Міністерство освіти і науки України  
Національний технічний університет України

«Київський політехнічний інститут  ім. І. Сікорського»

Кафедра інженерії програмного забезпечення в енергетиці

Практична робота № 4

з курсу: «Проектування кібер-фізичних систем»

**Виконав:**студент 2-го курсу,  
групи ТВ-32  
Щур Роман Ігорович

 Посилання на GitHub репозиторій:

https://github.com/RomanSchur/DCPS\_PR-4

Київ 2025

**Завдання**

Побудувати атрактор Лоренца або іншу модель хаосу та показати, яким чином вона доводить, що велика похибка призводить до зміни показників моделі (погоди), які критично впливають на результат підрахунків та псує коректність передбачення.

**Програмна реалізація**

Програма написана мовою програмування Python і складається з трьох функцій: Lorentz\_attractor, print\_figure та data\_input.

Функція Lorentz\_attractor призначена для обрахунку координат точки в певний момент часу. Для цього у функції використані формули з теоретичного матеріалу практичної роботи. Функція приймає значення попередньої точки та момент часу, а потім, використовуючи формули, виконує обчислення і повертає знайдені координати у функцію data\_input.

Функція data\_input, в свою чергу, призначена для отримання початкових координат точки (користувач вводить координати на свій розсуд) та виклику функцій Lorentz\_attractor і print\_figure. У функції також створено цикл для формування групи точок, які будуть використані для побудови атрактора. У функції є три масиви, кожен з яких відповідає за окрему координату (x, y, z).

Функція print\_figure приймає три масиви координат, запитує у користувача назву файлу для подальшого збереження графіка та будує 3D графік, використовуючи бібліотеку matplotlib.pyplot. Після цього вона зберігає цей графік за допомогою savefig, використовуючи введену назву файлу.

Також створено unit\_test , який призначений для перевірки коректності роботи функції Lorentz\_attractor (тест показав,що функція працює коректно).

Табл. №1 – Програмний код завдання.

import matplotlib.pyplot as plt  
def Lorentz\_attractor(x, y, z, time):  
 sigma = 10 # σ  
 lambdA = 28 # λ  
 beta = 8/3 # b  
  
 dx = sigma \* (y - x)  
 dy = x \* (lambdA - z) - y  
 dz = x \* y - beta \* z  
  
 x\_new = x + dx \* time  
 y\_new = y + dy \* time  
 z\_new = z + dz \* time  
  
 return x\_new, y\_new, z\_new  
  
def print\_figure(res\_x,res\_y,res\_z):  
 filename=input("Введіть назву файлу для збереження графіка (.png): ")  
  
 fig = plt.figure(figsize=(15, 10))  
 ax = fig.add\_subplot(111, projection='3d')  
 ax.plot(res\_x, res\_y, res\_z, lw=1, color='orange')  
 ax.set\_title("Аттрактор Лоренца", fontsize=20)  
 ax.set\_xlabel("X")  
 ax.set\_ylabel("Y")  
 ax.set\_zlabel("Z")  
 plt.savefig(filename, dpi=500)  
 plt.show()  
  
def data\_input():  
 x = float(input("Введіть значення для початкової кординати Х: "))  
 y = float(input("Введіть значення для початкової кординати Y: "))  
 z = float(input("Введіть значення для початкової кординати Z: "))  
 dt = 0.01  
  
 res\_x = []  
 res\_y = []  
 res\_z = []  
  
 for i in range(10000):  
 x, y, z = Lorentz\_attractor(x, y, z, dt)  
 res\_x.append(x)  
 res\_y.append(y)  
 res\_z.append(z)  
 print\_figure(res\_x,res\_y,res\_z)  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 data\_input()

Табл. №2 – Unit\_test для функції програми.

import unittest  
from PR4 import Lorentz\_attractor  
class TestLorentzAttractor(unittest.TestCase):  
 def test\_Lorentz\_attractor(self):  
 x, y, z = 1.0, 1.0, 1.0  
 dt = 0.01  
 x\_new, y\_new, z\_new = Lorentz\_attractor(x, y, z, dt)  
 self.assertAlmostEqual(x\_new, 1.0, delta=0.1)  
 self.assertAlmostEqual(y\_new, 1.26, delta=0.1)  
 self.assertAlmostEqual(z\_new, 0.99, delta=0.1)  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 unittest.main()

**Результати виконання**

Зображення, що містить малюнок, ескіз, схема, дизайн

Вміст, створений ШІ, може бути неправильним.

Рис №1 – Графік атрактора при початкових кординатах х=1 у=1 z=1

Зображення, що містить малюнок, ескіз, схема, дизайн

Вміст, створений ШІ, може бути неправильним.

Рис №2 – Графік атрактора при початкових кординатах х=1.01 у=1 z=1

Як висновок зазначу, що як і очікувалось,навіть невелика зміна початкових даних, призводить до кардинальних змін.