**Министерство науки и высшего образования Российской Федерации**

федеральное государственное автономное образовательное учреждение

высшего образования

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Инженерная школа природных ресурсов

Направление подготовки Химическая технология

Отделение химической инженерии

**ИТЕРАЦИОННЫЕ МЕТОДЫ РЕШЕНИЯ НЕЛИНЕЙНЫХ УРАВНЕНИЙ**

**Лабораторная работа по дисциплине «Углубленный курс информатики»**

Выполнил студент гр. Е.С. Мухина

(Подпись)

\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г.

Отчет принят:

Преподаватель

доцент ОХИ ИШПР, к.т.н. В.А. Чузлов

(Подпись)

\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г.

Томск 2020 г.

**Цель работы:** рассмотреть и составить программы с использованием итерационных методов решения нелинейных уравнений.

**Теоретическая часть**

**Итерация** в [программировании](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) — в широком смысле — организация обработки данных, при которой действия повторяются многократно, не приводя при этом к вызовам самих себя (в отличие от [рекурсии](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BA%D1%83%D1%80%D1%81%D0%B8%D1%8F)). В узком смысле — один шаг итерационного, циклического процесса[[1]](https://ru.wikipedia.org/wiki/%D0%98%D1%82%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D1%8F_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)#cite_note-1).

Когда какое-то действие необходимо повторить большое количество раз, в программировании используются [циклы](https://ru.wikipedia.org/wiki/%D0%A6%D0%B8%D0%BA%D0%BB_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)). Например, нужно вывести 200 раз на экран текст «[Hello, World!](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0_Hello_world" \o "Программа Hello world)». Вместо двухсоткратного повторения одной и той же команды вывода текста часто создается цикл, который повторяется 200 раз и 200 раз выполняет то, что написано в [теле цикла](https://ru.wikipedia.org/wiki/%D0%A6%D0%B8%D0%BA%D0%BB_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)#%D0%A2%D0%B5%D0%BB%D0%BE_%D1%86%D0%B8%D0%BA%D0%BB%D0%B0).

На практике уравнения не всегда можно решить прямыми методами. Для их решения применяются итерационные методы, т.е. методы последовательных приближений. Решение нелинейных уравнений итерационными методами проводится в 2 этапа:

1. Отделение корней, т.е. установление малых отрезков, в каждом из которых содержится только один корень уравнения.

2. Уточнение корней до некоторой заданной степени точности.

Самый простой – метод половинного деления, или иначе метод дихотомии. Метод дихотомии получил свое название от древнегреческого слова διχοτομία, что в переводе означает деление надвое. Его мы используем довольно часто. Допустим, играя в игру "Угадай число", где один игрок загадывает число от 1 до 100, а другой пытается его отгадать, руководствуясь подсказками "больше" или "меньше". Логично предположить, что первым числом будет названо 50, а вторым, в случае если оно меньше - 25, если больше - 75.

**Практическая часть**

**Задание 1**

**Исходные данные**:

Интервал [1; 2], допустимая точность 10-2

**Задание**

Составьте программу для решения нелинейных уравнений методом половинного деления, простых итераций и методом Ньютона.

**Программная реализация**

**1 способ:**

**Program** lab9;

**const**

eps = 1e-2;

**function** f(x: real);

**begin**

result:= exp(ln(x) \* 4) + 3 \* x - 20

**end**;

**function** dihotomy(a, b: real; eps: real): real;

**var**

x: real;

**begin**

**repeat**

x:= (a + b) / 2;

**if** f(a) \* f(x) > 0 **then**

a:= x

**else**

b:= x

**until** (abs(a - b) <= eps) **or** (f(x) = 0);

result:= x

**end**;

**begin**

writeln('Корень уравнения равен: ', dihotomy(1, 2, eps))

**end**.

**Ответ:**

Корень уравнения равен: 1.9453125

**2 способ:**

**Program** lab9;

**const**

eps = 1e-2;

**function** g(x: real): real;

**begin**

result := exp(ln(20 - 3 \* x) \* 1 / 4)

**end**;

**function** iterations(a, b: real; eps: real): real;

**var**

x: real;

**begin**

result := a;

**repeat**

x := g(result);

result := g(x)

**until** abs(result - x) <= eps;

**end**;

**begin**

writeln('Корень уравнения равен: ', iterations(1, 2, eps))

**end**.

**Ответ:**

Корень уравнения равен: 1.94037733840934

**3 способ:**

**Program** lab9;

**const**

eps = 1e-2;

**function** f(x: real): real;

**begin**

result := exp(ln(x) \* 4) + 3 \* x - 20

**end**;

**function** f1(x: real): real;

**begin**

result := 4 \* exp(ln(x) \* 3) + 3

**end**;

**function** f2(x: real): real;

**begin**

result := 12 \* exp(ln(x) \* 2)

**end**;

**function** newton(a, b: real; eps: real): real;

**var**

x: real;

**begin**

**if** f(a) \* f2(a) > 0 **then**

result := a

**else**

**if** f(b) \* f2(b) > 0 **then**

result := b

**else**

**begin**

writeln('Метод Ньютона, решений нет!');

**exit**

**end**;

**repeat**

x := result;

result := x - f(x) / f1(x)

**until** abs(result - x) <= eps;

**end**;

**begin**

writeln('Корень уравнения равен: ', newton(1, 2, eps))

**end**.

**Ответ:**

Корень уравнения равен: 1.94047935224908

**Задание 2**

**Исходные данные**:

Интервал [0; 1], допустимая точность 10-3

**Задание**

Составьте программу для решения нелинейных уравнений методом половинного деления, простых итераций и методом Ньютона.

**Программная реализация**

**1 способ:**

**Program** lab9;

**const**

eps = 1e-3;

**function** f(x: real);

**begin**

result:= exp(x) + x - 2

**end**;

**function** dihotomy(a, b: real; eps: real): real;

**var**

x: real;

**begin**

**repeat**

x:= (a + b) / 2;

**if** f(a) \* f(x) > 0 **then**

a:= x

**else**

b:= x

**until** (abs(a - b) <= eps) **or** (f(x) = 0);

result:= x

**end**;

**begin**

writeln('Корень уравнения равен: ', dihotomy(0, 1, eps))

**end**.

**Ответ:**

Корень уравнения равен: 0.4423828125

**2 способ:**

**Program** lab9;

**const**

eps = 1e-3;

**function** g(x: real): real;

**begin**

result:= ln(2 - x)

**end**;

**function** iterations(a, b: real; eps: real): real;

**var**

x: real;

**begin**

result:= a;

**repeat**

x:= g(result);

result:= g(x)

**until** abs(result - x) <= eps;

**end**;

**begin**

writeln('Корень уравнения равен: ', iterations(0, 1, eps))

**end**.

**Ответ:**

Корень уравнения равен: 0.442509950010955

**3 способ:**

**Program** lab9;

**const**

eps = 0.2 \* 1e-4;

**function** f(x: real): real;

**begin**

result := exp(x) + x - 2

**end**;

**function** f1(x: real): real;

**begin**

result := exp(x) + 1

**end**;

**function** f2(x: real): real;

**begin**

result := exp(x)

**end**;

**function** newton(a, b: real; eps: real): real;

**var**

x: real;

**begin**

**if** f(a) \* f2(a) > 0 **then**

result := a

**else**

**if** f(b) \* f2(b) > 0 **then**

result := b

**else**

**begin**

writeln('Метод Ньютона, решений нет!');

**exit**

**end**;

**repeat**

x := result;

result := x - f(x) / f1(x)

**until** abs(result - x) <= eps;

**end**;

**begin**

writeln('Корень уравнения равен: ', newton(0, 1, eps))

**end**.

**Ответ:**

Корень уравнения равен: 0.4423828125

**Задание 3**

**Исходные данные**:

Интервал [0.5; 1.5], допустимая точность 0.2\*10-4

**Задание**

Составьте программу для решения нелинейных уравнений методом половинного деления, простых итераций и методом Ньютона

**Программная реализация**

**1 способ:**

**Program** lab9;

**const**

eps = 0.2 \* 1e-4;

**function** f(x: real);

**begin**

result:= ln(x) + x

**end**;

**function** dihotomy(a, b: real; eps: real): real;

**var**

x: real;

**begin**

**repeat**

x:= (a + b) / 2;

**if** f(a) \* f(x) > 0 **then**

a:= x

**else**

b:= x

**until** (abs(a - b) <= eps) **or** (f(x) = 0);

result:= x

**end**;

**begin**

writeln('Корень уравнения равен: ', dihotomy(0.5, 1.5, eps))

**end**.

**Ответ:**

Корень уравнения равен: 0.567143290399369

**2 способ:**

**Program** lab9;

**const**

eps = 0.2 \* 1e-4;

**function** g(x: real): real;

**begin**

result:= exp(- x)

**end**;

**function** iterations(a, b: real; eps: real): real;

**var**

x: real;

**begin**

result:= a;

**repeat**

x:= g(result);

result:= g(x)

**until** abs(result - x) <= eps;

**end**;

**begin**

writeln('Корень уравнения равен: ', iterations(0.5, 1.5, eps))

**end**.

**Ответ:**

Корень уравнения равен: 0.567140763269807

**3 способ:**

**Program** lab9;

**const**

eps = 0.2 \* 1e-4;

**function** f(x: real): real;

**begin**

result := ln(x) + x

**end**;

**function** f1(x: real): real;

**begin**

result := 1 / x + 1

**end**;

**function** f2(x: real): real;

**begin**

result := -1 / (x \* x)

**end**;

**function** newton(a, b: real; eps: real): real;

**var**

x: real;

**begin**

**if** f(a) \* f2(a) > 0 **then**

result := a

**else**

**if** f(b) \* f2(b) > 0 **then**

result := b

**else**

**begin**

writeln('Метод Ньютона, решений нет!');

**exit**

**end**;

**repeat**

x := result;

result := x - f(x) / f1(x)

**until** abs(result - x) <= eps;

**end**;

**begin**

writeln('Корень уравнения равен: ', newton(0.5, 1.5, eps))

**end**.

**Ответ:**

Корень уравнения равен: 0.567143290399369

**Задание 4**

**Исходные данные**:

Интервал [0.2; 1.5], допустимая точность 0.5\*10-4

**Задание**

Составьте программу для решения нелинейных уравнений методом половинного деления, простых итераций и методом Ньютона.

**Программная реализация**

**1 способ:**

**Program** lab9;

**const**

eps = 0.5 \* 1e-4;

**function** f(x: real);

**begin**

result:= 2 \* x - exp(-0.1 \* x)

**end**;

**function** dihotomy(a, b: real; eps: real): real;

**var**

x: real;

**begin**

**repeat**

x:= (a + b) / 2;

**if** f(a) \* f(x) > 0 **then**

a:= x

**else**

b:= x

**until** (abs(a - b) <= eps) **or** (f(x) = 0);

result:= x

**end**;

**begin**

writeln('Корень уравнения равен: ', dihotomy(0.2, 1.5, eps))

**end**.

**Ответ:**

Корень уравнения равен: 0.476721637710739

**2 способ:**

**Program** lab9;

**const**

eps = 0.5 \* 1e-4;

**function** g(x: real): real;

**begin**

result:= exp(-0.1 \* x) / 2

**end**;

**function** iterations(a, b: real; eps: real): real;

**var**

x: real;

**begin**

result:= a;

**repeat**

x:= g(result);

result:= g(x)

**until** abs(result - x) <= eps;

**end**;

**begin**

writeln('Корень уравнения равен: ', iterations(0.2, 1.5, eps))

**end**.

**Ответ:**

Корень уравнения равен: 0.476721637710739

**3 способ:**

**Program** lab9;

**const**

eps = 0.5 \* 1e-4;

**function** f(x: real): real;

**begin**

result := 2 \* x - exp(-0.1 \* x)

**end**;

**function** f1(x: real): real;

**begin**

result := 2 + 0.1 \* exp(-0.1 \* x)

**end**;

**function** f2(x: real): real;

**begin**

result := 0.1 \* exp(-0.1 \* x)

**end**;

**function** newton(a, b: real; eps: real): real;

**var**

x: real;

**begin**

**if** f(a) \* f2(a) > 0 **then**

result := a

**else**

**if** f(b) \* f2(b) > 0 **then**

result := b

**else**

**begin**

writeln('Метод Ньютона, решений нет!');

**exit**

**end**;

**repeat**

x := result;

result := x - f(x) / f1(x)

**until** abs(result - x) <= eps;

**end**;

**begin**

writeln('Корень уравнения равен: ', newton(0.2, 1.5, eps))

**end**.

**Ответ:**

Корень уравнения равен: 0.476721637710739

**Выводы**

В ходе работы успешно обучилась составлять программы итерационных методов решения нелинейных уравнений.