**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

**СОСТАВЛЕНИЕ ПРОГРАММ С ИСПОЛЬЗОВАНИЕМ ПРОЦЕДУР И ФУНКЦИЙ**

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

Выполнил студент гр. 2Д93 Э.Ч. Жабанова

(Подпись)

11.05.2020 г.

Отчет принят:

Преподаватель

доцент ОХИ ИШПР, к.т.н. В.А. Чузлов

(Подпись)

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

Томск 2020 г.

**Цель работы:** изучить теорию, выполнить задания, составив программы для решения нелинейных уравнений методом половинного деления, простых итераций и методом Ньютона.

**Теоретическая часть**

**Метод простой итерации** — один из простейших численных методов решения уравнений. Метод основан на принципе сжимающего отображения, который применительно к численным методам в общем виде также может называться методом простой итерации или методом последовательных приближений. В частности, для систем линейных алгебраических уравнений существует аналогичный метод итерации. Идея метода простой итерации состоит в том, чтобы уравнение {\displaystyle f(x)=0} привести к эквивалентному уравнению{\displaystyle x=\varphi (x)} так, чтобы отображение {\displaystyle \varphi (x)} было сжимающим. Если это удаётся, то последовательность итераций {\displaystyle x\_{i+1}=\varphi (x\_{i})} сходится. Такое преобразование можно делать разными способами. {\displaystyle \varphi (x)=x-{\lambda }(x)f(x)\!,}Если {\displaystyle {\lambda }(x)\neq 0} на исследуемом отрезке. Оптимальным выбором является{\displaystyle \lambda (x)={\frac {1}{f'(x)}}}, что приводит к методу Ньютона, который является быстрым, но требует вычисления производной. Если {\displaystyle \lambda (x)} выбрать константу того же знака, что и производная в окрестности корня, то мы получаем простейший метод итерации.

**Метод Ньютона**, **алгоритм Ньютона** (также известный как **метод касательных**) — это итерационный численный метод нахождения корня (нуля) заданной функции. Метод был впервые предложен английским физиком, математиком и астрономом Исааком Ньютоном (1643—1727). Поиск решения осуществляется путём построения последовательных приближений и основан на принципах простой итерации. Метод обладает квадратичной сходимостью. Модификацией метода является метод хорд и касательных. Также метод Ньютона может быть использован для решения задач оптимизации, в которых требуется определить ноль первой производной либо градиента в случае многомерного пространства.

**Деление отрезка пополам** (метод дихотомии) — это численный метод нахождения (одного) решения **x** (с заданной точностью **ε**) нелинейного уравнения вида **f(x)=0**.

Суть метода деления отрезка пополам состоит в разбиении отрезка **[a,b]** (при условии **f(a)f(b)<0**) на два отрезка, определении знака функции **f(x)** в середине отрезка **(a+b)/2** и выборе отрезка, на котором функция меняет знак и содержит решение. Деление отрезка продолжается до достижения необходимой точности решения **ε**. Сначала находим отрезок **[a,b]** такой, что функция **f(x)** непрерывна и меняет знак на отрезке, то есть **f(a)·f(b)<0**. Далее применяем алгоритм решения.

**Практическая часть**

**Задание 1**

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

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

**Программная реализация**

1. Метод деления отрезка пополам:

**program** l91;

**const**

eps = 1e-2;

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

**begin**

result := exp(4 \* ln(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, 2, eps))

**end**.

**Ответ:** 1.9453125

2. Метод простых итераций:

**program** l912;

**const**

eps = 1e-2;

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

**begin**

result := x - (exp(4 \* ln(x)) + 3 \* x - 20) / 35

**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**.

**Ответ:** 1.94045649555801

3. Метод Ньютона:

**program** l913;

**const**

eps = 1e-2;

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

**begin**

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

**end**;

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

**begin**

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

**end**;

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

**begin**

result := 12 \* sqr(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('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:** 1.94047935224908

**Задание 2**

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

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

**Программная реализация**

1. Метод деления отрезка пополам:

**program** l921;

**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**.

**Ответ:** 0.4423828125

2. Метод простых итераций:

**program** l92;

**const**

eps = 1e-3;

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

**begin**

result := x - (exp(x) + x - 2) / 3.7183

**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**.

**Ответ:** 0.442795159460669

3. Метод Ньютона:

**program** l923;

**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('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:** 0.442854401004033

**Задание 3**

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

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

**Программная реализация**

1. Метод деления отрезка пополам:

**program** l931;

**const**

eps = 0.2e-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**.

**Ответ:** 0.567153930664063

2. Метод простых итераций:

**program** l932;

**const**

eps = 0.2e-4;

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

**begin**

result := x-(ln(x)+x)/3

**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**.

**Ответ:** 0.567153930664063

3. Метод Ньютона:

**program** l93;

**const**

eps = 0.2e-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 / sqr(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('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:** 0.567143290399369

**Задание 4**

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

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

**Программная реализация**

1. Метод деления отрезка пополам:

**program** l94;

**const**

eps = 0.5e-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**.

**Ответ:** 0.476718139648438

2. Метод простых итераций:

**program** l942;

**const**

eps = 0.5e-4;

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

**begin**

result := x-(2\*x-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**.

**Ответ:** 0.476721637710739

3. Метод Ньютона:

**program** l943;

**const**

eps = 0.5e-4;

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

**begin**

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

**end**;

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

**begin**

result := x+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('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:** 0.476741447263772

**Вывод**

В ходе работы я изучила теорию, а также составила программы для решения нелинейных уравнений с помощью метода половинного деления, простых итераций и метода Ньютона.