МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра АПУ

ОТЧЕТ
по лабораторной работе №2
по дисциплине «Теория автоматического управления»
Тема: Метод фазового пространства

Студент гр. 7392	____________________	И.В. Батура
Студент гр. 7392	____________________	Е.О. Амельченко
Студентка гр. 7392	____________________	А.Г. Цуркан
Преподаватель	____________________	А.М. Синица

Санкт-Петербург
2020

СОДЕРЖАНИЕ

1 ЦЕЛЬ РАБОТЫ

Знакомство с фазовыми портретами. Изучение инструментальных средств построения фазовых портретов с помощью библиотек языка программирование Python

2 ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ ПОЛОЖЕНИЯ

Метод фазовой плоскости дает возможность изобразить качественную картину всей совокупности свободных движений (процессов) для выбранной области начальных условий (состояний), а при необходимости — провести точные исследования интересующих типов движений. Через каждую точку фазового пространства при условии однозначности функции проходит только одна фазовая траектория. Единственность нарушается в особых точках, соответствующих точкам равновесия. Существует несколько основных типов особых точек (типов поведения в окрестности положения равновесия):

- устойчивый узел
- неустойчивый узел
- седло
- устойчивый фокус
- неустойчивый фокус
- узел

Заметим также, что особые точки являются частным случаем аттракторов — компактных подмножеств фазового пространства динамической системы, все траектории из некоторой окрестности которого стремятся к нему при времени, стремящемся к бесконечности. Кроме особых точек (точек равновесия) аттрактором могут быть замкнутые траектории (предельные циклы) или некоторая ограниченная область с неустойчивыми траекториями внутри (как у странного аттрактора).

3 ЗАДАЧА НА ЛАБОРАТОРНУЮ РАБОТУ

Изучение фазовых пространств, научиться успользовать инструметарий языка Python для решения дифф. уравнений. Изучить особые точки:

- устойчивый узел

- неустойчивый узел
- седло
- устойчивый фокус
- неустойчивый фокус
- узел

Решить следующие задачи:

- Построение фазовых портретов особых точек
- Построени фазовых портретов для нелинейных элементов
- Построение фазового портрета для маятникаустойчивый фокус
- Построение фазовго портрета осциллятора Ван дер Поля
- Построение фазового портрета аттрактора Лоренца

Решаем задачи исходя из данного нам варианта 1.

4 ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ

4.1 Изучение фазовых портретов особых точек

4.1.1 Система 1

Дана система следующего вида:

$$\begin{cases} x' = -4x + 0.1x^2 - 4y \\ y' = 1.5x + y - 0.2y^3 \end{cases}$$

Зададим данную системе в языке Python:

```
x, y = y
dydt = [(-4 * x) + (0.1 * variant * x * x) - (4 * y),
        (1.5 * x) + y - (0.2 * variant * y * y * y)]
return dydt
```

Построим ее фазовый портрет (1)

Линеаризируем систему, которая описана выше, и после данной операции мы получаем следующую систему:

$$\begin{cases} x' = -4x - 4y \\ y' = 1.5x + y \end{cases}$$

Зададим данную системе в языке Python:

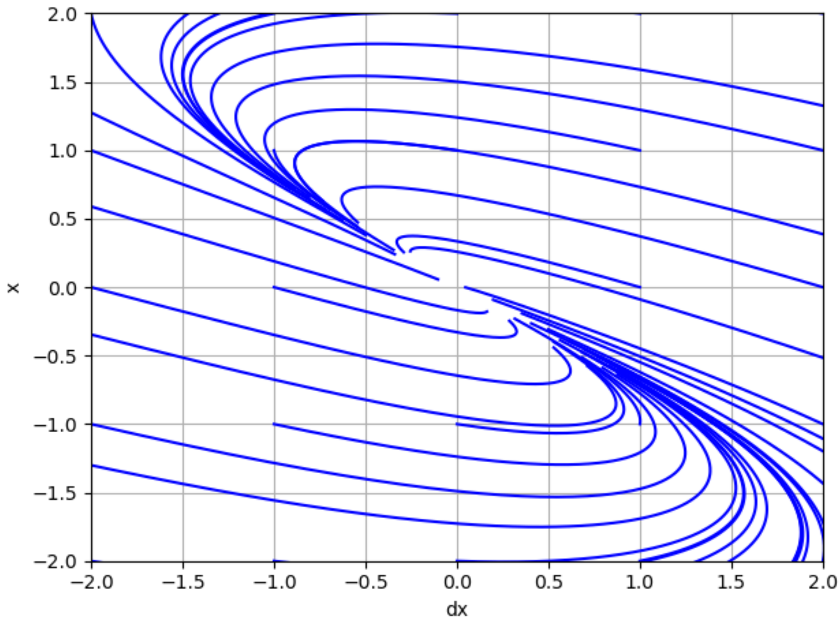

Рисунок 1 — Фазовый портрет системы №1

```
def ode(Y, t):
    x, y = Y
    dydt = [-4 * x - 4 * y, 1.5 * x + y]
    return dydt
```

Построим фазовый портрет линеаризованной системы (2)

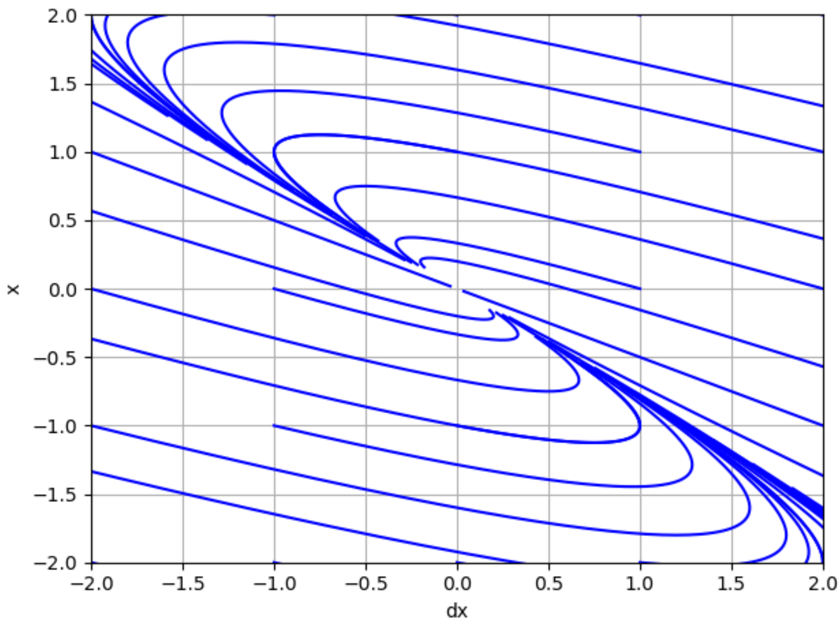

Рисунок 2 — Фазовый портрет линеаризованной системы №1

Полученный фазовый портрет соответсвует системе 2-го порядка. Собственные корни данной системы не равны друг другу и орицательны, то такое положение равновесия называется устойчивым узлом.

4.1.2 Система 2

Дана система следующего вида:

$$\begin{cases} x' = x + 0.5y - 0.1y^2 \\ y' = 0.5x - 0.2x^2 + y \end{cases}$$

Зададим данную системе в языке Python:

```
def ode(y, t, b, c) :
        x, y = y
        dydt = [x + (0.5 * y) - (0.1 * variant * y * y),
                (0.5 * x) - (0.2 * variant * x * x) + y]
        return dydt
```

Построим ее фазовый портрет (3)

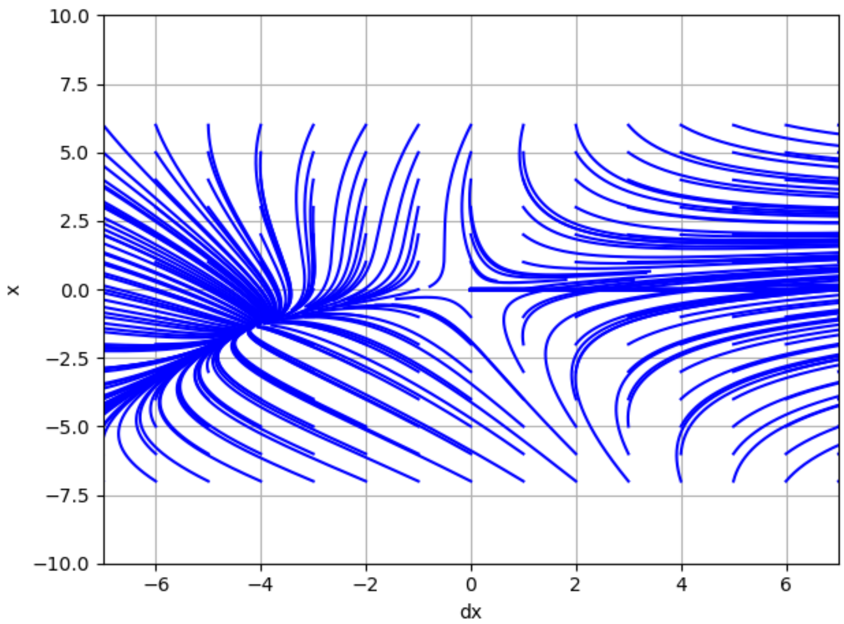

Рисунок 3 — Фазовый портрет системы №2

Линеаризируем систему, которая описана выше, и после данной операции мы получаем следующую систему:

$$\begin{cases} x' = x + 0.5y \\ y' = 0.5x + y \end{cases}$$

Зададим данную системе в языке Python:

```
def ode(Y, t):
        x, y = Y
        dydt = [x + 0.5 * y, 0.5 * x + y]
        return dydt
```

Построим фазовый портрет линеаризованной системы, смотри рисунок (4)

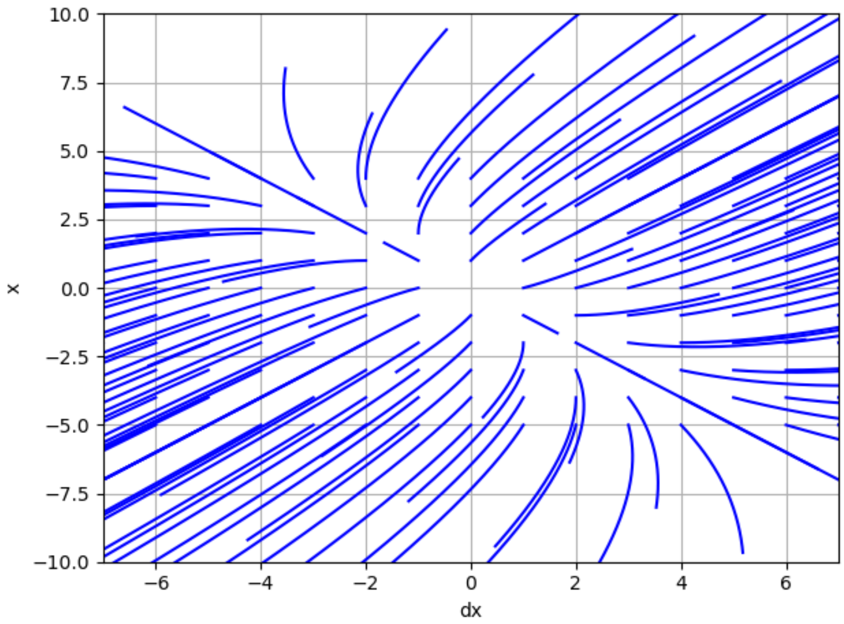

Рисунок 4 — Фазовый портрет линеаризованной системы №2

Данные фазовый портрет соотвествует системе 2-ого порядка, где собственные корни не равны друг другу и положительны. В этом случае точка называется неустойчивым узлом.

4.1.3 Система 3

Дана система следующего вида:

$$\begin{cases} x' = 2x + 0.2x^2 + y - 0.1y^2 \\ y' = x - 3y \end{cases}$$

Зададим данную системе в языке Python:

```
x, y = y
dydt = [(2*x) + (0.2*variant*x*x)+y-(0.1 * variant * y * y),
        x - 3 * y]
return dydt
```

Построим ее фазовый портрет (5)

Линеаризируем систему, которая описана выше, и после данной операции мы получаем следующую систему:

$$\begin{cases} x' = 2x + y \\ y' = x - 3y \end{cases}$$

Зададим данную системе в языке Python:

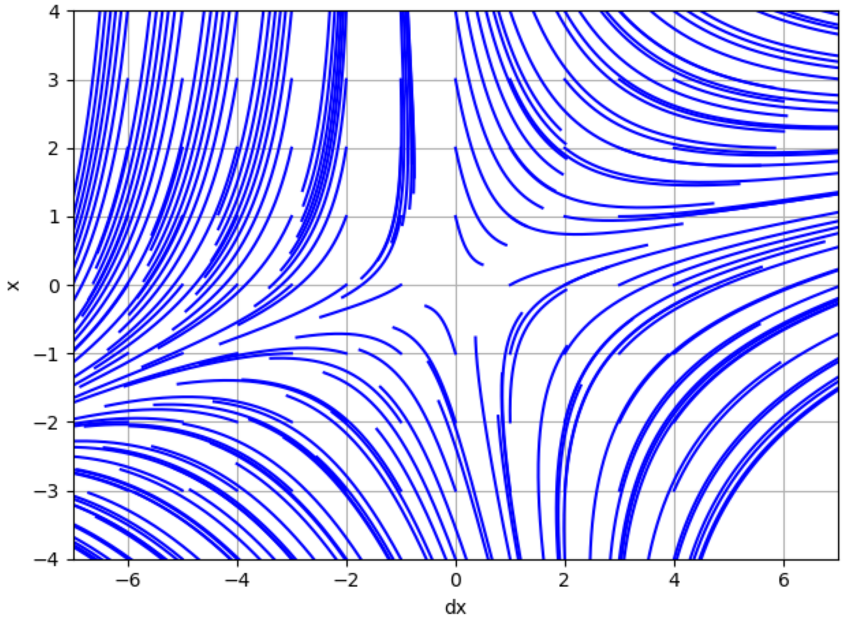

Рисунок 5 — Фазовый портрет системы №3

```
def ode(y, t) :
    x, y = y
    dydt = [(2 * x) + y, x - 3 * y]
    return dydt
```

Построим фазовый портрет линеаризованной системы, смотри рисунок (6)

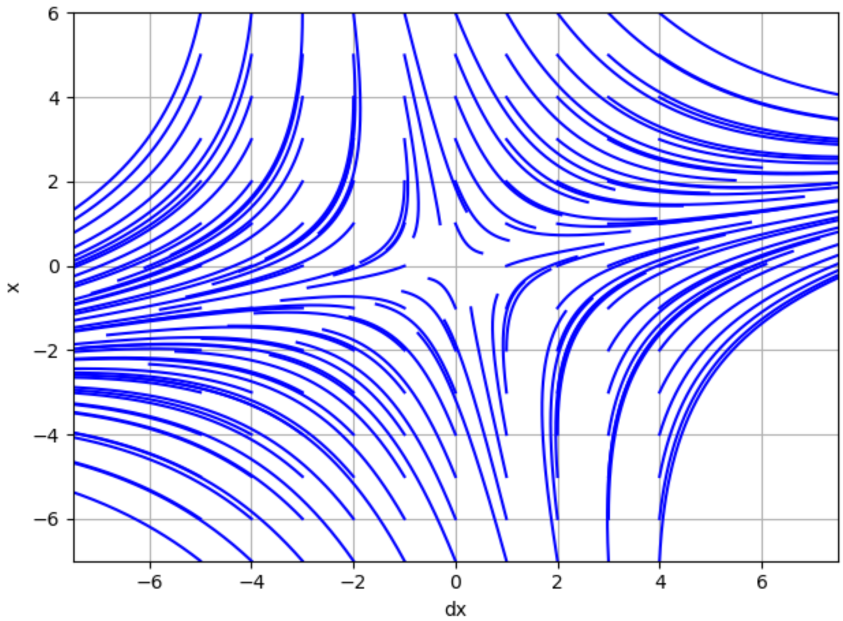

Рисунок 6 — Фазовый портрет линеаризованной системы №3

Как мы видим из рисунка 5 и 6, получился фазовый портрет характерный для системы второго порядка, с одним положительным корнем и одним отрицательным. Такое положение равновесия называется седлом. Даннам система, очевидно, неусточива, так как имеется положителный корень.

4.1.4 Система 4

Дана система следующего вида:

$$\begin{cases} x' = -0.1x^2 + 2y \\ y' = -3x - y \end{cases}$$

Зададим данную системе в языке Python:

```
    x, y = y
    dydt = [-0.1 * variant * x * x + 2 * y , -3 * x -y]
    return dydt
```

Построим ее фазовый портрет (7)

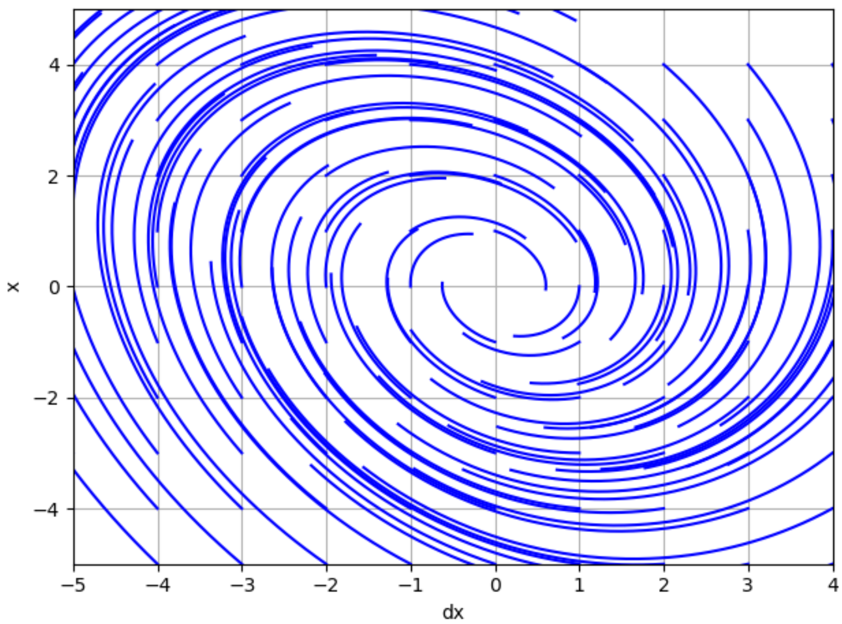

Рисунок 7 — Фазовый портрет системы №4

Линеаризируем систему, которая описана выше, и после данной операции мы получаем следующую систему:

$$\begin{cases} x' = 2y \\ y' = -3x - y \end{cases}$$

Зададим данную системе в языке Python:

```
def ode(y, t) :
    x, y = y
```

```
dydt = [2 * y , -3 * x - y]
return dydt
```

Построим фазовый портрет линеаризованной системы, смотри рисунок 8

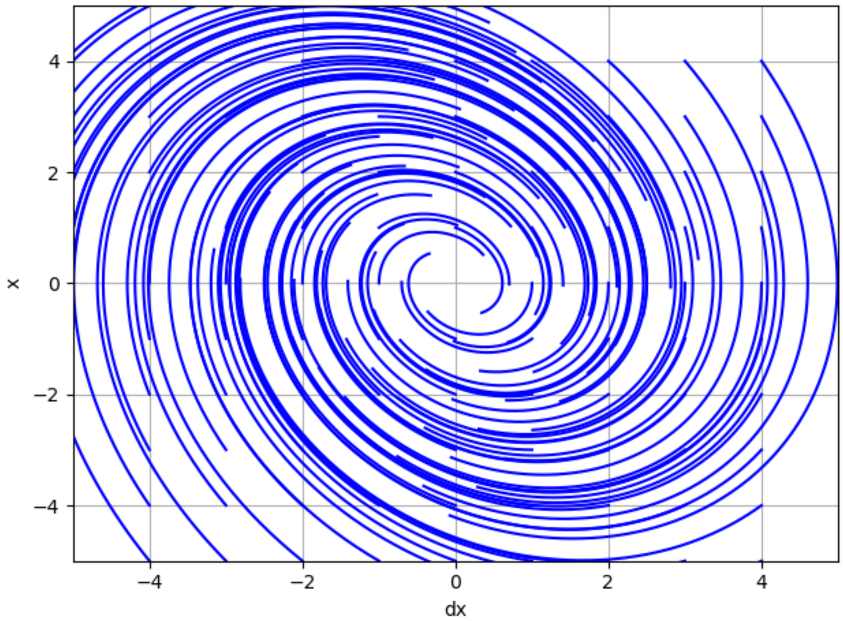

Рисунок 8 — Фазовый портрет линеаризованной системы №4

Фазовый портрет полученный на рисунке 7 и 8 характерен для систем второго порядка, где собственные числа матрицы состояний являются комплексными, действительные части которых не равны нулю. А комплексные корни будт представлены в виде комплексно-сопряженных чисел. Так же мы видим что корни находятся в левой полуплоскости, что свидетельствует о том что система устойчива. Такое положение равновесия называется устойчивым фокусом.

4.1.5 Система 5

Дана система следующего вида:

$$\begin{cases} x' = 0.1x - 4y \\ y' = 4x - 0.2x^2 + 0.1y \end{cases}$$

Зададим данную системе в языке Python:

```
x, y = y
dydt = [0.1 * x - 4 * y, 4 * x - 0.2 * variant * x * x
return dydt
```

Построим ее фазовый портрет (9)

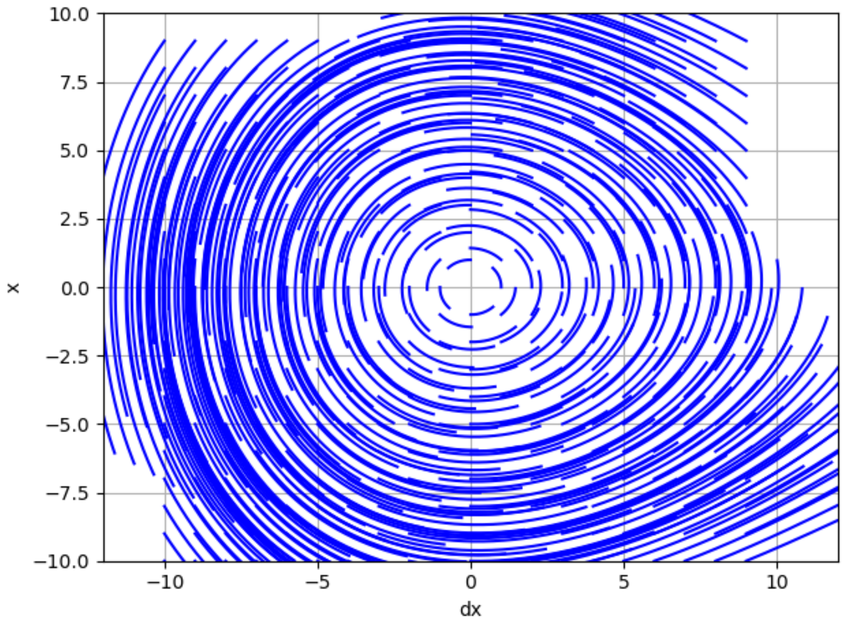

Рисунок 9 — Фазовый портрет системы №5

Линеаризируем систему, которая описана выше, и после данной операции мы получаем следующую систему:

$$\begin{cases} x' = 0.1x - 4y \\ y' = 4x + 0.1y \end{cases}$$

Зададим данную системе в языке Python:

```
def ode(y, t) :
    x, y = y
    dydt = [0.1 * x - 4 * y, 4 * x + 0.1 * y]
    return dydt
```

Построим фазовый портрет линеаризованной системы, смотри рисунок 10

Из построенного фазового портрета(см. рисунки 10, 10), можно заметить что он характерен система второго порядка, где корни являются комплексно-сопряженными и лежат в правой полуплоскости. Такое положение равновесия называется неустойчивым фокусом.

4.1.6 Система 6

Дана система следующего вида:

$$\begin{cases} x' = x - 0.1x^2 - 4y + 0.3y^2 \\ y' = 2x + 0.2x^2 - y - 0.3y^3 \end{cases}$$

Зададим данную системе в языке Python:

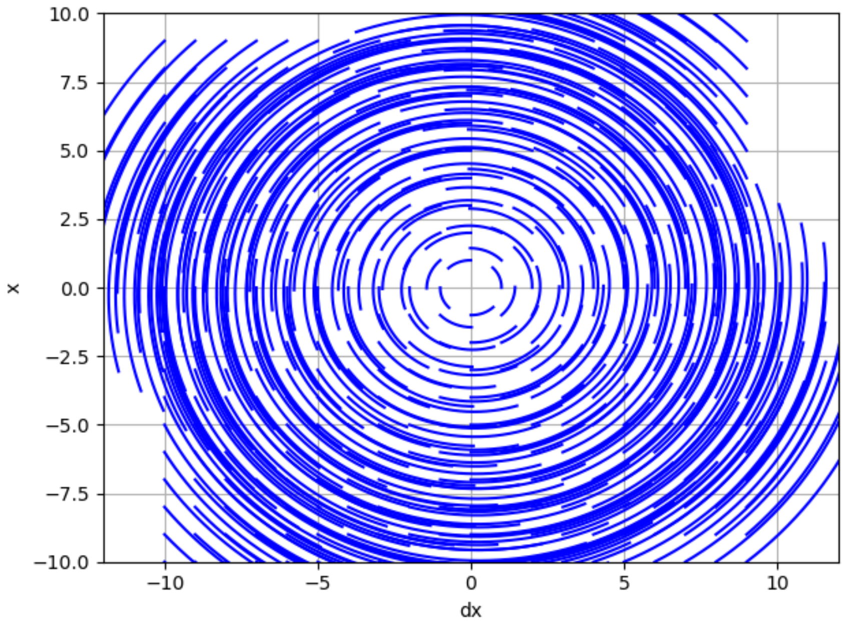

Рисунок 10 — Фазовый портрет линеаризованной системы №5

```
x, y = Y
dydt = [x - 0.1 * variant * x * x - 4 * y + 0.3 * varia
        2*x + 0.2*variant*x*x - y - 0.3*variant * y * y
return dydt
```

Построим ее фазовый портрет (11)

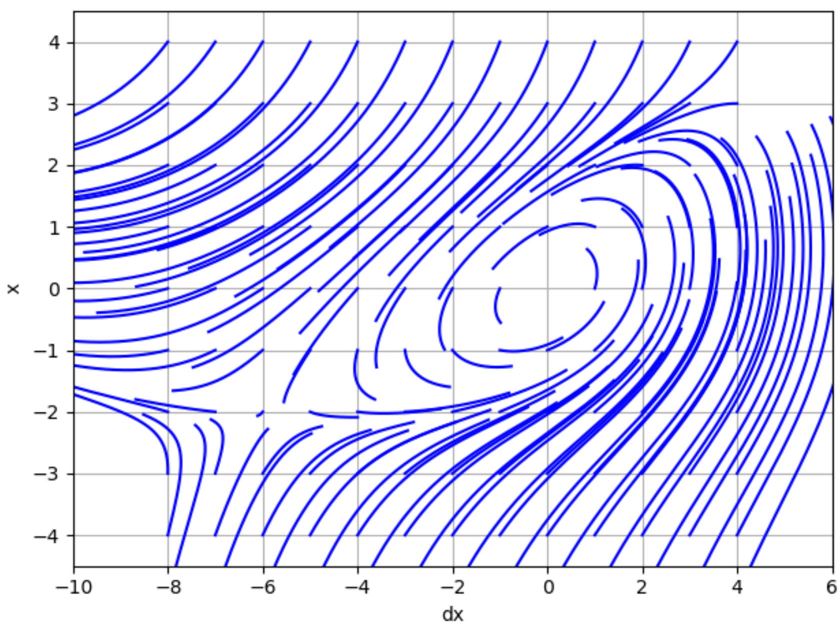

Рисунок 11 — Фазовый портрет системы №6

Линеаризируем систему, которая описана выше, и после данной операции мы получаем следующую систему:

$$\begin{cases} x' = x - 4y \\ y' = 2x - y \end{cases}$$

Зададим данную системе в языке Python:

```
def ode(Y, t) :
    x, y = Y
    dydt = [x - 4 * y, 2*x - y]
    return dydt
```

Построим фазовый портрет линеаризованной системы, смотри рисунок 12

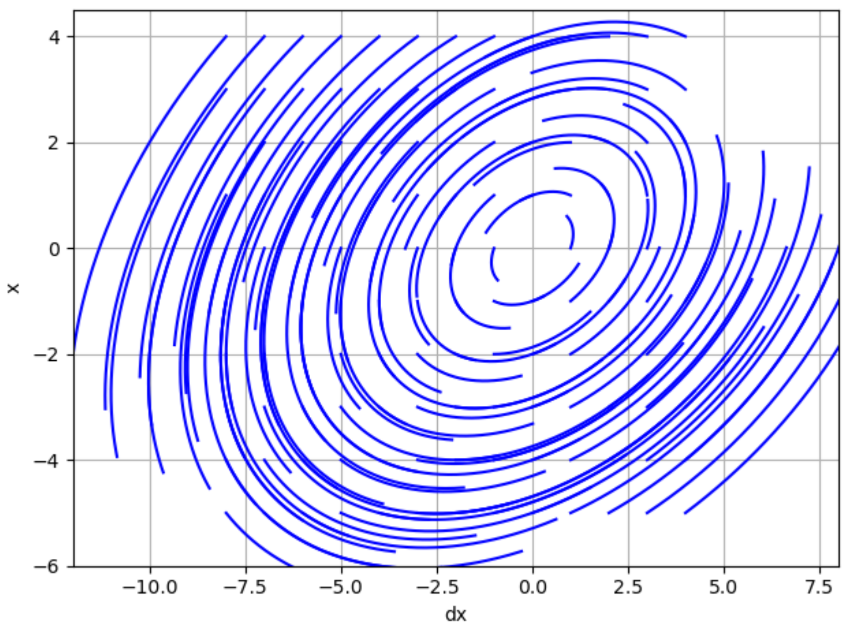

Рисунок 12 — Фазовый портрет линеаризованной системы №6

Фазовый портрет имеет вид особой точки, который называется центр. О такой системе нельзя судить только по фазовому портрету линеаризованной системы. Фазовый портреты этих двух систем сильно отличаются друг от друга, откуда следует выше написанный факт. Данная особая точка имеет только мнимые корни. Без линеаризации систем с особыми точками центр, невозоможно определитьточно вид особой точки, как в нашем примере.

4.2 Построени фазовых портретов для нелинейных элементов

По модели в виде блок-схемы(см. рисунок ??) постройте системы дифференциальных уравнений (в математической записи вместо нелинейных блоков можно определить соответствующие функции без их раскрытия).

Значения параметров нелинейностей: $0.2N_v + 0.2$, где $N_v = 1$ чт соответсвует варианту.

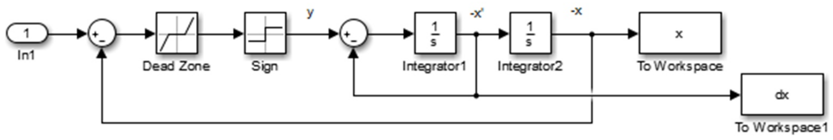

Рисунок 13 — Блок-схема заданной системы

Составим по данной схеме(см. рисунок 13) систему дифферинциальных уравнений, получится следующая система:

$$\begin{cases} x' = y \\ y' = np.sign(dead_zone(1 - x)) - y \end{cases}$$

Далее зададим полученную систему диффуринциальных уравнений в Python, во что получилось:

```
def ode(Y, t):
    x, y = Y
    dydt = [y, np.sign(dead_zone(1 - x)) - y]
    return dydt
```

Построим фазовый портрет данной системы(см.рисунок 14).

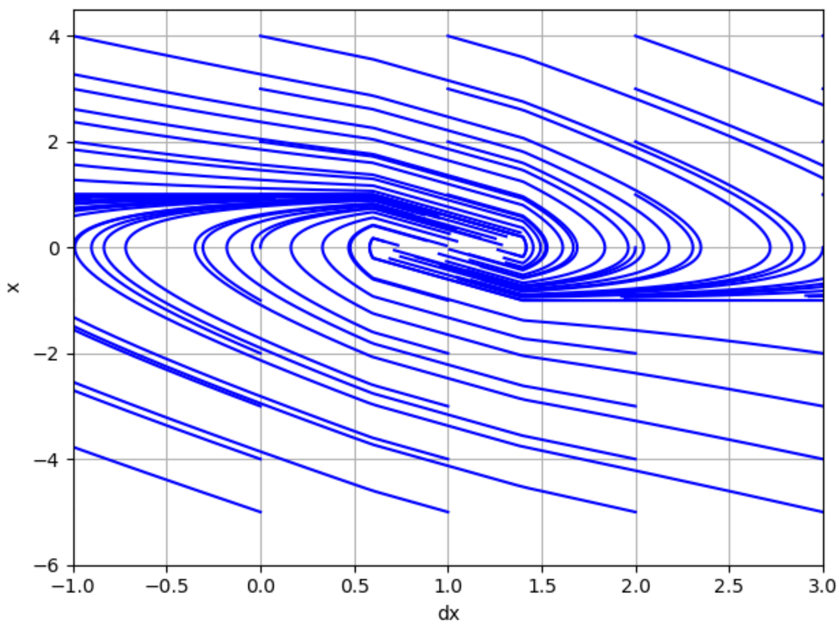

Рисунок 14 — Фазовый портрерт заданной блок-схемой системы

Проведем анализ полученного фазового портрета. Полученный фазовый портер схож с тем что мы получали на рисунке 1, только он нескоько изменен, но эти изменения показывают нам что он прошел через какую-то нелинейность, в данном случае (мертвая зоная, и реле). Таким образом мы можем сказать что система имеет вид особой точки устойчивый узел. Следовательно система имеет действительные корни, меньше нуля, и следовательно система устойчива.

4.3 Построение фазового портрета для маятника

В данном разделе мы построим фазовый портрет для мтематического мятника без и с взяким трением.

Система описывающая математический маятник без вязкого трения:

$$\begin{cases} \theta' = \omega \\ \omega' = -mglsin(\theta) \end{cases}$$

Сформулируем эту систему на языке программирования Python, получится следующий код:

```
def ode(y, t) :
    theta, omega = y
    dydt = [omega, -0.2 * 9.81 * 5 * np.sin(theta)]
    return dydt
```

После построем фазовый портрет даннной системы(см. рисунок 15)

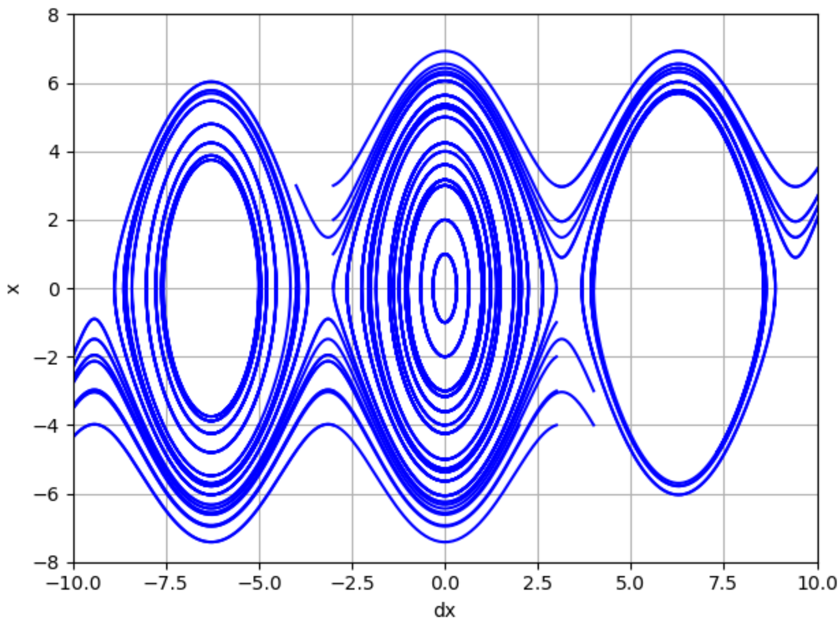

Рисунок 15 — Фазовый портрерт матемтического маятника без трения

Из рисунка 15 видно, что фазовый портрет математического маятника имеет бесконечно количество особых точек, которые чередются между точками седло и центр. В некоторой области система имеет минимые корни и переходный процес имеет кольбательный и незатухающий характер, после этого появляются как положительные, так и отрицательные корни.

Далее опишем математический маятник с вязким трением, система будет выглядить так:

$$\begin{cases} \theta' = \omega \\ \omega' = -b\omega - mglsin(\theta) \end{cases}$$

Опишем данную системы в Python:

```
def ode(y, t) :
    theta , omega = y
    dydt = [omega, (-b * omega) - (m * g * l * np.sin(theta))
    return dydt
```

Построем фазовой портрет описанной системы(см. рисунок 16):

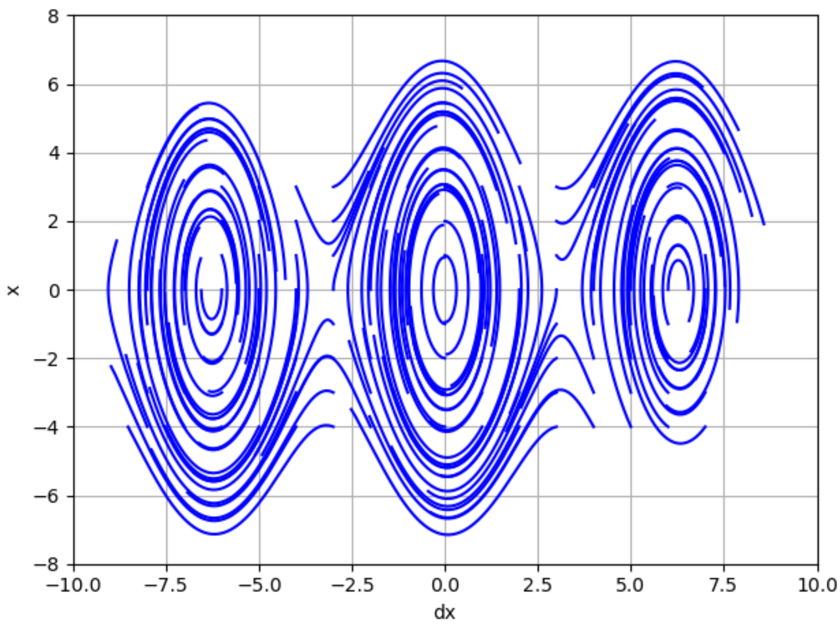

Рисунок 16 — Фазовый портрерт матемтического маятника с трением

Сразу же мы видим отличия рисунка 15 от рисунка 16, на рисунке 15 видно,что система в цместо центра, получает устойчивый узел, и таки образом у нее есть положение равновесия, к которому стремятся колебания в системе.

4.4 Построение фазовго портрета осциллятора Ван дер Поля

Осциллятор предложен голландским инженером и физиком Бальтазаром ван дер Полем, во время его работы в компании Philips. Ван дер Полем были найдены устойчивые колебания, которые были названы релаксационными, известные как «предельные циклы». В сентябре 1927 года Ван дер Поль и его коллега ван дер Марк сообщили, что на определенных частотах были зафиксированы шумы, всегда находящиеся рядом с собственными частотами волн. Это было одним из первых наблюдений детерминированного хаоса.

Уравнение Ван дер Поля применяется и в физике, и в биологии. Так, например, в биологии создана модель Фитц Хью-Нагумо. Данное уравнение также было использовано в сейсмологии для моделирования геологических разломов

$$\begin{cases} x' = y \\ y' = \mu(1 - x^2)y - x \end{cases}$$

Построем фазовые портреты с разными коэффициентами μ: $1, 0.5, 2$

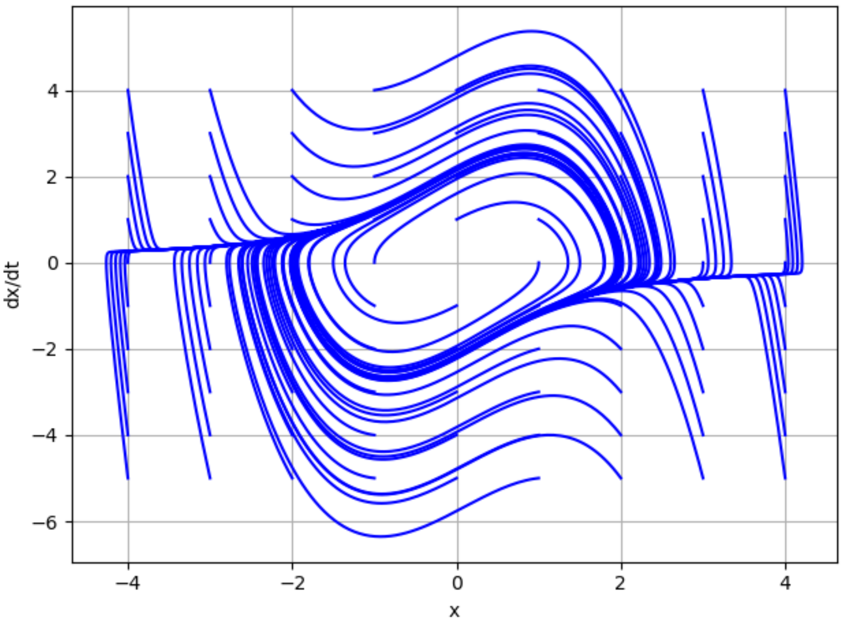

Рисунок 17 — Фазовый портрерт Ван дер Поля с 1

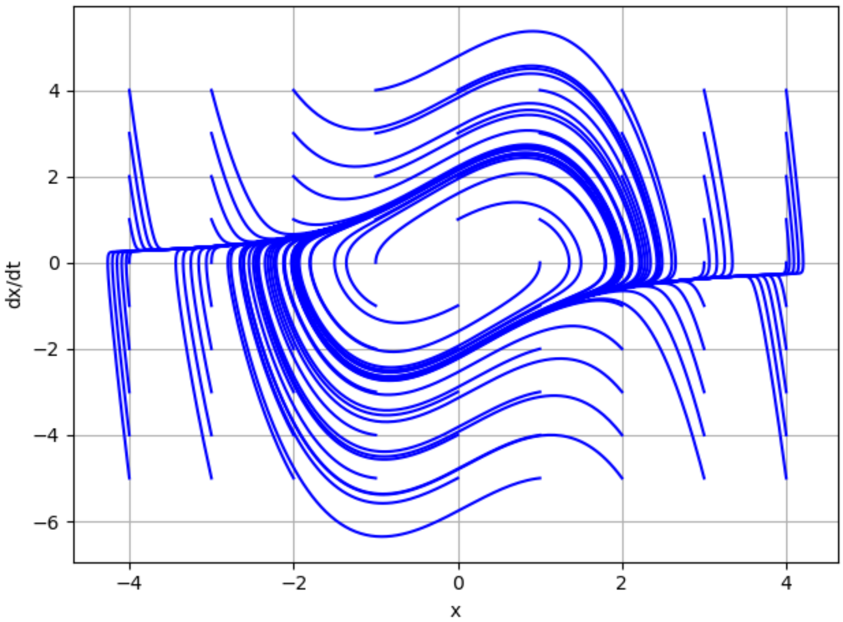

Рисунок 18 — Фазовый портрерт Ван дер Поля с 0,5

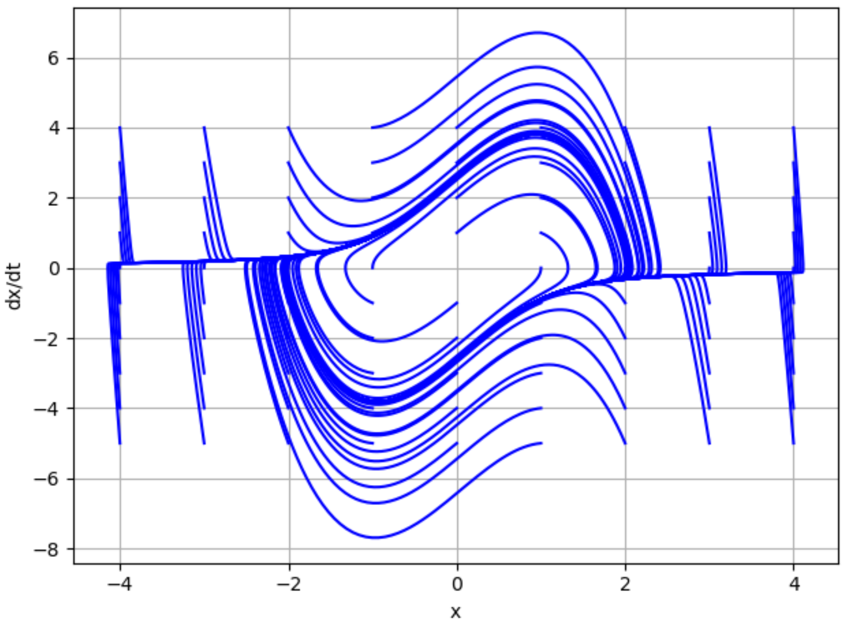

Рисунок 19 — Фазовый портрерт Ван дер Поля с 2

Фазовый портрет осцилятора Ван дер Поля обладает утсойчивым предельным циклом,что говорит о том что есть автоколебателный процесс. При изменени μ исходя из рисунко полученных выше, мы можем заметить что изменяется амплитуда, чем юольше форма, тем больше амплитуда.

4.5 Построение фазового портрета аттрактора Лоренца

$$\begin{cases} \dot{x} = \sigma(y - x) \\ \dot{y} = x(r - z) - y \\ \dot{z} = xy - bz \end{cases}$$

При следующих значения параметров: $\sigma = 10, r = 28, b = 8/3, x(0) = 1, y(0) = 0, z(0) = 0$, построить фазовый портрет. Опишем пример построение на языке Pyhton:

```
def ode(y, t, sigma, r, b):
x, y, z = y
dxdt = sigma * (y - x)
dydt = x * (r - z) - y
dzdt = x * y - b * z
return [dxdt, dydt, dzdt]


def calcODE(args, x, y, z, ts=10, nt=101):
        y0 = [x, y, z]
        t = np.linspace(0, ts, nt)
        sol = odeint(ode, y0, t, args)
        return sol



def drawPhasePortrait3D(args,
        deltaX=1, deltaY=1, deltaZ=1,
        startX=0, stopX=5,
        startY=0, stopY=5,
        startZ=0, stopZ=5,
        ts=10, nt=101):
        fig = plt.figure()
        ax = fig.add_subplot(2, 2, 1, projection='3d')
        ax.set_title("3D")
        plt.subplot(2, 2, 2)
        plt.title("X-Y")
        plt.grid()
        plt.subplot(2, 2, 3)
        plt.title("X-Z")
        plt.grid()
        plt.subplot(2, 2, 4)
        plt.title("Y-Z")
```

```
        plt.grid()

        for x in range(startX, stopX, deltaX):
        for y in range(startY, stopY, deltaY):
        for z in range(startZ, stopZ, deltaZ):
        sol = calcODE(args, x, y, z, ts, nt)

        ax.plot(sol[:, 0], sol[:, 1], sol[:, 2])
        plt.subplot(2, 2, 2)
        plt.plot(sol[:, 0], sol[:, 1])
        plt.subplot(2, 2, 3)
        plt.plot(sol[:, 0], sol[:, 2])
        plt.subplot(2, 2, 4)
        plt.plot(sol[:, 1], sol[:, 2])

        plt.show()


sigma = 10
r = 90
b = 8 / 3
args = (sigma, r, b)
drawPhasePortrait3D(args,
deltaX=8, deltaY=8, deltaZ=8,
startX=-10, stopX=10,
startY=-10, stopY=10,
startZ=-10, stopZ=10,
ts=0.5, nt=500)
```

Теперь посмотрим на фазовый портрет полученый написанной функцией, с задаными параметрами(см. рисунок 20)

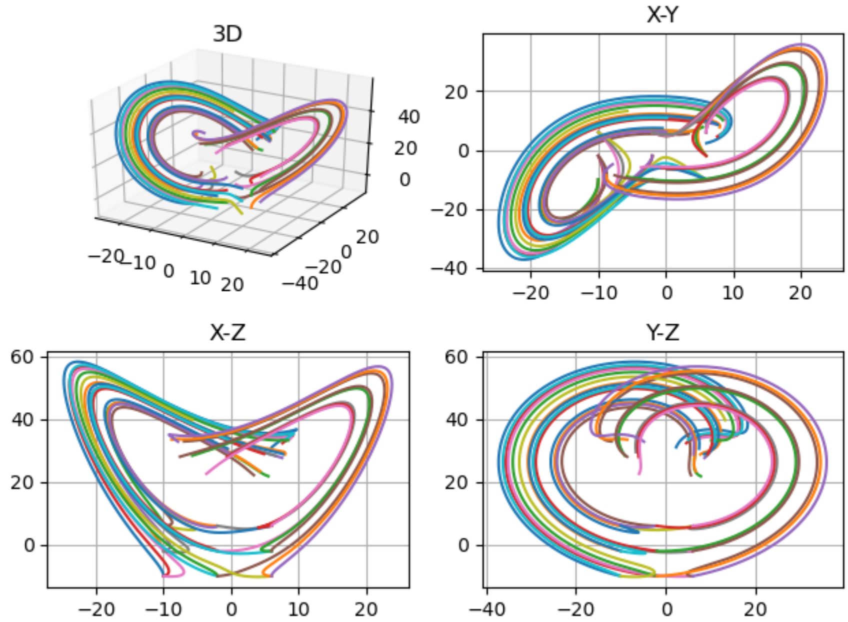

Рисунок 20 — Фазовый портрет аттрактора Лоренца

Фазовые траектории ведут не к устойчивым точкам, а асимптотически приближаются к неустойчивым предельным циклом.

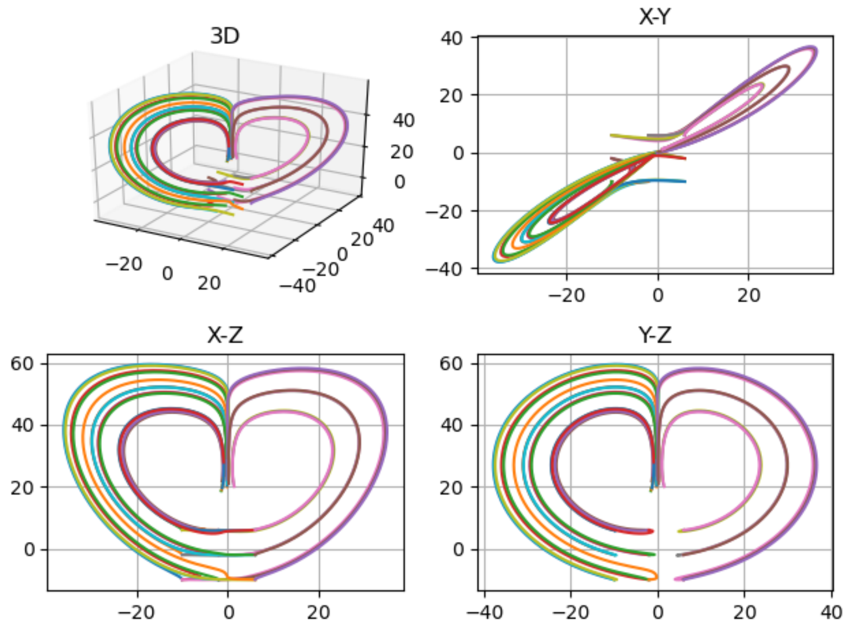

Рисунок 21 — Фазовый портрет аттрактора Лоренца при увелечение σ

Как можно заметить из рисунков 23 и 20, чт система "разрывается"при увеличение σ. Так же можно заметить что она отвечает за наклон графика к плоскостям.

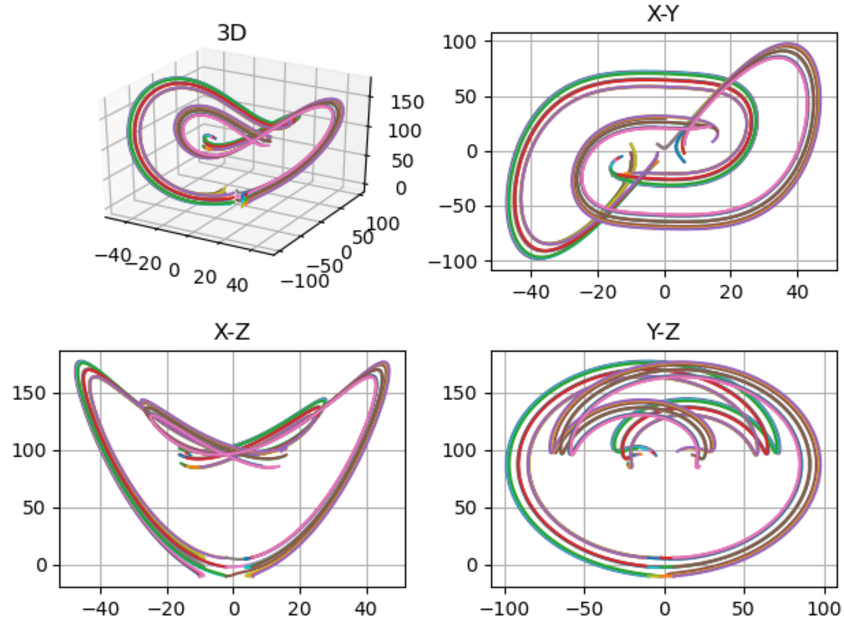

Рисунок 22 — Фазовый портрерт аттрактора Лоренца при увелечение r

При увеличение r мы видим что траектория изменяется, мы видим что в системе присутствуют колебания и потом они переходя в некое подобие центра.

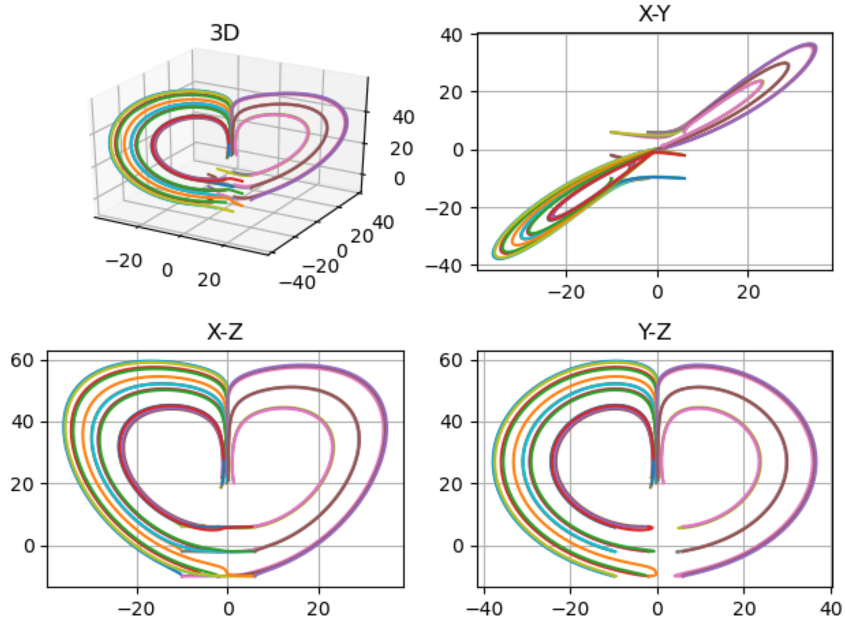

Рисунок 23 — Фазовый портрерт аттрактора Лоренца при увелечение σ

При изменение параметра b, мы наблюдаем следующие.

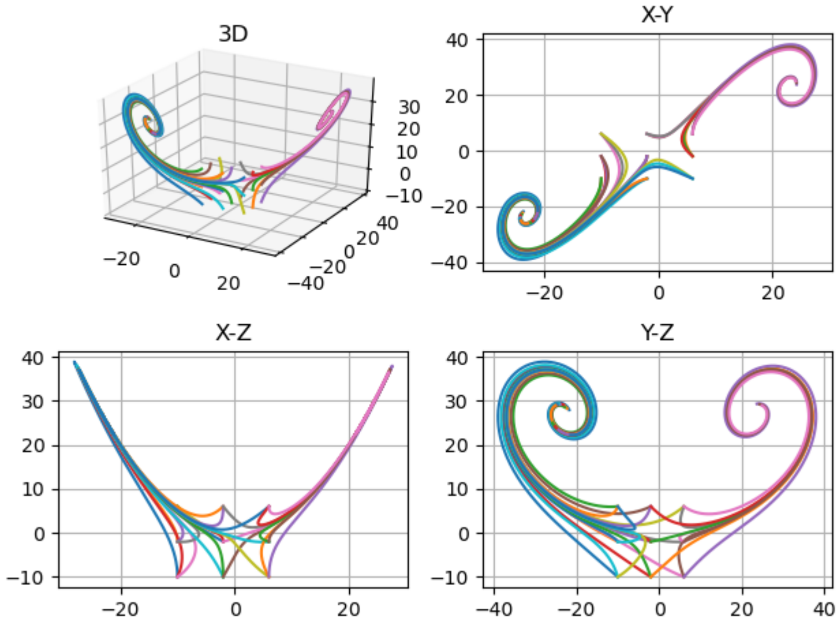

Рисунок 24 — Фазовый портрерт аттрактора Лоренца при увелечение b

Мы видим опять же "разрывание"системы, но так же мы видим что очень сильно изменяются фазовые траектория(в олиие от первоначального(см. рисунок 20)), и чем больше b, тем больше изменение траекторий.

5 ВЫВОД

Были изучены способы построения фазовых портретов, построены фазовые портреты для типовых особых точек систем второго порядка, рассмотрены случаи построения фазовых портретов нелинейных систем, а так же был исследован аттрактор Лоренца, показывающий особую точку системы 3-ого порядка. Метод фазвых портретов позволяет исследовать ту или иную систему,и ее поведения, не решаю дифференциальные уравнения, которые порой сложно решаемы.