

Чернівецький національний університет імені Юрія Федьковича
Навчально-науковий інститут фізико-технічних та комп'ютерних наук
Відділ комп'ютерних технологій
Кафедра математичних проблем управління і кібернетики

Звіт
про виконання лабораторної роботи №5

Тема: “ Апроксимації гладких ліній вищих порядків. Фрактали.”

з дисципліни
“ Літня обчислювальна практика” Варіант
№ 13

Виконав:
ст. гр. 241 Українець Д.
Прийняв:
доц. Лазорик В. В.

Чернівці – 2025

Лабораторна роботи №5.

Виконання лабораторної роботи

1. Зайти в свій обліковий запис на github.com.
2. Клонувати репозиторій <https://classroom.github.com/a/Berdcd6> в свій обліковий запис на github.com.
3. Розв'язати завдання.
4. Вихідних код записати в створений репозиторій.

Завдання

1. Створити проект Windows Form (C++ або C#).
2. Розробити програму вирішення задач згідно варіантів(варіанти згідно списку) .

Задача 1.1

На площині задано дві точки $P1(X1,Y1)$, $P2(X2,Y2)$ та два вектора $V1(Vx1,Vy1)$, $V2(Vx2,Vy2)$. Розробити програму для відображення параметричної лінії Ерміта від точки $P1$ до точки $P2$ з початковим вектором $V1$ та кінцевим вектором $V2$.

Блок-схема:

Початок програми

↓

Виклик функції `draw_hermite(p1, p2, v1, v2, scale)`

↓

Обчислити параметричні значення t від 0 до 1

↓

Обчислити базисні функції Ерміта:

- $h_1 = 2t^3 - 3t^2 + 1$
- $h_2 = -2t^3 + 3t^2$
- $h_3 = t^3 - 2t^2 + t$
- $h_4 = t^3 - t^2$

↓

Обчислити координати кривої:

- $x = h_1 \cdot P1.x + h_2 \cdot P2.x + h_3 \cdot V1.x + h_4 \cdot V2.x$
- $y = h_1 \cdot P1.y + h_2 \cdot P2.y + h_3 \cdot V1.y + h_4 \cdot V2.y$

↓

Масштабувати координати (x, y) та точки $P1, P2, V1, V2$



Побудувати графік:

- Побудова кривої Ерміта
- Відображення точок P1, P2 (червоні)
- Відображення векторів V1, V2 (зелені/помаранчеві стрілки)



Показати графік (plt.show())



Кінець програми

```
import numpy as np
import matplotlib.pyplot as plt

def compute_hermite_points(p1, p2, v1, v2, resolution=100):
    t = np.linspace(0, 1, resolution)
    h = [
        2 * t**3 - 3 * t**2 + 1,
        -2 * t**3 + 3 * t**2,
        t**3 - 2 * t**2 + t,
        t**3 - t**2
    ]
    x_vals = h[0] * p1[0] + h[1] * p2[0] + h[2] * v1[0] + h[3] * v2[0]
    y_vals = h[0] * p1[1] + h[1] * p2[1] + h[2] * v1[1] + h[3] * v2[1]
    return x_vals, y_vals

def draw_hermite(p1, p2, v1, v2, scale=1.5):
    x, y = compute_hermite_points(p1, p2, v1, v2)

    x *= scale
    y *= scale

    points = {
        'P1': np.array(p1) * scale,
        'P2': np.array(p2) * scale,
        'V1': np.array(v1) * scale,
        'V2': np.array(v2) * scale
    }

    plt.figure(figsize=(8, 6))
    plt.plot(x, y, 'b', label='Крива Ерміта')
    plt.scatter([points['P1'][0], points['P2'][0]], [points['P1'][1], points['P2'][1]], color='red', label='Точки')
    plt.quiver(*points['P1'], *points['V1'], angles='xy', scale_units='xy', scale=1, color='green', label='Вектор V1')
    plt.quiver(*points['P2'], *points['V2'], angles='xy', scale_units='xy', scale=1, color='orange', label='Вектор V2')

    plt.title("Гладка крива Ерміта")
    plt.legend()
    plt.axis('equal')
    plt.grid(True)
    plt.show()

# Демонстрація
if __name__ == "__main__":
    draw_hermite((0, 1), (4, 4), (2, -5), (4, -2), scale=-3)
```

Рис. 1 – Код завдання 1

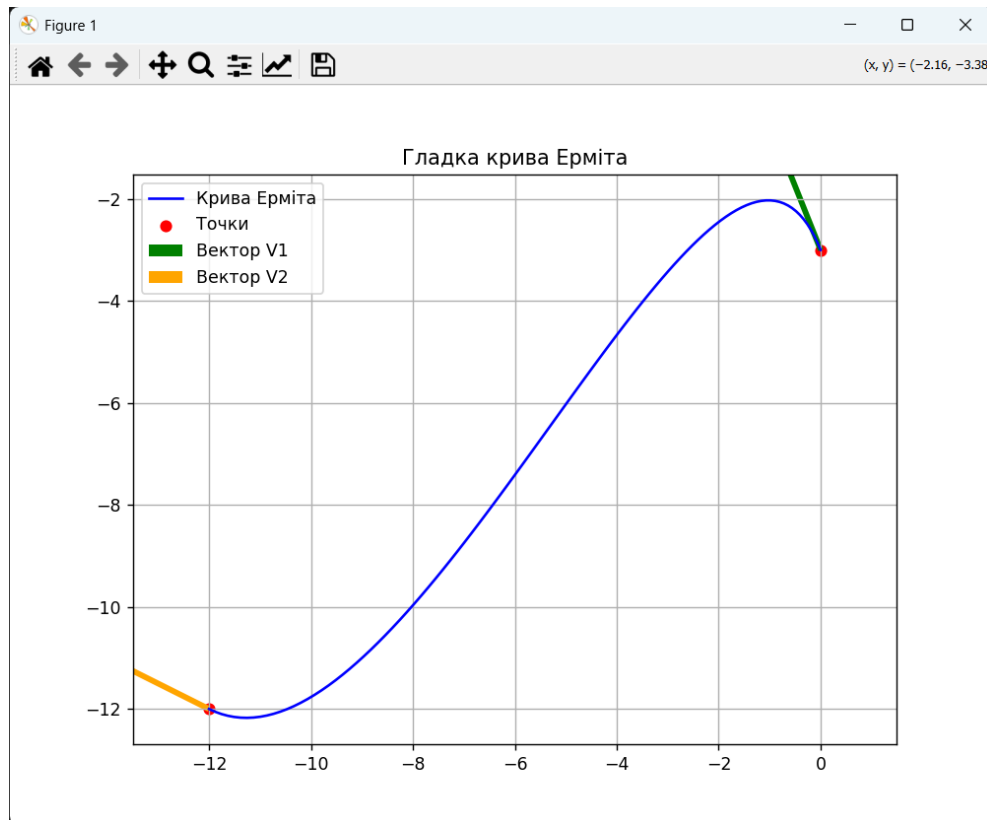


Рис. 2 – Результат виконання завдання 1

Задача 2.5

Для «пучка» з A відрізків з одним спільним кінцем реалізувати фрактали «Папороть» порядку K з різним кольором для кожного «листка».

Блок-схема:

Початок програми

↓

Запуск функції `draw()`

↓

Для кожного кута з 6 обертів (листіків):

↓

Виклик `fern(n)` для генерації точок

↓

Для кожної точки:

- Генерується випадкове число
- Застосовується одна з 4 афінних трансформацій
- Зберігається нова точка

↓

Обертання точки на відповідний кут

↓
Виведення точок певного кольору

↓
Повторити для всіх 6 листків

↓
Сховати осі та показати графік

↓
Кінець програми

```
import matplotlib.pyplot as plt
import numpy as np
import random

plt.figure(figsize=(6, 6))
for i, a in enumerate(np.linspace(0, 2*np.pi, 6, endpoint=False)):
    x = y = 0
    points = []
    for _ in range(1500):
        r = random.random()
        if r < 0.01: x, y = 0, 0.16*y
        elif r < 0.86: x, y = 0.85*x + 0.04*y, -0.04*x + 0.85*y + 1.6
        elif r < 0.93: x, y = 0.2*x - 0.26*y, 0.23*x + 0.22*y + 1.6
        else: x, y = -0.15*x + 0.28*y, 0.26*x + 0.24*y + 0.44
        points.append((x, y))
    pts = np.array(points)
    rot = np.array([[np.cos(a), -np.sin(a)], [np.sin(a), np.cos(a)]])
    rot_pts = pts @ rot
    plt.plot(rot_pts[:,0], rot_pts[:,1], '.', ms=0.5, color=plt.cm.hsv(i/6))

plt.axis('off')
plt.show()
```

Рис. 3 – Код завдання 2

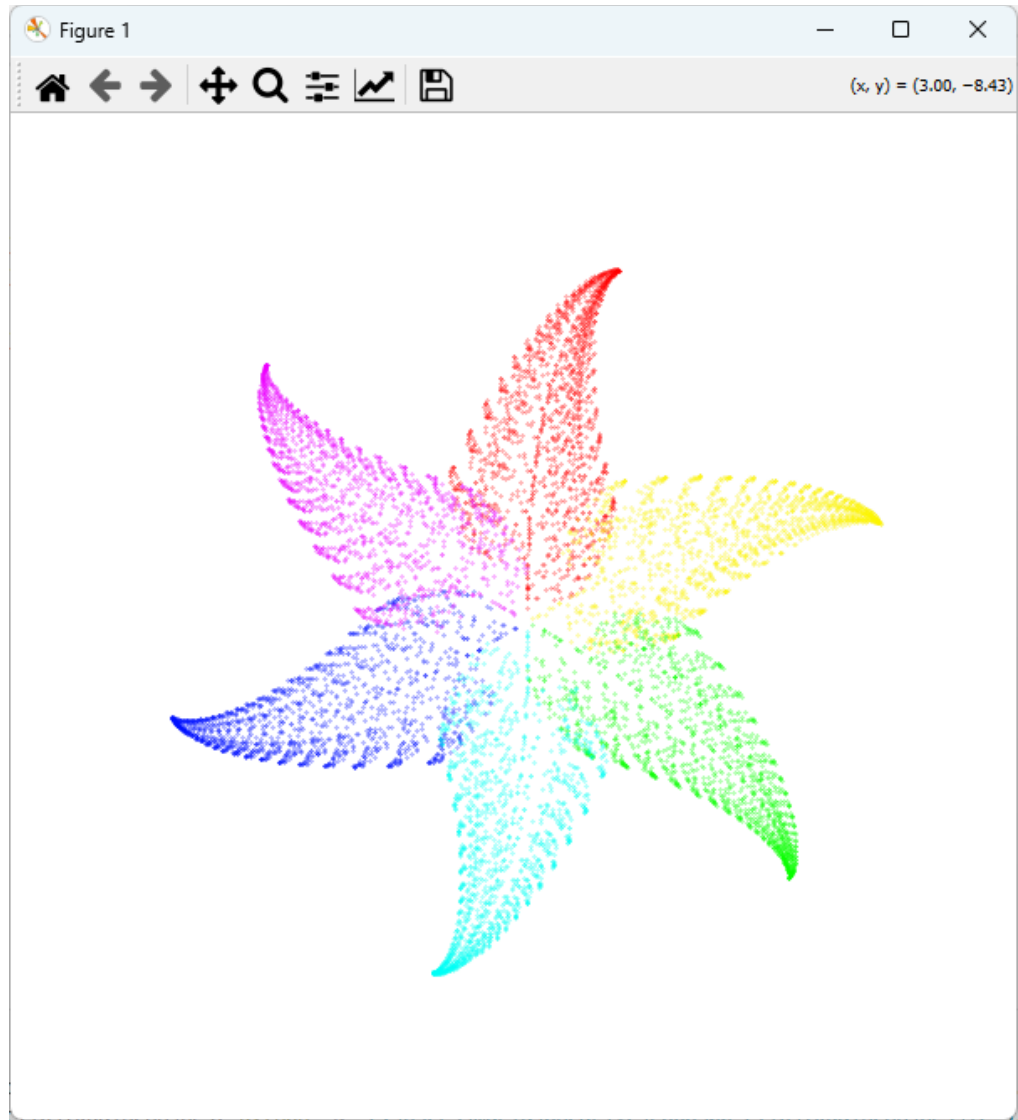


Рис. 4 – Результат виконання завдання 2

Висновок: Під час виконання лабораторної роботи було реалізовано два графічних завдання на мові Python. Перше — побудова кривої Ерміта між двома точками з урахуванням напрямків (векторів). Друге — генерація фракталу «Папороть» у вигляді пучка з кількох листків різного кольору.