**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. 2Д93 Филатова О.В.

(Подпись)

\_\_1\_\_\_ \_\_\_\_мая\_\_\_\_\_\_\_\_\_ 2020 г.

Отчет принят:

Преподаватель

доцент ОХИ ИШПР, к.т.н. В.А. Чузлов

(Подпись)

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

Томск 2020 г.

**Цель работы:** научиться использовать итерационные методы решения нелинейных уравнений

**Теоретическая часть**

Существуют три итерационных метода решения нелинейных уравнений:

1. Метод деления отрезка пополам
2. Метод простых итераций
3. Метод Ньютона (касательных)

**Практическая часть**

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

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

**Программная реализация:**

Половинное деление:

**program** lab\_09\_example;

**const**

eps = 1e-2;

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

**begin**

result := x\*x\*x\*x+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, 3, eps))

**end**.

Ответ: 1.9453125

Метод простых интеграций:

**program** lab\_09\_example\_2;

**const**

eps = 1e-2;

// эквивалентная функция

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

**begin**

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

**end**;

// реализация метода простых итераций

**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**;

// основная программа

// вызов функции

**begin**

writeln(iterations(1,2, eps))

**end**.

Ответ: 1.94037733840934

Метод Ньютона:

**program** lab\_09\_example\_3;

**const**

eps = 1e-2;

// исходная функция

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

**begin**

result := x\*x\*x\*x+3\*x-20

**end**;

// первая производная

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

**begin**

result := 4\*x\*x\*x+3

**end**;

// вторая производная

**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**;

// основная программа, вызов функции

**begin**

writeln(newton(1, 2, eps))

**end**.

Ответ: 1.94047935224908

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

**Программная реализация:**

Половинное деление:

**program** lab\_09\_example;

**const**

eps = 1e-3;

// уравнение, корень которого нужно определить

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

**begin**

result := exp(x)+x-2

**end**;

// реализация метода деления отрезка пополам

**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**.

Ответ: 0.4423828125

Метод простых интеграций:

**program** lab\_09\_example\_2;

**const**

eps = 1e-3;

// эквивалентная функция

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

**begin**

result := ln(2-x)

**end**;

// реализация метода простых итераций

**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**;

// основная программа

// вызов функции

**begin**

writeln(iterations(0,1, eps))

**end**.

Ответ: 0.442509950010955

Метод Ньютона:

**begin**

result := exp(x)+1

**end**;

// вторая производная

**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**;

// основная программа, вызов функции

**begin**

writeln(newton(0, 1, eps))

**end**.

Ответ: 0.442854401004033

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

**Программная реализация:**

Половинное деление:

**program** lab\_09\_example;

**const**

eps = 0.2e-4;

// уравнение, корень которого нужно определить

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

**begin**

result := ln(x)+x

**end**;

// реализация метода деления отрезка пополам

**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**.

Ответ: 0.567153930664063

Метод простых интеграций:

**program** lab\_09\_example\_2;

**const**

eps = 0.2e-4;

// эквивалентная функция

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

**begin**

result := exp(-x)

**end**;

// реализация метода простых итераций

**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**;

// основная программа

// вызов функции

**begin**

writeln(iterations(0.5,1.5, eps))

**end**.

Ответ: 0.567140763269807

Метод Ньютона:

**program** lab\_09\_example\_3;

**const**

eps = 0.2e-4;

// исходная функция

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

**begin**

result := ln(x)+x

**end**;

// первая производная

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

**begin**

result := 1/x+1

**end**;

// вторая производная

**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**;

// основная программа, вызов функции

**begin**

writeln(newton(0.5, 1.5, eps))

**end**.

Ответ: 0.567143290399369

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

**Программная реализация:**

Половинное деление:

**program** lab\_09\_example;

**const**

eps = 0.5e-4;

// уравнение, корень которого нужно определить

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

**begin**

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

**end**;

// реализация метода деления отрезка пополам

**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**.

Ответ: 0.476718139648438

Метод простых интеграций:

**program** lab\_09\_example\_2;

**const**

eps = 0.5e-4;

// эквивалентная функция

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

**begin**

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

**end**;

// реализация метода простых итераций

**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**;

// основная программа

// вызов функции

**begin**

writeln(iterations(0.2,1.5, eps))

**end**.

Ответ: 0.476721637710739

Метод Ньютона:

**program** lab\_09\_example\_3;

**const**

eps = 0.5e-4;

// исходная функция

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

**begin**

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

**end**;

// первая производная

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

**begin**

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

**end**;

// вторая производная

**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**;

// основная программа, вызов функции

**begin**

writeln(newton(0.2, 1.5, eps))

**end**.

Ответ: 0.476723086001294

**Вывод:** в ходе лабораторной работы научились использовать три итерационных метода решения нелинейных уравнений.