

**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-102Б-21
Кажекин Д.А.

Руководитель проекта:
Никулин С.П.

Дата сдачи:

Москва, 2021

СОДЕРЖАНИЕ

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

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

№	Уравнение	Отрезок, содержащий корень	Базовый метод	Приближенное значение корня
16	$3 \sin \sqrt{x} + 0.35x - 3.8 = 0$	[2, 3]	итераций	2.2985
17	$0,25x^3 + x - 1,2502 = 0$	[0, 2]	Ньютона	1.0001

Алгоритм

Опишем функции, отвечающие за искомые уравнения 16 и 17, функции, отвечающие за производные искомых уравнений. После этого пишем функции вывода решения методами Ньютона, итераций и дихотомии

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:

- ☐ function_1, function_2 – уравнения 16 и 17 соответственно.
- ☐ function_1_2, function_2_2 – уравнения вида $x = f(x)$;
- ☐ proizv_1, proizv_2 – производные первого порядка уравнений 16 и 17 соответственно
- ☐ dichotomy, iteration, newton – вывод решения методами дихотомии, итераций и Ньютона соответственно.

Протокол

```
#include <stdio.h>

#include <math.h>

#include <stdlib.h>

double function_1(double x) {
    return (3.0 * sin(sqrt(x)) + 0.35 * x - 3.8);
```

```

}

double function_2(double x) {
    return (0.25 * (x * x * x) + x - 1.2502);
}

double function_1_2(double x) {
    return ((3.8 - 3.0 * sin(sqrt(x)))/0.35);
}

double function_2_2(double x) {
    return (1.2502 - 0.25 * (x * x * x));
}

double proizv_1(double x) {
    return (((3.0 * cos(sqrt(x)))/(2.0 * sqrt(x))) + 0.35);
}

double proizv_2(double x) {
    return (0.75 * (x * x) + 1.0);
}

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 = 0.0000000000000001;

    printf("Машинное эпислон = %.15f\n\n", eps);
    printf("-----\n");
    printf("|          Уравнение          | Отрезок | Метод | Результат \n");
    printf("-----\n");
    printf("| 3.0 * sin(sqrt(x)) + 0.35 * x - 3.8      | [2.0,3.0] | дихотомия | %.17f  \n",
dichotomy(function_1,2.0,3.0,eps));
    printf("|          |          | итерации | %.17f  \n", iteration(function_1_2,2.0,3.0,eps));
    printf("|          |          | ньютон  | %.17f  \n", newton(function_1, proziv_1,2.0,3.0,eps));
    printf("-----\n");
    printf("| 0.25 * (x * x * x) + x - 1.2502          | [0,2.0] | дихотомия | %.17f  \n",
dichotomy(function_2,0,2.0,eps));
    printf("|          |          | итерации | %.17f  \n", iteration(function_2_2,0,2.0,eps));
    printf("|          |          | ньютон  | %.17f  \n", newton(function_2, proziv_2,0,2.0,eps));
    printf("-----\n");
    return 0;
}

```

Уравнение	Отрезок	Метод	Результат
3.0 * sin(sqrt(x)) + 0.35 * x - 3.8	[2.0, 3.0]	дихотомия	2.29853617092284157
		итерации	2.29853617092284246
		ньютон	2.29853617092284246
0.25 * (x * x * x) + x - 1.2502	[0, 2.0]	дихотомия	1.00011428011695269
		итерации	1.00011428011695269
		ньютон	1.00011428011695314

Вывод

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