Лабораторная работа 3: Дифференциальные уравнения и имитационное моделирование в Python

Цель работы:

Дифференциальные уравнения:

Дифференциальное уравнение - это уравнение, которое содержит производные неизвестной функции. Оно используется для описания зависимостей между некоторой неизвестной функцией и ее производными от одной или нескольких переменных.

Обычно дифференциальные уравнения используются для моделирования процессов изменения во времени, где скорость изменения какой-либо величины зависит от текущего значения этой величины. Они широко применяются во многих областях, таких как физика, биология, экономика и инженерия, для описания различных процессов и явлений.

Аттракторы в дифференциальных уравнениях:

В дифференциальных уравнениях аттрактор обозначает конечное или бесконечное множество значений, в котором движется решение системы дифференциальных уравнений во времени. Он представляет собой тип поведения, при котором решение системы стремится к определенному набору значений с течением времени, независимо от начальных условий.

Аттракторы могут иметь различные формы, такие как точки (аттракторы-точки), кривые (аттракторы-циклы) или сложные геометрические фигуры (аттракторы-странные аттракторы), и они играют важную роль в описании поведения сложных динамических систем.

В Python существует несколько библиотек, которые предоставляют методы для решения дифференциальных уравнений и нахождения аттракторов. Вот несколько популярных библиотек и методов:

– SciPy - это библиотека для научных вычислений в Python, которая содержит модуль scipy.integrate для численного интегрирования и решения обыкновенных дифференциальных уравнений (ODE). Он включает методы, такие как solve_ivp и odeint, которые позволяют численно интегрировать системы дифференциальных уравнений.

– SymPy - это библиотека символьных вычислений в Python, которая предоставляет возможность работы с символьными выражениями, включая решение дифференциальных уравнений. Модуль sympy.ode содержит методы для символьного решения обыкновенных дифференциальных уравнений.

– NumPy - это фундаментальная библиотека для научных вычислений в Python, которая предоставляет высокопроизводительные структуры данных и операции над ними. Хотя NumPy сам по себе не решает дифференциальные уравнения, его массивы и функции могут использоваться вместе с другими библиотеками для численного решения.

– Matplotlib - это библиотека для визуализации данных в Python, которая может использоваться для визуализации результатов решения дифференциальных уравнений и аттракторов. Модуль matplotlib.pyplot предоставляет широкие возможности по созданию графиков и визуализации данных.

Решим и визуализируем аттрактор Лоренца в трехмерном пространстве с использованием вышеописанных библиотек:

```
import numpy as np
from scipy.integrate import solve_ivp
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

# Определение функции для аттрактора Лоренца
def lorenz(t, xyz, sigma=10, beta=8/3, rho=28):
    x, y, z = xyz
    dxdt = sigma * (y - x)
    dydt = x * (rho - z) - y
```

```
    dzdt = x * y - beta * z
    return [dxdt, dydt, dzdt]

# Задание начальных условий
xyz0 = [0, 1, 1.05]
t_span = [0, 100]
t = np.linspace(t_span[0], t_span[1], 10000)

# Решение системы уравнений Лоренца
sol = solve_ivp(lorenz, t_span, xyz0, t_eval=t)

# Визуализация аттрактора Лоренца
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.plot(sol.y[0], sol.y[1], sol.y[2], 'b', alpha=0.7)
ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')
ax.set_title('Аттрактор Лоренца')
plt.show()
```

Результат представлен на рисунке 1.

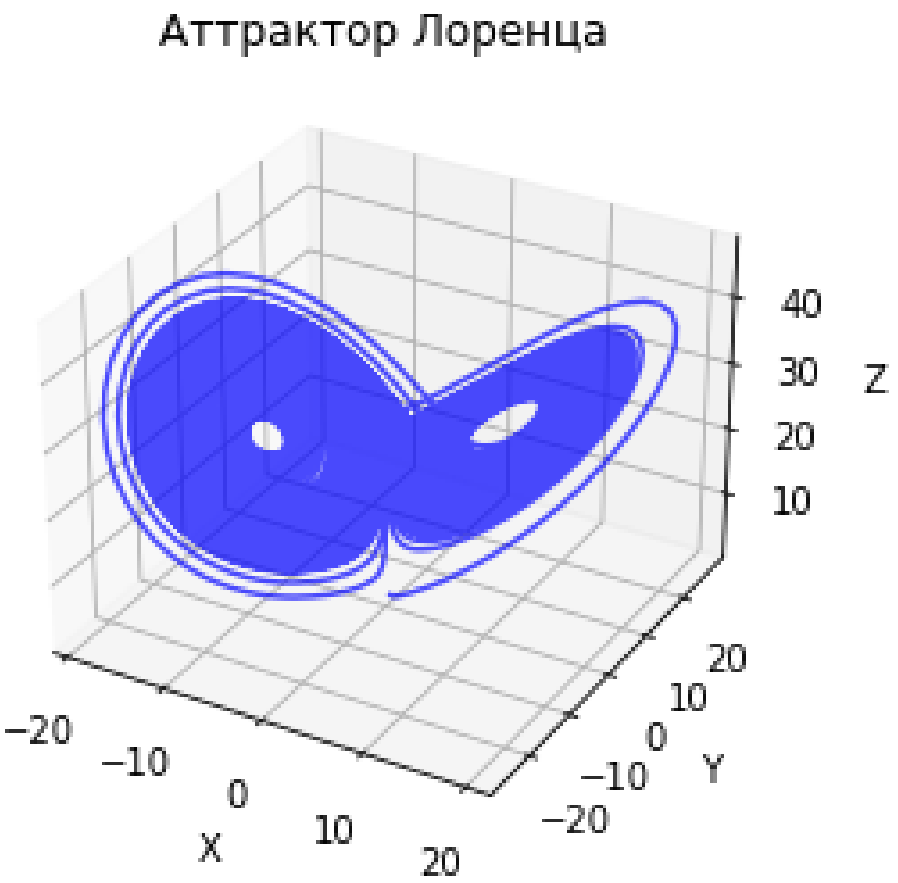

Рисунок 1 — Результат

Аттрактор Лоренца имеет важное прикладное значение в различных областях, связанных с анализом временных данных. Несколько способов, которыми аттрактор Лоренца может быть полезен в работе с временными данными, включают в себя:

1 Прогнозирование временных рядов: Аттрактор Лоренца используется для прогнозирования динамики сложных систем. Путем анализа структуры аттрактора и его параметров можно получить представление о будущем поведении системы, основываясь на её текущем состоянии.

2 Анализ динамических систем: Использование аттрактора Лоренца позволяет исследовать сложные динамические системы и определять характер их поведения. Это может быть полезно для выявления

особенностей временных данных, таких как тренды, циклы, и хаотическое поведение.

3 Обнаружение аномалий: Анализ аттрактора Лоренца может помочь в обнаружении аномалий или необычных паттернов во временных данных. Отклонения от типичного поведения аттрактора могут свидетельствовать о наличии аномальных событий или нештатных ситуаций.

4 Моделирование и управление сложными системами: Аттрактор Лоренца может быть использован для создания математических моделей сложных систем и планирования управления такими системами. Это особенно полезно в таких областях, как физика, метеорология, финансы, и т.д.

В целом, аттрактор Лоренца является мощным инструментом для анализа сложных временных данных и может быть использован для получения глубокого понимания динамики систем, что позволяет принимать более обоснованные решения и прогнозировать будущие события.

Как видно из кода, для аттрактора Лоренца мы используем библиотеки numpy, scipy и matplotlib, чтобы определить дифференциальное уравнение Лоренца, решить его численно и визуализировать результаты.

Для других систем дифференциальных уравнений принцип аналогичен - вы определяете функцию, описывающую систему, задаете начальные условия, решаете уравнения численно и затем визуализируете результаты.

Задание

1 рассчитать численность населения 20 значений за разные периоды времени(, 1000 г.н.э. , 1500 н.э и так по 1970.) и сравнить их с имеющимися в наличии данными . Построить графики расчетных т имеющихся значений

2 рассчитать ВВП мировой 20 значений за разные периоды времени(, 1000 г.н.э. , 1500 н.э и так по 1970.) и сравнить их с имеющимися в наличии данными . Построить графики расчетных т имеющихся значений

3 Для выполнения работы получите свои варианты у преподавателя и решите по аналогичному принципу дифференциальные уравнения согласно вашим вариантам, исследуйте есть ли аттрактор. Затем сделайте выводы на основе полученных данных, не забудьте указать, как на ваш взгляд решение уравнения может пригодиться в работе с временными данными. Варианты представлены в таблице 1. Если уравнение не получается визуализировать , то сделать аттрактор Лоренца , но с другими данными (оценка ниже)

Таблица 1 — Варианты с функциями

Номер варианта	Название функции	Вид функции
1	Уравнение охлаждения Ньютона:	$\mathrm{d}T/\mathrm{d}t=-k(T-T_a)$
2	Уравнение Гельмгольца для распространения звука:	$\nabla^2 p+k^2 p=0$
3	Уравнение распада радиоактивного изотопа:	$dN/dt=-\lambda N$
4	Уравнение колебаний пружинного маятника:	$d^2x/ dt^2+(k/\mathrm{m})x=0$
5	Уравнение экспоненциального роста:	$dN/dt=rN(1-KN)$
6	Уравнение переноса:	$\partial u/\partial t+c(\partial u/\partial x)=0$

7	Уравнение роста бактерий с ограниченными ресурсами:	$dN/dt = rN(1-N/K) - \alpha N^2$
8	Уравнение теплопроводности:	$\partial u/\partial t = \alpha \nabla^2 u$
9	Уравнение Клейна-Гордона для поля скалярных частиц:	$\partial^2\phi/\partial t^2 - c^2\nabla^2\phi + m^2\phi = 0$
10	Уравнение Гаусса для электрического поля:	$\nabla \cdot E = \rho/\varepsilon^0$
11	Уравнение Шредингера для квантовой механики:	$H\psi = i\hbar * \partial\psi/\partial t$
12	Уравнение Бюргерса для гидродинамики:	$\partial u/\partial t + u(\partial u/\partial x) = \nu\partial^2 u/\partial x^2$
13	Уравнение Навье-Стокса для течения несжимаемой жидкости:	$\rho(\partial \mathbf{v}/\partial t + \mathbf{v}\cdot\nabla\mathbf{v}) = -\nabla p + \mu\nabla^2\mathbf{v} + \mathbf{f}$
14	Уравнение Фредгольма в теории операторов:	$(I+K)u = f$
15	Уравнение Кинга-Фергюсона для моделирования химических реакций:	$d\theta/dt = k_1(1-\theta)^{n1} - k_2\theta^{n2}$
16	Уравнение роста популяции	$dN/dt = rN$

	(модель Мальтуса):	
17	Уравнение Больцмана для статистической механики:	$\partial f/\partial t+v\cdot\nabla xf=Q(f,f)$
18	Уравнение Эйнштейна для тензора энергии-импульса:	$G_{\mu\nu}+\Lambda g_{\mu\nu}=8\pi GT_{\mu\nu}\ /\ c^4$
19	Уравнение дисперсии в оптике:	$(\partial^2E/\partial t^2)-c^2(\partial^2E/\partial x^2)=\partial^2E/\partial y^2=\partial^2E/\partial z^2$
20	Уравнение Ван дер Поля	$d^2x/dt^2-\mu(1-x^2)*dx/dt+x=0$