Московский Авиационный институт

(Национальный исследовательский университет)

**Курсовой проект**

**по курсам**

**“Архитектура компьютера”, “Программные и аппаратные средства информатики”**

**1 семестр**

**Задание 4**

Студент: Белоносов К.А.

Группа: М8О-103Б-21

Руководитель: Севастьянов В. С.

Оценка:

Дата:

Подпись:

**Содержание**

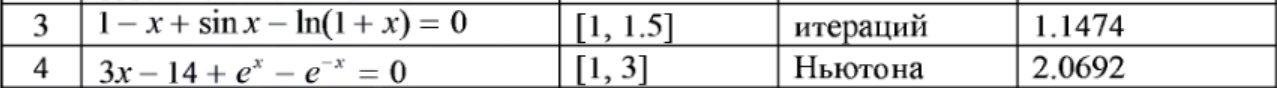
1. Введение 3
2. Вариант 3
3. Программное обеспечение 3
4. Описание работы, алгоритм 4
5. Описание переменных 4
6. Проверка программы 5
7. Вывод 6
8. Исходный код 6

**Введение**

Составить программу на Си с процедурами решения трансцендентных алгебраических уравнений различными численными методами (итераций, Ньютона и половинного деления – дихотомии). Нелинейные уравнения оформить как параметры-функции, разрешив относительно неизвестной величины в случае необходимости. Применить каждую процедуру к решению двух уравнений, заданных двумя строками таблицы, начиная с варианта с заданным номером. Если метод неприменим, дать математическое обоснование и графическую иллюстрацию, например, с использованием gnuplot.

**Вариант**

Вариант 3-4



**Программное обеспечение**

Операционная система семейства: linux, наименование: ubuntu, версия 20.04

Интерпретатор команд: bash версия 5.0.17(1)

Система программирования VS Code, редактор текстов emacs версия 25.2.2

**Описание работы, алгоритм**

Я создал несколько функций, которые служат для работы основных функций вычисления корня уравнений. Всего 3 типа функций – первый возвращает значение исходной функции по заданному x, второй – то же самое, но для производной исходной функции, а третий – для функции типа x = f(x). Они выступают в качестве аргументов трех основных функций, вычисляющих ответ методом дихотомии, итераций и Ньютона. Пользователь вводит значение эпсилона, которое используется для регулировки точности ответа. Далее программа вычисляет ответ и выводит его в виде таблицы.

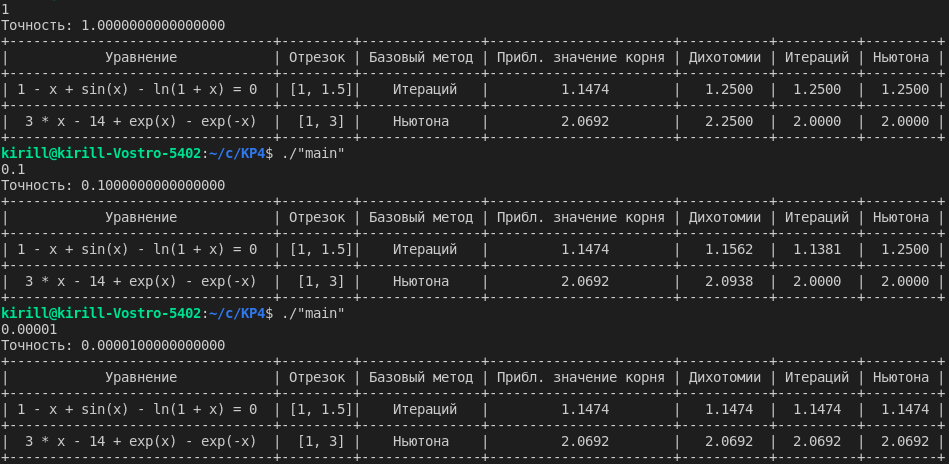
**Описание переменных**

|  |  |  |
| --- | --- | --- |
| **Переменная** | **Тип** | **Назначение** |
| eps | double | Эпсилон |
| a | Левая граница отрезка |
| b | Правая граница отрезка |
| x | Хранение решений функций |
| buf | Промежуточное хранение вычислений |

**Проверка программы**

Входные данные – эпсилон. В тестах eps = 1; 0.1; 0,00001.

Выходные данные:



**Вывод**

В ходе работы я написал и протестировал программу на языке Си, которая вычисляет ответ тремя методами – дихотомии, итераций и Ньютона. Можно сделать вывод, что приведенные алгоритмы позволяют довольно точно вычислять значения и различия в ответе уменьшаются при уменьшении эпсилон

**Исходный код**

#include <stdio.h>

#include <math.h>

double f1(double x) {

return 1 - x + sin(x) - log(1 + x);

}

double f2(double x) {

return 3 \* x - 14 + exp(x) - exp(-1 \* x);

}

double f1\_d(double x) {

return -1 + cos(x) - 1 / (1 + x);

}

double f2\_d(double x) {

return 3 + exp(x) + exp(-1 \* x);

}

double f1\_v(double x) {

return 1 + sin(x) - log(1 + x);

}

double f2\_v(double x) {

return log(14 - 3 \* x + exp(-1 \* x));

}

double dichotomy(double (\*f)(double x), double eps, double a, double b) {

double buf;

while (!(fabs(a - b) < eps)) {

if (f(a) \* f((a + b) / 2) > 0) {

a = (a + b) / 2;

} else {

b = (a + b) / 2;

}

}

return (a + b) / 2;

}

double iter(double (\*f)(double x), double eps, double a, double b) {

double x = (a + b) / 2;

double x\_n = f(x);

while (!(fabs(x\_n - x) < eps)) {

x = f(x);

x\_n = f(x);

}

return x;

}

double newton(double (\*f)(double x), double (\*f\_d)(double x), double eps, double a, double b) {

double x = (a + b) / 2;

while (!(fabs(f(x) / f\_d(x)) < eps)) {

x = x - f(x) / f\_d(x);

}

return x;

}

int main(void) {

double eps;

scanf("%lf", &eps);

printf("Точность: %.16f\n", eps);

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

printf("| Уравнение | Отрезок | Базовый метод | Прибл. значение корня | Дихотомии | Итераций | Ньютона |\n");

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

printf("| 1 - x + sin(x) - ln(1 + x) = 0 | [1, 1.5]| Итераций | 1.1474 | %.4f | %.4f | %.4f |\n", dichotomy(f1, eps, 1, 1.5), iter(f1\_v, eps, 1, 1.5),newton(f1, f1\_d, eps, 1, 1.5));

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

printf("| 3 \* x - 14 + exp(x) - exp(-x) | [1, 3] | Ньютона | 2.0692 | %.4f | %.4f | %.4f |\n", dichotomy(f2, eps, 1, 3), iter(f2\_v, eps, 1, 3), newton(f2, f2\_d, eps, 1, 3));

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

return 0;

}