МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Московский Авиационный Институт» (Национальный Исследовательский Университет)

Институт: №8 «Информационные технологии и прикладная математика»

Кафедра: 806 «Вычислительная математика и программирование»

Курсовая работа

по курсу «Фундаментальная информатика» I семестр

Задание 4

«Процедуры и функции в качестве параметров»

|  |  |
| --- | --- |
| Группа | М8О-109Б-22 |
| Студент | Яшин В.А. |
| Преподаватель | Сысоев М.А. |
| Оценка |  |
| Дата |  |

Москва, 2022

**Постановка задачи**

Составить программу на Си с процедурами решения трансцендентных

алгебраических уравнений резличными 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 7:**

Функция:



Отрезок содержащий корень: [1, 3]

Метод Ньютона

**Вариант 8:**

Функция:



Отрезок содержащий корень: [2, 3]

Метод Дихотомии

**Теоретическая часть**

**метод Ньютона**



**Метод дихотомии (половинного деления)**



**Описание алгоритма**

Составляю программу для нахождения корня с помощью метода Ньютона и

проверяю найденный корень, либо вывожу, что метод не применим.

Аналогично поступаю и с методом дихотомии.

**Исходный код программы:**

**7**

#include <stdio.h>

#include <math.h>

// Функция, данная в задании

long double func ( long double x ) {

return 0.6 \* pow(3, x) - 2.3 \* x - 3;

}

// Функция модуль

long double my\_abs( long double x) {

if ( x < 0 ) {

return -x;

}

return x;

}

int main() {

long double eps = 1.0l, a = 2.0l, b = 3.0l;

// Вычисление машинного эпсилона

while (1.0l + eps / 2.0l > 1.0l) {

eps /= 2.0l;

}

eps \*= 100;

printf("Машинное эпсилон для типа double = %.16Le\n", eps);

// Вычисление методом дихотомии (метода половинного деления)

while (my\_abs(a - b) > eps) {

if (( func(a) \* func((a + b) / 2.0l) ) > 0) {

a = ( a + b ) / 2.0l;

} else {

if ((func(b) \* func((a + b) / 2.0l)) > 0) {

b = ( a + b ) / 2.0l;

} else {

break;

}

}

}

// Вывод ответа

printf("Приближенное значение корня, полученного при помощи метода дихотомии (метода половинного деления) равно: %Le\n", (a + b) / 2);

}

**8**

#include <stdio.h>

#include <math.h>

// Функция, данная в задании

long double func ( long double x ) {

return 3 \* log(x) \* log(x) + 6 \* log(x) - 5;

}

// Производная функции, данной в задании

long double dif\_func (long double x) {

return 6 \* log(x) / x + 6 / x;

}

// Вторая производная функции, данной в задании

long double dif\_dif\_func (long double x) {

return -6 \* log(x) / x / x;

}

// Функция модуль

long double my\_abs( long double x) {

if ( x < 0 ) {

return -x;

}

return x;

}

int main() {

long double eps = 1.0l, a = 1.0l, b = 2.0l, x, x1 = 0;

// Вычисление машинного эпсилона

while (1.0l + eps / 2.0l > 1.0l) {

eps /= 2.0l;

}

// Изначальное эпсилон слишком маленькое. Поэтому, в вычисление происходит зацикливание. Можно чуть - чуть увеличить эпсилон, и метод Ньютона будет работать.

eps \*= 50;

printf("Машинное эпсилон для типа double = %.16Le\n", eps);

// Проверка на сходимость

// Для этого проходимся циклом от левой границе к правой, с шагом 1/1000000, проверяя, выполнение условие сходимости |F(x) \* F''(x)| < (F'(x))^2, если условие не выполнятеся, мы завершаем программу

for (long long i = 1000000; i <= 2000000; i++ ) {

x = i / 1000000.0l;

if (my\_abs(func(x) \* dif\_dif\_func(x)) - dif\_func(x) \* dif\_func(x) >= eps) {

printf("Невозможно вычислить значение методом Ньютона, метод не сходится.\n");

return 0;

}

}

// Изначально, х равен середине отрезка

x = (a + b) / 2;

// Вычисление корня

while (my\_abs(x - x1) >= eps) {

x1 = x;

x = x1 - func(x) / dif\_func(x);

}

// Вывод ответа

printf("Приближенное значение корня, полученного при помощи метода Ньютона равно: %Le\n", x);

}

**Входные данные**

-----

**Выходные данные**

Программа должна вывести для первого уравнения сходится метод или нет.

В случае, если сходится, вывести его значение. Для второго уравнения

вывести найденный корень и значение уравнения при таком корне.

**Протокол исполнения и тесты**

**7**

Машинное эпсилон для типа double = 1.0842021724855044e-17

Приближенное значение корня, полученного при помощи метода дихотомии (метода половинн�го деления) равно: 2.419982e+00

**8**

Машинное эпсилон для типа double = 5.4210108624275222e-18

Приближенное значение корня, полученного при помощи метода Ньютона равно: 1.883239e+00

**Вывод**

В работе описаны и использованы различные численные методы для

решения трансцендентных алгебраических уравнений. Даны обоснования

сходимости и расходимости тех или иных методов. Имплементирована

функция вычисления производной от заданной функции в точке. На

основе алгоритма составлена программа на языке Си, сделана проверка

полученных значений путем подстановки. Работа представляется

довольно полезной для понимания принципов работы численных методов

и способов их имплементации.

**Список литературы**

1. Численное дифференецирование – URL:

<https://ru.wikipedia.org/wiki/Численное_дифференцирование>

2. Конечная разность – URL:

<https://ru.wikipedia.org/wiki/Численное_дифференцирование>