Министерство образования Республики Беларусь
Учреждение образования «Белорусский государственный университет информатики и радиоэлектроники»

Факультет компьютерных систем и сетей
Кафедра информатики
Дисциплина «Методы численного анализа»

ОТЧЕТ

к лабораторной работе №4

на тему:

«РЕШЕНИЕ НЕЛИНЕЙНЫХ УРАВНЕНИЙ»

БГУИР 1-40 04 01

Выполнил студент группы 253505
Форинов Егор Вячеславович

(дата, подпись студента)

Проверил доцент кафедры
информатики
АНИСИМОВ Владимир Яковлевич

(дата, подпись преподавателя)

Минск 2023

Содержание

Цель работы

- изучить метод простых итераций и метод Ньютона решения нелинейных уравнений;
- составить программу решения нелинейных уравнений указанными методами, применимую для организации вычислений на ЭВМ;
- выполнить тестовые примеры и проверить правильность работы программы

ЗАДАНИЕ. Решить систему нелинейных уравнений:

$$\begin{cases} tg(xy + m) = x \\ ax^2 + 2y^2 = 1, \end{cases}$$ где x>0, y>0,

с точностью до 0,0001 методами простых итераций и Ньютона,

принимая для номера варианта k значения параметров a и m из таблицы:

k	1	2	3	4	5	6	7	8	9	10	11	12	13	14
m	0,0	0,1	0,1	0,2	0,2	0,3	0,3	0,4	0,4	0,1	0,2	0,3	0,2	0,2
a	0,5	0,6	0,7	0,8	0,9	1,0	0,5	0,6	0,7	0,8	0,9	1,0	0,7	0,5

Начальные приближения найти графически. Сравнить скорость сходимости методов.

Вариант 12

Программная реализация

Для проверки решения подставим найденный корень в функцию и найдем ее значение.

Исходные данные

Система, полученная в результате подстановки в ***m*** и ***a***:

-x+tan(x*y+0.3)=0

x^2+2y^2=1

Код метода простых итераций:

```
(double[],int) Iterations(double[] x)
{
    int iterations = 0;
    var eps = 1e-5;
    double error = 1;

    while (error > eps)
    {
        var x1 = x[0];
        var x2 = x[1];
```

```
        x = EvaluateFunctions(x);
        var x11 = x[0];
        var x21 = x[1];
        error = Math.Max(Math.Abs(x1 - x11), Math.Abs(x2 - x21));
        ++iterations;
    }

    return (x, iterations);
}

double[] EvaluateFunctions(double[] x)
{
    x[0] = Math.Tan(x[0] * x[1] + 0.3);
    x[1] = Math.Sqrt((1 - x[0] * x[0]) / 2);
    return x;
}
```

Код метода Ньютона:

```
(double[],int) Newton(double[] x)
{
    int iterations = 0;
    var eps = 1e-5;
    double error = 1;

    while (error > eps)
    {
        var x1 = x[0];
        var x2 = x[1];
        x = EvaluatePhi(x);
        var x11 = x[0];
        var x21 = x[1];
        error = Math.Max(Math.Abs(x1 - x11), Math.Abs(x2 - x21));
        ++iterations;
    }

    return (x, iterations);
}

Func<double, double, double>[,] Jacobi()
{
    var f11 = (double x, double y) => y / Math.Pow(Math.Cos(x * y + 0.3), 2) - 1;
    var f12 = (double x, double y) => x / Math.Pow(Math.Cos(x * y + 0.3), 2);
    var f21 = (double x, double y) => 2 * x;
    var f22 = (double x, double y) => 4 * y;

    return new[,] { { f11, f12 }, { f21, f22 } };
}

Func<double, double, double>[] System()
{
    var f1 = (double x, double y) => Math.Tan(x * y + 0.3) - x;
    var f2 = (double x, double y) => Math.Pow(x, 2) + 2 * Math.Pow(y, 2) - 1;

    return new[] { f1, f2 };
}

double[] EvaluatePhi(double[] x)
{
    var system = System();
    var jacobi = Jacobi();
    var xVec = Vector<double>.Build.DenseOfArray(x);
    var bSource = new double[2,2];

    for (var i = 0; i < 2; i++)
        for (var j = 0; j < 2; j++)
            bSource[i, j] = jacobi[i, j](x[0], x[1]);
```

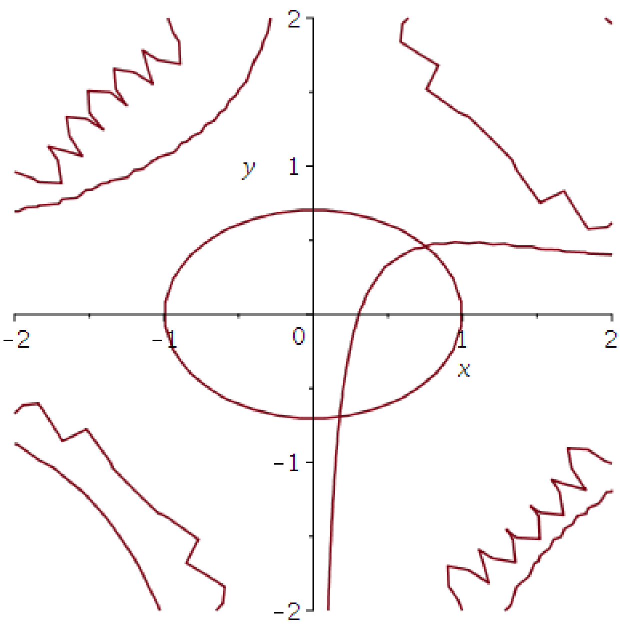
```
    var b = Matrix<double>.Build.DenseOfArray(bSource);
    var a = -1 * b.Inverse();

    var fSource = new double[2];
    for (var i = 0; i < 2; i++)
        fSource[i] = system[i](x[0], x[1]);
    var f = Vector<double>.Build.DenseOfArray(fSource);

    return (xVec + a.Multiply(f)).ToArray();
}
```

Полученные результаты

С помощью пакета Maple построим график системы нелинейных уравнений, чтобы найти начальные приближения:



Начальные приближения:

x=0.5, y=0.2;

Метод простых итераций:

```
0,7592495575974431
0,46018480488173824
Iterations: 9
```

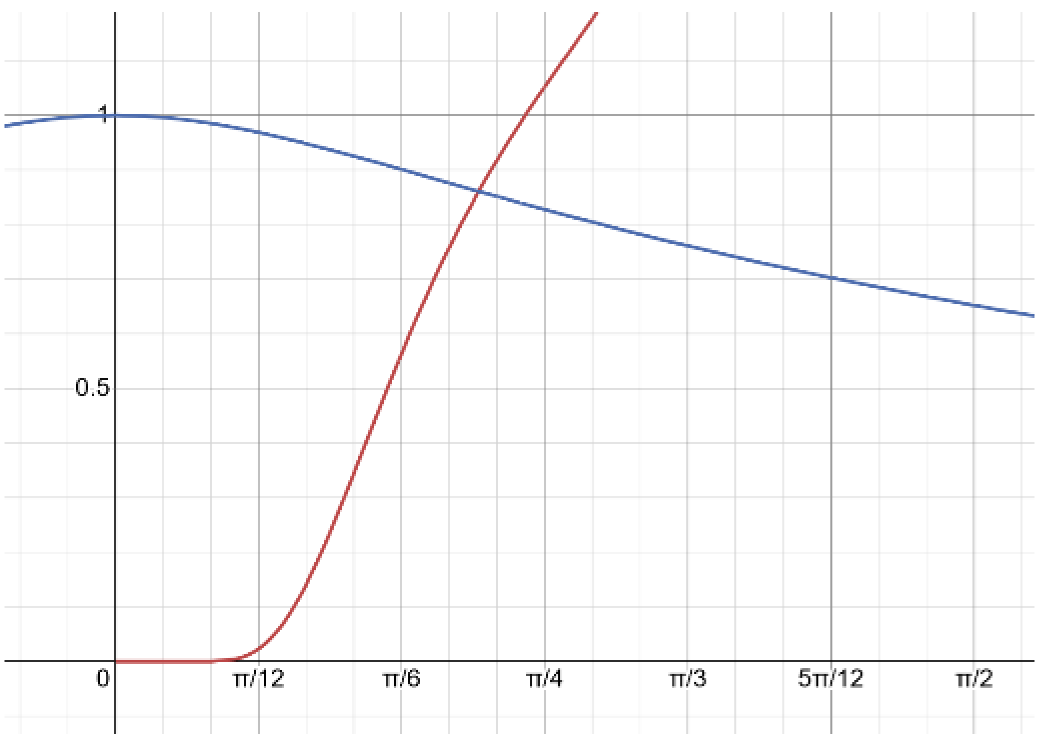
Метод Ньютона:

```
0,759250204970354
0,46018427083749586
Iterations: 6
```

Тестовый пример 1.

```
Исходная система уравнений:
-x + sin(x*y)**x = 0
-y + cos(x*y)**y = 0
Приближение:
[0.6, 0.8]
Метод простых итераций:
Кол-во итераций 7
Вектор решений:
Matrix([[0.665316078894085], [0.860954006618983]])
Значение f1 в точке: -9.85763470140455e-6
Значение f2 в точке: -2.94765634878402e-6
Метод Ньютона:
Кол-во итераций: 2
Вектор решений:
Matrix([[0.665304672854928], [0.860955196779156]])
Значение f1 в точке: 6.29988283762373e-11
Значение f2 в точке: -6.07131012131390e-11
```

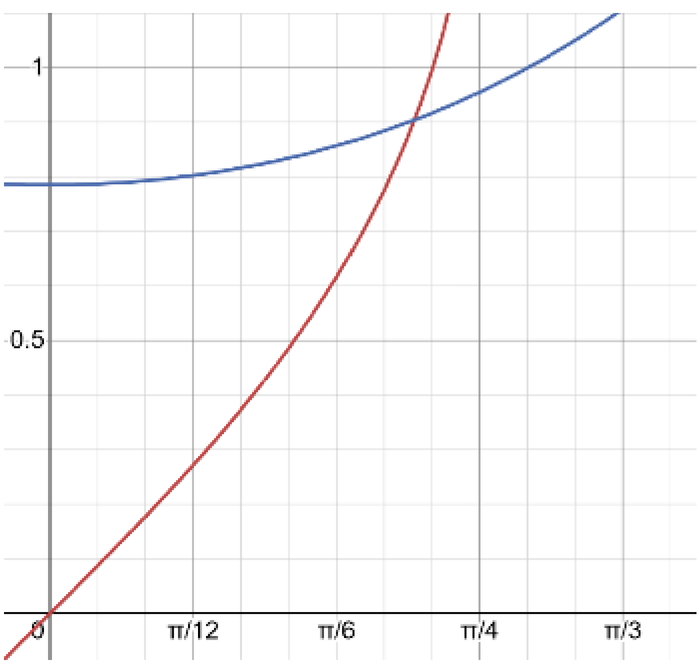
График функций:



Тестовый пример 2.

```
Исходная система уравнений:
-sin(y) + tan(x) = 0
-cos(x) + cot(y) = 0
Приближение:
[0.7, 0.9]
Метод простых итераций:
Кол-во итераций 7
Вектор решений:
Matrix([[0.666255074360238], [0.904554823195875]])
Значение f1 в точке: 2.65894007531742e-5
Значение f2 в точке: 1.30184282509660e-5
Метод Ньютона:
Кол-во итераций: 2
Вектор решений:
Matrix([[0.666239432492970], [0.904556894302416]])
Значение f1 в точке: 7.13540337926588e-13
Значение f2 в точке: 2.24043006369357e-13
```

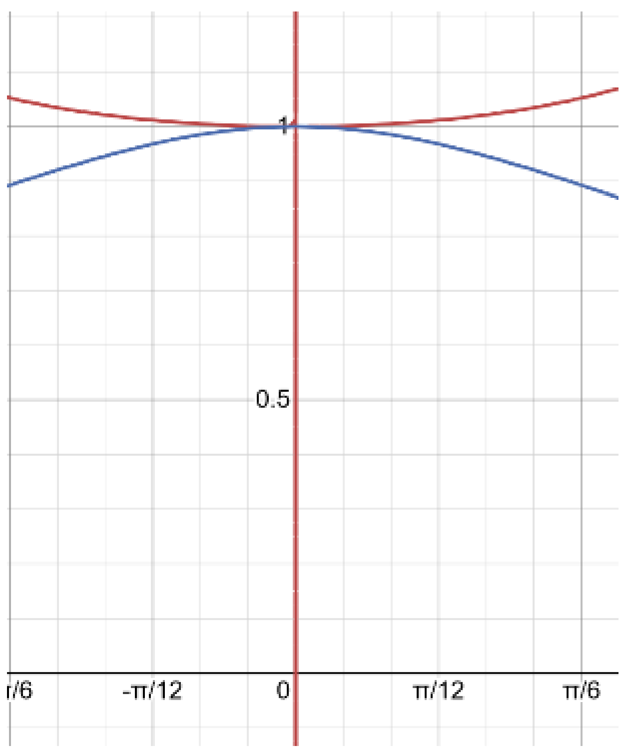
График функций:



Тестовый пример 3.

```
Исходная система уравнений:
-x + sin(x*y) = 0
-y + cos(x*y) = 0
Приближение:
[0.1, 0.9]
Метод простых итераций:
Кол-во итераций 174
Вектор решений:
Matrix([[0.0529522903669825], [0.998597043328735]])
Значение f1 в точке: -9.89282035142106e-5
Значение f2 в точке: 5.24092061415793e-6
Метод Ньютона:
Кол-во итераций: 15
Вектор решений:
Matrix([[0.000115903951080631], [0.999999995458787]])
Значение f1 в точке: -7.85848125396844e-13
Значение f2 в точке: -2.17565021554122e-9
```

График функций:



Вывод

В ходе выполнения лабораторной работы я изучил метод простых итераций и метод Ньютона решения нелинейных уравнений, написал программу их реализации на языке C#, правильность работы программы проверил на тестовых примерах.

На основании тестов можно сделать следующие выводы:

- Программа позволяет получить решения системы с заданной точностью (заданная точность в условиях лабораторной работы 10^-4);
- Метод Ньютона эффективнее по сравнению с методом простых итераций, так как затрачивает меньшее число итераций;
- Оптимальным способом решения нелинейных уравнений является применение метода Ньютона, так скорость сходимости в этом методе почти всегда квадратичная.
- Основной недостаток методов – малая область сходимости ($\bar{x}_0$ должен быть достаточно близок к решению уравнения).