

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ  
ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
“Московский Авиационный Институт  
(Национальный Исследовательский Университет)”  
Факультет информационных технологий и прикладной математики  
Кафедра 806 “Вычислительная математика и программирование”

Курсовой проект  
по курсу “Фундаментальная информатика”  
1 семестр  
Задание 4. Процедуры и функции в качестве параметров

Студент: Сибирцев Р. Д.  
Группа: М8О-108Б-2  
Руководитель: Сахарин Н.А.  
Дата: 09.01.23  
Оценка:

Москва, 2023

## Задание

Составить программу на языке Си с процедурами решения трансцендентных алгебраических уравнений различными способами (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.

## Общий метод решения

Вычисление приближенных значений функций при помощи метода дихотомии, метода итераций и метода Ньютона.

Рассматривается уравнение вида  $F(x) = 0$ . Предполагается, что функция  $F(x)$  достаточно гладкая, монотонная на этом отрезке и существует единственный корень уравнения  $x^* \in [a, b]$ . на отрезке  $[a, b]$  ищется приближенное решение  $x$  с точностью  $\varepsilon$ , т.е. такое, что  $|x - x^*| < \varepsilon$ .

Метод дихотомии - деление отрезка пополам с учётом того, что знак функции на концах отрезка должен быть разным:  $F(a) * F(b) < 0$ . До тех пор, пока длина отрезка не будет меньше значения  $\varepsilon$ , процесс деления будет выполняться. Приближенное значение корня к моменту окончания итерационного процесса будет находиться примерно в середине заданного отрезка.

Метод итераций заключается в замене исходного уравнения  $F(x) = 0$  уравнением  $f(x) = x$ . Начальным приближенным значением корня является середина заданного отрезка  $x(0) = (a + b)/2$ . Итерационный процесс имеет вид:  $x(k+1) = f(x(k))$ . Процесс выполняется пока  $|x(k+1) - x(k-1)| < \varepsilon$

Метод Ньютона - частный случай метода итераций. Итерационный процесс представляет собой:  $x(k+1) = x(k) - F(x(k))/F'(x(k))$ .

## Назначение

Программа предназначена для высокоточного вычисления вещественного значения трансцендентных функций в алгебраической форме с использованием ряда Тейлора и при помощи встроенных программных функций библиотеки языка Си.

## Описание переменных и функций

Имя	Тип	Назначение
eps	double	Машинный эпсилон
a	double	Левая граница отрезка
b	double	Правая граница отрезка
x	double	Текущее значение аргумента
x_old	double	Предыдущее значение аргумента

Имя	Тип	Назначение
mech_eps	double	Вычисление машинного эпсилона
func	double	Вычисление значения функции
der	double	Вычисление значения производной функции
sec_der	double	Вычисление значения второй производной функции

## Листинг программы

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

double mech_eps(void) {
    double e = 1.0f;

    while (1.0f + e / 2.0f > 1.0f)
        e /= 2.0f;
    return e;
}

double func(double x) {
    double res = 0.25*x*x*x + x - 1.2502;
    return res;
}

double der(double x) {
    double res = 0.25*3*x*x + 1;
    return res;
}

double sec_der(double x) {
    double res = 0.25*3*2*x;
    return res;
}

double f2 (double x) {
    return (1.2502 - 0.25*x*x*x);
}

void newton(double a, double b) {
    double x = a, x_old = -1, eps = sqrt(mech_eps());
    while (x <= b) {
```

```

if (fabs(func(x)*sec_der(x)) < (der(x)*der(x))) {
    x_old = x;
    x = x_old - func(x)/der(x);
    if (fabs(x - x_old) <= eps) {
        printf("Newton's method\n");
        printf("%.16f\n", x);
        printf("%.16f\n", func(x));
        break;
    }
}
else {
    printf("doesn't converge\n");
    break;
}
}
}

```

```

void half(double a, double b) {
    double x = a, mid, eps = sqrt(mech_eps());
    while (b - a > eps) {
        mid = (a + b) / 2.;
        if (func(a) * func(mid) < 0) b = mid;
        if (func(mid) * func(b) < 0) a = mid;
    }
    x = (a + b) / 2;
    printf("dichotomy method\n");
    printf("%.16f\n", x);
    printf("%.16f\n", func(x));
}

```

```

void iter(double a, double b) {
    double x = a, x_old = -1, eps = sqrt(mech_eps());
    while (fabs(x_old - x) > eps) {
        x_old = x;
        x = f2(x);
    }
}

```

```

    }
    printf("iterations method\n");
    printf("%.16f\n", x);
    printf("%.16f\n", func(x));
}

```

```

int main(void) {
    double a = 0, b = 2;
    newton(a, b);
    printf("\n");
    half(a, b);
    printf("\n");
    iter(a, b);

}

```

## Выходные данные

Newton's method

1.0001142801169531

-0.0000000000000000

dichotomy method

1.0001142801484093

0.00000000000550537

iterations method

1.0001142800038547

-0.0000000001979416

## Дневник отладки

Дата	Место	Событие	Действие по исправлению

## **Вывод**

В результате выполнения данного курсового проекта были получены навыки работы с процедурами и функциями. Также было освоено практическое применение методов дихотомии, итераций и Ньютона для нахождения приближенного значения корней уравнений  $F(x) = 0$ . При решении заданий ответ, полученный программой, совпал с ответом в варианте.