Министерство образования Республики Беларусь
Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей
Кафедра информатики
Дисциплина: Методы численного анализа

ОТЧЁТ

к лабораторной работе
на тему

Решение систем нелинейных уравнений

Выполнил: студент группы 153503
Кончик Денис Сергеевич

Проверил: Анисимов Владимир Яковлевич

Минск 2022

Содержание

1. ЦЕЛЬ РАБОТЫ

1) Изучить методы численного решения систем нелинейных уравнений (метод простой итерации, метод Ньютона).
2) Составить алгоритм и программу численного решения нелинейных уравнений методами простой итерации и Ньютона.
3) Проверить правильность работы программы на тестовых примерах.
4) Численно решить систему нелинейных уравнений заданного варианта.
5) Сравнить число итераций (трудоемкость), необходимых для достижения заданной точности вычисления разными методами.

2. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Краткие теоретические сведения. Пусть дана система нелинейных уравнений (система не линейна, если хотя бы одно из входящих в нее уравнений не линейно):

$$\begin{cases} f_1(x_1,\ldots,x_n)=0 \\ f_2(x_1,\ldots,x_n)=0 \\ \ldots\ldots\ldots\ldots\ldots \\ f_n(x_1,\ldots,x_n)=0 \end{cases}$$

Мы можем записать систему в более компактной векторной форме

$$\boldsymbol{f(x)=0}, \qquad (4.1)$$

где $\boldsymbol{f} = (f, \ldots, f)$, $\boldsymbol{x}=(x, \ldots, x)^T$.

Для решения системы (4.1) иногда можно применить метод последовательного исключения неизвестных, который приводит решение системы к решению одного уравнения с одним неизвестным. Однако в подавляющем большинстве случаев систему уравнений (4.1) решают итерационными методами. Для решения системы (4.1) существует набор методов, из которых рассмотрим простейшие методы: метод простых итераций, базирующийся на принципе сжимающих отображений и метод Ньютона (многомерный аналог метода касательных).

Метод простых итераций. Чтобы воспользоваться методом простых итераций, необходимо предварительно привести систему к следующему виду:

$$\begin{cases} x_1 = \varphi_1(x_1,...,x_n) \\ x_2 = \varphi_2(x_1,...,x_n) \\ .....\\ x_n = \varphi_{n1}(x_1,...,x_n) \end{cases}$$

Или в векторной форме

$$\bar{x} = \varphi(\bar{x}) \qquad (4.2)$$

где $\vec{\varphi} = (\varphi_1, \varphi_2, ..., \varphi_n).$

Исходя из некоторого начального приближения $\bar{x}^0$, построим итерационную последовательность точек

$\bar{x}^k = \varphi(\bar{x}^{k-1})$, $k=1,2,....$

Пусть точка $\bar{x}^0$ есть некоторое начальное приближение к решению. Рассмотрим δ-окрестность $U_\delta(\bar{x})$ этой точки. В силу принципа сжимающих отображений итерационная последовательность

$\bar{x}^k = \varphi(\bar{x}^{k-1})$, $k=1,2,....$

будет сходиться, если существует число q<1 такое, что

выполнено условие:

$$\left\| \varphi(\bar{x}^1) - \varphi(\bar{x}^2) \right\| \le q \left\| \bar{x}^1 - \bar{x}^2 \right\|, \quad \forall \bar{x}^1, \bar{x}^2 \in U_\delta(\bar{x}^0), \qquad (4.3)$$

называемое условием сжатия для отображения φ. В частности, это условие всегда выполняется, если векторная функция φ непрерывно дифференцируема и норма матрицы производных функции φ удовлетворяет неравенству:

$$\left\| \frac{\partial \varphi}{\partial x}(\bar{x}) \right\| \le q$$

во всех точках *x* из δ-окрестности $U_\delta(\bar{x})$ точки $\bar{x}^0$.

Следующая теорема дает условие сходимости и оценку скорости сходимости метода простых итераций.

Теорема. Пусть отображение φ является сжатием в $U_\delta(\overline{x^0})$ *и пусть*

$$\left\|\varphi(\bar{x}^0) - \bar{x}^0\right\| < \delta(1-q).$$

Тогда итерационная последовательность:

$$\bar{x}^k = \varphi(\bar{x}^{k-1}),$$

с начальной точкой $\bar{x}^0$*, сходится к решению* $\bar{x}^*$ *системы (1). При этом справедлива следующая оценка погрешности:*

$$\left\|\bar{x}^k - \bar{x}^*\right\| \le \frac{q^k}{1-q}\left\|\varphi(\bar{x}^0) - \bar{x}^0\right\|.$$

Отметим, что начальное приближение $\bar{x}^0$ выбирают экспериментально. (Например, на основе грубого графического решения системы, если порядок системы не высок. По точкам строят график первого уравнения, потом второго и ищут приблизительно точку их пересечения).

Метод Ньютона.

Пусть дана система нелинейных уравнений :

$$\begin{cases} f_1(x_1,\ldots,x_n) = 0 \\ f_2(x_1,\ldots,x_n) = 0 \\ \ldots\ldots\ldots\ldots\ldots \\ f_n(x_1,\ldots,x_n) = 0 \end{cases}$$

Запишем ее в векторной форме:

$$f(\bar{x}) = 0 \qquad (4.1)$$

Найдем начальное приближение $\bar{x}^0$.

Будем предполагать, что векторная функция f непрерывна дифференцируема в некоторой окрестности начального приближения. Вместо системы (4.1) будем искать решение соответствующей ей линеаризованной системы

$$f(\bar{x}^0) + \frac{\partial f}{\partial x}(\bar{x}^0)(\bar{x} - \bar{x}^0) = 0 \Rightarrow \quad f(\bar{x}^0) + J(\bar{x}^0)(\bar{x} - \bar{x}^0) = 0,$$

где через $J(\bar{x}^0)$ обозначена для удобства записи матрица производных векторной функции f в точке $\bar{x}^0$ (матрица Якоби системы (4.1) в этой точке).

При этом при применении метода Ньютона предполагается, что $\det J(\bar{x}^0) \neq 0$ в окрестности точки $\bar{x}^0$.

Тогда из линеаризованной системы, которая линейна относительно переменных x, можно найти первое приближение

$$\bar{x}^k = \bar{x}^{k-1} - J^{-1}(\bar{x}^{k-1}) f(\bar{x}^{k-1}), k = 1,...$$

Рассматривая линеаризованную систему в точках при $k=1,2,....$, найдем к-ое приближение

$$\bar{x}^k = \bar{x}^{k-1} - J^{-1}(\bar{x}^{k-1}) f(\bar{x}^{k-1}), k = 1,...$$

Построенная таким способом рекуррентная последовательность Ньютона сходится при определенных дополнительных условиях к решению системы (4.1). Легко видеть, что рассматриваемый метод совпадает с методом касательных в случае $n=1$, т.е. является многомерным вариантом метода касательных.

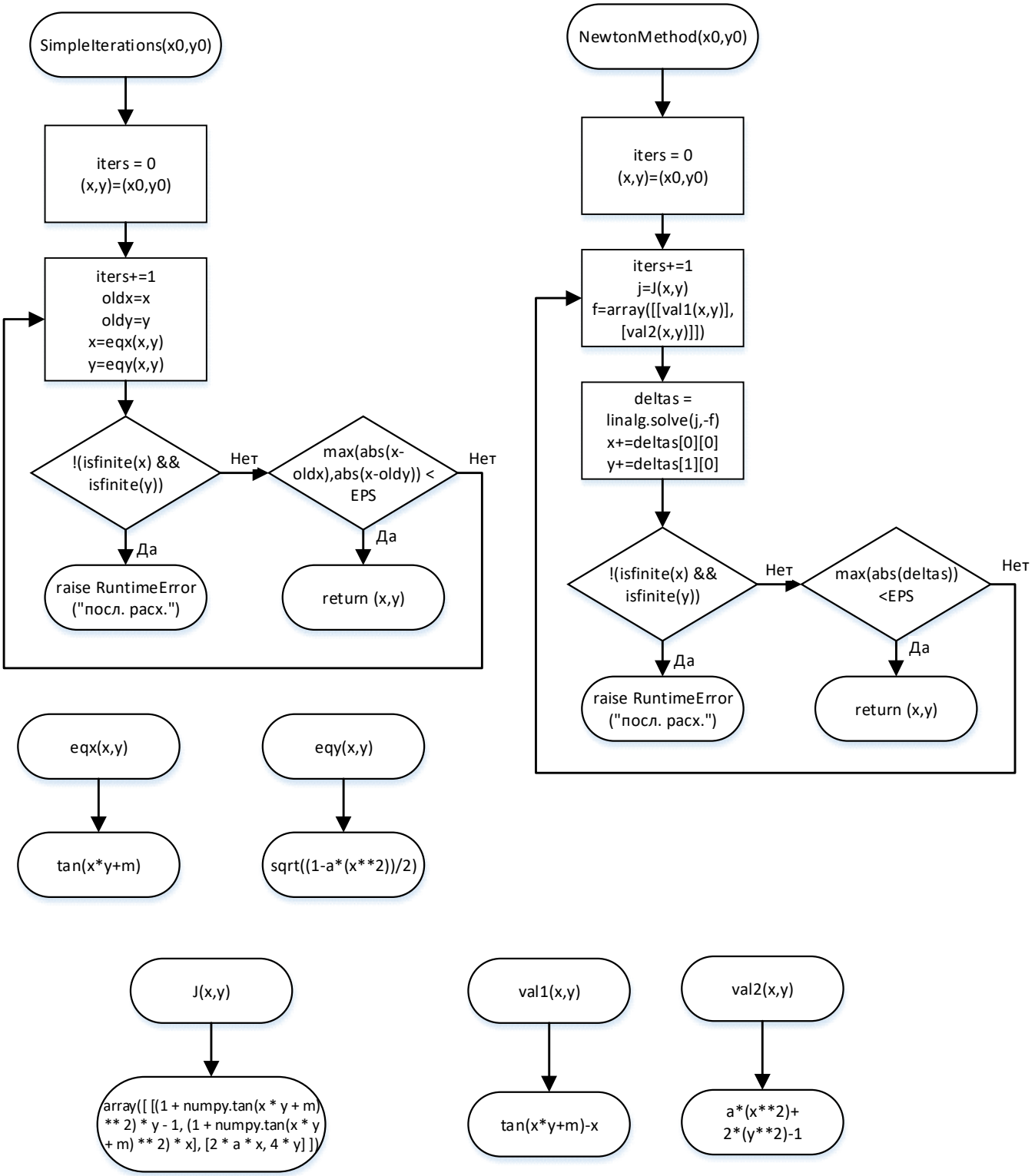
На практике обратную матрицу не считают, а на каждом шаге решают линеаризованную систему:

$$f(\bar{x}^{k-1}) + J(\bar{x}^{k-1})(\bar{x} - \bar{x}^{k-1}) = 0 \Rightarrow \bar{x} = \bar{x}^k$$

Теорема. При сделанных выше предположениях, последовательность Ньютона сходится к решению системы (4.1), если начальное приближение выбрано достаточно близко к решению.

Отметим в заключение, что метод Ньютона сходится достаточно быстро (скорость сходимости квадратичная), если начальное приближение выбрано удачно. На практике итерационный процесс заканчивают, когда норма разности двух последовательных приближений меньше заданной точности вычисления решения.

3. АЛГОРИТМ РЕШЕНИЯ



4. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ

```
import numpy
import sympy


m = 0.2
a = 0.9
iters = 0
EPS = 10.0 ** -4

# Исходные уравнения в формате f(x,y) = 0
(x, y) = sympy.symbols("x y")
eq1 = sympy.tan(x * y + m) - x
eq2 = a * (x ** 2) + 2 * (y ** 2) - 1

print("Система нелинейных уравнений:")
print(eq1, "= 0")
print(eq2, "= 0")
print()

# Вычисление исходных уравнений: f1(x,y), f2(x,y)
def val1(x, y):
    return numpy.tan(x * y + m) - x
def val2(x, y):
    return a * (x ** 2) + 2 * (y ** 2) - 1

# Вычисление из исходных уравнений: x = phi(x,y), y = phi(x,y)
def eqx(x, y):
    return numpy.tan(x * y + m)
def eqy(x, y):
    return numpy.sqrt((1 - a * (x ** 2)) / 2)

# Вычисление матрицы Якоби
def J(x, y):
    return numpy.array([
        [(1 + numpy.tan(x * y + m) ** 2) * y - 1, (1 + numpy.tan(x * y + m)
** 2) * x],
        [2 * a * x, 4 * y]
    ])


# Графики исходных уравнений
plots = sympy.plot_implicit(sympy.Eq(eq1, 0), (x, -2, 2), (y, -2, 2),
line_color = "blue", show = False)
plots.extend(sympy.plot_implicit(sympy.Eq(eq2, 0), (x, -2, 2), (y, -2, 2),
line_color = "red", show = False))
```

```
# Метод простых итераций
def SimpleIterations(x0, y0):
    global iters
    iters = 0
    (x, y) = (x0, y0)
    while True:
        iters += 1
        oldx = x
        oldy = y
        x = eqx(x, y)
        y = eqy(x, y)
        if (not (numpy.isfinite(x) and numpy.isfinite(y))):
            raise RuntimeError("Последовательность {x} расходящаяся")
        if (max(abs(x - oldx), abs(y - oldy)) < EPS):
            return (x, y)

# Метод Ньютона
def NewtonMethod(x0, y0):
    global iters
    iters = 0
    (x, y) = (x0, y0)
    while True:
        iters += 1
        j = J(x, y)
        f = numpy.array([[val1(x, y)], [val2(x, y)]])
        deltas = numpy.linalg.solve(j, -f)
        x += deltas[0][0]
        y += deltas[1][0]
        if (not (numpy.isfinite(x) and numpy.isfinite(y))):
            raise RuntimeError("Последовательность {x} расходящаяся")
        if (max(abs(deltas)) < EPS):
            return (x, y)


x0 = 0.9
y0 = 0.5
print("Начальное приближение: \n(x0, y0) = ", (x0, y0))
print()

def solute(method):
    global iters
    try:
        (x, y) = method(x0, y0)
        print(f"{method.__name__} (итераций: {iters})")
        print(f"(x, y) = ({x:.4f}, {y:.4f})")
    except Exception as ex:
        print(f"Ошибка: {ex} - в {method.__name__} (итераций: {iters})")
    print()

SimpleIterations.__name__ = "Метод простых итераций"
NewtonMethod.__name__ = "Метод Ньютона"
solute(SimpleIterations)
solute(NewtonMethod)

plots.show()
```

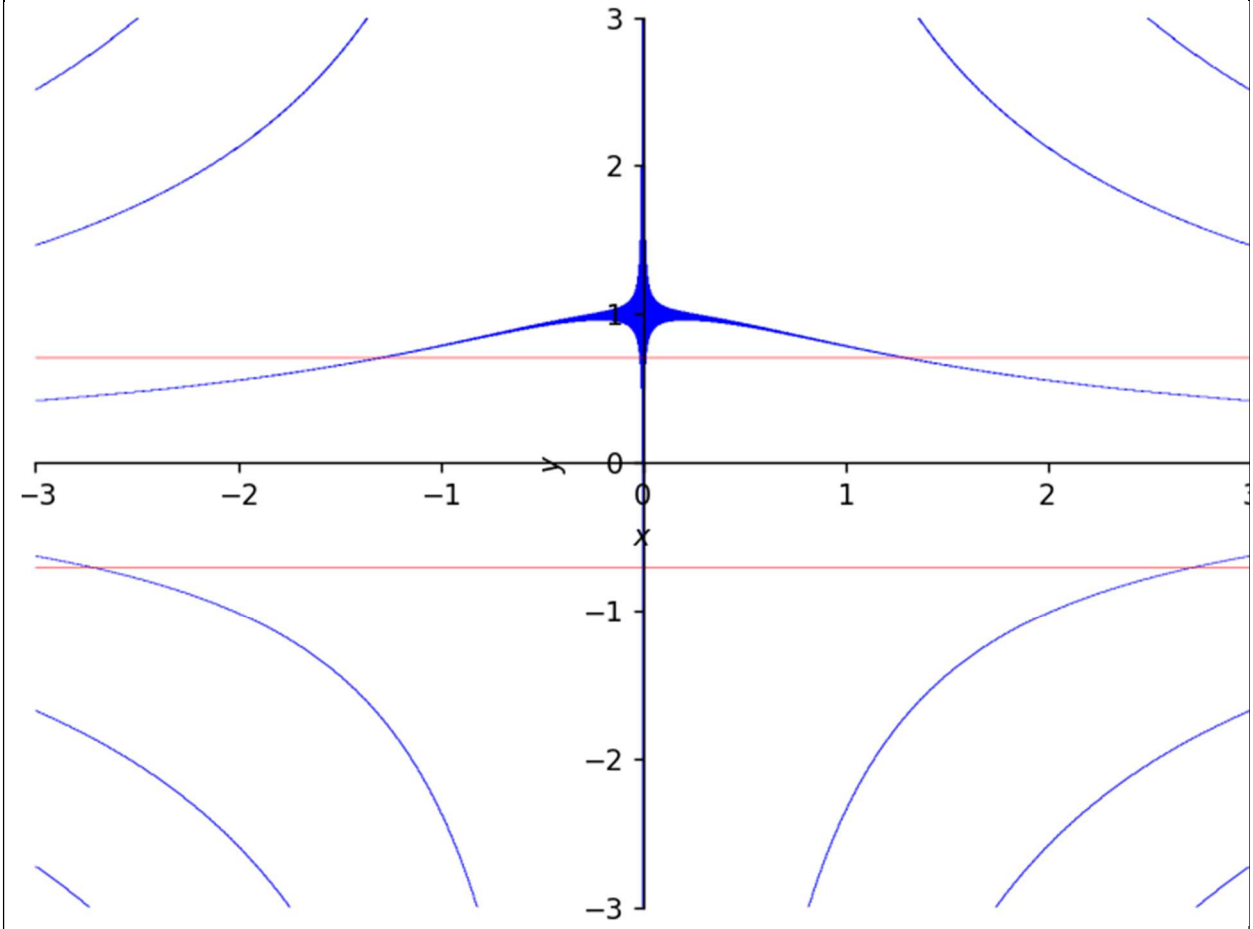
5. ТЕСТОВЫЕ ПРИМЕРЫ

Тестовый пример 1

Решить систему нелинейных уравнений с точностью до 0,0001 методами простых итераций и Ньютона:

$$\begin{cases} tg(xy) = x, \\ 2y^2 = 1; \end{cases}$$

Ответ:



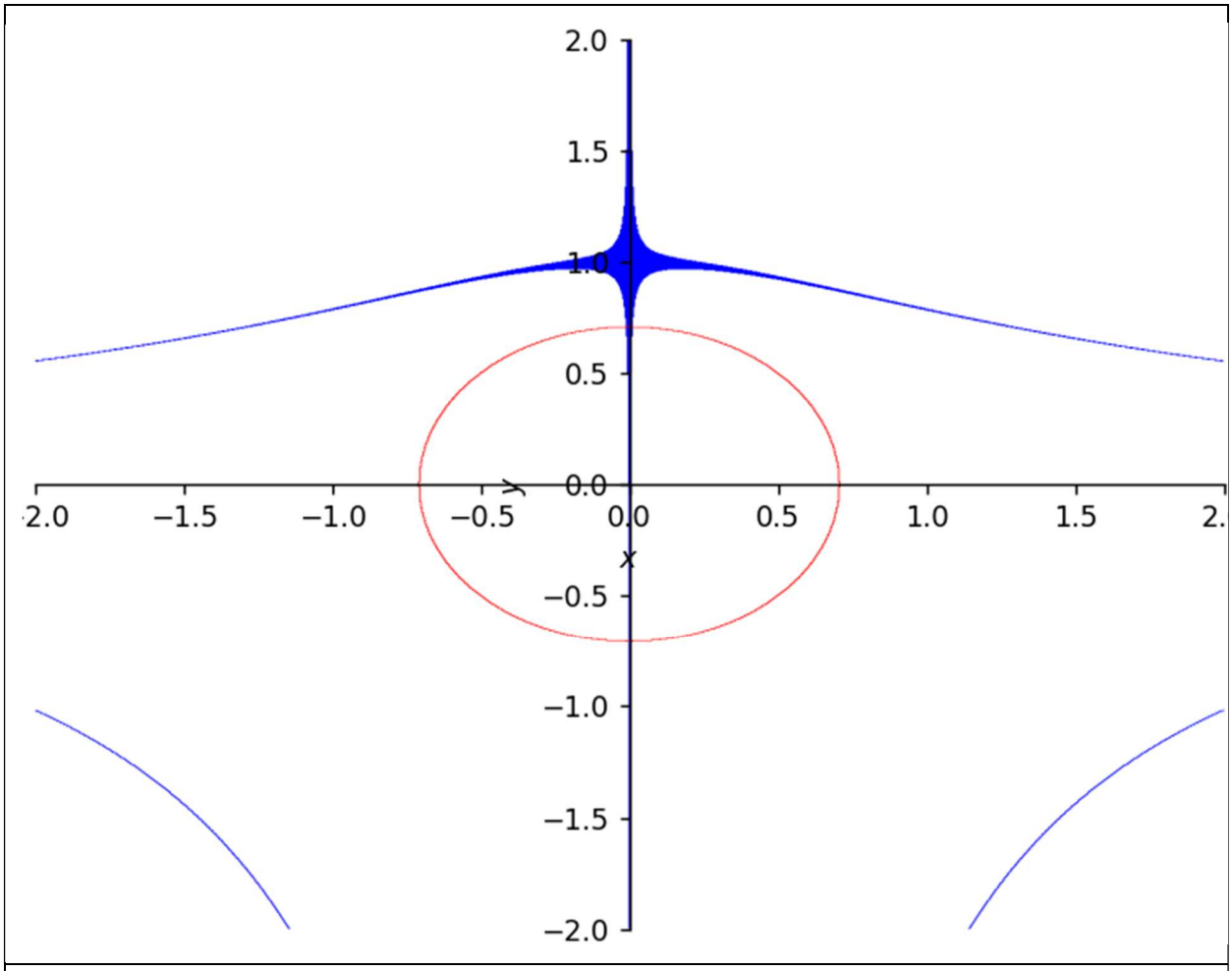
Начальное приближение: $\begin{cases} x = 0.1 \\ y = 0.9 \end{cases}$	
Метод простых итераций	Метод Ньютона
$\begin{cases} x = 0.0002 \\ y = 0.7071 \end{cases}$	$\begin{cases} x = 0.0002 \\ y = 0.7071 \end{cases}$
Количество итераций	
19	4

Тестовый пример 2

Решить систему нелинейных уравнений с точностью до 0,0001 методами простых итераций и Ньютона:

$$\begin{cases} tg(xy) = x, \\ 2x^2 + 2y^2 = 1; \end{cases}$$

Ответ:

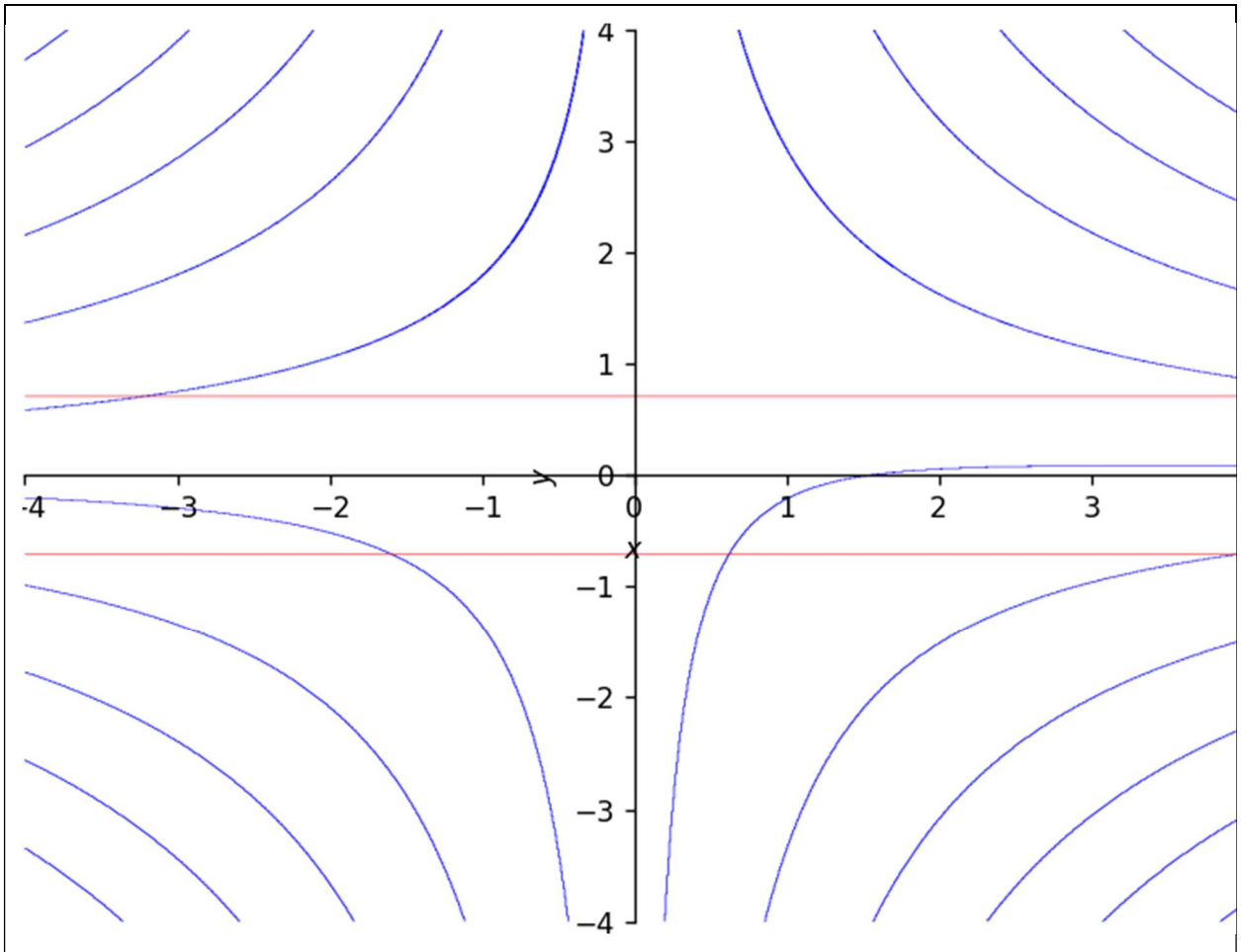
<table>
<tr><td colspan="2"> 
</td></tr>
<tr><td colspan="2">Начальное приближение: $\begin{cases} x = 0.1 \\ y = 0.9 \end{cases}$</td></tr>
<tr><td>Метод простых итераций</td><td>Метод Ньютона</td></tr>
<tr><td>$\begin{cases} x = 0.0002 \\ y = 0.7071 \end{cases}$</td><td>$\begin{cases} x = -0.0000 \\ y = 0.7071 \end{cases}$</td></tr>
<tr><td colspan="2">Количество итераций</td></tr>
<tr><td>19</td><td>5</td></tr>
</table>

Тестовый пример 3

Решить систему нелинейных уравнений с точностью до 0,0001 методами простых итераций и Ньютона:

$$\begin{cases} tg(xy+1) = x, \\ 2y^2 = 1; \end{cases}$$

Ответ:



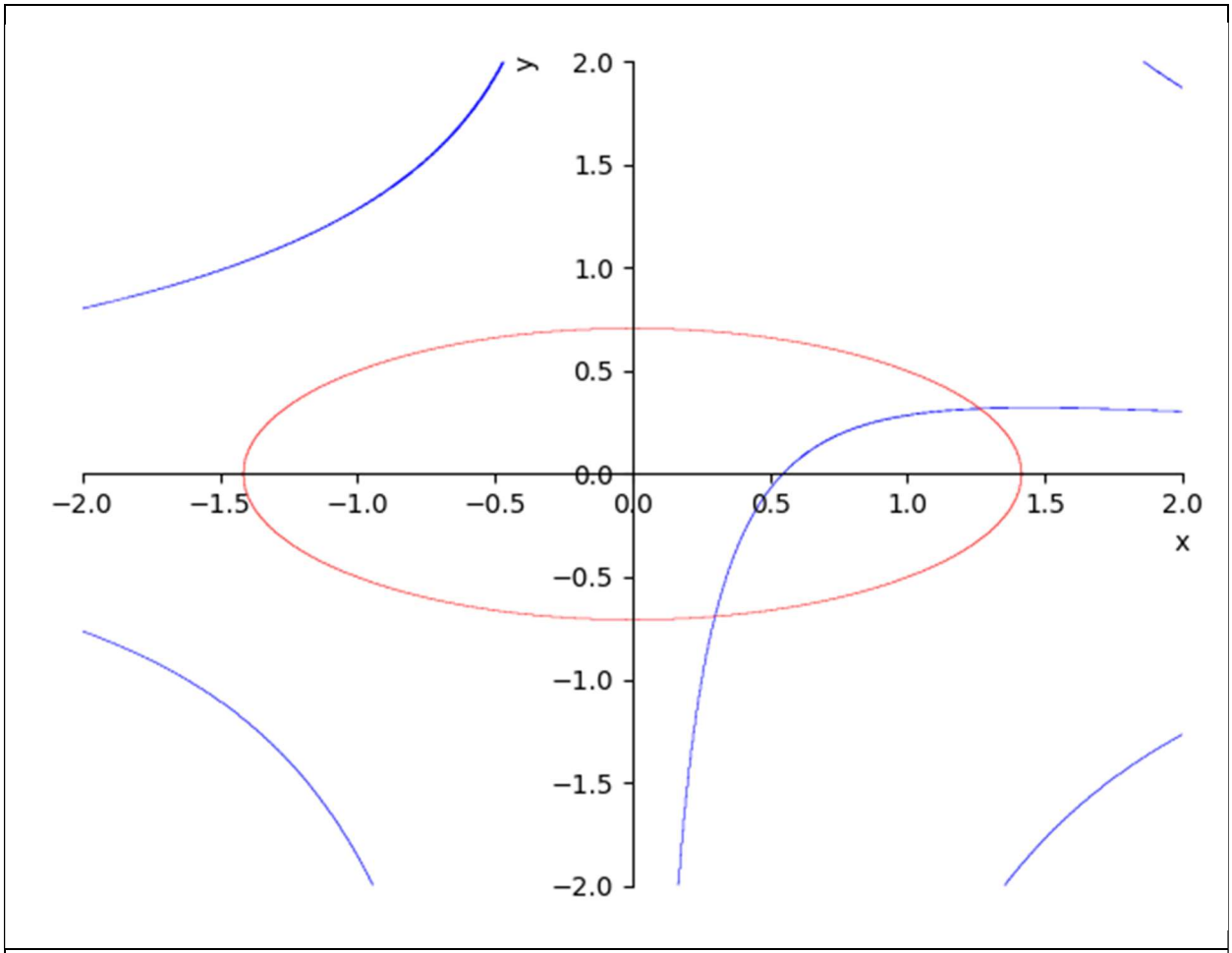
Начальное приближение: $\begin{cases} x = 0.5 \\ y = -1 \end{cases}$	
Метод простых итераций	Метод Ньютона
$\begin{cases} x = -3.2083 \\ y = 0.7071 \end{cases}$	$\begin{cases} x = 0.6246 \\ y = -0.7071 \end{cases}$
Количество итераций	
423561	4

Тестовый пример 4

Решить систему нелинейных уравнений с точностью до 0,0001 методами простых итераций и Ньютона:

$$\begin{cases} tg(xy+3) = x, \\ 3x^2 + 2y^2 = 1; \end{cases}$$

Ответ:

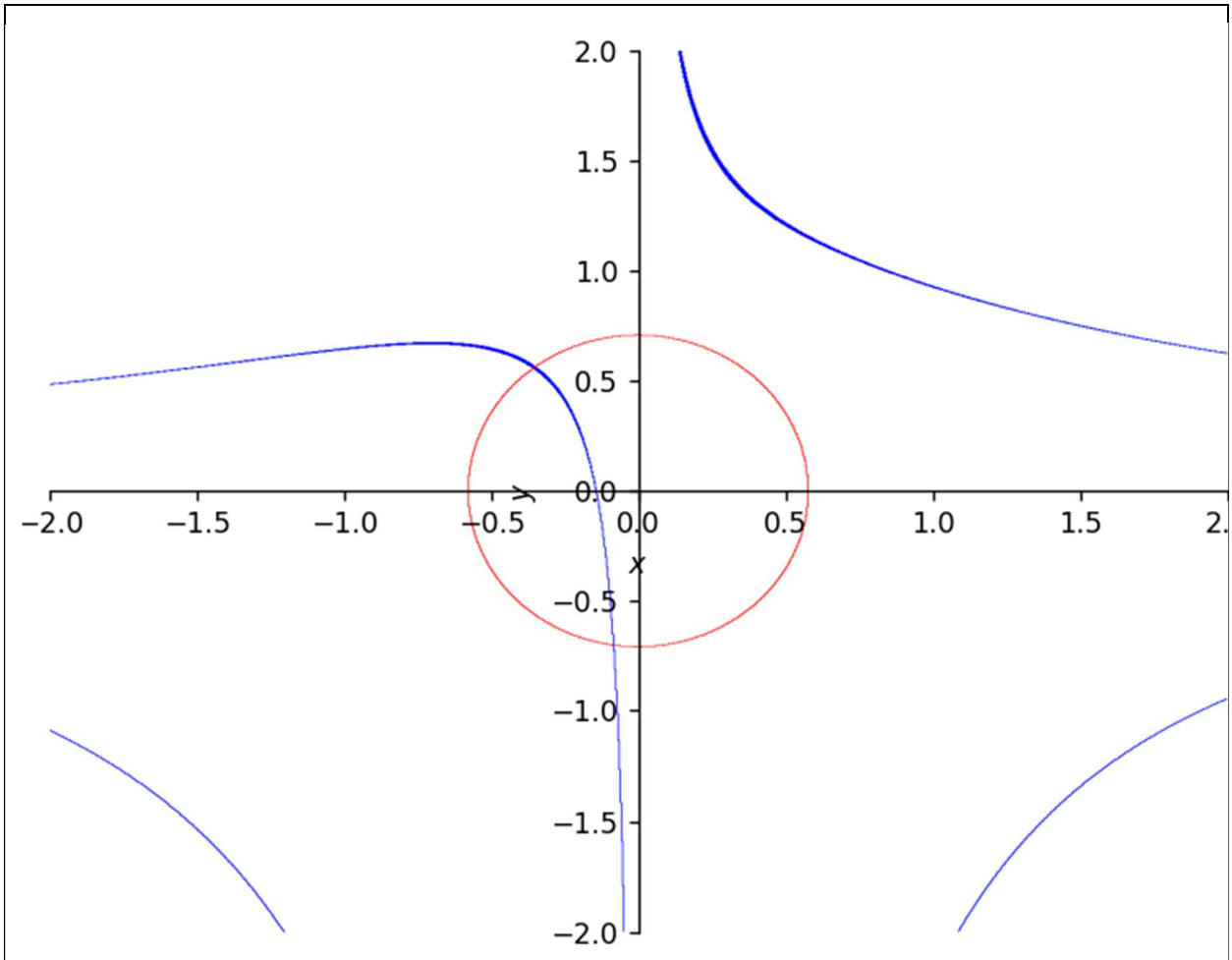
<table>
<tr><td colspan="2"> 
</td></tr>
<tr><td colspan="2">Начальное приближение: $\begin{cases} x = -0.5 \\ y = 0.5 \end{cases}$</td></tr>
<tr><td>Метод простых итераций</td><td>Метод Ньютона</td></tr>
<tr><td>$\begin{cases} x = -0.3527 \\ y = 0.5598 \end{cases}$</td><td>$\begin{cases} x = -0.3526 \\ y = 0.5599 \end{cases}$</td></tr>
<tr><td colspan="2">Количество итераций</td></tr>
<tr><td>6</td><td>4</td></tr>
</table>

Тестовый пример 5

Решить систему нелинейных уравнений с точностью до 0,0001 методами простых итераций и Ньютона:

$$\begin{cases} tg(xy+3) = x, \\ 3x^2 + 2y^2 = 1; \end{cases}$$

Ответ:

<table>
<tr><td colspan="2"> 
</td></tr>
<tr><td colspan="2">Начальное приближение: $\begin{cases} x = 5 \\ y = 5 \end{cases}$</td></tr>
<tr><td>Метод простых итераций</td><td>Метод Ньютона</td></tr>
<tr><td>$\begin{cases} x = -0.3526 \\ y = 0.5599 \end{cases}$</td><td>$\begin{cases} x = -0.0834 \\ y = -0.6997 \end{cases}$</td></tr>
<tr><td colspan="2">Количество итераций</td></tr>
<tr><td>6</td><td>1337</td></tr>
</table>

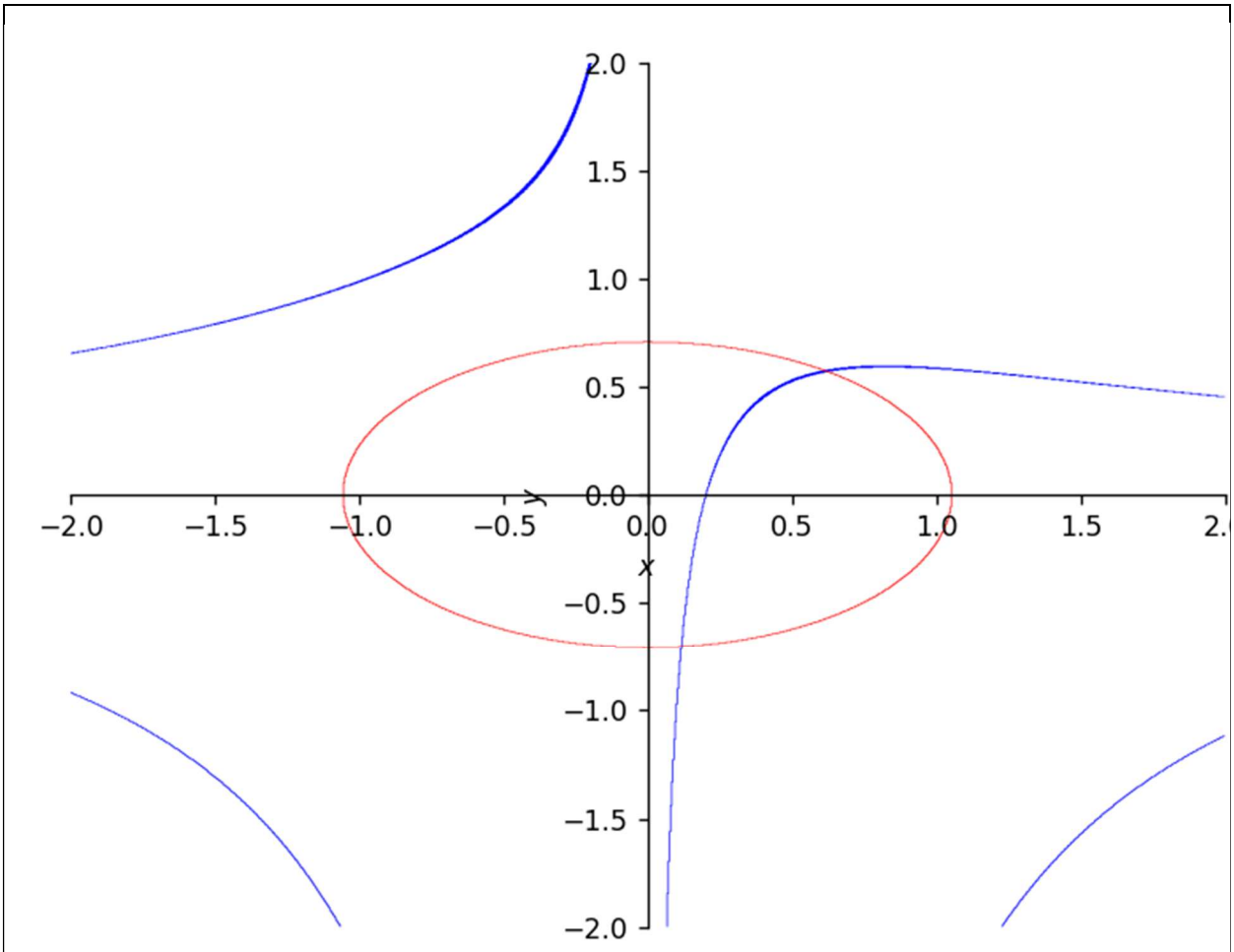
6. ЗАДАНИЕ

Вариант 11

Решить систему нелинейных уравнений с точностью до 0,0001 методами простых итераций и Ньютона:

$$\begin{cases} tg(xy + 0.2) = x, \\ 0.9x^2 + 2y^2 = 1; \end{cases} \text{ где } x > 0, y > 0$$

Ответ:



Начальное приближение: $\begin{cases} x = 0.5 \\ y = 0.5 \end{cases}$	
Метод простых итераций	Метод Ньютона
$\begin{cases} x = 0.6190 \\ y = 0.5724 \end{cases}$	$\begin{cases} x = 0.6190 \\ y = 0.5723 \end{cases}$
Количество итераций	
10	4

7. ВЫВОД

Таким образом, в ходе выполнения лабораторной работы были изучены методы численного решения систем нелинейных уравнений (метод простой итерации, метод Ньютона), составлен алгоритм и программа численного решения систем нелинейных уравнений методами простой итерации и Ньютона, проверена правильность работы программы на тестовых примерах, численно решена система уравнений заданного варианта.

Исходя из полученных результатов, можно сделать вывод о большей трудоемкости метода простых итераций (тестовые примеры 1,2,3,4). Метод простых итераций обладает линейной скоростью сходимости. Метод Ньютона сходится достаточно быстро (квадратичная скорость сходимости), если начальное приближение выбрано удачно (в примере 5 выбрано неудачно).