**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. А.В. Исаева

(Подпись)

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

Отчет принят:

Преподаватель

доцент ОХИ ИШПР, к.т.н. В.А. Чузлов

(Подпись)

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

Томск 2020 г.

**Цель работы:** составление программ с использованием итерационного метода решения нелинейных уравнений.

**Теоретическая часть**

Нелинейные уравнения можно разделить на 2 класса - алгебраические и трансцендентные. *Алгебраическими уравнениями* называют уравнения, содержащие только алгебраические функции (целые, рациональные, иррациональные). В частности, многочлен является целой алгебраической функцией. Уравнения, содержащие другие функции (тригонометрические, показательные, логарифмические и другие) называются *трансцендентными.*

Методы решения нелинейных уравнений делятся на две группы:

1. *точные методы*;
2. *итерационные методы*.

*Точные методы* позволяют записать корни в виде некоторого конечного соотношения (формулы). Из школьного курса алгебры известны такие методы для решения тригонометрических, логарифмических, показательных, а также простейших алгебраических уравнений.

Как известно, многие уравнения и системы уравнений не имеют аналитических решений. В первую очередь это относится к большинству трансцендентных уравнений. Доказано также, что нельзя построить формулу, по которой можно было бы решить произвольное алгебраическое уравнение степени выше четвертой. Кроме того, в некоторых случаях уравнение содержит коэффициенты, известные лишь приблизительно, и, следовательно, сама задача о точном определении корней уравнения теряет смысл. Для их решения используются *итерационные методы*с заданной степенью точности.

**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. **program** lab9\_11;

**const**

eps = 1e-2;

**function** f(x: real): 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(dihotomy(1, 2, eps))

**end**.

1. **program** lab9\_12;

**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(iterations(1, 2, eps))

**end**.

1. **program** lab9\_13;

**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 \* 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(newton(1, 2, eps))

**end**.

**Ответ**

1. 1.9453125
2. 1.94037733840934
3. 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. **program** lab9\_21;

**const**

eps = 1e-3;

**function** f(x: real): 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(dihotomy(0, 1, eps))

**end**.

1. **program** lab9\_22;

**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(iterations(0, 1, eps))

**end**.

1. **program** lab9\_23;

**const**

eps = 1e-3;

**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(newton(0, 1, eps))

**end**.

**Ответ**

1. 0.4423828125
2. 0.442509950010955
3. 0.442854401004033

**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. **program** lab9\_31;

**const**

eps = 0.2 \* 1e-4;

**function** f(x: real): 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(dihotomy(0.5, 1.5, eps))

**end**.

1. **program** lab9\_32;

**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(iterations(0.5, 1.5, eps))

**end**.

1. **program** lab9\_33;

**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 + 1 / x

**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(newton(0.5, 1.5, eps))

**end**.

**Ответ**

1. 0.567153930664063
2. 0.567140763269807
3. 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. **program** lab9\_41;

**const**

eps = 0.5 \* 1e-4;

**function** f(x: real): 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(dihotomy(0.2, 1.5, eps))

**end**.

1. **program** lab9\_42;

**const**

eps = 1e-2;

**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(iterations(0.2, 1.5, eps))

**end**.

1. **program** lab9\_43;

**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 := -0.01 \* 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(newton(0.2, 1.5, eps))

**end**.

**Ответ**

1. 0.476718139648438
2. 0.476721637710739
3. 0.476723086001294

**Выводы**

В ходе работы были составленные программы с использованием итерационного метода решения нелинейных уравнений.