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	Федоров А. А.
Преподаватель	Сысоев М. А.
Оценка	
Дата	

Москва, 2022

Постановка задачи

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

Варианты 5, 6.

№	Уравнение	Отрезок, содержащий корень	Базовый метод	Приближенное значение корня
5	$\sqrt{1-x} - \mathrm{tg}x = 0$	[0, 1]	дихотомии	0.5768
6	$x + \cos(x^{0,52} + 2) = 0$	[0.5, 1]	итераций	0.9892

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

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 $\left|a^{(k)}-b^{(k)}\right|<\varepsilon$.

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

Численное дифференцирование — Так как возможности компьютера не позволяют проводить вычисления с бесконечно малыми, для расчетов будем брать просто очень маленькие значения. Так, для вычисления производной через предел возьмем prib равное 1e-6

Описание алгоритма

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

Исходный код программы:

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

long double MachineEps() {
    long double eps = 1.0l;
    while (1.0l + eps > 1.0l) {
        eps *= 0.5l;
    }
    return eps;
}

long double fun5(long double x) {
    return sqrtl(1 - x) - tanl(x);
}

long double fun6(long double x) {
    return -cosl(powl(x, 0.52) + 2);
}

long double absl(long double x) {
    if(x < 0) {
        return -x;
    }
    return x;
}

long double dichotomy_method(long double (*f) (long double), long double a,
long double b, long double eps) {
    long double root;
    while (fabsl(a - b) > eps) {
        root = (a + b) / 2.0;
        if (f(root) * f(a) < 0) {
            b = root;
        } else {
            a = root;
        }
    }
    return root;
}

long double iter_method(long double (*f) (long double), long double a, long
double b, long double eps) {
    long double x0 = (a + b) / 2.0, x = f(x0), diff = x - x0;
    while (absl(diff) >= eps) {
        x = f(x0);
        diff = x - x0;
        x0 = x;
    }
    return x;
}

long double derive(long double (*f)(long double), long double x0) {
    long double delta = 1e-6;
    return (f(x0 + delta) - f(x0)) / delta;
}

int check_is_iter(long double (*f)(long double), long double a, long double b)
{
    long double s = (b - a) / 1e4;
    for (long double x = a; x <= b;) {
        if (derive(f, x) >= 1) {
            return 0;
        }
        x += s;
```

```c
    }
    return 1;
}

int main() {
    long double eps = MachineEps();

    // вариант 5
    long double a = 0.0, b = 1.0;
    long double root = dichotomy_method(fun5, a, b, eps);
    printf("Method: dichotomy, root is: %.20Lf\n", root);

    printf("\n");
    // вариант 6
    a = 0.5, b = 1.0;
    if (check_is_iter(fun6, a, b)) {
        root = iter_method(fun6, a, b, eps);
        printf("Method: iteration, can be applied, root is: %.20Lf\n", root);
    } else {
        printf("Iteration method cannot be applied\n");
    }
}
```

Входные данные

Нет

Выходные данные

Для первого уравнения программа должна вывести значение корня. Для второго уравнения программа должна вывести сообщение о том, сходится метод или нет. В случае, если сходится, вывести его значение.

Тест №1

```
Method: dichotomy, root is: 0.57676980757075159927

Method: iteration, can be applied, root is: 0.98918073496336520454
```

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)