**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Московский Авиационный Институт»**

**(Национальный Исследовательский Университет)**

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

# **Кафедра 806 «Прикладная математика и информатика»**

Курсовой проект

по курсу «Вычислительные системы»

1 семестр

Задание 4

**Автор работы:**

студент 1 курса, группа М8О-102Б-21

Богуж Владислав Андреевич

**Руководитель проекта:**

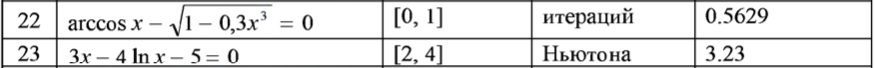
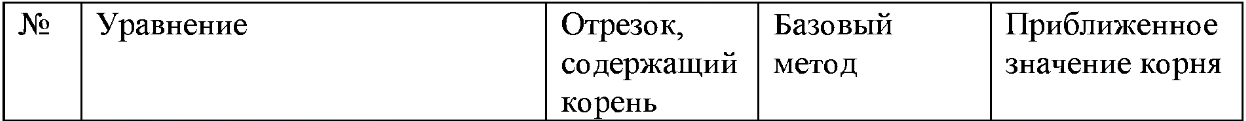
Доцент кафедры 806 Никулин С.П.

**Дата сдачи:**

**Цель работы**

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 №22**



**Алгоритм**

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

После этого результаты выводятся на экран в виде таблицы.

**Описание программы**

Входные данные: отсутствуют.

Выходные данные: значение машинного эпсилон, таблица с 2 функциями, соответствующие им отрезки и результат вычисления этих функций 3 способами - методами итераций, Ньютона и дихотомии.

**Программа:**

/\* Богуж Владислав Андреевич, М8О-102Б-21

КП 4. Вариант 22. \*/

#include <stdio.h>

#include <math.h>

double function\_1(double x) {

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

}

double function\_2(double x) {

return acos(x) - sqrt(1 - 0.3 \* x \* x \* x);

}

double function\_1\_2(double x) {

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

}

double function\_2\_2(double x) {

return cos(sqrt(1 - 0.3 \* x \* x \* x));

}

double derivative\_1(double x) {

return 3 - (4 / x);

}

double derivative\_2(double x) {

return (-1) / sqrt(1 - x \* x) + (9 \* x \* x) / (20 \* sqrt(1 - (3 \* x \* x \* x) / 10));

}

double dichotomy(double function(double), double l, double r, double eps) {

double res;

while (fabs(l - r) > eps) {

res = (r + l) / 2;

if (function(l) \* function(res) > 0) {

l = res;

} else {

r = res;

}

}

return res;

}

double iteration(double function(double), double l, double r, double eps) {

double x = (l + r) / 2;

double previousX = 0;

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

previousX = x;

x = function(x);

}

return x;

}

double newton(double function(double), double derivative(double), double l, double r, double eps) {

double x = (l + r) / 2;

double previousX = 0;

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

previousX = x;

x = x - function(x) / derivative(x);

}

return x;

}

int main() {

double eps = 1;

while (1 + eps / 2 > 1) {

eps /= 2;

}

printf("Машинное эпсилон = %.20f\n\n", eps);

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

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

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

printf("| 3 \* x - 4 \* log(x) - 5 = 0 | [2,4] | дихотомия | %.17f |\n",

dichotomy(function\_1, 2, 4, eps));

printf("| | | итерации | %.17f |\n",

iteration(function\_1\_2, 2, 4, eps));

printf("| | | ньютон | %.17f |\n",

newton(function\_1, derivative\_1, 2, 4, eps));

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

printf("| acos(x) - sqrt(1 - 0,3\*x^3) = 0 | [0,1] | дихотомия | %.17f |\n",

dichotomy(function\_2, 0, 1, eps));

printf("| | | итерации | %.17f |\n",

iteration(function\_2\_2, 0, 1, eps));

printf("| | | ньютон | %.17f |\n",

newton(function\_2, derivative\_2, 0, 1, eps));

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

return 0;

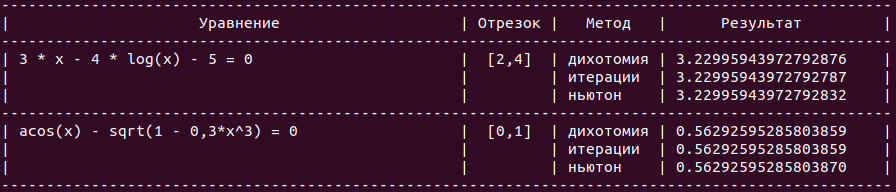
}

**Результат:**

vlad@vlad:~/Documents$ gcc kp4\_var22.c -lm

vlad@vlad:~/Documents$ ./a.out

Машинное эпсилон = 0.00000000000000022204



**Заключение:**

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