**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**"Уфимский университет науки и технологий"**

**Кафедра** Высокопроизводительных вычислительных технологий и систем

**Дисциплина:** Математическое моделирование

**Отчет по лабораторной работе № 4**

**Тема:** «Исследование эволюции нелинейной диссипативной динамической системы»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Группа ПМ-457 | Фамилия И.О. | Подпись | Дата | Оценка |
| Студент | Акмурзин М.Э. |  |  |  |
| Принял | Лукащук С. Ю. |  |  |  |

**Уфа 2024**

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

**Задание на лабораторную работу**

Работа выполнена согласно варианту № 2.

Рассматривается нелинейная двухпараметрическая автономная динамическая система

Для заданной системы выполнить следующие задания.

1. Определить области изменения параметров *a* и *b*, 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 *a* и *b*, при которых в системе появляется странный аттрактор.
5. Написать вычислительную программу на языке программирования Cи++, 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. Построить траектории системы в окрестности стационарных точек. Определить численно значения параметров *a* и *b,* при которых в системе существует странный аттрактор и при которых система переходит в режим автоколебаний.

**Практическая часть**

Исходная система имеет вид

1. **Область изменения параметров.**

Динамическая система является диссипативной, если

Следовательно, система будет диссипативной, если .

1. **Поиск стационарных точек.**

Данная система имеет тривиальное решение .

1. **Исследование стационарных точек на асимптотическую устойчивость по первому приближению.**

Выполним линеаризацию в окрестностях стационарных точек

малые возмущения.

Система после разложения в ряд Тейлора в стационарной точке

Система после разложения в точке :

Матрица системы

0

Получим

Коэффициенты при степенях :

Характеристический многочлен имеет вид

Для дальнейшего исследования воспользуемся критерием Рауса – Гурвица.

Матрица Гурвица:

Должны выполняться следующие условия:

Получим следующие ограничения:

На рисунке 1 показано как эти ограничения накладываются друг на друга.

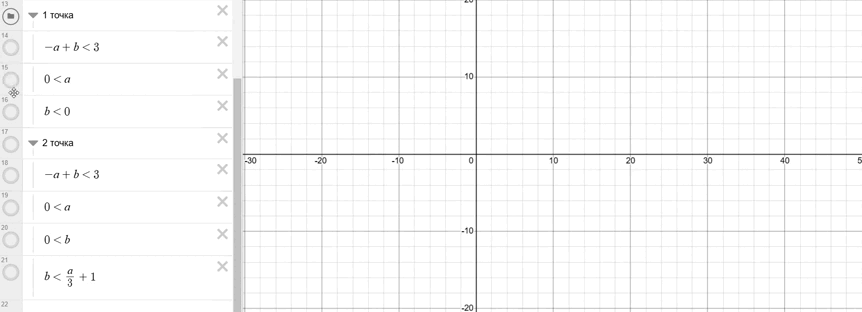


Рисунок 1 – Область значений параметров и , при которых точка устойчива.

Система после разложения в точке

Матрица системы

0

Получим

Коэффициенты при степенях :

Характеристический многочлен имеет вид

Для дальнейшего исследования воспользуемся критерием Рауса – Гурвица.

Матрица Гурвица:

Должны выполняться следующие условия:

Получим следующие ограничения:

На рисунке 2 показано как эти ограничения накладываются друг на друга.

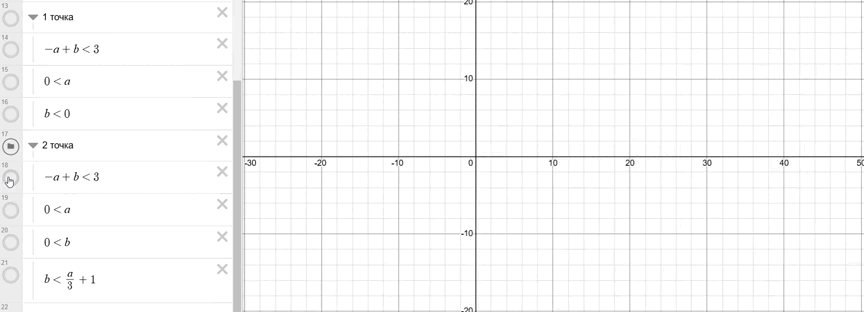
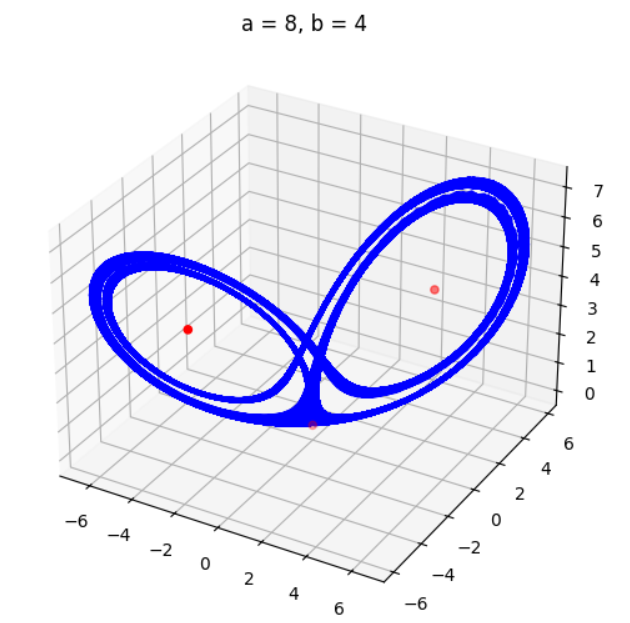


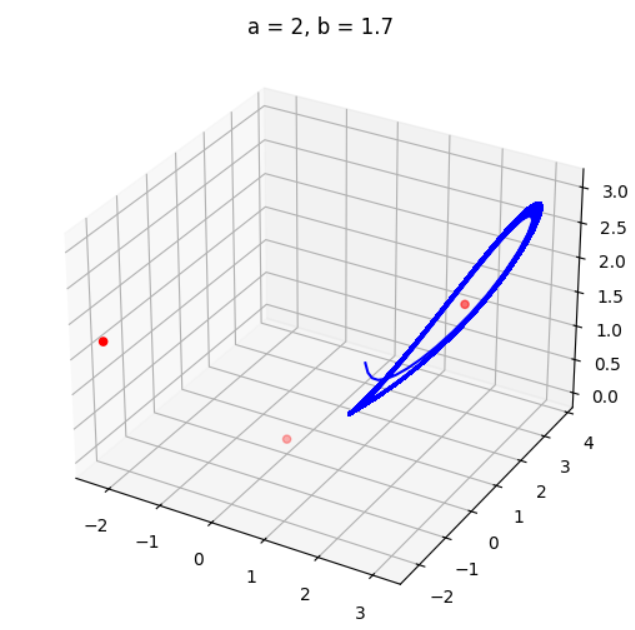
Рисунок 2 – Область значений параметров и , при которых точка устойчива.

Так как характеристические полиномы для точек то области устойчивости стационарных точек совпадают.

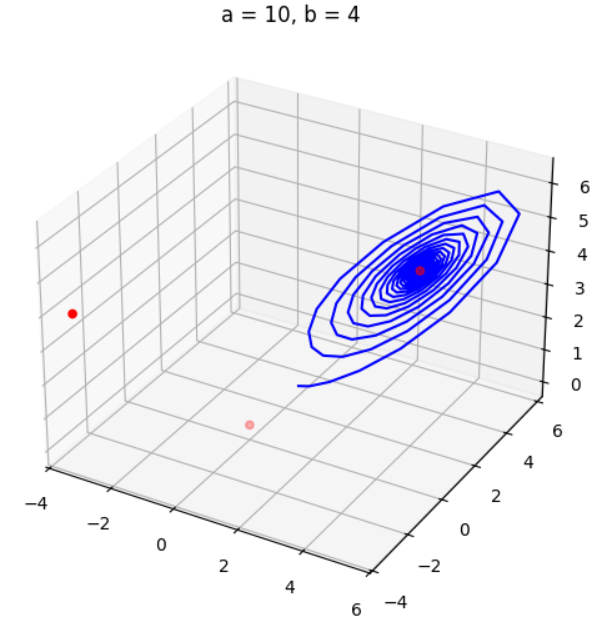
1. **Определение значений параметров a и b, при которых в системе появляется странный аттрактор.**

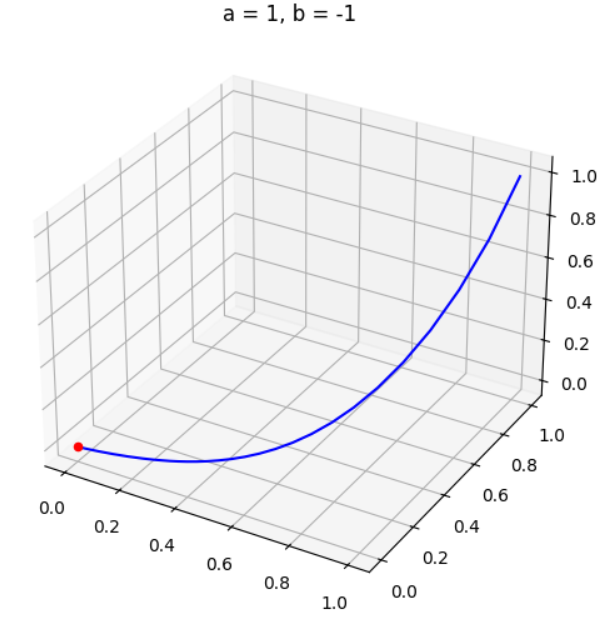
Странный аттрактор возникает, когда все стационарные точки неустойчивы.





Сходимость к стационарном точкам в области устойчивости

****

****

**Вывод**

В данной лабораторной работе были получены навыки численного исследования динамики нелинейной диссипативной динамической системы, обладающей странным аттрактором. Были определены области изменения параметров a и b, в которых данная динамическая система является диссипативной, также были найдены параметры a и b при которых в системе существует странный аттрактор и при которых система переходит в режим автоколебаний.

**Приложение**

Листинг программы:

#include <iostream>

#include <stdio.h>

#include <vector>

#include <iomanip>

#include <fstream>

#include <time.h>

#include <math.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 \* y - x \* z;

}

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

return -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;

std::ofstream fout\_stat;

fout.open("../labs/lab4/misha/result/result.txt");

fout\_stat.open("../labs/lab4/misha/result/stat\_points.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\_stat << a << " " << b << std::endl;

fout\_stat << sqrt(3\*b) << " " << sqrt(3\*b) << " " << b << std::endl;

fout\_stat << -sqrt(3\*b) << " " << -sqrt(3\*b) << " " << b << std::endl;

fout\_stat << 0 << " " << 0 << " " << 0 << std::endl;

fout\_stat.close();

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

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

}

fout.close();

}

int main()

{

double step = 0.1;

int n = 100000;

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

std::cout << "Введите a и b: " << std::endl;

double a = - 0.01 , b = 2.9;

std::cin >> a >> b;

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);

// system("python ../labs/lab4/misha/src/draw.py");

return 0;

}