**Федеральное государственное бюджетное образовательное учреждение высшего образования  
“Уфимский университет науки и технологий”**

**Кафедра** Высокопроизводительных вычислительных технологий и систем

**Дисциплина:** Математическое моделирование

Лабораторная работа №4

**Тема:** “Исследование эволюции нелинейной диссипативной динамической системы.”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Группа ПМ-455 | ФИО | Подпись | Дата | Оценка |
| Студент | Наумова Т.И. |  |  |  |
| Преподаватель | Лукащук С.Ю. |  |  |  |

Уфа 2023

**Цель**: получить навык численного исследования динамики нелинейной диссипативной динамической системы, обладающей странным аттрактором.

**Задание**

Рассматривается нелинейная двухпараметрическая автономная динамическая система

Для заданной системы выполнить следующие задания:

1. Определить области изменения параметров и , в которых данная динамическая система является диссипативной.
2. Определить стационарные точки диссипативной системы.
3. Исследовать стационарные точки на асимптотическую устойчивость по первому приближению.
4. Определить значения параметров и , при которых в системе появляется странный аттрактор.
5. Написать вычислительную программу на языке программирования С++, реализующую процедуру численного интегрирования исходной диссипативной системы по методу Рунге-Кутта 4-го порядка точности.
6. С использованием вычислительной программы провести серию вычислительных экспериментов, демонстрирующих различные виды динамики системы. Определить численно значения параметров и , при которых в системе существует странный аттрактор и при которых система переходит в режим автоколебаний.

**Практическая часть**

Дана система

1. Чтобы система была диссипативной, необходимо чтобы
2. Стационарные точки системы

Причем, для точек и появится условие для параметра b:

1. Исследование на асимптотическую устойчивость по первому приближению.  
   Условием устойчивости будет

Сделаем замену переменных:

Где – координаты стационарной точки. – новые координаты.

и разложим в ряд Тейлора в окрестностях точек:

Обобщенная матрица правой части примет вид:

1. Рассмотрим точку :

Соответствующее характеристическое уравнение:

По критерию Рауса-Гурвица, необходимым и достаточным условием отрицательности действительных частей всех корней многочлена является положительность всех главных диагональных миноров матрицы Рауса-Гурвица.

Для многочлена

Матрица Рауса-Гурвица выглядит следующим образом:

Для нашего случая:

Оценим его главные диагональные миноры:

Для того, чтобы точка было устойчива, необходимо:

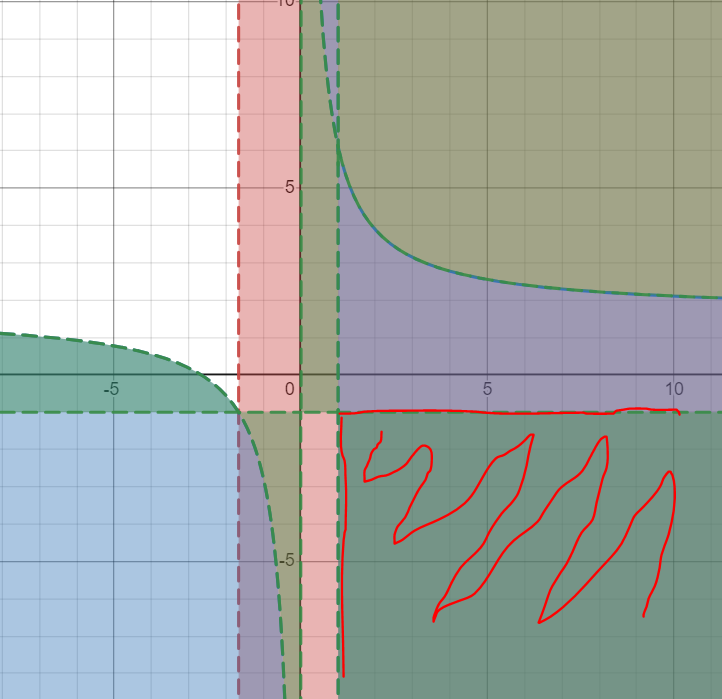


Рисунок 1 – область устойчивости стационарной точки

1. Рассмотрим точку

Матрица правой части в этой точке:

По критерию Рауса-Гурвица, необходимым и достаточным условием отрицательности действительных частей всех корней многочлена является положительность всех главных диагональных миноров матрицы Рауса-Гурвица.

Для многочлена

Матрица Рауса-Гурвица выглядит следующим образом:

Для нашего случая:

Запишем необходимые условия:

Для того, чтобы точка было устойчива, необходимо:

С учетом всех приведенных выше условий, стационарная точка устойчива в области:

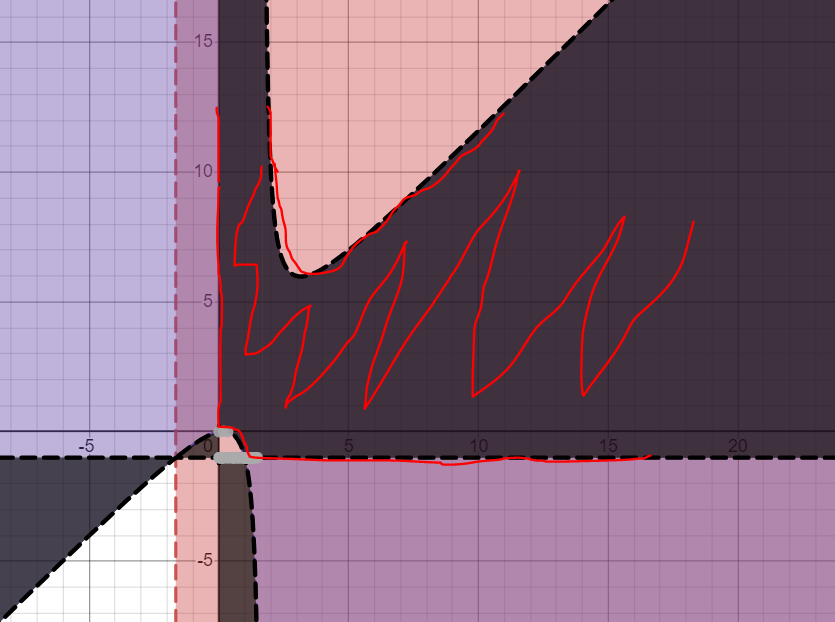


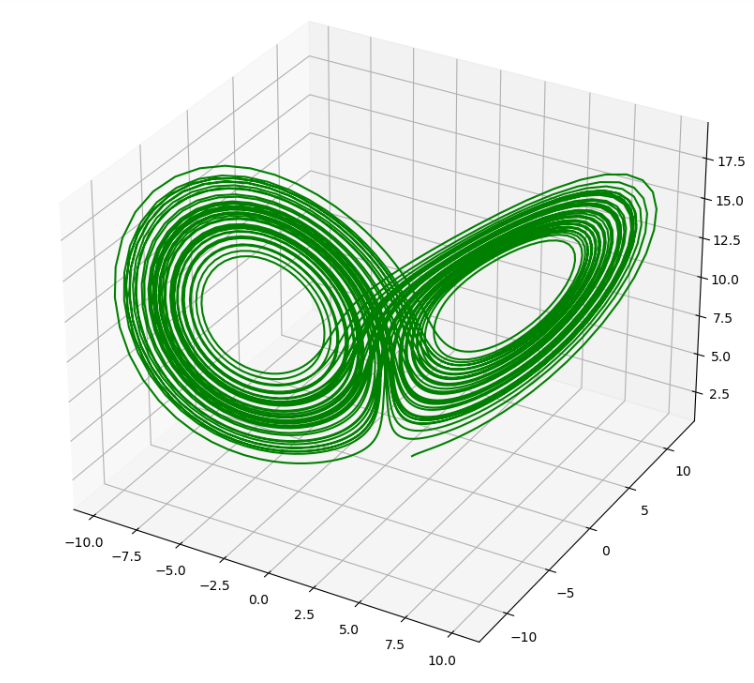
Рисунок 2 – область устойчивости стационарной точки

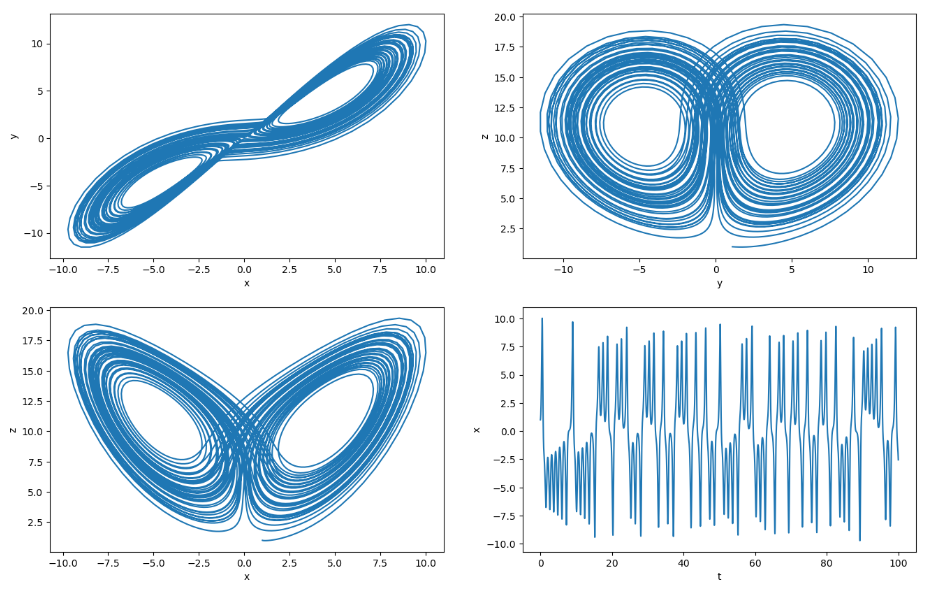
1. Рассмотрим точку

Матрица правой части системы в этой точке:

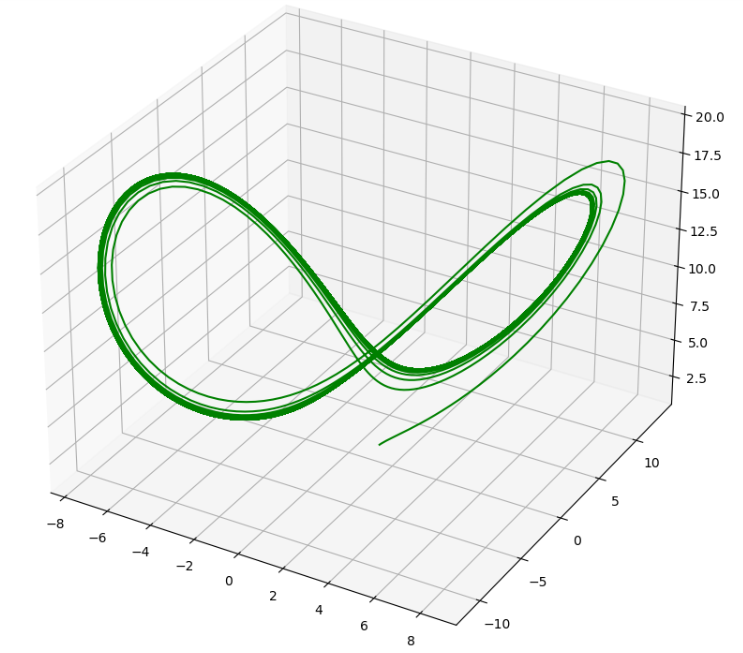
Характеристический полином совпадает с полиномом точки а значит и области устойчивости стационарной точки совпадают.

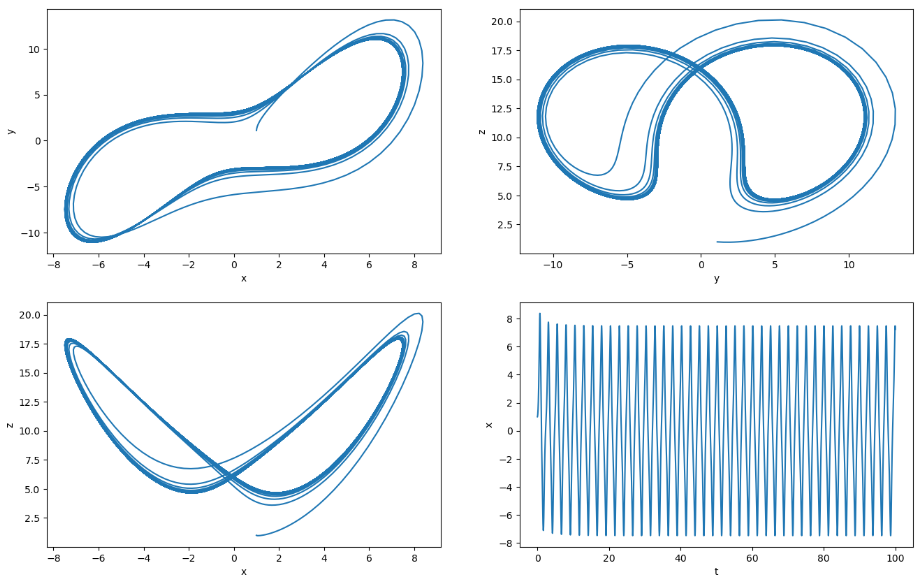
1. Странный аттрактор появляется при



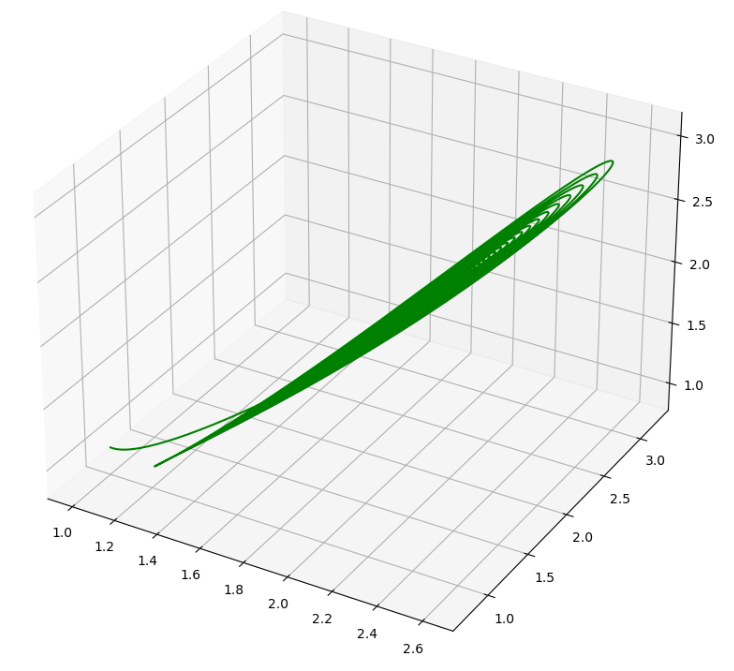


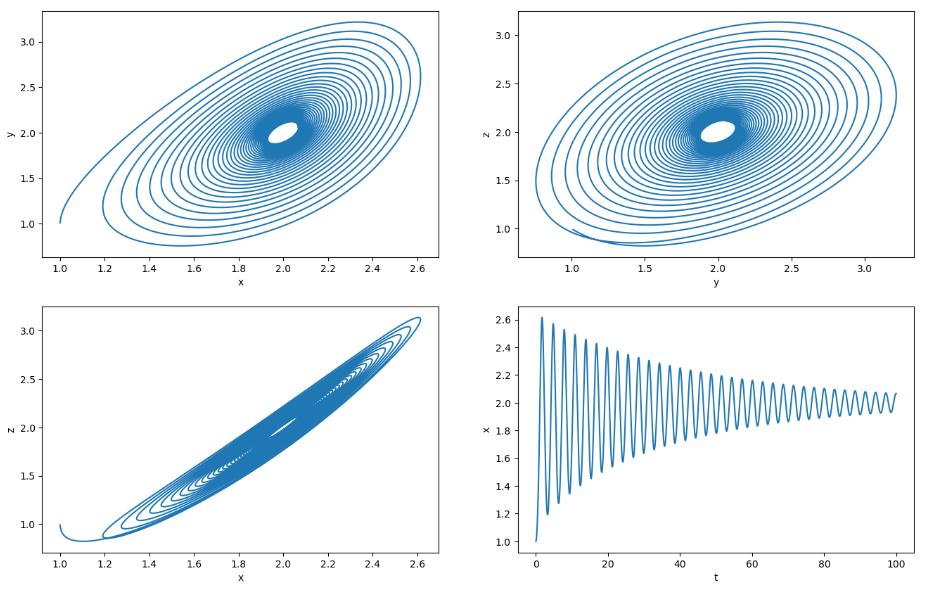
1. Автоколебания появляются при a=3, b=10





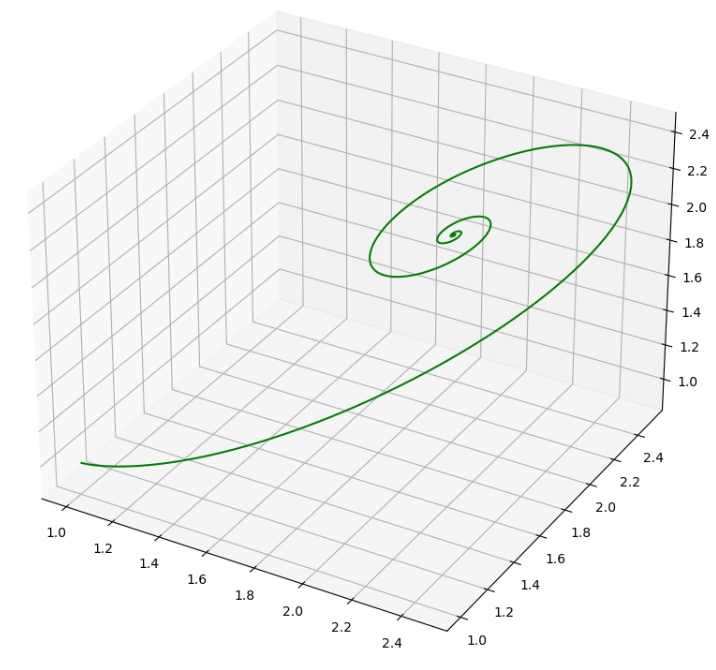
Сходимость к точкам при а=1,5 и b=1

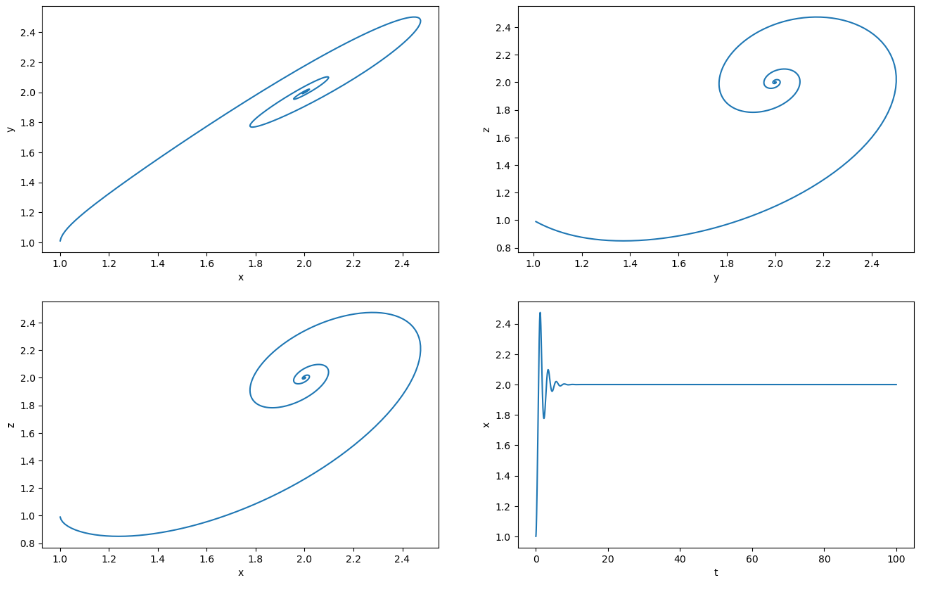




устойчива

При a=10, b=1





устойчива

**Вывод:** в ходе выполнения лабораторной работы получен навык численного исследования динамики нелинейной диссипативной динамической системы, обладающей странным аттрактором. Определены области изменения параметров a и b, в которых данная динамическая система является диссипативной. Проведен поиск стационарных точек. Найдены параметры a и b, при которых в системе существует странный аттрактор, а также получены различные виды динамики системы. Для расчета траекторий системы при различных параметрах a и b на языке программирования С++ реализован метод Рунге-Кутта четвертого порядка.

Листинг программы:

C++:

#include <iostream>

#include <stdio.h>

#include <vector>

#include <iomanip>

#include <fstream>

#include <time.h>

double f(double t, double x, double y, double z, double a, double b) {

    return a \* (y – x);

}

double g(double t, double x, double y, double z, double a, double b) {

    return b \* x + y – x \* z;

}

double h(double t, double x, double y, double z, double a, double b) {

    return -8/3 \* z + x \* y;

}

std::vector<double> new\_point(double f(double, double, double, double, double, double),

                            double g(double, double, double, double, double, double),

                            double h(double, double, double, double, double, double),

                            double t, double x, double y, double z,

                            double step, double a, double b)

{

    double kx0, ky0, kz0, kx1, ky1, kz1, kx2, ky2, kz2, kx3, ky3, kz3;

    kx0 = f(t, x, y ,z, a, b);

    ky0 = g(t, x, y, z, a, b);

    kz0 = h(t, x, y, z, a, b);

    kx1 = f(t + (step / 2), x + (step / 2) \* kx0, y + (step / 2) \* ky0, z + (step / 2) \* kz0, a, b);

    ky1 = g(t + (step / 2), x + (step / 2) \* kx0, y + (step / 2) \* ky0, z + (step / 2) \* kz0, a, b);

    kz1 = h(t + (step / 2), x + (step / 2) \* kx0, y + (step / 2) \* ky0, z + (step / 2) \* kz0, a, b);

    kx2 = f(t + (step / 2), x + (step / 2) \* kx1, y + (step / 2) \* ky1, z + (step / 2) \* kz1, a, b);

    ky2 = g(t + (step / 2), x + (step / 2) \* kx1, y + (step / 2) \* ky1, z + (step / 2) \* kz1, a, b);

    kz2 = h(t + (step / 2), x + (step / 2) \* kx1, y + (step / 2) \* ky1, z + (step / 2) \* kz1, a, b);

    kx3 = f(t + step, x + step \* kx2, y + step \* ky2, z + step \* kz2, a, b);

    ky3 = g(t + step, x + step \* kx2, y + step \* ky2, z + step \* kz2, a, b);

    kz3 = h(t + step, x + step \* kx2, y + step \* ky2, z + step \* kz2, a, b);

    x = x + (step / 6) \* (kx0 + 2 \* kx1 + 2 \* kx2 + kx3);

    y = y + (step / 6) \* (ky0 + 2 \* ky1 + 2 \* ky2 + ky3);

    z = z + (step / 6) \* (kz0 + 2 \* kz1 + 2 \* kz2 + kz3);

    return std::vector<double>{x, y, z};

}

void write\_file(double a, double b,

                std::vector<double> t,

                std::vector<double> x,

                std::vector<double> y,

                std::vector<double> z)

{

    std::ofstream fout;

    fout.open("result.txt");

    if (t.size() != x.size() || t.size() != y.size() || t.size() != z.size()) {

        std::cout << "Dimension t is not equal with other dimensions" << std::endl;

        fout.close();

        throw "dimensions error";

    }

    fout << a << " " << b << "\n";

    for (int i = 0; i < x.size(); i++)

    {

        fout << t[i] << " " << x[i] << " " << y[i] << " " << z[i] << "\n";

    }

    fout.close();

}

int main()

{

    double step = 0.01;

    int n = 10000;

    double t0 = 0, t1 = t0 + n \* step;

    double a = 1, b = 10; //

    double x0 = 1, y0 = 1, z0 = 1;

    std::vector<double> t(n), x(n), y(n), z(n);

    std::vector<double> point(3);

    double t\_;

    x[0] = x0; y[0] = y0; z[0] = z0;

    for (int i = 0; i < n - 1; i++)

    {

        t\_ = i \* step;

        point = new\_point(f, g, h, t\_, x[i], y[i], z[i], step, a, b);

        t[i + 1] = t\_;

        x[i + 1] = point[0];

        y[i + 1] = point[1];

        z[i + 1] = point[2];

    }

    write\_file(a, b, t, x, y, z);

    return 0;

}

Python:

import numpy as np

import matplotlib.pyplot as plt

import scipy

f = open(C:/Users/Acer/source/repos/Project3/result.txt', 'r')

a, b = map(float, f.readline().split())

txyz = []

for point in f:

    txyz.append(list(map(float, f.readline().split())))

txyz = np.array(txyz)

print(txyz.shape)

print(txyz)

fig, axs = plt.subplots(2, 2, figsize=(16,10))

axs[0,0].plot(txyz[:, 1], txyz[:, 2])

axs[0,0].set\_xlabel('x')

axs[0,0].set\_ylabel('y')

axs[0,1].plot(txyz[:, 2], txyz[:, 3])

axs[0,1].set\_xlabel('y')

axs[0,1].set\_ylabel('z')

axs[1,0].plot(txyz[:, 1], txyz[:, 3])

axs[1,0].set\_xlabel('x')

axs[1,0].set\_ylabel('z')

axs[1,1].plot(txyz[:, 0], txyz[:, 1])

axs[1,1].set\_xlabel('t')

axs[1,1].set\_ylabel('x')

plt.show()

fig = plt.figure(figsize=(16, 10))

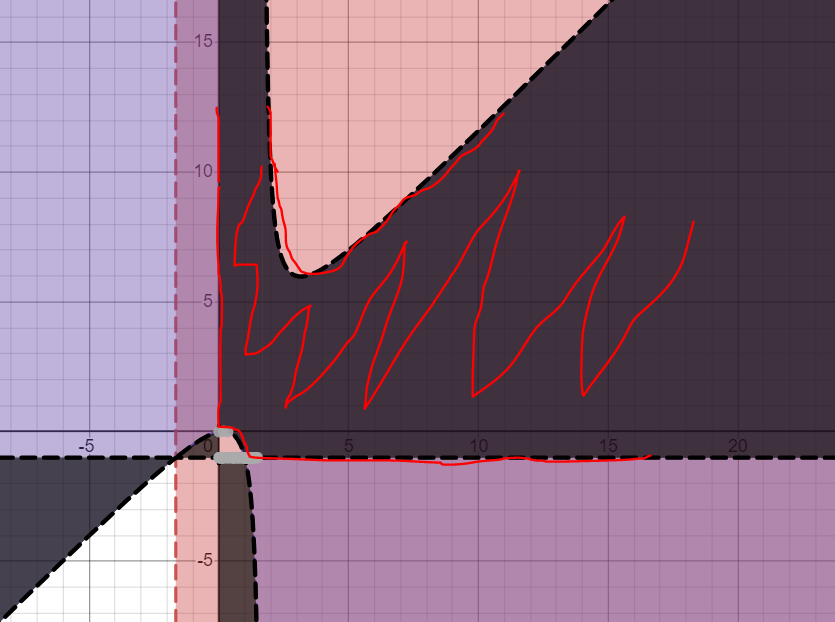
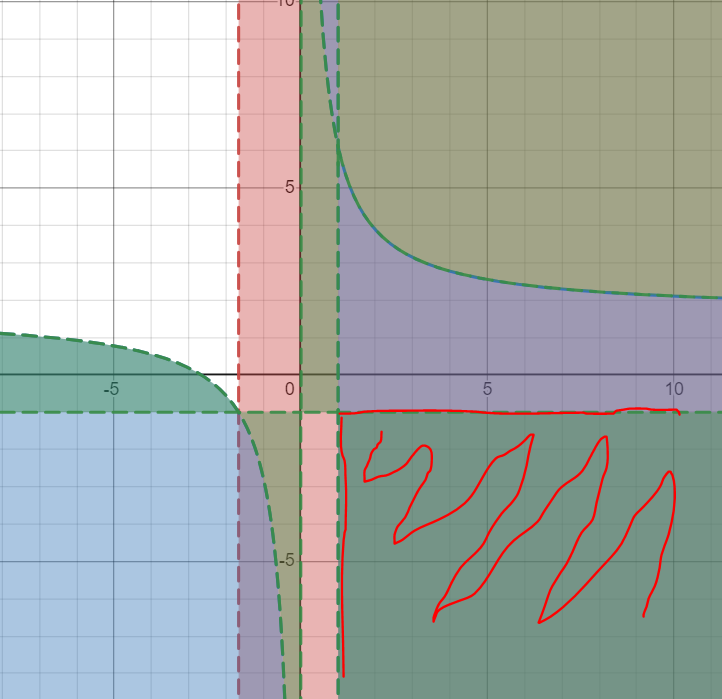
# syntax for 3-D projection

ax = plt.axes(projection ='3d')

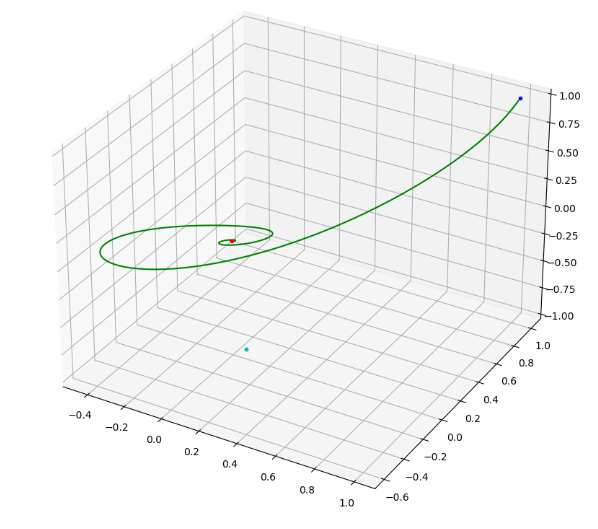
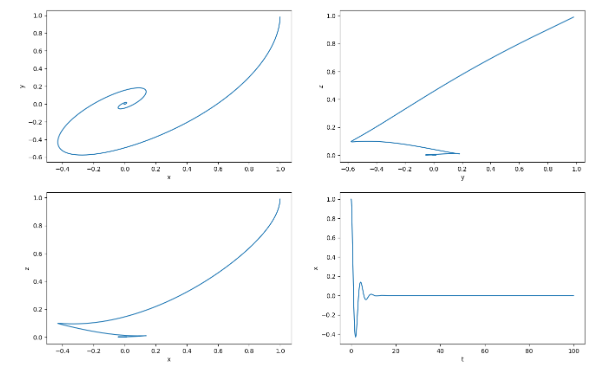
# plotting

ax.plot3D(txyz[:, 1], txyz[:, 2], txyz[:, 3], 'green')

plt.show()

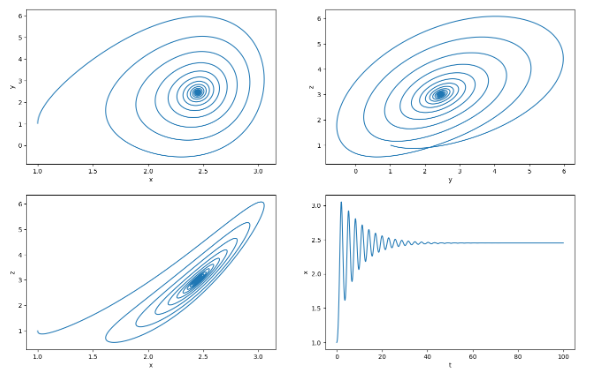


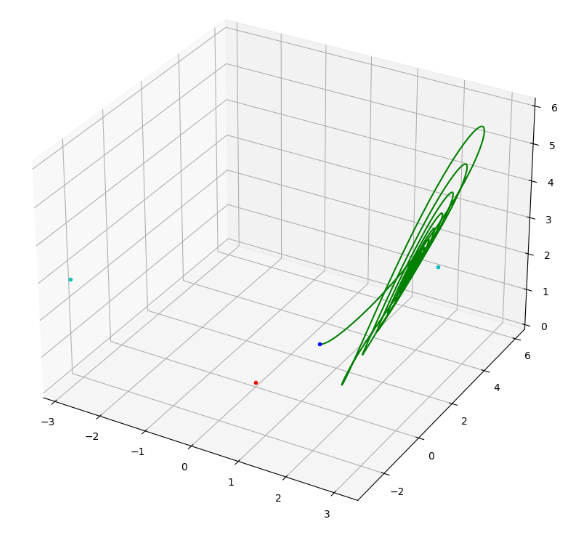
при а=2 и б=-2



Точка О1 устойчива О2 и О3 неустойчивы

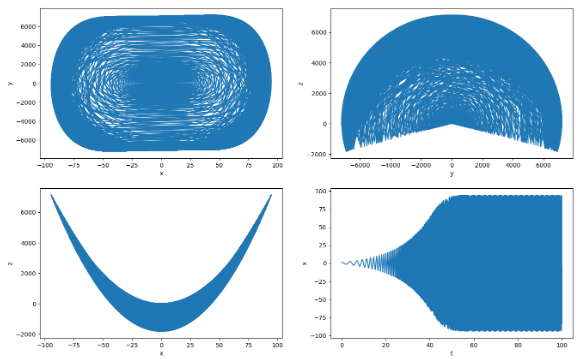
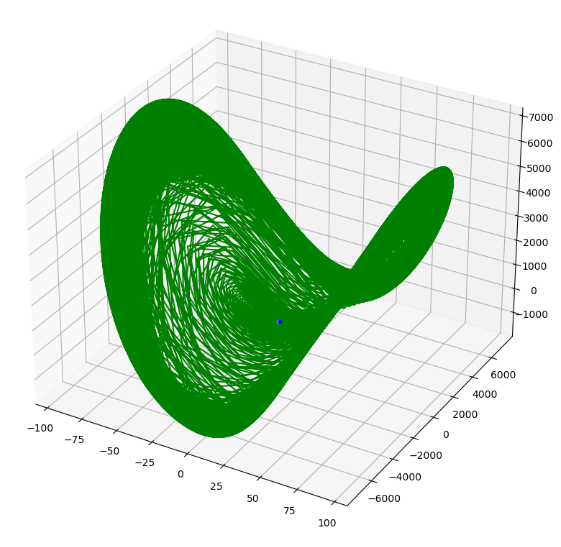
При а=0.5 и б=2





О1 неуст и О3 уст

При а=0,5и б=-5



Все точки неустойчивы