double x) { return 1./x + sin(x); }

double dichotomy(double f(double), double a, double b)

{

double prevX = b, x = (a + b) / 2.;

while (fabs(prevX - x) > eps)

{

if (f(x) \* f(a) > 0)

a = x;

else

b = x;

prevX = x;

x = (a + b) / 2.;

}

return x;

}

double iteration(double f(double), double a, double b)

{

double prevX = b + a \* 0.1, x = f(prevX);

while (fabs(x - prevX) > eps)

{

prevX = x;

x = f(x);

}

return x;

}

double tangent(double f(double), double F(double), double a, double b)

{

double prevX = a + b / 2., x = prevX - f(prevX) / F(prevX);

while (fabs(prevX - x) > eps)

{

prevX = x;

x = prevX - f(prevX) / F(prevX);

}

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("| \t 1 | [2;4] | Дихотомии | %.4f |\n", dichotomy(f1, 2, 4));

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

printf("| \t 1 | [2;4] | Ньютона | %.4f |\n", tangent(f1, Fp1, 2, 4));

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

printf("| \t 1 | [2;4] | Итераций | %.4f |\n", iteration(F1, 2, 4));

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

printf("| \t 1 | [2;4] | Хорд | %.4f |\n", chord(f1, 2, 4));

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

printf("| \t 2 | [0;1] | Дихотомии | %.4f |\n", dichotomy(f2, 0, 1));

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

printf("| \t 2 | [0;1] | Ньютона | %.4f |\n", tangent(f2, Fp2, 0, 1));

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

printf("| \t 2 | [0;1] | Итераций | %.4f |\n", iteration(F2, 0, 1));

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

printf("| \t 2 | [0;1] | Хорд | %.4f |\n", chord(f2, 0, 1));

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

printf("| \t 3 | [0;1] | Дихотомии | %.4f |\n", dichotomy(f3, 0, 1));

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

printf("| \t 3 | [0;1] | Ньютона | %.4f |\n", tangent(f3, Fp3, 0, 1));

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

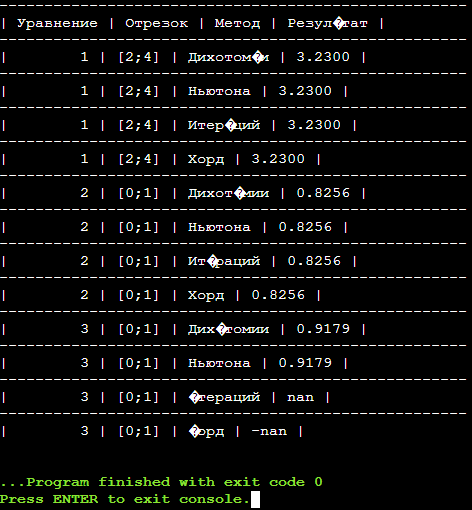
printf("| \t 3 | [0;1] | Итераций | %.4f |\n", iteration(F3, 0, 1));

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

printf("| \t 3 | [0;1] | Хорд | %.4f |\n", chord(f3, 0, 1));

return 0;

}

****

**Вывод**

Я научился составлять программы на языке СИ, решающие уравнение методом итераций, дихотомии, Ньютона и хорд. Полученные значения являются корнями уравнений на заданных отрезках.