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	Ефименко К.И.
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.

Вариант 21:

Функция:

$$\operatorname{tg} x - \frac{1}{3}\operatorname{tg}^3 x + \frac{1}{5}\operatorname{tg}^5 x - \frac{1}{3} = 0$$

Отрезок содержащий корень: [0.0, 0.8]

Вариант 22:

Функция:

$$\arccos x - \sqrt{1 - 0{,}3x^3} = 0$$

Отрезок содержащий корень: [0.0, 1.0]

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

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 eps равное LDBL_EPSILON

Описание алгоритма

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

Использованные в программе переменные

Название переменной	Тип переменной	Смысл переменной
eps	double	Маленькое значение
iter	double	Шаг для проверки
a	double	Левая граница отрезка
b	double	Правая граница отрезка
Prev_x	double	Следующее значение x

Исходный код программы:

Метод дихотомии

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

double func(double x) {
    double res;
    res = tan(x) - 1.0/3.0 * pow(tan(x), 3) + 1.0/5.0 * pow(tan(x), 5) - 1.0/3.0;
    return res;
}

void solution(double a, double b, double eps) {
    double x = 0;
    int count = 0;

    if (func( x: a) * func( x: b) < 0) {
        printf("The convergence condition is met");

        while ((b - a) > eps) {
            x = (a + b) / 2.0;
            count++;

            if (func( x: a) * func(x) < 0) {
                b = x;
            } else {
                a = x;
            }
        }

        printf("x = %f\n", x);
        printf("number of iterations = %d", count);

    } else {
        printf("The convergence condition is not met");
    }
```

```c

}

int main() {
    double eps = LDBL_EPSILON;
    double a = 0.0;
    double b = 0.8;

    solution(a, b, eps);

    return 0;
}
```

Метод итераций:

```c
#include <stdio.h>
#include <math.h>
#include <float.h>
#include <stdbool.h>

typedef double (*func)(double);

typedef struct result {
    double answ;
    int iters;
    bool check;
} result;

double function(double x) {
    return acos(x) - sqrt(1 - 0.3 * pow(x, 3));
}

double iter(double x) {
    func f = function;
    return x - f(x) * -1;
}

result iterationsMethod(func iterF, double a, double b) {
    double prevX;
    double x = (a + b) / 2;
    double eps = LDBL_EPSILON;
    result res  = { .answ: 0,  .iters: 0,  .check: 0};

    do {
        res.check = fabs((iterF(x + eps) - iterF(x - eps)) / (2 * eps)) < 1;
        if (!res.check) return res;
        prevX = x;
        x = iterF(prevX);
        res.iters++;
        res.answ = x;
    } while (fabs(x - prevX) > eps);

    return res;
}

void printResearch(func iter_f, double a, double b) {
    result res = iterationsMethod( iterF: iter_f, a, b);

    printf(" x =  %.16lf\n number of iterations = %d  ", res.answ, res.iters);

}

int main() {

    printResearch( iter_f: iter,  a: 0.0,  b: 1.0);

    return 0;
}
```

Входные данные

Нет

Выходные данные

Программа должна вывести для второго уравнения сходится метод или нет. В случае, если сходится, вывести его значение. Для первого уравнения вывести его значение.

Тест №1

Метод дихотомии:

```
The convergence condition is met
x = 0.333255
number of iterations = 55
```

Метод итераций:

```
x = 0.5629259528580387
number of iterations = 14
```

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:

   Численное дифференцирование — Википедия (wikipedia.org)

2. Конечная разность – URL:

   Численное дифференцирование — Википедия (wikipedia.org)