МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
Навчально-науковий інститут прикладного системного аналізу

ЛАБОРАТОРНА РОБОТА №3
з дисципліни «Просторове моделювання та візуалізація»
на тему: «Побудова фракталів»
Варіант №7

Виконала:
Аспірантка гр. КН-в31ф
Старовойт Тетяна Василівна
Перевірила:
д.т.н., доц. Аушева Н.М.

Київ – 2025

Побудова фракталів

Мета: *володіти навичками побудови фракталів методами IFS, L-систем та алгебраїчних рівнів.*

Частина А. Побудова фракталу методом IFS (Iterated Function System)

Теоретичні відомості:

Метод IFS (Iterated Function System) — це спосіб побудови фракталів шляхом багаторазового застосування афінних перетворень до точок площини. Кожне перетворення визначається множиною коефіцієнтів, які визначають масштабування, обертання, зсув і ймовірність вибору перетворення при кожній ітерації.

Згідно з варіантом 7, надано один рядок параметрів (тобто одне афінне перетворення):

- a=0.745455
- b=−0.45909
- c=1.460279
- d=0.406061
- e=0.887121
- f=0.691072
- p=0.912675

Теоретичні основи.

Фрактал — це геометрична фігура, яка володіє властивістю **самоподібності**, тобто її частини мають подібну структуру на різних масштабах. Це може бути точна або статистична самоподібність.

Основні властивості фракталів:

- **Самоподібність (Self-similarity):** форма об'єкта повторюється при масштабуванні.
- **Деталізованість:** фрактали мають складну структуру, яка не зникає при збільшенні масштабу.
- **Фракційна розмірність:** фрактали мають дробову розмірність, яка відрізняється від традиційних евклідових (лінія — 1, площа — 2, тіло — 3).

- **Рекурсивність:** багато фракталів визначаються рекурсивними або ітеративними правилами.

Класифікація фракталів

1. **Геометричні (детерміновані) фрактали**:

- Побудовані за чітко визначеним математичним правилом.
- Приклади: крива Коха, крива Дракона, трикутник Серпінського.

2. **Стохастичні фрактали**:

- Створюються за ймовірнісними правилами, мають статистичну самоподібність.
- Приклади: моделювання берегової лінії, хмар, дерев.

3. **Алгебраїчні фрактали**:

- Визначаються алгебраїчними або комплексними рівняннями (наприклад, множина Мандельброта, множина Жулія).

4. **Фрактали на основі Ітераційних Систем Функцій (IFS)**:

- Створюються шляхом багаторазового застосування афінних перетворень з ймовірностями.

```python
# Fractal construction using the IFS method
import matplotlib.pyplot as plt
import random

# Transformation parameters
transformations = [
    {
        "a": 0.745455, "b": -0.45909, "c": 1.460279,
        "d": 0.406061, "e": 0.887121, "f": 0.691072,
        "p": 0.912675
    }
]

# Point generation
x, y = 0.0, 0.0
X, Y = [], []
for _ in range(10000):
    t = random.choices(transformations, weights=[t["p"] for t in transformations])[0]
    x, y = t["a"] * x + t["b"] * y + t["c"], t["d"] * x + t["e"] * y + t["f"]
    X.append(x)
    Y.append(y)

plt.figure(figsize=(6, 6))
plt.scatter(X, Y, s=0.2, color='black')
plt.axis('equal')
plt.title("Fractal IFS (option 7)")
plt.show()
```

На графіку нижче зображено побудований фрактал за методом IFS. Візуально видно спіралеподібну структуру з фокусом у центральній області.

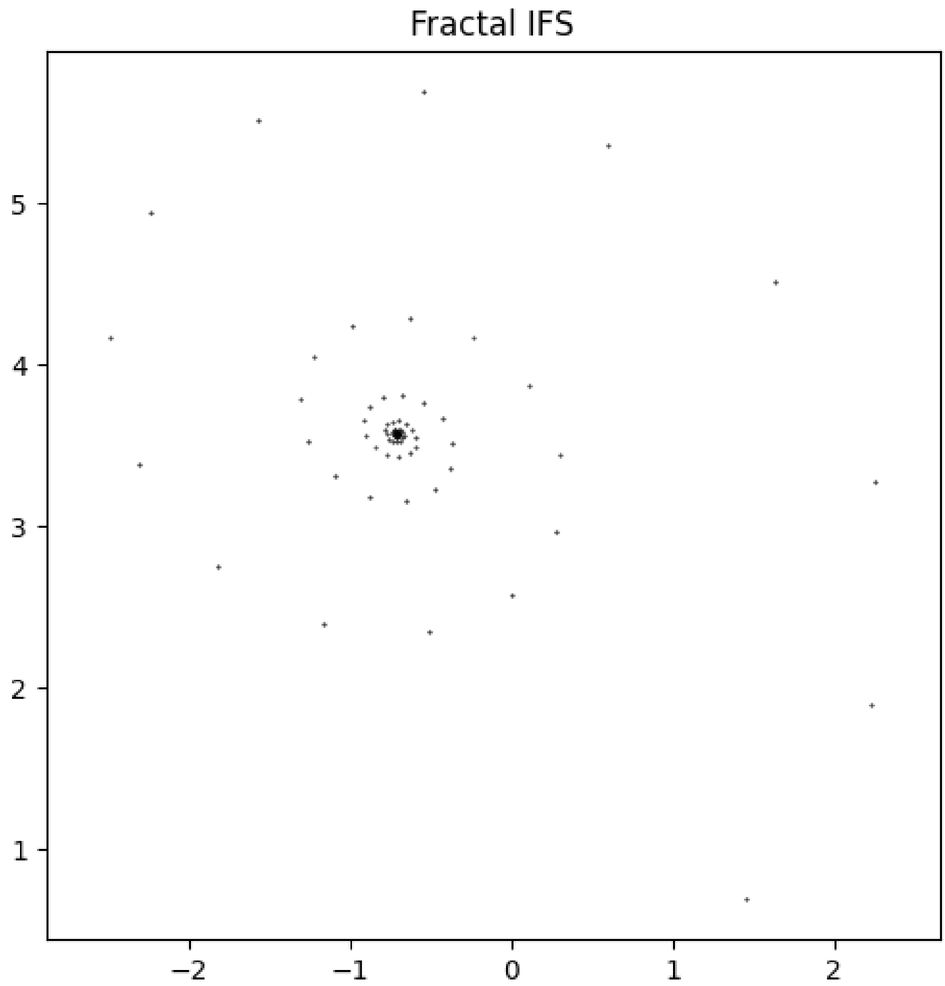


Рисунок 1. – Фрактал за методом IFS

Застосування методу IFS у гідравлічному моделюванні

У гідравлічному моделюванні часто потрібні синтетичні топології для:

- тестування алгоритмів оптимізації;
- перевірки моделей втрат води;
- генерації сценаріїв у відсутності повних геоданих.

Метод IFS може генерувати синтетичні мережі, які зберігають фрактальну логіку зростання:

- вузли — центри перетворень;
- труби — з'єднання між ітераціями;
- кожен наступний рівень — менш масштабна копія попереднього, з модифікованим кутом/довжиною.

IFS може слугувати моделлю для визначення зон стратегічного моніторингу, якщо:

- сенсори/датчики розміщуються в ієрархічній системі;
- кожне нове “розгалуження” містить сенсор з певною ймовірністю.

Це дозволяє імітувати фрактальну структуру датчиків, що забезпечує оптимальне покриття з мінімальними ресурсами.

Частина Б. Побудова фракталу методом L-System (Lindenmayer System)

Метод L-System базується на рекурсивному переписуванні символів за певними правилами (продукціями). На кожному кроці ітерації початковий рядок (аксіома) розширюється згідно з правилами заміни, утворюючи все складніший ланцюг інструкцій, який інтерпретується як послідовність команд для «черепашки» — графічного інтерпретатора.

Використані параметри:

- **Аксіома (початковий рядок):** +R Цей символ означає: «поверни праворуч на кут» і почни з символу R.
- **Кут повороту:** angle = 60°Використовується для повороту направо (+) або наліво (−) у площині побудови.

Кількість ітерацій: iterations = 3

З кожною ітерацією розмірність фрактала зростає експоненціально.

Алгоритм побудови:

1. **Генерація рядка інструкцій:**
    - Кожна ітерація проходить по рядку і замінює кожен символ згідно з правилами.
    - Після 3 ітерацій аксіома +R перетворюється на складну послідовність інструкцій (довжиною в сотні символів).
2. **Інтерпретація команд (черепашка):**
    - F або G: рух вперед (додається нова точка);
    - +: поворот вправо на angle;
    - -: поворот вліво на angle;
    - [і]: збереження / повернення позиції (не використовувались у цьому прикладі, але є в більш складних L-System).
3. **Обчислення координат:**

- Координати нової точки обчислюються через cos/sin кута;
- Кожен крок додає точку до coords;
- Графік будується як ламана лінія (plt.plot(xs, ys)).

```
import matplotlib.pyplot as plt
import math

# L-System settings
axiom = "X"
rules = {
    "F": "FF",
    "X": "F[+X]F[-X]+X"
}
angle = 20
iterations = 5
```

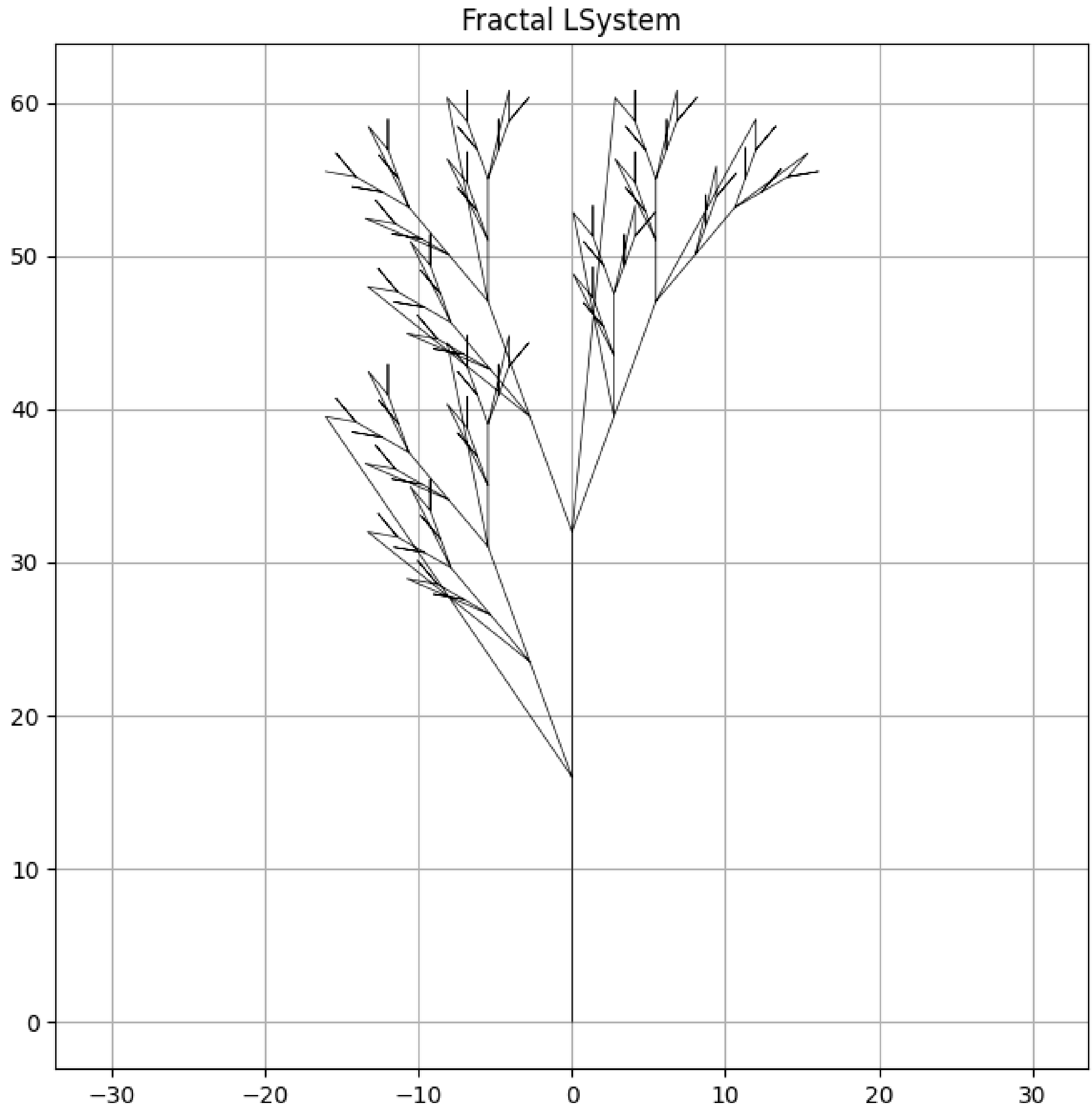
```
def generate_lsystem(axiom, rules, depth):
    current = axiom
    for _ in range(depth):
        next_seq = ""
        for ch in current:
            next_seq += rules.get(ch, ch)
        current = next_seq
    return current
```

На виході отримано розгалужену геометричну структуру, яка демонструє самоподібність і локальну симетрію. Вона нагадує сніжинку Коха або дерево Серпінського, але з більш довільною гіллястістю.

Ця структура імітує фрактальну мережу розподілу — аналог водопровідної системи з розгалуженням на багато споживачів.

```
sequence = generate_lsystem(axiom, rules, iterations)
points = draw_lsystem(sequence, angle, step=1)


X, Y = zip(*points)
plt.figure(figsize=(8, 8))
plt.plot(X, Y, color='black', linewidth=0.5)
plt.title("Fractal LSystem")
plt.axis('equal')
plt.grid(True)
plt.show()
```



Практичне застосування:

- Генерація синтетичних водопровідних топологій;
- Моделювання розширень мереж або невідомих ділянок;
- Створення бенчмарків для GNN (graph neural networks);
- Імітація розгалуженої інфраструктури, де немає реальної карти.

Частина С. Algebraic fractal (Julia or Mandelbrot set)

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


width, height = 800, 800
zoom = 1
move_x, move_y = 0.0, 0.0
max_iter = 300
c = complex(-0.7, 0.27015)

# Complex plane grid
x = np.linspace(-1.5, 1.5, width)
y = np.linspace(-1.5, 1.5, height)
X, Y = np.meshgrid(x, y)
Z = X + 1j * Y

# Iterative process
def julia(z, c, max_iter):
    output = np.zeros(z.shape, dtype=int)
    for i in range(max_iter):
        z = z*z + c
        mask = np.abs(z) < 1000
        output += mask
    return output


plt.figure(figsize=(6, 6))
plt.imshow(julia(Z, c, max_iter), extent=(-1.5, 1.5, -1.5, 1.5), cmap='inferno')
plt.title("Plural Julia")
plt.colorbar()
plt.show()
```

Множина Жулія (Julia Set) базується на повторному застосуванні комплексного відображення:

$$Z_{k+1} = Z_k^2 + C$$

- $Z \in \mathbb{C}$ — змінна на комплексній площині,
- $C \in \mathbb{C}$ — фіксований комплексний параметр,
- послідовність Z_k або прямує до нескінченності, або залишається обмеженою.

Множина Жулія — це геометричне відображення початкових значень Z, які не втікають у нескінченність при нескінченному ітеративному застосуванні функції.

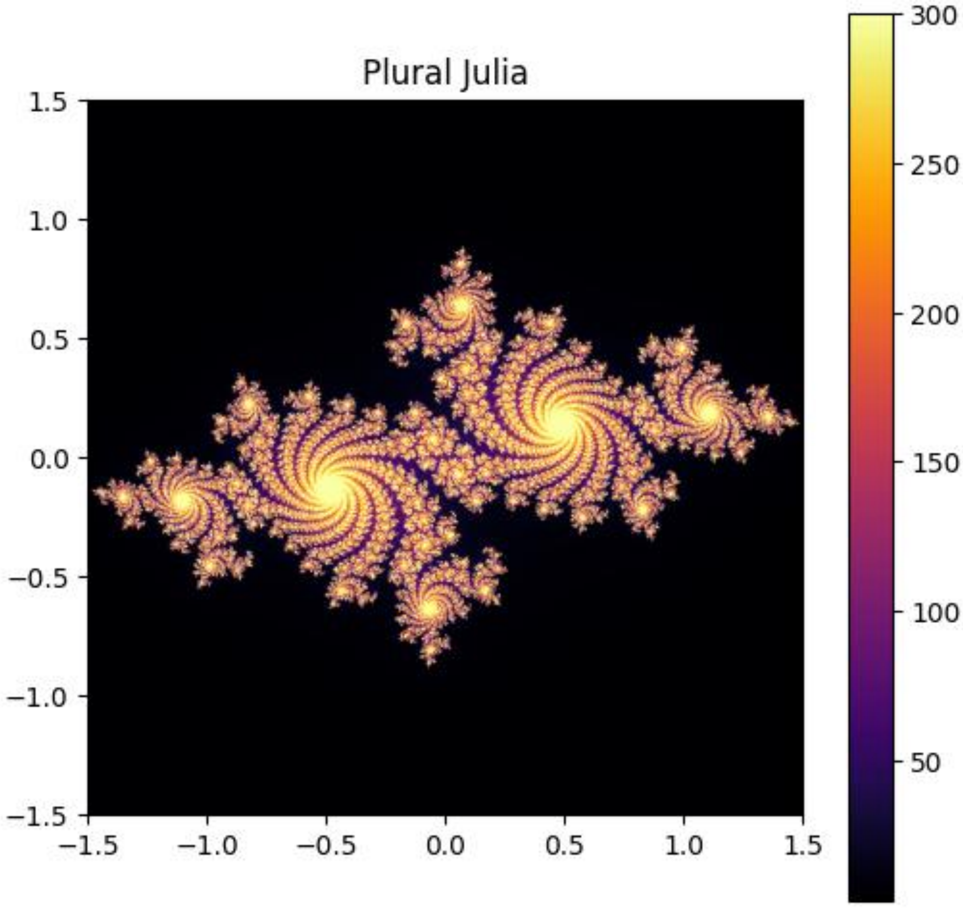
Параметри реалізації:

- **Комплексна константа:**

C=0.36+0.36i — задає форму і складність фракталу.

Область побудови: $\boldsymbol{X \in [-2, 1], Y \in [-1.5, 1.5]}$

— двовимірна сітка на комплексній площині.



Результат: Побудована складна симетрична фрактальна структура з самоподібністю на всіх масштабах.

Висновки:

У результаті виконання лабораторної роботи були реалізовані три методи побудови фракталів: метод ітераційної функціональної системи (IFS), метод формальної граматики L-System та алгебраїчний фрактал на основі множини Жулія. Для кожного з підходів було розроблено відповідну програмну реалізацію з візуалізацією фрактальної структури.

1. **Метод IFS** дозволяє будувати самоподібні структури за допомогою афінних перетворень. На прикладі варіанту 7 було реалізовано фрактал, параметри якого задаються коефіцієнтами лінійних перетворень. Цей метод особливо ефективний для генерації геометрично симетричних форм, таких як дерева, папороті, сніжинки тощо.

2. **Метод L-System** реалізує фрактали як систему продукцій, що імітує ріст рослинних структур. Створений фрактал має виражену розгалужену топологію, що відображає принципи побудови природних об’єктів. Цей

метод дозволяє інтуїтивно контролювати складність структури шляхом зміни правил граматики або кількості ітерацій.

3. **Множина Жулія** як приклад алгебраїчного фракталу демонструє складну топологію на комплексній площині, де кожна точка визначає динаміку ітеративного процесу. Зафіксоване значення C=0.36+0.36i дозволило отримати візуально складну фрактальну структуру з високою щільністю самоподібних деталей.

4. Особливу цінність фрактальні методи мають у задачах моделювання природних, технічних та інженерних систем. Наприклад, множина Жулія потенційно може бути використана як аналітична модель для моделювання нестійкої поведінки реагентів у трубопровідних мережах або апроксимації зон, структура яких невідома.

5. Візуалізація отриманих фракталів підтверджує здатність простих математичних правил породжувати складну та самоподібну геометрію, що має як естетичну, так і прикладну цінність у галузях комп’ютерної графіки, біоінженерії, гідравлічного моделювання та штучного інтелекту.