

Московский Авиационный Институт
(Национальный исследовательский университет)

Институт №8 “Информационных технологий и прикладной математики”

Курсовой проект
по курсу
«Вычислительные системы»
I семестр

Задание 4

Студент: Ядров А. Л.

Группа: М8О-101Б-20

Руководитель: Никулин С. П.

Оценка: _____

Дата: _____

Подпись преподавателя: _____

1. Задача

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

2. Вариант

№	Уравнение	Отрезок, содержащий корень	Базовый метод	Приближенное значение корня
1	$e^x + \ln(x) + 10x$	[3, 4]	Ньютона	3.5265
2	$\cos(x) - e^{\frac{-x^2}{2}} + x + 1$	[1, 2]	дихотомии	1.0804

3. Общий метод решения

Для начала необходимо описать возможностями языка Си вычисление приближенного значения функции, нахождения корня методом итераций, методом Ньютона и методом дихотомии. Далее подставить заданные значения и вывести высчитанные корни, затем сравнить с приближенными значениями x .

4. Общие сведения о программе

Аппаратное обеспечение: домашний ноутбук

Операционная система: Fedora 33

Язык и система программирования: C, CLion

Число строк программы: 71

Местонахождение файлов: /CLionProjects/kursach/main.c

Компиляция программы в консоли Fedora: gcc -lm -g main.c

Вызов программы: ./a.out

5. Функциональное назначение

Программа предназначена для решения трансцендентных алгебраических уравнений различными численными методами.

6. Описание логической структуры

Программа получает на вход требуемый отрезок, далее находит корень уравнения $F(x) = 0$ различными методами и выводит значение на экран.

7. Описание переменных и констант

Имя	Тип	Назначение
eps	double	Машинное ε

8. Описание функций

function F1(x : double) : double

Функция, вычисляющая значение функции $e^x + \ln(x) + 10x$

Имя	Тип	Вид	Назначение
x	double	Входной параметр	Значение аргумента для вычисления значения функции $e^x + \ln(x) + 10x$

function F2(x : double) : double

Функция, вычисляющая значение функции $\cos(x) - e^{\frac{-x^2}{2}} + x + 1$

Имя	Тип	Вид	Назначение
x	double	Входной параметр	Значение аргумента для вычисления значения функции $\cos(x) - e^{\frac{-x^2}{2}} + x + 1$

function F1_1(x : double) : double

Функция, вычисляющая значение производной функции $e^x + \ln(x) + 10x$

Имя	Тип	Вид	Назначение
x	double	Входной параметр	Значение аргумента для вычисления значения производной функции $e^x + \ln(x) + 10x$

function F2_2(x : double) : double

Функция, вычисляющая значение производной функции $\cos(x) - e^{\frac{-x^2}{2}} + x + 1$

Имя	Тип	Вид	Назначение
x	double	Входной параметр	Значение аргумента для вычисления значения функции $\cos(x) - e^{\frac{-x^2}{2}} + x + 1$

function f1(x : double) : double

Функция, вычисляющая выраженный x для функции $e^x + \ln(x) + 10x$

Имя	Тип	Вид	Назначение
x	double	Входной параметр	Значение аргумента для вычисления выраженного x для функции $e^x + \ln(x) + 10x$

function f2(x : double) : double

Функция, вычисляющая выраженный x для функции $\cos(x) - e^{\frac{-x^2}{2}} + x + 1$

Имя	Тип	Вид	Назначение
x	double	Входной параметр	Значение аргумента для вычисления выраженного x для функции $\cos(x) - e^{\frac{-x^2}{2}} + x + 1$

Function dth(f(double), a, b, eps : double) : double

Функция, вычисляющая корень уравнения $f(x) = 0$ методом дихотомии

Имя	Тип	Вид	Назначение
f(double)	double	Входной параметр	Функция, для которой вычисляется корень
a			Левая граница отрезка
b			Правая граница отрезка
eps			Машинное ϵ

Function iter(f(double), a, b, eps : double) : double

Функция, вычисляющая корень уравнения $f(x) = 0$ методом итераций

Имя	Тип	Вид	Назначение
f(double)	double	Входной параметр	Функция выраженного x
a			Левая граница отрезка
b			Правая граница отрезка
eps			Машинное ϵ

Function newton(F(double), F1(double), a, b, eps : double) : double

Функция, вычисляющая корень уравнения $f(x) = 0$ методом Ньютона

Имя	Тип	Вид	Назначение
F(double)	double	Входной параметр	Функция $F(x)$
F1(double)			Функция производной функции $F(x)$
a			Левая граница отрезка
b			Правая граница отрезка
eps			Машинное ϵ

function ans(F(double), F1(double), f(double), a, b, eps: double): void

Функция вывода ответа на уравнение $F(x) = 0$, решенное 3 различными методами

Имя	Тип	Вид	Назначение
F(double)	double	Входной параметр	Функция $F(x)$
F1(double)			Функция производной функции $F(x)$
f(double)			Функция выраженного x
a			Левая граница отрезка
b			Правая граница отрезка
eps			Машинное ϵ

9. Входные данные

Отсутствуют

10. Выходные данные

Функция $\exp(x) + \ln(x) - 10x$

Корень, полученный методом дихотомии: 3.5264980

Корень, полученный методом итераций: 3.5264980

Корень, полученный методом Ньютона: 3.5264980

Функция $\cos(x) - \exp(-0.5 \cdot x^2) + x - 1$

Корень, полученный методом дихотомии: 1.0894428

Корень, полученный методом итераций: 1.0894428

Корень, полученный методом Ньютона: 1.0894428

11. Тестовые примеры

Не предусмотрены.

12. Дневник отладки

Дата	Место	Событие	Действие по исправлению

13. Выводы по задаче

Я составил программу на языке Си с процедурами решения трансцендентных алгебраических уравнений различными численными методами, и, следовательно, научился пользоваться этими методами.

14. Протокол

```
[Temi4@localhost kursach]$ cat main.c
```

```
#include <stdio.h>
```

```
#include <math.h>
```

```
double F1(double x) {  
    return exp(x) + log(x) - 10 * x;  
}
```

```
double F2(double x) {  
    return cos(x) - exp(-pow(x, 2) / 2) + x - 1;  
}
```

```
double f1(double x) {  
    return log(10*x-log(x));  
}
```

```
double f2(double x) {  
    return 1 + exp(-pow(x, 2) / 2) - cos(x);  
}
```

```
double F1_1(double x) {  
    return exp(x) + 1 / x - 10;  
}
```

```
double F2_1(double x) {  
    return -sin(x) + x * exp(-pow(x, 2) / 2) + 1;  
}
```

```
double dth(double f(double), double a, double b, double eps) {  
    double x;  
    while (fabs(a - b) > eps) {  
        x = (a + b) / 2;  
        if (f(x) * f(a) < 0) {  
            b = x;  
        } else {  
            a = x;  
        }  
    }  
    return (a+b)/2;  
}
```

```
double itter(double f(double), double a, double b, double eps) {
    double x = (a + b) / 2;
    while (fabs(f(x) - x) > eps) {
        x = f(x);
    }
    return x;
}
```

```
double newton(double F(double), double F1(double), double a, double b, double eps) {
    double x = (a + b / 2);
    while (fabs(F(x) / F1(x)) > eps) {
        x -= F(x)/F1(x);
    }
    return x;
}
```

```
void ans(double F(double), double F1(double), double f(double), double a, double b, double eps){
    printf("Корень, полученный методом дихотомии: %11.7f\n", dth(F, a, b, eps));
    printf("Корень, полученный методом итераций: %11.7f\n", itter(f, a, b, eps));
    printf("Корень, полученный методом Ньютона: %11.7f\n", newton(F, F1, a, b, eps));
}
```

```
int main() {
    double eps = 1;
    while (1 + eps/2 > 1){
        eps /= 2;
    }
    printf("Функция exp(x) + ln(x) - 10x\n");
    ans(F1, F1_1, f1, 3, 4, eps);
    printf("Функция cos(x) - exp(-0.5*x^2) + x - 1\n");
    ans(F2, F2_1, f2, 1, 2, eps);
    return 0;
}
```

```
}[Temi4@localhost kursach]$ gcc -lm -g main.c
```

```
[Temi4@localhost kursach]$ ./a.out
```

Функция $\exp(x) + \ln(x) - 10x$

Корень, полученный методом дихотомии: 3.5264980

Корень, полученный методом итераций: 3.5264980

Корень, полученный методом Ньютона: 3.5264980

Функция $\cos(x) - \exp(-0.5 \cdot x^2) + x - 1$

Корень, полученный методом дихотомии: 1.0894428

Корень, полученный методом итераций: 1.0894428

Корень, полученный методом Ньютона: 1.0894428

[Temi4@localhost kursach]\$