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) |

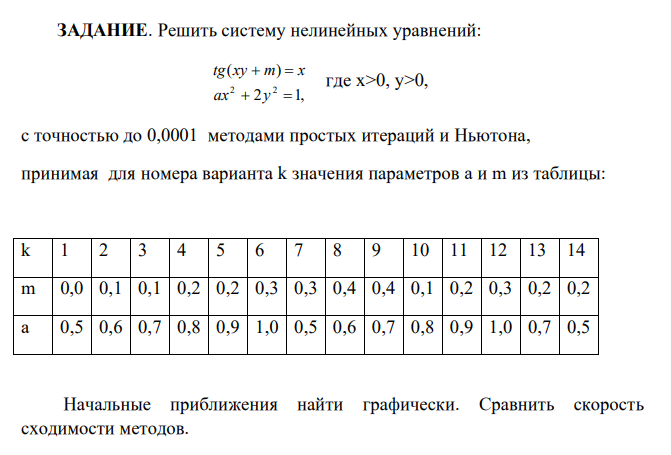
Минск 2023

**Содержание**

1. Цель работы
2. Задание
3. Программная реализация
4. Полученные результаты
5. Оценка полученных результатов
6. Вывод

**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



**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;  
}

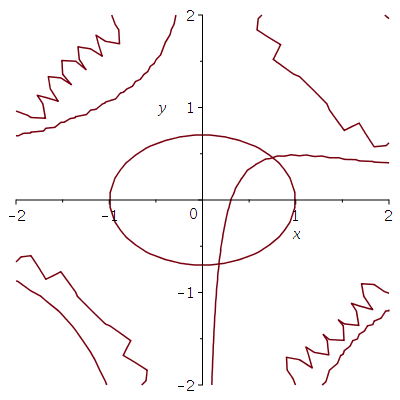
Код метода Ньютона:

(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**

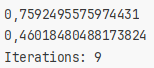
С помощью пакета Maple построим график системы нелинейных уравнений, чтобы найти начальные приближения:



*Начальные приближения:*

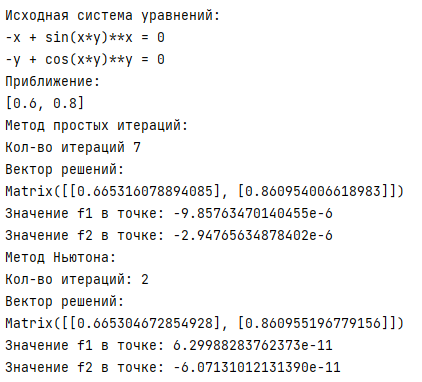
*x=0.5, y=0.2;*

*Метод простых итераций:*

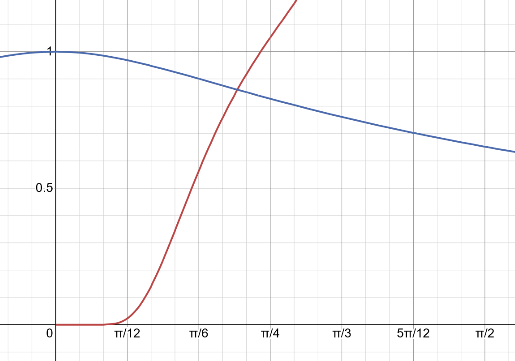
*Метод Ньютона:*

**

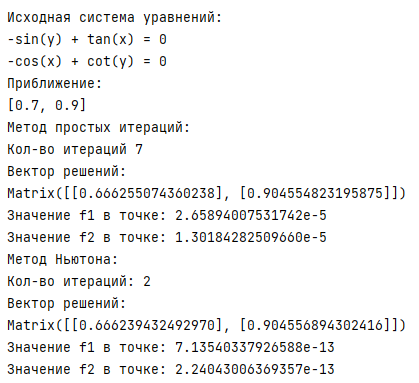
*Тестовый пример 1.*



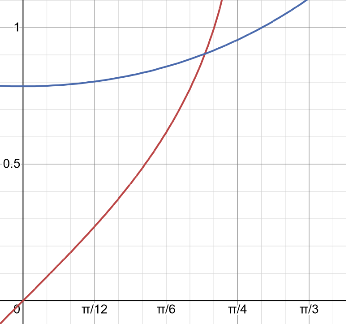
*График функций:*



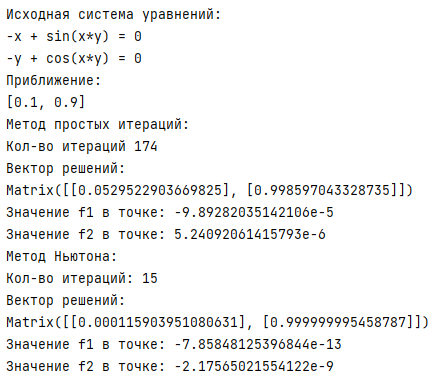
*Тестовый пример 2.*



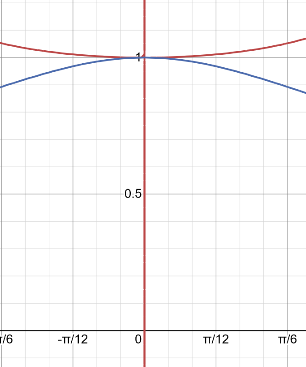
*График функций:*



*Тестовый пример 3.*



*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 С#, 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 (о должен быть достаточно близок к решению уравнения).