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	Мозговой Н.Е.
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 11:

Функция:

11	$e^x + \sqrt{1 + e^{2x}} - 2 = 0$

Отрезок содержащий корень: [- 1.0, 0.0]

Вариант 12:

Функция:

12	$\ln x - x + 1{,}8 = 0$

Отрезок содержащий корень: [2.0, 3.0]

Теоретическая часть

Метод итераций

Идея метода заключается в замене исходного уравнения $F(x)=0$ уравнением вида $x=f(x)$.

Достаточное условие сходимости метода: $|f'(x)|<1, x\in[a,b]$. Это условие необходимо проверить перед началом решения задачи, так как функция $f(x)$ может быть выбрана неоднозначно, причем в случае неверного выбора указанной функции метод расходится.

Начальное приближение корня: $x^{(0)}=(a+b)/2$ (середина исходного отрезка).

Итерационный процесс: $x^{(k+1)}=f(x^{(k)})$.

Условие окончания: $|x^{(k)}-x^{(k-1)}|<\varepsilon$.

Приближенное значение корня: $x^{*}\approx x^{(конечное)}$.

Метод дихотомии (половинного деления)

Очевидно, что если на отрезке $[a,b]$ существует корень уравнения, то значения функции на концах отрезка имеют разные знаки: $F(a)\cdot F(b)<0$. Метод заключается в делении отрезка пополам и его сужении в два раза на каждом шаге итерационного процесса в зависимости от знака функции в середине отрезка.

Итерационный процесс строится следующим образом: за начальное приближение принимаются границы исходного отрезка $a^{(0)}=a$, $b^{(0)}=b$. Далее вычисления проводятся по формулам: $a^{(k+1)}=(a^{(k)}+b^{(k)})/2$, $b^{(k+1)}=b^{(k)}$, если $F(a^{(k)})\cdot F((a^{(k)}+b^{(k)})/2)>0$; или по формулам: $a^{(k+1)}=a^{(k)}$, $b^{(k+1)}=(a^{(k)}+b^{(k)})/2$, если $F(b^{(k)})\cdot F((a^{(k)}+b^{(k)})/2)>0$.

Процесс повторяется до тех пор, пока не будет выполнено условие окончания $|a^{(k)}-b^{(k)}|<\varepsilon$.

Приближенное значение корня к моменту окончания итерационного процесса получается следующим образом $x^{*}\approx(a^{(конечное)}+b^{(конечное)})/2$.

Описание алгоритма

Делаем функцию для высчитывания корня методом дихотомии. После чего выводим его значение. Аналогично поступаем и для метода итераций, но для него выгодно будет сделать проверку отдельно.

Использованные в програме переменные

Название переменной	Тип переменной	Смысл переменной
xk	long double	Следующее значение x
x	long double	Начальное значение x
a	long double	Левая граница отрезка
b	long double	Правая граница отрезка

Исходный код программы:

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

long double function11(long double x) {
    return exp(x) + powl(1+exp(2*x), 0.5) - 2;
}

long double function12(long double x) {
    return log(x)+ 1.8;
}

long double derivative(long double x) {
    return 1/x;
}

long double move(long double (*f)(long double), long double a, long double b) {
    long double x = (a + b) / 2.0;
    long double xk = function12(x);
    while (fabsl(xk - x) > LDBL_EPSILON) {
        x = xk;
        xk = function12(x);
    }
    return xk;
}

int main() {
    long double a = -1.0;
    long double b = 0.0;

    if (function11(a) * function11(b) > 0) {
        printf("11. NO ROOTS\n");
    }
    else {
        while (fabsl(a - b) > LDBL_EPSILON) {
            if (function11(a) * function11((a + b) / 2) > 0) {
                a = (a + b) / 2;
                b = b;
                break;
            }
            else if (function11(b) * function11((a + b) / 2) > 0) {
                a = a;
                b = (a + b) / 2;
                break;
            }
        }

        printf("11. Our root is: %.4Lf\n", (a + b) / 2);
    }

    a = 2.0;
```

```c
    b = 3.0;

    if (fabsl(derivative(a)) < 1 || fabsl(derivative(b)) < 1) {
        printf("12. Our x = %.4Lf\n", move(function12, a, b));
    }
    else {
        printf("12. No");
    }

    return 0;
}
```

Входные данные

Нет

Выходные данные

Полученные значения

Тест №1

```
11. Our root is: -0.2500
12. Our x = 2.8459
```

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 дифференцирование – URL:

   Численное дифференцирование — Википедия (wikipedia.org)

2. Конечная разность – URL:

   Численное дифференцирование — Википедия (wikipedia.org)