МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Тихоокеанский государственный университет»

Кафедра «Программное обеспечение вычислительной техники и автоматизированных систем»

Решение системы нелинейных уравнений

Лабораторная работа №3

по дисциплине «Вычислительная математика»

Выполнил студент Пшеничный Д. О.

Факультет, группа ФКФН, ПО(аб)-81

Проверил Резак Е.В.

Хабаровск – 2020г.

Задание: Используя метод Ньютона, решить систему нелинейных уравнений с точностью 0.0001.

Вариант 2 

**Метод Ньютона для решения систем нелинейных уравнений**

Формула для нахождения решения системы нелинейных уравнений:

(1)

Где - матрица Якоби.

Так как процесс вычисления обратной матрицы является трудоемким, преобразуем (1) следующим образом:

В результате получена система линейных алгебраических уравнений относительно поправки ε. После ее определения вычисляется следующее приближение .

**Алгоритм метода Ньютона**

1. Задать начальное приближение и точность. Положить *k = 0*.

2. Решить систему линейных алгебраических уравнений относительно поправки ε:

3. Вычислить следующее приближение: .

4. Если норма разности векторов и , равная , меньше либо равна заданной точности, процесс закончить и положить ответ равным . Иначе положить *k = k+1* и перейти к пункту 2.

**Теорема о сходимости метода Ньютона**

Пусть f() определены, непрерывны и имеют непрерывные первые и вторые производные в области Ω.

Пусть:

1) для начальной точки выполняется условие

2) существует , причем

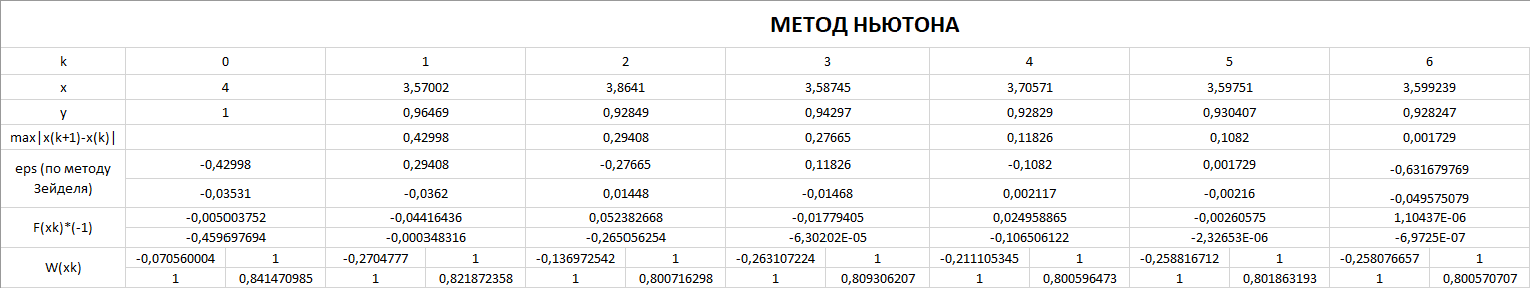
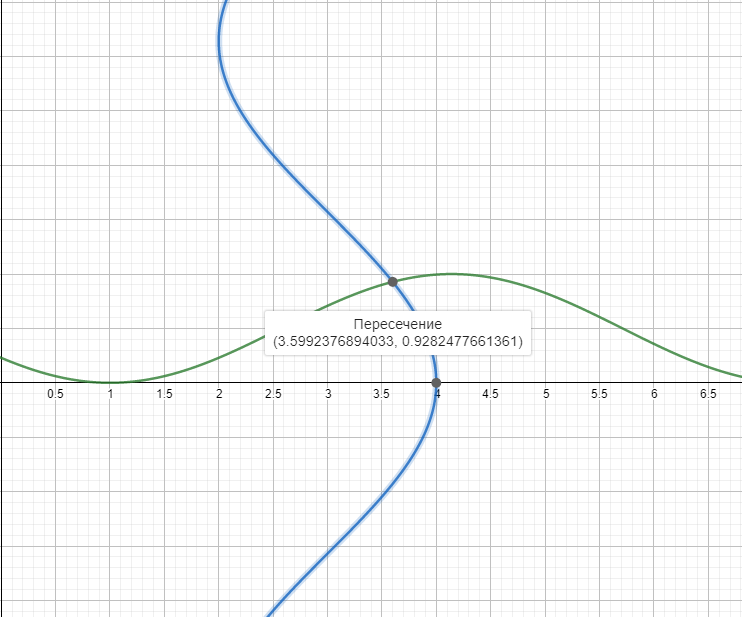
3)

4)

5) удовлетворяют условию

Тогда метод Ньютона сходится к точному решению

**Ручной расчет**

****

**Листинг**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace program

{

public partial class Form1 : Form

{

double accuracy = 0.0001;

public Form1()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

}

private void button2\_Click(object sender, EventArgs e)

{

double[,] W = new double[2, 2];

double[] FVector = new double[2];

double[] roots = new double[2] { 4, 1 };

double[] deltaX = new double[2];

string res = "";

int k = 1;

while(true)

{

FVector = FVectorCount(roots);

W = WCount(roots);

deltaX = Zeidel(new double[2, 2] { { W[1, 0], W[1, 1] }, {W[0, 0], W[0, 1] } }, new double[2] { FVector[1], FVector[0] });

roots[0] += deltaX[0];

roots[1] += deltaX[1];

res += k.ToString() + ") " + roots[0].ToString() + " " + roots[1].ToString() + "\n";

if(Math.Abs(deltaX[0]) < accuracy)

{

MessageBox.Show(deltaX[0].ToString());

MessageBox.Show(res);

break;

}

k++;

}

}

private double[] Zeidel(double [,] A, double[] b)

{

double[,] C = new double[2, 2];

double[] d = new double[2];

for (int i = 0; i < 2; i++)

{

for (int j = 0; j < 2; j++)

{

if (i != j)

C[i, j] = Math.Round(A[i, j] / A[i, i] \* -1, 5);

else

C[i, j] = 0;

}

d[i] = Math.Round(b[i] / A[i, i], 5);

}

//MessageBox.Show(C[0, 0].ToString() + " " + C[0, 1].ToString() + "\n" + C[1, 0].ToString() + " " + C[1, 1].ToString());

double[] x = new double[2] { d[0], d[1] };

double[] X = new double[2] { 0, 0 };

int k = 1;

string result = "";

while (true)

{

X[0] = C[0, 1] \* x[1] + d[0];

X[1] = C[1, 0] \* X[0] + d[1];

result += "Итерация " + k + ":\n X1 = " + Math.Round(X[0], 5).ToString() + "\n X2 = " + Math.Round(X[1], 5).ToString() + "\n";

if (Math.Abs(SumOfVec(X) - SumOfVec(x)) <= accuracy)

{

// MessageBox.Show(result);

// MessageBox.Show("Ответ:\n X1 = " + Math.Round(X[0], 5).ToString() + "\n X2 = " + Math.Round(X[1], 5).ToString());

return X;

}

k++;

x[0] = X[0];

x[1] = X[1];

}

}

private double SumOfVec(double[] vector)

{

double result = 0;

for(int i = 0; i < vector.Length; i++)

{

result += vector[i];

}

return result;

}

private double[] FVectorCount(double[] roots)

{

double[] res = new double[2];

res[0] = -1 \* (Math.Cos(roots[0] - 1) / 2 + roots[1] - 0.5);

res[1] = -1 \* (roots[0] - Math.Cos(roots[1]) - 3);

//MessageBox.Show(res[0].ToString() + "\n" + res[1].ToString());

return res;

}

private double[,] WCount(double[] roots)

{

double[,] res = new double[2, 2];

res[0, 0] = -Math.Sin(roots[0] - 1)/2;

res[0, 1] = 1;

res[1, 0] = 1;

res[1, 1] = Math.Sin(roots[1]);

//MessageBox.Show(res[0, 0].ToString() + " " + res[0, 1].ToString() + "\n" + res[1, 0].ToString() + " " + res[1, 1].ToString());

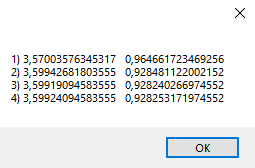
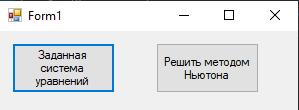
return res;

}

}

}

**Вывод программы**

****

**Вывод**

В ходе данной лабораторной работы был изучен метод Ньютона для решения систем нелинейных уравнений. На основе теоретических данных была написана программа, результаты которой совпали с ручным расчетом.