

Міністерство освіти і науки України  
Український державний університет науки і технологій


Факультет Комп'ютерні технології та системи  
Кафедра Комп'ютерні інформаційні технології

**Пояснювальна записка**  
до кваліфікаційної роботи  
магістра


на тему: «Визначення фрактальних розмірностей часткових геометричних  
фракталів»

за освітньою програмою **12 Інженерія програмного забезпечення**  
зі спеціальності: **121 Інженерія програмного забезпечення**

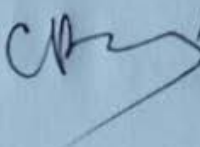
Виконав: студент групи ПЗ2222:

 / Віра МАСЛЮК /

Керівник:

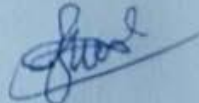
 / Віктор ШИНКАРЕНКО /

Нормоконтролер:

 / Світлана ВОЛКОВА /

Засвідчую, що у цій роботі немає  
запозичень з праць інших авторів  
без відповідних посилань.

Студент



Дніпро – 2024 рік

Ministry of Education and Science of Ukraine

Ukrainian State University of Science and Technologies

Faculty Computer technologies and systems  
Department Computer information technology

## **Explanatory Note**

to Master's Thesis

on the topic: «Determination of fractal dimensions of partial geometric fractals»

according to educational curriculum **12 software engineering**  
in the Speciality: **121 software engineering**

Done by the student of the group  
PZ2222:

/ Vira MASLIUK /

Scientific Supervisor:

/ Viktor SHYNKARENKO /

Normative controller:

/ Svitlana VOLKOVA /

Міністерство освіти і науки України  
Український державний університет науки і технологій

Факультет: Комп'ютерних технологій і систем  
Кафедра: Комп'ютерні інформаційні технології  
Рівень вищої освіти: магістр  
Освітня програма: Інженерія програмного забезпечення  
Спеціальність: Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ  
Завідувач кафедри КІТ  
Вадим ГОРЯЧКІН  
\_\_\_\_\_

### ЗАВДАННЯ

На кваліфікаційну роботу Магістр  
студенту МАСЛЮК ВІРІ ОЛЕКСІЇВНІ

1. Тема дипломної роботи: Визначення фрактальних розмірностей часткових геометричних фракталів.

Керівник роботи: ШИНКАРЕНКО ВІКТОР ІВАНОВИЧ

затверджені наказом 1196 ст від 05.12.2022 року

2. Строк подання студентом роботи 25.01.2024 року
3. Вихідні дані до дипломної роботи:  
пояснювальна записка, створений додаток для виконання досліджень.
4. Зміст пояснювальної записки (перелік питань до розробки):
  - 4.1. огляд предметної галузі;
  - 4.2. методи дослідження та їх обґрунтування;
  - 4.3. проектування й розробка;

- 4.4. дослідження розмірності часткових фракталів;
- 4.5. висновки.
5. Перелік демонстраційного матеріалу:
- 5.1. презентація;
- 5.2. демонстраційне відео.

### КАЛЕНДАРНИЙ ПЛАН

| №  | Назва етапів кваліфікаційної роботи  | Строк виконання етапів | Примітка |
|----|--|------------------------|----------|
| 1  | Вступ  | 03.03.23 – 14.04.23    |          |
| 2  | Аналіз сучасного стану дослідження проблеми за науковими літературними джерелами                                       | 15.04.23 – 22.08.23    | 10%      |
| 3  | Аналіз сучасного стану програмно-апаратного забезпечення, яке потребує вдосконалення для вирішення проблем дослідження | 23.08.23 – 18.10.23    |          |
| 4  | Постановка задачі, технічне завдання   | 19.10.23 – 20.10.23    | 30%      |
| 5  | Розробка інструментальних засобів дослідження  | 21.10.23 – 13.11.23    |          |
| 6  | Виконання досліджень   | 14.11.23 – 28.11.23    | 60%      |
| 7  | Оформлення пояснювальної записки   | 29.11.23 – 15.01.24    |          |
| 8  | Розробка демонстраційних матеріалів  | 16.01.24 – 17.01.24    | 100%     |
| 9  | Подання кваліфікаційної роботи до кафедри  | 20.01.24               |          |
| 10 | Захист кваліфікаційної роботи на засіданні Екзаменаційної комісії  | 25.01.24               |          |

Студент \_\_\_\_\_ / Віра МАСЛЮК /

Керівник роботи \_\_\_\_\_ / Віктор ШИНКАРЕНКО /

## РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи магістра.

66с., 41 рис., 18 табл., 4 додатки, 14 джерел.

Об'єктом дослідження є поведінка розмірності часткових фракталів при різній кількості внесених дефектів.

Основною метою дослідження є оцінка впливу дефектів на фрактальну розмірність часткових фракталів. Дослідження може спрямовуватися на виявлення закономірностей або залежностей між кількістю дефектів і зміною фрактальної розмірності. Також може бути важливим визначення методу додавання дефектів, які мають значущий вплив на структуру фракталів.

Пояснювальна записка складається зі вступу, 4 розділів, висновків, бібліографічного списку та 4 додатків:

- вступ – опис загальної суті проблеми дослідження, 2 сторінки;
- огляд предметної галузі – опис поточного стану проблеми відносно інших досліджень, постановка задачі до дослідження, 12 сторінок;
- методи дослідження та їх обґрунтування – опис використаних методів дослідження, 17 сторінок;
- проектування й розробка – описує кроки розробки та планування проєкту, 10 сторінок;
- дослідження розмірності часткових фракталів – опис ходу досліджень та їх результатів, 19 сторінок;
- додатки містять: тези конференції, технічне завдання, керівництво користувача, текст програми.

Ключові слова: ФРАКТАЛ, FRACTAL, ФРАКТАЛЬНА РОЗМІРНІСТЬ, ЧАСТКОВИЙ ФРАКТАЛ, WINDOWS FORMS, C#.

## ЗМІСТ

|   |    |
|---|----|
| РЕФЕРАТ .....   | 1  |
| ВСТУП.....  | 4  |
| 1 ОГЛЯД ПРЕДМЕТНОЇ ГАЛУЗІ.....  | 6  |
| 1.1 Аналіз сучасного стану дослідження розмірності часткових фракталів..... | 6  |
| 1.2 Огляд відомостей про фрактали.....                                      | 6  |
| 1.3 Огляд відомостей про фрактальну розмірність .....                       | 8  |
| 1.4 Аналіз стану програмно-апаратного забезпечення .....                    | 10 |
| 1.5 Аналіз вимог до системи та постановка задачі.....                       | 16 |
| Висновки до першого розділу.....  | 17 |
| 2 МЕТОДИ ДОСЛІДЖЕННЯ ТА ЇХ ОБҐРУНТУВАННЯ .....                              | 18 |
| 2.1 Обґрунтування експериментального методу дослідження .....               | 18 |
| 2.2 Метод представлення фракталу .....                                      | 19 |
| 2.3 Метод розрахунку розмірності.....                                       | 28 |
| Висновки до другого розділу .....   | 33 |
| 3 ПРОЕКТУВАННЯ Й РОЗРОБКА .....   | 34 |
| 3.1 Формалізація задачі .....   | 34 |
| 3.2 Базова архітектура системи .....  | 36 |
| 3.3 Внутрішнє програмування.....  | 37 |
| 3.3.1 Вибір мови та платформи для розробки.....                             | 37 |
| 3.3.2 Вибір формату збереження графічного представлення фракталів....       | 38 |
| 3.3.3 Ієрархія та взаємодія класів системи .....                            | 38 |
| 3.3.4 Використані принципи проектування.....                                | 39 |
| 3.3.5 Використані шаблони проектування .....                                | 40 |
| 3.4 Розробка інтерфейсу користувача.....                                    | 41 |
| 3.5 Тестування та налагодження програми .....                               | 42 |
| Висновки до третього розділу.....   | 44 |
| 4 ДОСЛІДЖЕННЯ РОЗМІРНОСТІ ЧАСТКОВИХ ФРАКТАЛІВ.....                          | 45 |
| 4.1 Підготовка до експериментів.....  | 45 |

|       |   |    |
|-------|---|----|
| 4.2   | Дослідження розмірності часткових фракталів сформованих першим методом .....                                      | 45 |
| 4.2.1 | Вплив ступеня пошкодження першим методом фракталів сніжинки Коха на фрактальну розмірність .....                  | 46 |
| 4.2.2 | Вплив ступеня пошкодження першим методом фракталів квадратного острова на фрактальну розмірність.....             | 47 |
| 4.2.3 | Вплив ступеня пошкодження першим методом фракталів кривої Серпінського на фрактальну розмірність.....             | 49 |
| 4.2.4 | Вплив ступеня пошкодження першим методом фракталів кривої дракона на фрактальну розмірність .....                 | 50 |
| 4.2.5 | Вплив ступеня пошкодження першим методом фракталів кривої Леві на фрактальну розмірність .....                    | 52 |
| 4.3   | Дослідження розмірності часткових фракталів сформованих другим методом .....                                      | 53 |
| 4.3.1 | Залежність фрактальної розмірності від ступеня пошкодження другим методом фракталів сніжинки Коха .....           | 53 |
| 4.3.2 | Залежність фрактальної розмірності від ступеня пошкодження другим методом фракталів квадратного острова Коха..... | 55 |
| 4.3.3 | Залежність фрактальної розмірності від ступеня пошкодження другим методом фракталів кривої Серпінського .....     | 57 |
| 4.3.4 | Залежність фрактальної розмірності від ступеня пошкодження другим методом фракталів кривої дракона.....           | 58 |
| 4.3.5 | Залежність фрактальної розмірності від ступеня пошкодження другим методом фракталів кривої Леві.....              | 60 |
| 4.4   | Результати експериментів .....  | 61 |
|       | Висновки до четвертого розділу .....  | 62 |
|       | ВИСНОВКИ.....   | 63 |
|       | БІБЛІОГРАФІЧНИЙ СПИСОК .....  | 64 |
|       | ДОДАТКИ.....  | 66 |

## ВСТУП

Сучасна наука активно знаходить напрямки застосування для математичних структур з фрактальними ознаками. Фрактальна розмірність є однією з важливих властивостей, для дослідження фракталів. Вона має неоднозначне трактування та не один спосіб розрахунку, кожен з яких може описати досліджуваний об'єкт з різних сторін. Актуальність роботи полягає в недостатній вивченості розмірностей фігур, що відносяться до змінних деяким чином регулярних фракталів.

Завдання дослідження:

- аналіз впливу дефектів: вивчення, як різні типи та кількості дефектів впливають на структуру фракталів;
- генерація часткових фракталів: розробка методів формування часткових фракталів та внесення дефектів;
- розрахунок фрактальної розмірності: визначення змін у фрактальній розмірності при різних рівнях пошкодженості регулярних фракталів;
- аналіз результатів: перевірка отриманих результатів через проведення експериментів та порівняння їх із наявними теоретичними концепціями.

Основним методом дослідження є проведення деякої кількості експериментів з генерації часткових фракталів та розрахунок їх розмірностей, шляхом аналізу графічного зображення досліджуваної фігури.

Наукова новизна роботи представлена в двох аспектах:

- 1) введено і описане поняття часткового фракталу, який за методом свого формування не є одним і тим самим з випадковим фракталом;
- 2) проведення дослідження поведінки фрактальної розмірності в залежності з ступенем пошкодженості регулярного фракталу.

Результати дослідження можуть мати практичне застосування в різних областях, включаючи матеріалознавство, технології виробництва, та природознавство.

Реалізація апробації результатів передбачає представлення висновків на науковій конференції для обговорення в науковій громадськості. Це допоможе підвищити відомість та визнання в науковому середовищі.

## 1 ОГЛЯД ПРЕДМЕТНОЇ ГАЛУЗИ

### 1.1 Аналіз сучасного стану дослідження розмірності часткових фракталів

Застосування фракталів на сьогоднішній момент є дуже широким. У дослідженнях та виробництві іноді виникають проблеми опису математичних залежностей або геометричних форм не правильної структури. В більшості випадків можна просто округлювати дані до більш правильних та звичних, і виконувати розрахунки з деякою похибкою. Але якщо мова йде про процеси, що чуттєві до похибок, то актуальним стає вивчення фрактальних множин неправильної структури.

Серед популярних застосувань знань про фрактали, є:

- 1) опис і аналіз складних математичних структур, що не можуть бути описані класичними методами;
- 2) створення реалістичних візуальних ефектів в комп'ютерній графіці;
- 3) опис складних фізичних явищ;
- 4) вивчення біологічних систем та аналіз медичних зображень;
- 5) дослідження коливань цін та фінансових ринків;
- 6) покращення систем управління в енергетиці та логістиці;
- 7) моделювання складних молекулярних структур та реакцій;
- 8) інше.

Кожна з представлених галузей має ряд досліджень, що представляють собою корисне або ні застосування фракталів та їх властивостей. Не всі результати таких досліджень на даний час можуть мати раціональне застосування, але кожне з них несе деяку потенційну користь для науки.

### 1.2 Огляд відомостей про фрактали

За Мальдебротом [1] термін «фрактал» походить від латинського слова «fractus», що означає «неправильний за формою», «зламаний». За іншою версією слово походить з англійського «fractional» – «дробовий». Визначення цього тер-

міну не є однозначним, оскільки він охоплює різні аспекти науки та сфери застосування. Загальне трактування зосереджене на властивості самоподібності (означає схожість деякої структури самої на себе) на будь-якому масштабі та визначає фрактал як геометричну структуру [1]. Окрім самоподібності присутні також наступні властивості:

- 1) масштабованість – здатність зберігати форму при змінненні масштабу;
- 2) фрактальні розміри – це група понять (фрактальна розмірність, Гаусдорфова розмірність та ін.), за допомогою яких можна описати складність та структуру фігури.
- 3) хаотичність – для деяких видів фракталів, що проявляють випадкові (хаотичні) властивості;
- 4) ітераційність – для формування таких структур на кожному кроці (ітерації) використовується деяке правило;

Фрактали бувають:

- 1) геометричні;
- 2) алгебраїчні;
- 3) стохастичні;
- 4) системи ітеруючих функцій.

Геометричні фрактали – тип фракталів, що отримується за допомогою простих геометричних перетворень [2]. Алгебраїчні фрактали – структури, що утворені на основі алгебраїчних формул, є найпоширенішою групою фракталів. Стохастичні – виникають тоді, коли у процесі формування фрактала випадковим чином змінюються будь-які його параметри. Системи ітеруючих функцій (IFS – Iterated Function Systems) – група фракталів, що застосовується в методах кодування зображень, запропонованих Майклом Барнслі [2]. Ця робота стосується геометричних фракталів.

Фрактали також можна поділити [3] на природні та рукотворні. Для природних існують обмеження при яких спостерігаються фрактальні властивості.

Рукотворні фрактали винайдені вченими та зберігають свої властивості про будь-якому масштабі.

Фрактали розділюють на:

- 1) конструктивні – є результатом лінійних перетворень подібності [4];
- 2) динамічні – володіють масштабною інваріантністю наближено [4] та утворюються в нелінійних динамічних системах.

Оскільки природні явища та об'єкти не можуть бути ідеальними, то для отримання їх представлення, в регулярні фрактали вносяться деякі випадковості. Наприклад, вносячи випадкові зміни в деревовидний фрактал, можна отримати більшу схожість з природним деревом. Такі фрактали, відповідно до свого методу генерації, називаються випадковими. Для таких структур, на відміну від регулярних, властивість самоподібності дійсна лише після знаходження середніх значень по усім незалежним статистичним реалізаціям, при тому, що збільшений фрагмент фігури може не бути точно таким, як вхідний.

Під терміном «частковий фрактал» в даній роботі розглянуто такий геометричний фрактал, що був частково пошкоджений в процесі формування. Використані методи пошкодження фракталів описані в наступному розділі. Основна відмінність таких фігур від випадкових фракталів це те, що відсутня випадковість виконання того чи іншого кроку формування. Тобто, для випадкового фракталу є випадковий вибір дії, а для часткового – деякі випадково обрані дії під час формування можуть бути взагалі відсутні.

Приклади часткових фракталів подані у наступних розділах.

### 1.3 Огляд відомостей про фрактальну розмірність

У загальному випадку розмірність фігури – це значення, що визначає складність чи об'єм фігури у просторі. Для точки це значення дорівнює 0, для кола або відрізка – 1, для куба або сфери – 2. На відміну від простих геометричних фігур, фрактали мають дробове значення розмірності [3].

Розмірність буває:

- 1) лінійна – рахується як кількість незалежних змінних, необхідних

для опису фігури;

2) нелінійна – існує для опису складних та нерегулярних форм.

Коли мова йде про фрактали, то мається на увазі нелінійна розмірність, бо лінійна описує тільки прості фігури.

Для пояснення фрактальної розмірності можна використати термін "топологічна розмірність". Топологічна розмірність [5] множини в лінійному просторі визначається кількістю лінійно незалежних координат у цьому просторі. Фрактальна розмірність множини визначає розмірність простору, який повністю заповнюється даною множиною. Під цим терміном розуміють різні величини:

- 1) розмірність Мінковського;
- 2) розмірність Гаусдорфа-Безиковича;
- 3) розмірність самоподібності;
- 4) інші.

Серед великої кількості різних варіантів розрахунку фрактальних розмірностей найстарішою є розмірність Гаусдорфа. Її основна перевага серед інших це те, що вона визначена [5] для будь-якої множини. Розмірність Хаусдорфа-Безиковича [5] – це міра розділення об'єкта на частини деякого розміру з наступним підрахунком кількості частин, що покривають об'єкт.

Також до розмірностей відносять евклідову (вкладену) розмірність (embedding dimension), яка визначає об'єкти як набори крапок [5], поміщених в простір.

Основоположником розрахунків розмірності фракталів також став математик і фізик Бенуа Мандельброт. – вивчав нерегулярні та самоподібні структури в природі та інших областях [1], адаптовуючи їх за допомогою математичних термінів, таких як фрактальна розмірність. Його роботи відкрили новий погляд на розуміння складних форм у природі.

Розмірність Мінковського є одним із підходів до вимірювання фрактальної розмірності, що був введений математиком і фізиком Германом Мінковсь-

ким. Основна ідея полягає в тому, щоб виміряти, наскільки множина "займає місце" в просторі, використовуючи величину бокс-зліплення. З визначення, розмірністю Мінковського для обмеженої множини у метричному просторі є така границя [6]:

$$\lim_{\varepsilon \rightarrow 0} \frac{\ln N_\varepsilon}{-\ln \varepsilon}, \quad (1.1)$$

де  $N_\varepsilon$  — найменше число множин діаметра  $\varepsilon$ , що покривають фрактал. Якщо такої границі для множини не існує, то знаходять нижню й верхню границю цього відношення, що називаються відповідно верхньою та нижньою розмірностями Мінковського.

Розмірність Мінковського має такі особливості:

- 1) чутливість до деталей структури фракталу – тобто значення розмірності може змінюватись для різної деталізації об'єкту дослідження;
- 2) невизначеність масштабу – можливість вимірювати розмірність множини для різних її масштабів;
- 3) опис заповненості простору для нелінійних структур;
- 4) виявлення фрактальної властивості об'єкта.

Виявлення розмірності Мінковського допомагає краще розуміти властивості фрактальних об'єктів і має важливі практичні застосування в різноманітних областях, сприяючи розвитку нових методів аналізу та визначення структур в різних системах.

#### 1.4 Аналіз стану програмно-апаратного забезпечення

На сьогодні існує доволі багато різних програм та плагінів, що можуть згенерувати та дослідити фрактал за заданими властивостями. Є серед них більш специфічні, тобто ті, що вузько направлені на дослідження конкретних структур.

Програма Mandelbrot Set Explorer, яка була спеціально розроблена для візуалізації фрактала Мандельброта. Серед функцій цієї програми можна виділити наступне:

- задання ітерації;
- згладжування;
- візуалізація фракталу;
- можливість масштабування;
- кольорова анімація;
- існує версія як для браузеру, так і для смартфона.

Загалом програма може бути використана для візуального ознайомлення з поведінкою фрактальної структури при різних ітераціях та масштабуванні.

Редактор ChaosPro – генератор фрактальних зображень з великим спектром налаштувань. Серед налаштувань доступні наприклад:

- кількість ітерацій;
- розміри зображення;
- колірна палітра;
- особливості проектування;
- розмиття;
- інше.

Програма дозволяє одночасну роботу з формування декількох фракталів в різних вікнах. Також доступне створення тривимірних фракталів на основі двовимірних зображень.

Програма BoxCount – програмне забезпечення з відкритим кодом, сумісне як з Windows, так і з Unix-подібними системами. Окрім генерації правильного фракталу потрібної ітерації, має функцію підрахунку розмірності кривих на площині методом «box-counting», тобто використовує покриття фігури квадратами. Етапи роботи програми [7] представлені на рисунку 1.1.

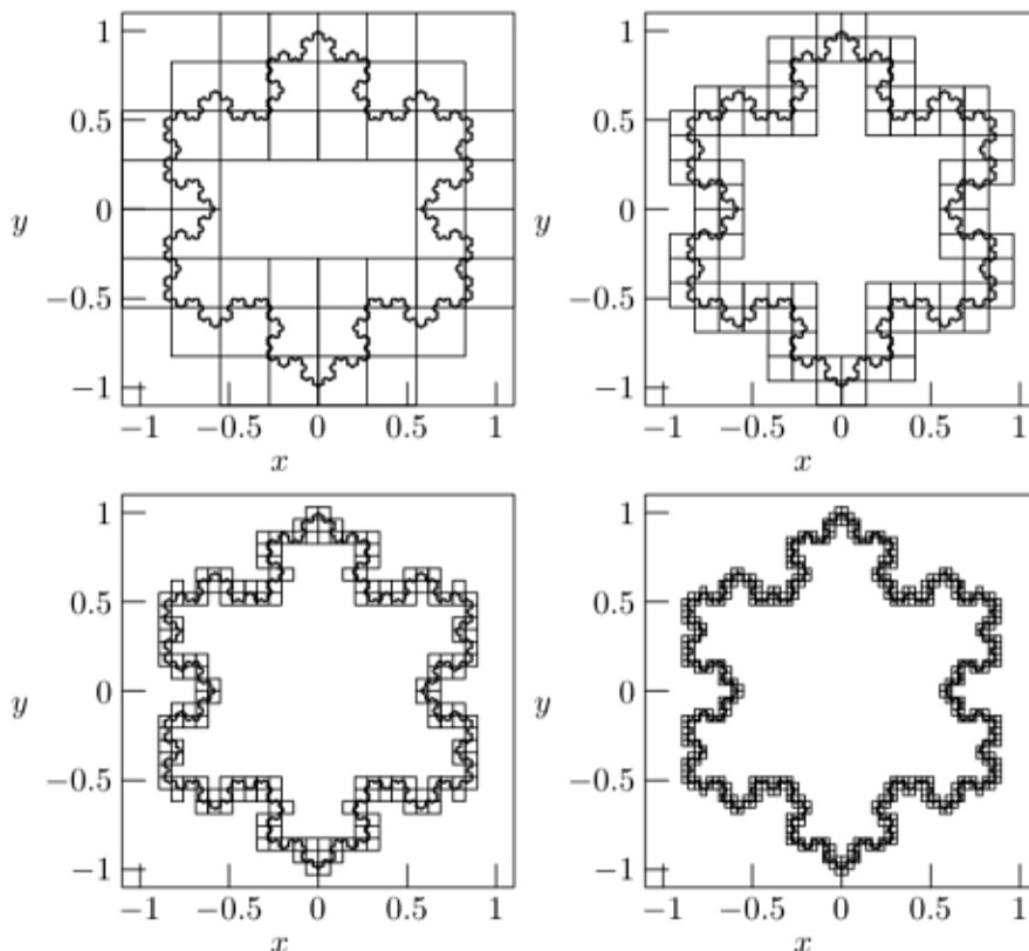


Рисунок 1.1 – Приклад роботи програми VoxCount

Chaotica – програма нового покоління, що дозволяє швидко створити, заанімувати або підготувати для друку фрактальні зображення високої якості. Серед переваг [8] можна виділити підтримку різноманітних фракталів та можливість створення складних аполонійських кривих. Загалом програма більше творча, ніж науковоспрямована.

Fyre — це кросплатформний інструмент із відкритим кодом для створення зображень на основі гістограм ітерованих хаотичних функцій [9].

Програмне забезпечення Ultra Fractal [10] за інтерфейсом (рисунок 1.3) нагадує Photoshop і має детально проілюстровану документацію. Згенеровані в цій програмі фрактали можна зберегти у високому розширенні та в будь-якому з популярних форматів, включаючи векторні. Фрактали генеруються на основі формул, які можна задати самому або обрати з пропонованих. Параметри формул можна редагувати.

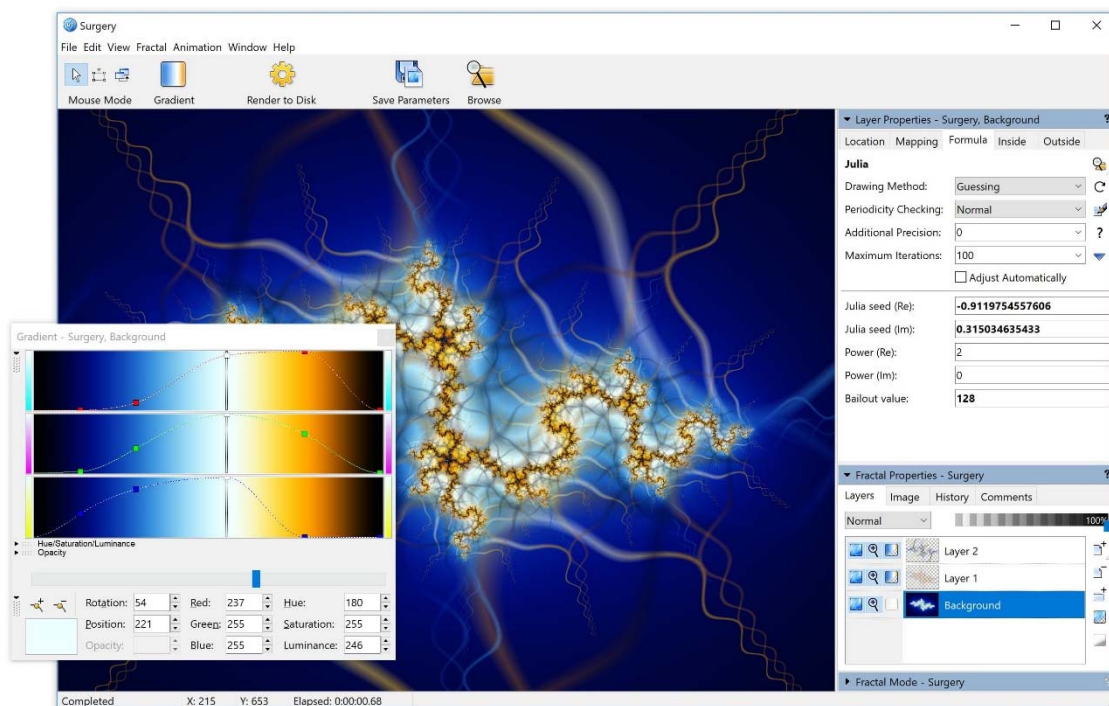


Рисунок 1.2 – Інтерфейс програми Ultra Fractal

Ще один з популярних редакторів для фракталів – Fractal Explorer [11]. Безкоштовний графічний редактор для творення та перегляду фрактальних зображень, який дозволяє швидко генерувати унікальні фрактали на основі різноманітних математичних формул. Надає можливість створювати 2D-фрактали, IFS-фрактали, атрактори, кватерніони та інші геометричні фігури за допомогою різних математичних формул та зазначених параметрів для генерації. Програма має готові шаблони, які зберігаються в її каталозі, для створення нових фігур. Також вона включає графічні інструменти, такі як палітра кольорів, масштабування, спеціальні фільтри та візуальні ефекти. Має функцію створення фрактальних ландшафтів. Fractal Explorer підтримує роботу з різними форматами фрактальної графіки, що робить його універсальним і зручним для творчих експериментів та наукових досліджень.

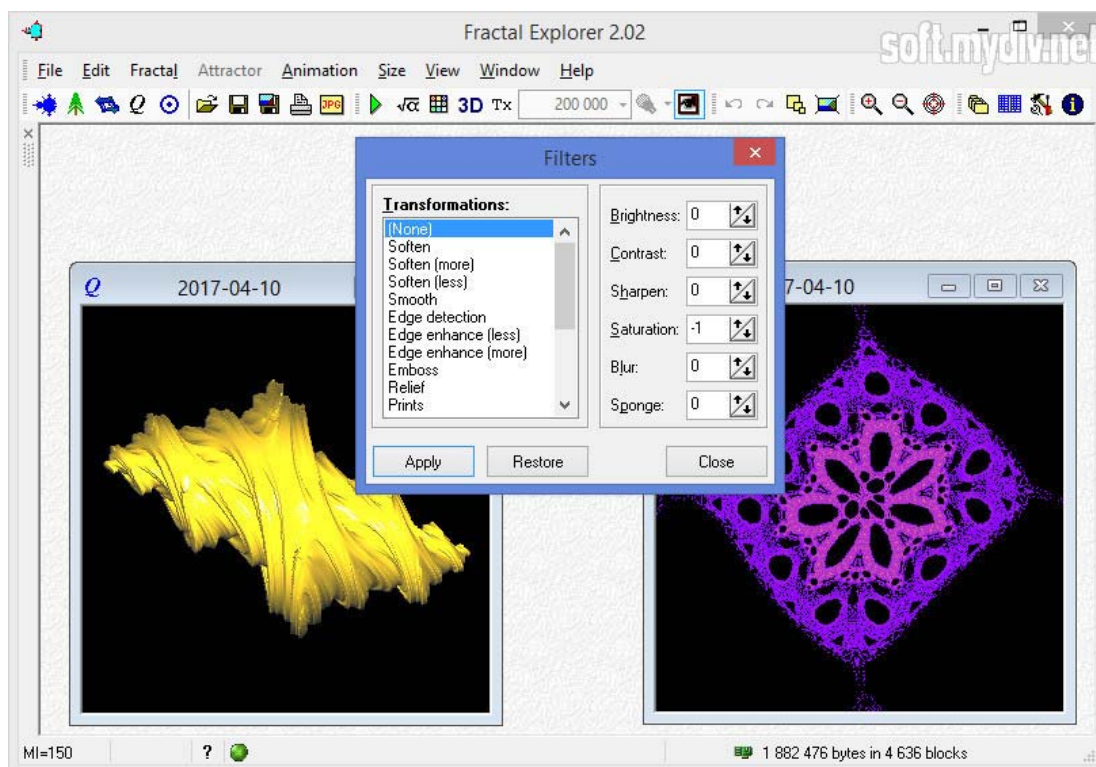


Рисунок 1.3 – Інтерфейс програми Fractal Explorer

Порівняльна характеристика за загальними властивостями описаних програм представлена в таблиці 1.1.

Одним з інструментів, який можна використати для дослідження фрактальної розмірності є Matlab. За допомогою засобів обробки зображень та математичних операцій в Matlab можна виконувати аналіз фракталів. Це дає можливість не лише візуалізувати фрактали, але і математично оцінювати їхню складність та розмірність для подальших наукових досліджень. Але для цього треба добре володіти математичними аспектами дослідження фракталів, тобто дана програма лише інструмент, алгоритм розрахунку реалізується самотужки дослідником.

Більшість програм, які можна було б використати для наукового дослідження написані або дуже давно, або дуже специфічні і налаштовані на якісь конкретні типи фракталів. Сучасне програмне забезпечення в цій області спрямоване на візуальний аспект і творчість, тобто в них відсутні які небуть функції розрахунку розмірностей. Широкий діапазон генераторів з великим вибором налаштувань дозволяють виконати перший крок – створення будь-якого фракталу,

а от дослідження треба виконати або в окремих програмних рішеннях або вручну.

Таблиця 1.1 – Порівняльна характеристика програм для роботи з фракталами

|                               |   |  |  |   |  |   |  |
|-------------------------------|---|--|--|---|--|---|--|
| Функціональність /Програма    | Mandelbrot Set Explorer   | ChaosPro                                       | Chaotica                                       | Fyre                                      | Ultra Fractal                                  | Fractal Explorer  | BoxCount                                       |
| Тип фрактала                  | Мандельброт   | Різноманітні фрактали                          | Аполонійські криві, диференційовані фрактали   | Частина популяції генеруються             | Різноманітні фрактали                          | Мандельброт   | Різноманітні фрактали                          |
| Інтерфейс користувача         | Простий, інтуїтивний  | Складний, велика кількість налаштувань         | Простий, інтуїтивний                           | Простий                                   | Складний, багато налаштувань і можливостей     | Простий, інтуїтивний  | Простий  |
| Можливості налаштувань        | Базові і продвинуті   | Велика кількість, але складно користуватися    | Велика кількість, але легко користуватися      | Обмежені                                  | Велика кількість можливостей                   | Базові і продвинуті   | Обмежені                                       |
| Експорт та збереження         | Збереження у зображення, можливість експорту у векторні формати | Збереження у різні формати, включаючи векторні | Збереження у різні формати, включаючи векторні | Збереження у зображення, векторні формати | Збереження у різні формати, включаючи векторні | Збереження у зображення, анімації, імпорт та імпорт зображень | Збереження у зображення та таблиці результатів |
| Підтримка анімацій            | Так   | Так  | Так  | Так                                       | Так  | Обмежена  | Так  |
| Вбудовані алгоритми генерації | Так   | Так  | Так  | Так                                       | Так  | Так   | Так  |
| Відкритий код                 | Ні  | Ні   | Ні   | Так                                       | Ні   | Так   | Так  |
| Операційна система            | Windows, MacOS  | Windows  | Windows, MacOS, Linux                          | Windows, MacOS, Linux                     | Windows, MacOS                                 | Windows   | Windows  |

### 1.5 Аналіз вимог до системи та постановка задачі

Підсумовуючи відомості про фрактальну розмірність та огляд вже наявних інструментів по роботі з фракталами, можна сформуванати вимоги до системи та завдання, що мають бути виконані для дослідження.

Завдання:

- 1) розробити методи формування зображення часткових фракталів та обрати зручний для їх реалізації підхід;
- 2) визначити метод розрахунку розмірності часткових фракталів та адаптувати його до автоматичного розрахунку за допомогою програми;
- 3) обрати вхідні дані для досліджень;
- 4) зібрати дані про розмірності часткових фракталів в ході дослідження, для подальшого аналізу;
- 5) дослідити поведінку розмірності при різних відсотках дефектів, внесених у фрактали.

Вимоги до системи:

- 1) генерація фракталу та розрахунки розмірності зібрані в одну програму;
- 2) зручний мінімізований інтерфейс, що не буде навантажувати систему додатковою графікою;
- 3) реалізація роботи з декількома типами геометричних фракталів;
- 4) можливість автоматизації дослідів.

## Висновки до першого розділу

У першому розділі розглянуте і описане поняття фракталів, їх властивості та класифікацію. Особливий акцент зроблено на геометричних фракталах, які отримуються за допомогою простих геометричних перетворень. Також розглянуто часткові фрактали, які є геометричними фракталами, що були частково пошкоджені в процесі формування.

Розглянуте поняття розмірності об'єкту. Фрактальна розмірність відіграє визначальну роль у дослідженні та описі складних форм. Для фракталів, у відмінну від простих геометричних фігур, характерною є нелінійна розмірність, що точно визначає їх складні та нерегулярні форми. Розмірність фігури в загальному визначає її складність або об'єм у просторі, і у випадку фракталів ця величина має дробове значення.

Щодо програм для генерації та дослідження фракталів, оглянуті та проаналізовані різноманітні інструменти, такі як Mandelbrot Set Explorer, ChaosPro, BoxCount, Chaotica, Fyre, Ultra Fractal, Fractal Explorer, кожен з яких має свої функції для візуального вивчення чи творчого процесу генерації фракталів.

На основі представлених теоретичних даних за темою дослідження, сформувані завдання для подальшого виконання.

## 2 МЕТОДИ ДОСЛІДЖЕННЯ ТА ЇХ ОБҐРУНТУВАННЯ

### 2.1 Обґрунтування експериментального методу дослідження

Частково пошкоджений фрактал можливо дослідити за допомогою обробки його графічного представлення. При внесенні дефектів, лінії фігури вестимуть себе не передбачувано. Тому для дослідження треба приймати до уваги середні значення деякої кількості порахованих розмірностей для відповідного відсотку пошкоджень.

Розмірність дуже чуттєва до різних вхідних даних. Оскільки в даній роботі виконується обробка растрового зображення фракталу, то потрібно врахувати і розмір ліній, з яких він складається. Чим більший розмір ліній фігури, тим менше на неї впливає пікселізація (ефект «сходинок» - недолік растрових зображень).

Також розмірність може змінювати своє значення в залежності від ітерації фракталу, оскільки чим більша ітерація, тим більше простору заповнює собою фігура. Для зручності у роботі початковий вигляд фігури представлений нульовою ітерацією (рисунок 2.1)

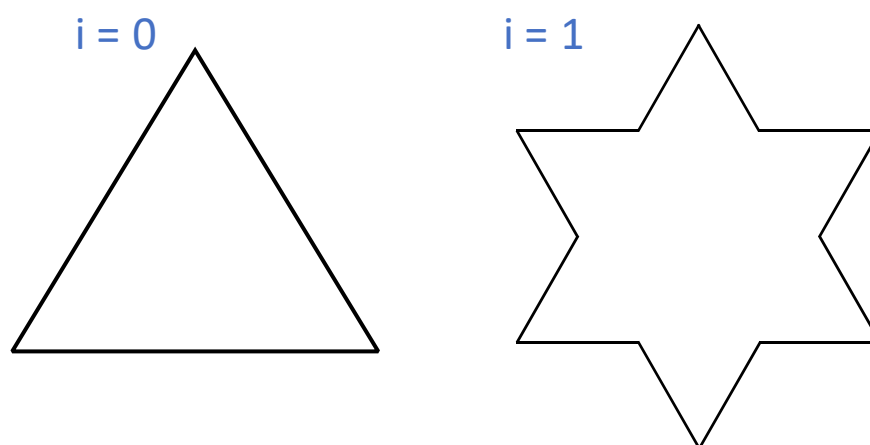


Рисунок 2.1 – Вигляд нульової та першої ітерації Сніжинки Коха

В залежності від ітерації та розміру лінії, наприклад, можна отримати такі значення для регулярного фракталу сніжинки Коха (табл. 2.1):

Таблиця 2.1 Значення розмірності фрактала сніжинки Коха за різної ітерації та довжини лінії

| Розмір лінії (в пікселях) | Ітерація    |             |
|---------------------------|-------------|-------------|
|                           | 2           | 3           |
| 20                        | 1,0613      | 1,1936      |
| 30                        | 1,0965      | 1,2087      |
| 40                        | 1,0971      | 1,2291      |
| 50                        | 1,0921      | 1,2180      |
| 100                       | 1,1097      | 1,2287      |
| 140                       | 1,1088      | 1,2294      |
| 160                       | 1,1055      | 1,2264      |
| Середнє                   | 1,101616667 | 1,219128571 |

Для перших чотирьох ітерацій сніжинка Коха має розмірність Гаусдорфа рівну 1,26. Тому можна зробити висновок, що для більш наближених досліджень треба фрактали великого розміру. Але також врахувати технічні можливості до збереження та обробки зображень великого розширення.

## 2.2 Метод представлення фракталу

Першим етапом вирішення поставленої задачі є графічне зображення часткового фракталу. Зазвичай для зображення правильного геометричного фракталу використовують рекурсивні методи.

Рекурсія – це процес, при якому функція або алгоритм викликає сам себе з певними аргументами, досягаючи базового випадку. У контексті геометричних фракталів, рекурсія використовується для повторення одного чи кількох базових об'єктів з певними правилами. Основна ідея використання рекурсії полягає в тому, щоб взяти базовий геометричний об'єкт (наприклад, лінію, криву, трикутник) і застосовувати до нього перетворення. Кожна отримана частина після перетворення стає новим базовим об'єктом, до якого знову застосовуються ті ж самі правила. Метод простий, але для реалізації додавання дефектів не підходить, оскільки це буде дуже незручно.

Другий метод, що застосовується для вирішення таких задач – це використання так званих L-систем (Lindenmayer system). L-системи (L-system) були винайдені Аристидом Линденмаєром в 1968 р. з ціллю моделювання біологічного

росту. Основна ідея систем полягає у визначенні правил для поетапної заміни символів у рядку. Кожен символ в L-системі представляє конкретну дію або зміну в геометричній конструкції. Цільовий рядок потім використовується для формування відповідного зображення або структури.

Для цієї системи характерна наявність таких елементів: алфавіт, аксіома, правило. Алфавіт – це множина символів, які використовує L-система для позначення деяких дій. Наприклад:

- 1) «F» – йти вперед;
- 2) «+» – повернути направо;
- 3) «-» – повернути наліво;
- 4) інші.

В таблиці 2.2 подані аксіоми та правила для деяких фракталів.

Таблиця 2.2 – Аксіоми та правила L-системи для деяких фракталів

| Фрактал                    | Аксіома   | Правило   |
|----------------------------|-----------|---|
| Сніжинка Коха              | F--F--F   | "F": "F+F--F+F"   |
| Квадратний острів Коха     | F+F+F+F   | "F": "F-F+F+FFF-F-F+F"  |
| Крива Леві                 | F         | "F": "+F--F+"   |
| Крива Серпінського         | F+XF+F+XF | "X": "XF-F+F-XF+F+XF-F+F-X"   |
| Крива дракона              | FX        | "X": "X+YF+",<br>"Y": "-FX-Y"   |
| Квадратний фрактал Госпера | YF        | "X": "XFX-YF-YF+FX+FX-YF-YFFX+YF+FXFX-YF-FX+YF+FXFX+YF-FXYF-YF-FX+FX+YFYF-",<br>"Y": "+FXFX-YF-YF+FX+FX-YF+YFYF-FX-YF+FX-YFYF-YFFX+FX+YF-YF-FX+FX+YFYF" |
| Крива Мура                 | LFL-F-LFL | "L": "+RF-LFL-FR+",<br>"R": "-LF+RFR+FL-"   |
| Решітка Серпінського       | YF        | "X": "YF+XF+Y",<br>"Y": "XF-YF-X"   |

Аксіома – базовий рядок для генерації, що складається з літер алфавіту. Правило – інструкції, що описують те, яким чином будуть замінюватися символи

при перетворені рядка на наступну ітерацію.

Вигляд запису повного сформованого рядка для другої ітерації відповідно до аксіом та правил сніжинки Коха починаючи з базового рядка подано в таблиці 2.3.

Таблиця 2.3 – Формування рядка інструкцій другої ітерації сніжинки Коха

| Ітерація | Вигляд рядка   |
|----------|--|
| 0        | F--F--F  |
| 1        | F+F--F+F--F+F--F+F--F+F  |
| 2        | F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F--<br>F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F |

Графічне представлення згідно цього рядка на рисунку 2.2.

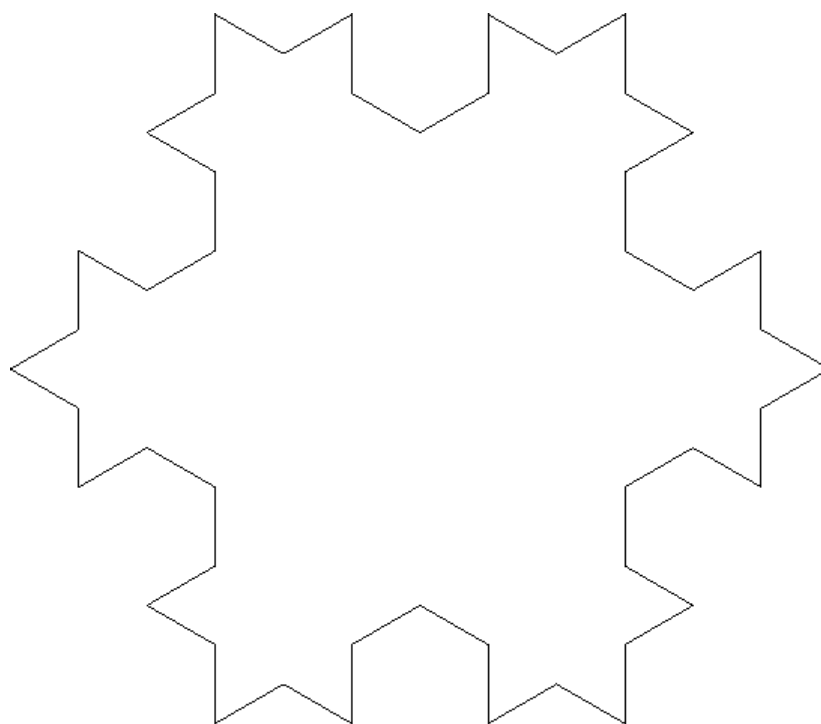


Рисунок 2.2 – Друга ітерація Сніжинки Коха

Оскільки усі елементи фігури представлені символами, то не складно буде ввести в цей рядок деякі зміни, для отримання частково пошкодженого фракталу, що й треба було виконати. Дефектом (змiнами) буде виступати деяка кількість випадково видалених символів з рядка. Цими символами можуть виявитись як і ті, що відповідають за малювання лінії, так і ті, що відповідають за по-

ворот. Кількість символів для видалення буде визначена в залежності від бажаного відсотку дефектів.

Описані вище пошкодження в фрактал можна ввести двома способами:

- 1) видаляти символи з вже сформованого рядка заданої ітерації фрактала;
- 2) сформувати рядок ітерації на одну менше заданої, ввести пошкодження і знову виконати заміни згідно правила.

За першим способом формування часткового фракталу, один з варіантів рядка з прикладу вище для Сніжинки Коха з видаленими двадцятьма відсотками символів: «F-F+F+F+F-FF--F+-+F+F+F--F+F-F+F--+F+F+F--FF-F+-FFF+--F+F--F+F-F+F++F--+F--+F--F+F++F--F+F». Графічне представлення рядка (рисунок 2.3):

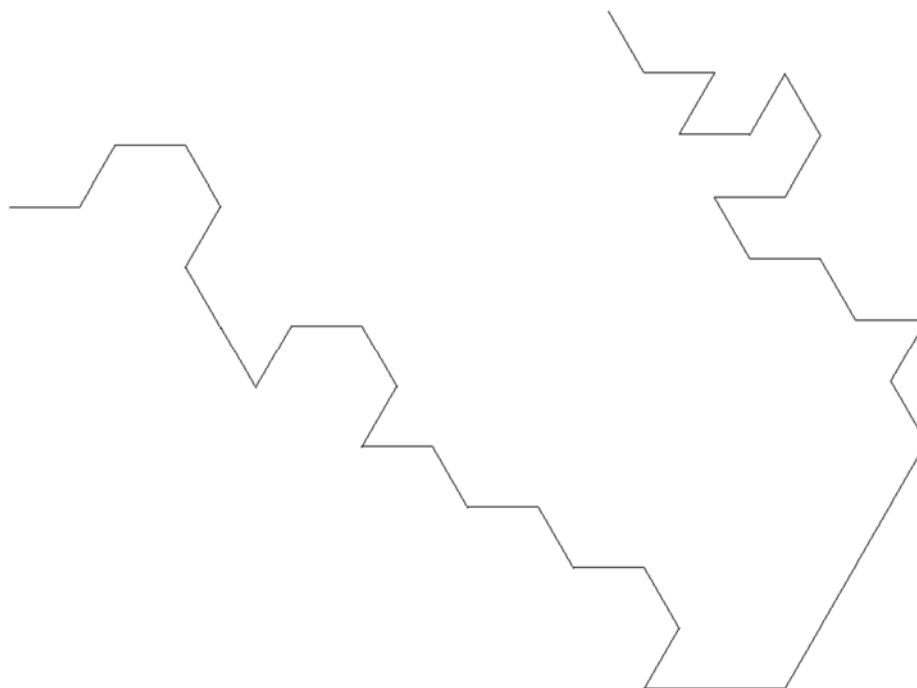


Рисунок 2.3 – Графічне представлення рядку фрактала сніжинки Коха з видаленими двадцятьма відсотками символів

Через те, що зміни вносяться не рівномірно, то фінальне зображення може мати дуже багато різних варіацій (рисунок 2.4 – 2.5).



Рисунок 2.4 – Частковий фрактал сніжинки Коха



Рисунок 2.5 – Частковий фрактал сніжинки Коха

В деяких місцях частково пошкодженого фракталу сніжинки Коха можна побачити схожість з елементами початкової фігури. Це можна спостерігати і для інших фракталів, але не завжди (рисунки 2.6 – 2.7). На схожість можуть впливати такі фактори:

- 1) відсоток дефектів – чим він вищий, тим більша вірогідність згенерувати фігуру, що майже не схожа на свій початковий фрактал;

- 2) серед видалених символів, більшість – повороти або лінії, тому припускається, що в такому випадку фігура починає поводити себе більш незвично.

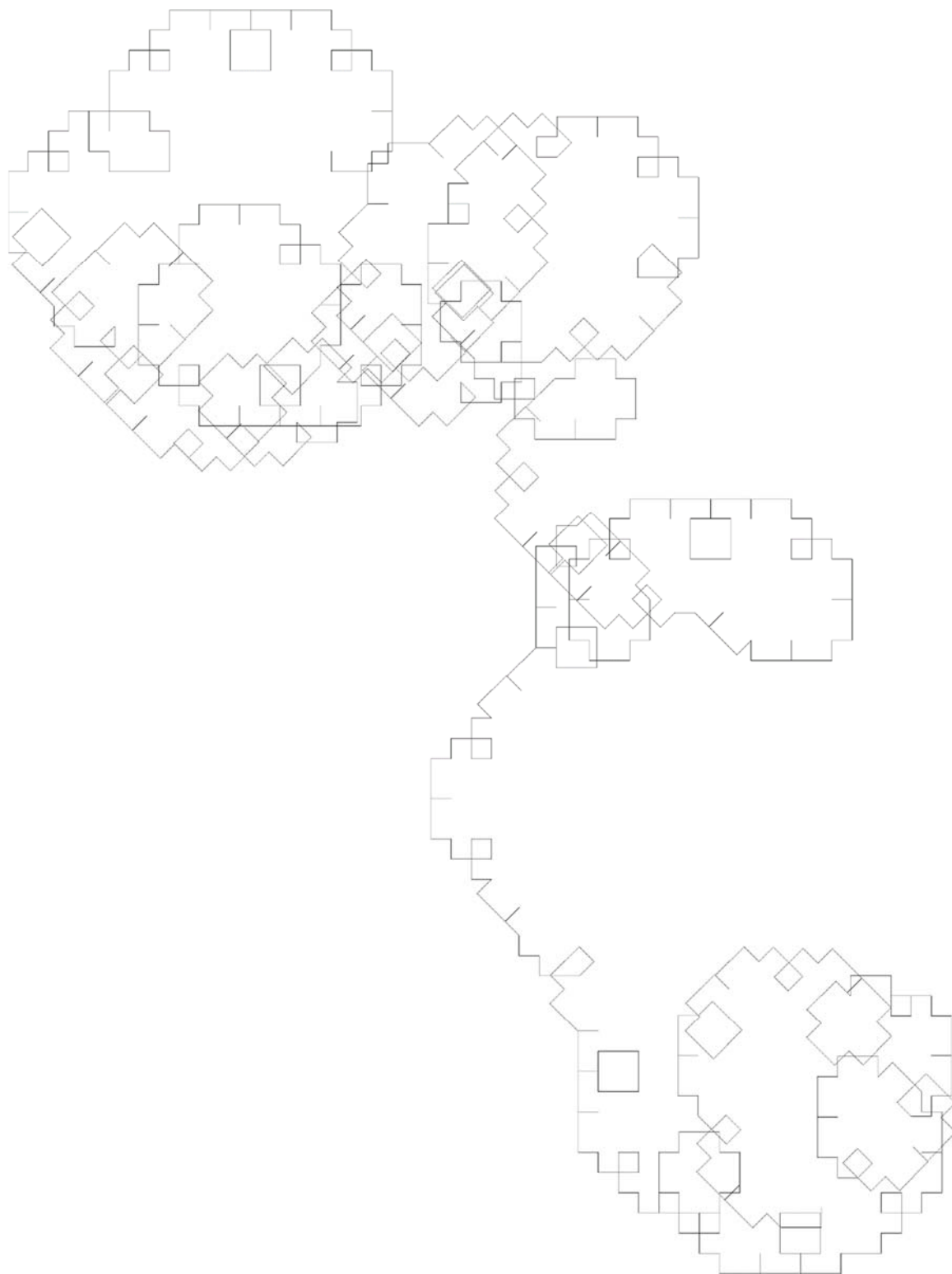


Рисунок 2.6 – Частковий фрактал кривої Леві

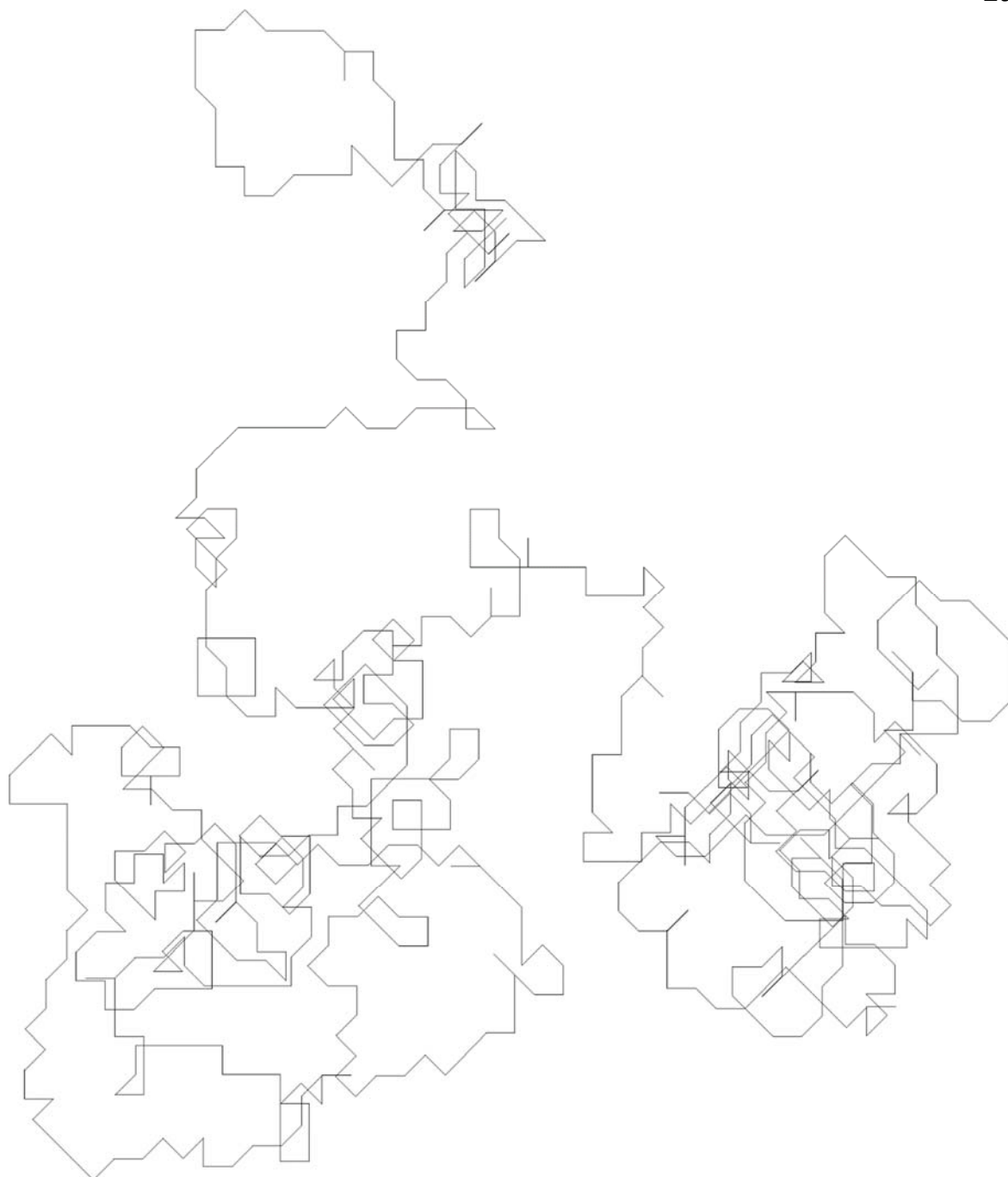


Рисунок 2.7 – Частковий фрактал кривої Леві

За другим способом формування часткового фракталу, один з варіантів рядка сніжинки Коха третьої ітерації з видаленими двадцятьма відсотками символів на етапі другої ітерації в таблиці 2.4.

Таблиця 2.4 – Формування інструкції для другого способу формування часткового фракталу сніжинки Коха

| Етап                           | Рядок L-системи   |
|--------------------------------|---|
| Друга ітерація до видалення    | F+F--F+F+F+F--F+F--F+F--<br>F+F+F+F--F+F--F+F--F+F+F+F--<br>F+F--F+F--F+F+F+F--F+F--F+F--<br>-F+F+F+F--F+F--F+F--F+F+F+F--<br>-F+F  |
| Друга ітерація після видалення | F+F--F+F+F+F--F+F--F+F--<br>F+F++F--F+F--F+F-F+F+F+F--<br>F+F--F+F--F+F+F+F--F+F--F+F--<br>-F+F+F+F--F+F--FF--F+F+F+F--<br>F+F  |
| Третя ітерація                 | F+F--F+F+F+F--F+F--F+F--<br>F+F+F+F--F+F+F+F--F+F+F+F--<br>F+F--F+F--F+F+F+F--F+F--F+F--<br>-F+F+F+F--F+F--F+F--F+F+F+F--<br>-F+F++F+F--F+F--F+F--<br>F+F+F+F--F+F--F+F--F+F+F+F--<br>F+F-F+F--F+F+F+F--F+F+F+F--<br>F+F+F+F--F+F--F+F--F+F+F+F--<br>F+F--F+F--F+F+F+F--F+F--F+F--<br>-F+F+F+F--F+F+F+F--F+F+F+F--<br>-F+F--F+F--F+F+F+F--F+F--<br>F+F--F+F+F+F--F+F--F+F--<br>F+F+F+F--F+F+F+F--F+F+F+F--<br>F+F--F+F--F+F+F+F--F+F--F+F--<br>-F+FF+F--F+F--F+F--F+F+F+F--<br>F+F+F+F--F+F+F+F--F+F--F+F--<br>F+F+F+F--F+F |

Графічний вигляд цього рядка представлено на рисунку 2.8.

Як можна спостерігати з даного прикладу, візуальна схожість з регулярним фракталом сніжинки Коха вища, ніж в першому способі, оскільки остання ітерація виконується вже після внесення дефектів.

Приклад формування другим способом для кривої Серпінського третьої ітерації з відсотком дефектів рівним 3% представлено на рисунку 2.9.

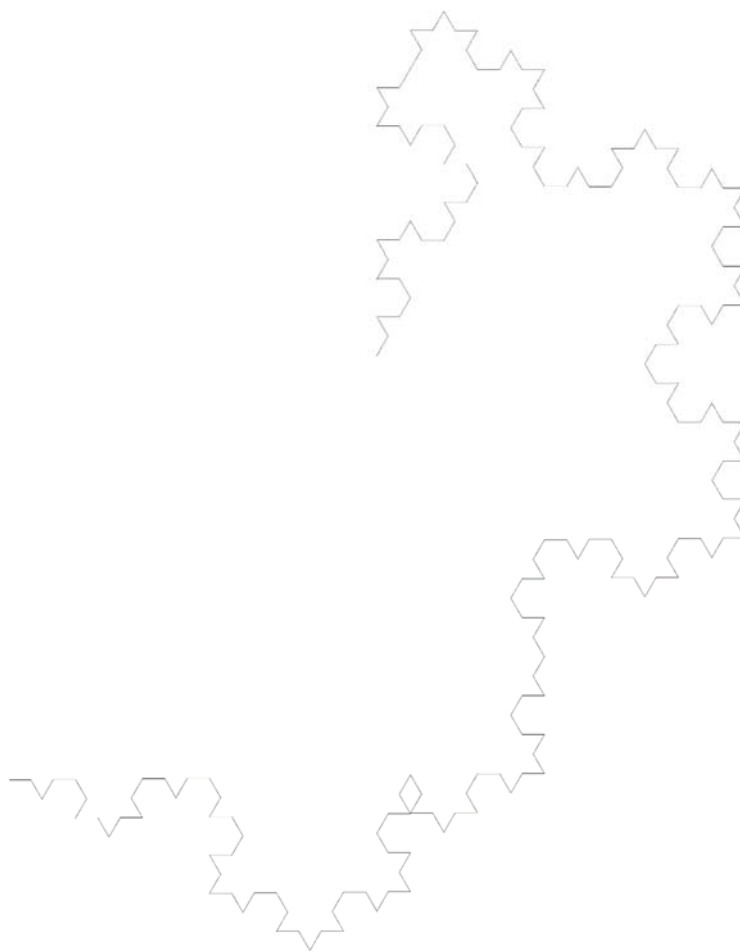


Рисунок 2.8 – Частковий фрактал сніжинки Коха

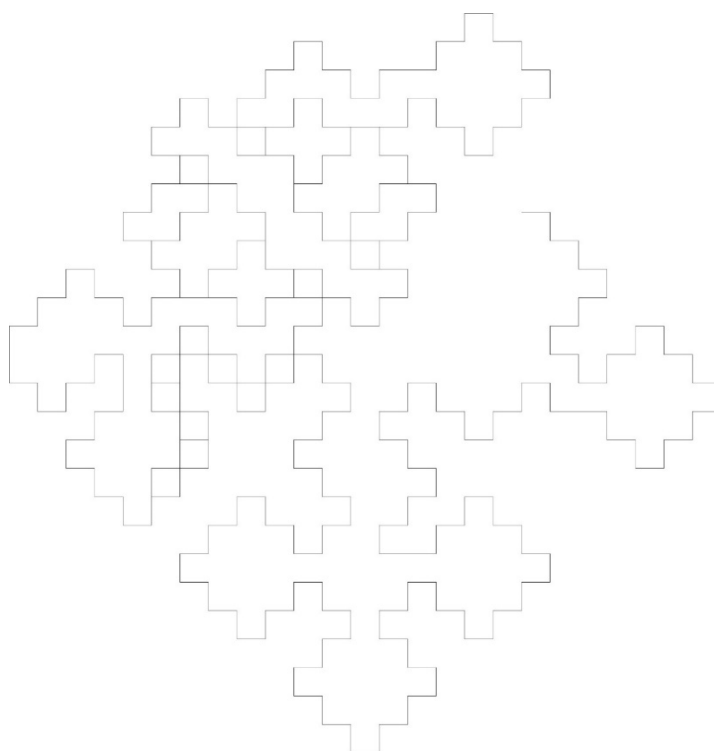


Рисунок 2.9 – Частковий фрактал кривої Серпінського

Вище вже було сказано, зображення фракталу виконується у високому розширенні. Для того, щоб при генерації уникнути виходу за допустимі межі зображення, треба брати до уваги розмір ліній і їх кількість залежно від типу та ітерації фракталу.

Оскільки для підрахунку розмірності нам треба лише лінія фракталу, то увесь незадіяний простір зображення потрібно обрізати, залишивши тільки ту частину де присутня фігура.

### 2.3 Метод розрахунку розмірності

Для розрахунку розмірності використано метод Мінковського – зображення фракталу покривається деякою кількістю однакових квадратів. Операція повторюється десять разів зі зміною довжини сторони квадрата. Оскільки в розпорядженні є лише піксельне зображення фігури, то для виконання описаної процедури умовно розрізаємо зображення на квадрати потрібного розміру. Ті квадрати, що містять у собі лінію фігури (тобто мають хоча б один чорний піксель), зараховуються як «покриваючі». Візуальне представлення одного з етапів цього процесу показано на рисунку 2.10.

Довжина сторони квадрата для покриття розраховується наступним чином:

- визначається менша сторона зображення (після обрізання зображення по розмірам фігури);
- за стартове значення довжини сторони квадрата для покриття береться двадцять відсотків від меншої довжини зображення;
- кожне наступне значення знаходиться шляхом ділення поточного значення на  $1/3$ , округлюючи до меншого.

Наприклад, для зображення розміром  $1060 \times 2503$  значення довжин сторін покриваючих квадратів представлені в таблиці 2.5.

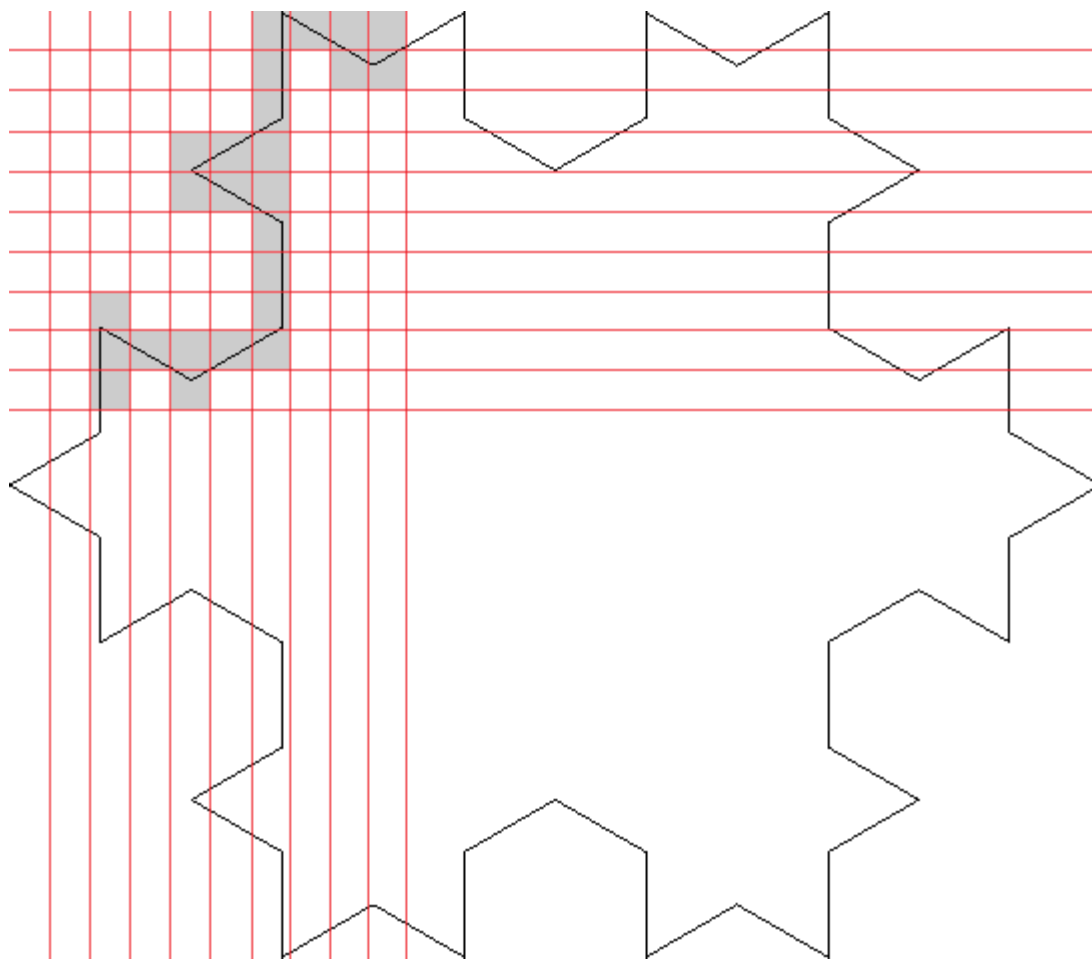


Рисунок 2.10 – Візуалізація розрізання зображення на квадрати та виділення тих квадратів, що порашовані за покриваючі

Таблиця 2.5 – Значення довжин сторін покриваючих квадратів для прикладу

| Ітерація покриття        | 1   | 2   | 3   | 4  | 5  | 6  | 7  | 8  | 9  | 10 |
|--------------------------|-----|-----|-----|----|----|----|----|----|----|----|
| Довжина сторони квадрата | 212 | 163 | 125 | 96 | 73 | 56 | 43 | 33 | 25 | 19 |

Наступний крок – розрахунок розмірності за формулою [12] (2.1):

$$\log M = \log c - s \cdot \log \delta, \quad (2.1)$$

де  $M$  – це кількість покриваючих квадратів,  $c$  – це деякий коефіцієнт,  $s$  – це шукана розмірність,  $\delta$  – це довжина сторони покриваючого квадрата.

Виконаємо у формулі наступну заміну:

$$z = \log M; \quad (2.2)$$

$$b = \log c; \quad (2.3)$$

$$x = \log \delta; \quad (2.4)$$

$$a = -s. \quad (2.5)$$

Виходячи з (2.1), отримаємо рівняння:

$$z = b + ax. \quad (2.6)$$

Оскільки, різниця між теоретичними та експериментальними значеннями під час усіх  $n$  поділів на квадрати має бути мінімальною, то:

$$\sum_{i=1}^n (z_i^T - z_i^E)^2 \rightarrow \min. \quad (2.7)$$

Тоді:

$$\varphi = \sum_{i=1}^n (b + a \cdot x_i^E - z_i^E)^2 \rightarrow \min; \quad (2.8)$$

$$\frac{\partial \varphi}{\partial a} = 0 \cdot \frac{\partial \varphi}{\partial b} = 0. \quad (2.9)$$

Отже маємо систему рівнянь:

$$\begin{cases} \sum_{i=1}^n 2 \cdot x_i^E (b + a \cdot x_i^E - z_i^E) = 0; \\ \sum_{i=1}^n 2 \cdot (b + a \cdot x_i^E - z_i^E) = 0. \end{cases} \quad (2.10)$$

Перетворимо рівняння системи:

$$\begin{cases} b \cdot \sum_{i=1}^n x_i^E + a \cdot \sum_{i=1}^n x_i^2 = \sum_{i=1}^n z_i^E \cdot x_i^E; \\ b \cdot n + a \cdot \sum_{i=1}^n x_i^E = \sum_{i=1}^n z_i^E. \end{cases} \quad (2.11)$$

З отриманої системи з двох рівнянь та двома невідомими знаходимо значення  $a$ , і відповідно саму розмірність  $s$ .

Результати одиничного дослідження поведінки розмірності за різної кількості дефектів представлені на рисунках 2.11 – 2.13. Вхідні дані для дослідів були:

- 1) ітерація самоподоби – 3;

- 2) довжина лінії – 122 пікселі;
- 3) тип фракталу – сніжинка Коха.

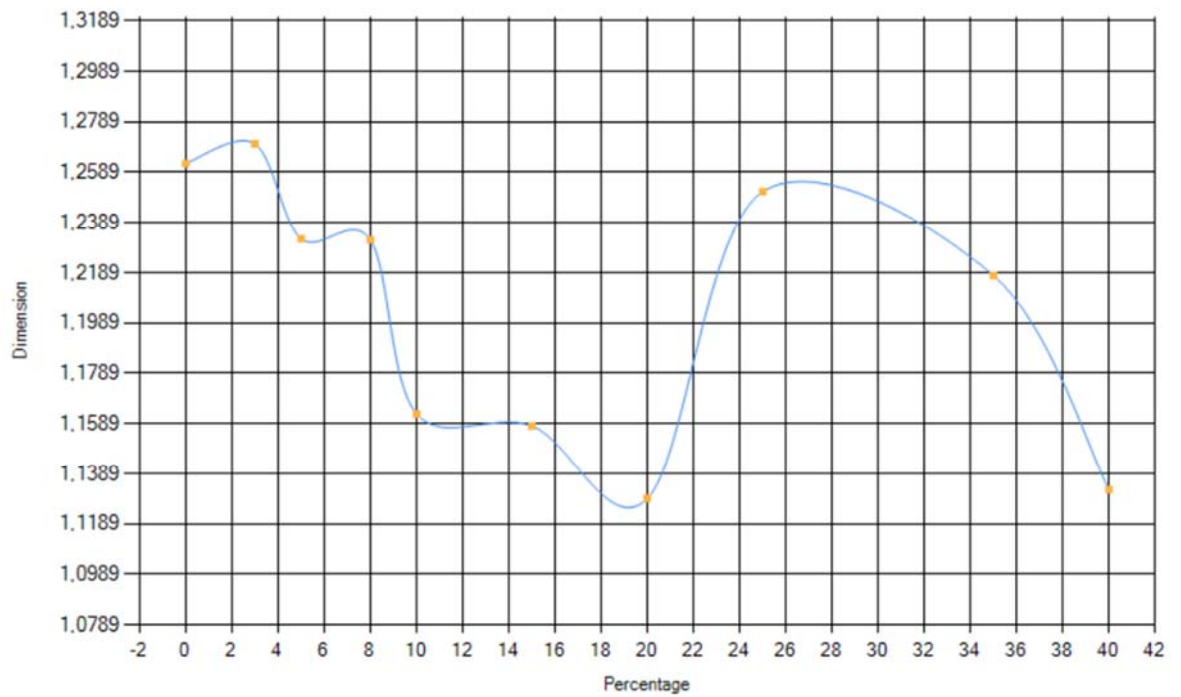


Рисунок 2.11 – Перший результат одиничного досліду

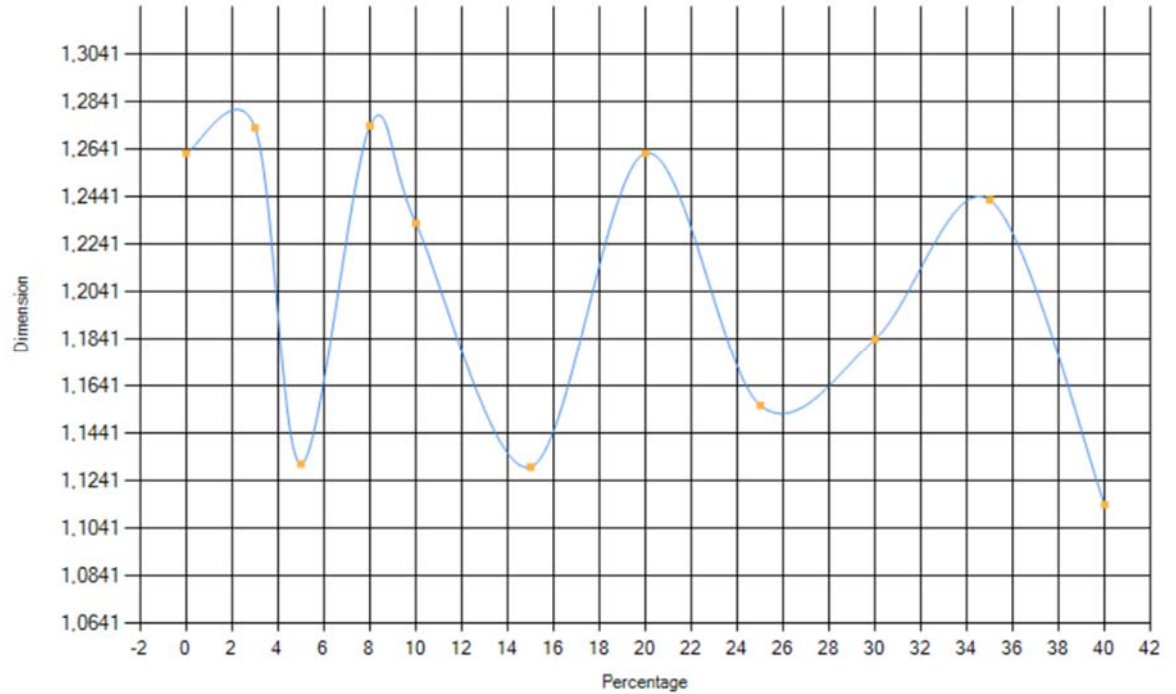


Рисунок 2.12 – Другий результат одиничного досліду

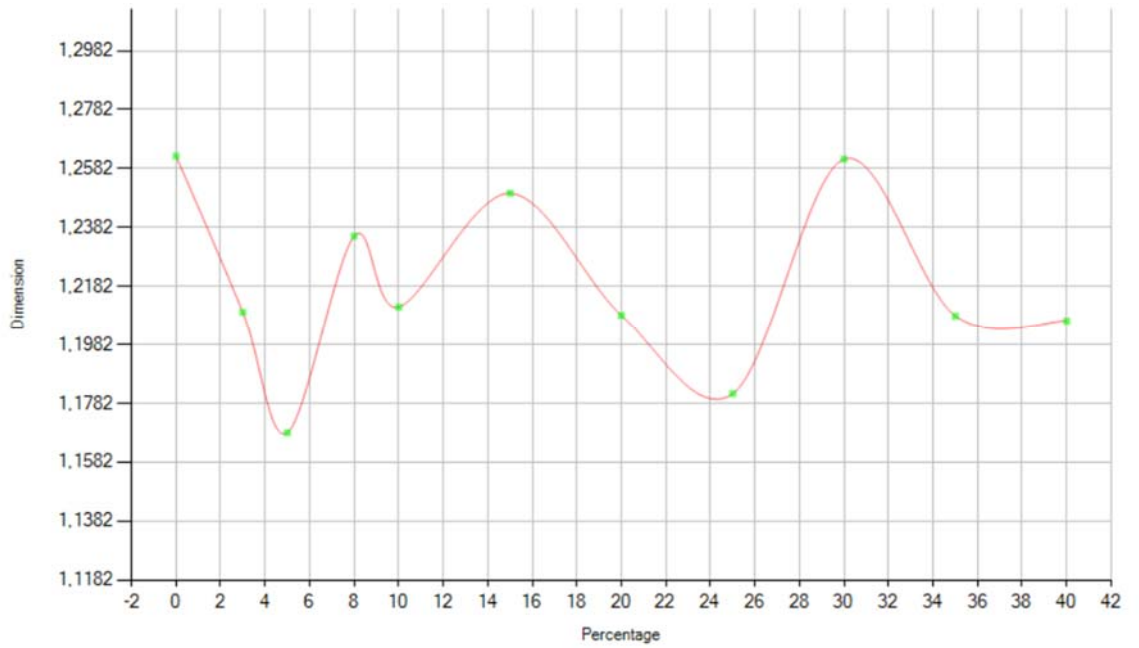


Рисунок 2.13 – Третій результат одиничного дослідю

Як видно з графіків, якоїсь чіткої залежності одиничні дослідю не дають, тому треба провести їх декілька та розділити отримані результати на максимальні, мінімальні та середні.

### Висновки до другого розділу

Другий розділ даної дипломної роботи представляє огляд методів, пояснення та приклади вирішення поставлених завдань дослідження розмірностей часкових геометричних фракталів. Описано два принципи формування часткових фракталів. Представлено метод для виконання розрахунку розмірностей таких фігур.

На прикладі декількох одиничних дослідів показано, що для отримання більш чітких результатів, потрібно виконати велику кількість експериментів та показати залежності через середні значення.

### 3 ПРОЕКТУВАННЯ Й РОРОБКА

#### 3.1 Формалізація задачі

Для формалізації задачі сформовано діаграми прецедентів, що показують взаємодію між користувачем та програмою за декількома сценаріями (рисунки 3.1 – 3.2).

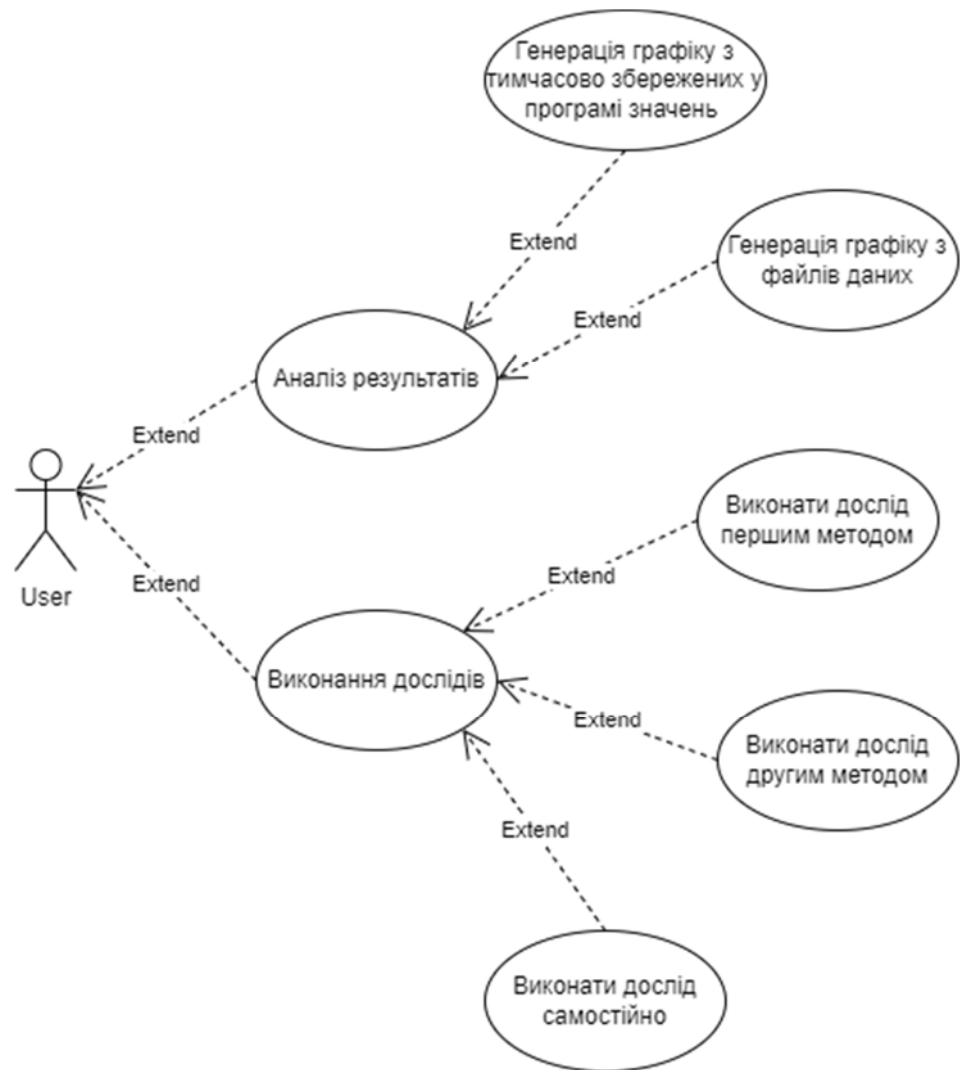


Рисунок 3.1 – Діаграма прецедентів, робота користувача з методами дослідження та аналізом результатів

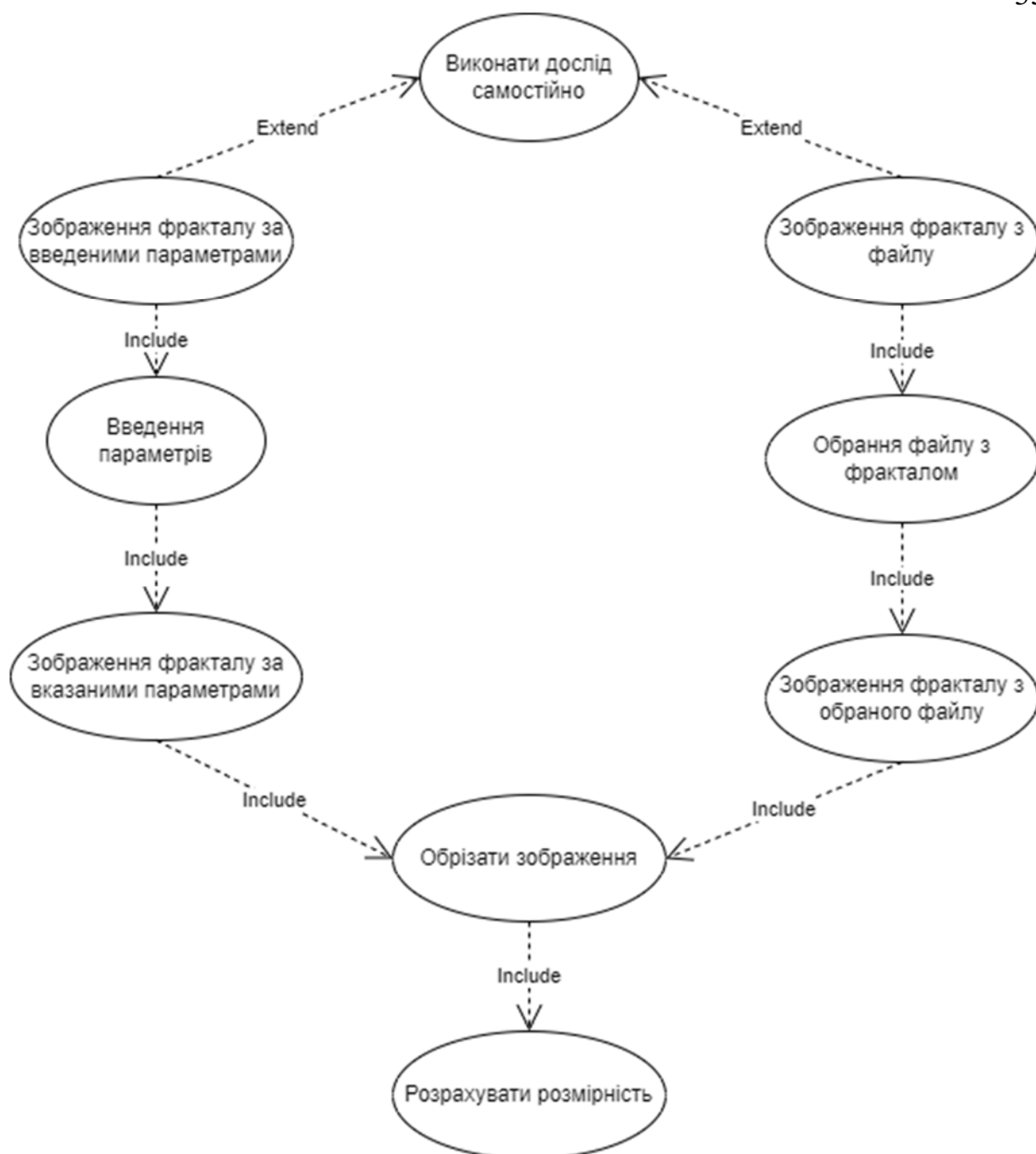


Рисунок 3.2 – Діаграма прецедентів, виконання користувачем дослід самостійно

Оскільки взаємодія користувача з програмою відбувається за допомогою графічного інтерфейсу, то приведемо особливості інструменту, що реалізує таку взаємодію. Windows Forms (WinForms) є одним із інструментів для створення графічних користувацьких інтерфейсів (GUI) в програмах для операційної системи Windows. Цей інструмент входить до складу платформи розробки Microsoft .NET Framework.

Основні властивості Windows Forms:

- 1) візуальний дизайнер: створення GUI за допомогою візуального дизайнера, перетягуючи та розміщуючи елементи інтерфейсу користувача на формах;
- 2) елементи управління (Controls): WinForms надає різноманітні елементи управління, такі як кнопки, тексти, списки, таблиці, вкладки і багато інших для побудови функціональних інтерфейсів;
- 3) обробка подій (Event Handling): можливість визначати обробники подій для взаємодії з користувачем або реагувати на події від інших частин програми;
- 4) можливості многозадачності: Windows Forms дозволяє створювати багатовіконні програми та взаємодіяти з іншими частинами операційної системи;
- 5) взаємодія з іншими технологіями .NET: можливість використовувати інші технології .NET Framework разом з WinForms, такі як ADO.NET для роботи з базами даних, ASP.NET для веб-розробки та інші;
- 6) мови програмування: можливість використовувати різні мови програмування, такі як C# або VB.NET, для створення програм з використанням Windows Forms.

### 3.2 Базова архітектура системи

Архітектуру системи представляє патерн Model-View-Controller (MVC).

Цей патерн є підходом до розробки програмного забезпечення, що розділяє його на три основні компоненти: Model (Модель), View (Вид) і Controller (Контролер).

Кожен з цих компонентів виконує конкретну роль у взаємодії з додатком.

Основні функції компоненту Model (Модель):

- 1) відповідає за представлення даних та бізнес-логіки додатка;
- 2) обробляє запити з контролера та оновлює стан даних;
- 3) не залежить від інших компонентів системи.

Функції компоненту View (Вид):

- 1) відповідає за відображення інформації користувачеві;
- 2) виводить дані, які отримує від моделі, та пересилає події в контролер при взаємодії користувача.

Controller (Контролер) виконує наступні задачі:

- 1) обробляє введення користувача та інші події;
- 2) взаємодіє з моделлю, ініціюючи зміни в даних;
- 3) оновлює вигляд відповідно до змін у моделі.

Основна ідея паттерну MVC полягає в тому, що він дозволяє відокремити представлення даних, бізнес-логіку та логіку введення. Це полегшує розробку, підтримку та тестування коду, оскільки кожен компонент відповідає за свою частину функціональності. Паттерн MVC використовується в багатьох технологіях та рамках розробки, включаючи веб-додатки, десктоп-застосунки та мобільні додатки.

### 3.3 Внутрішнє програмування

#### 3.3.1 Вибір мови та платформи для розробки

Для виконання поставленого завдання цієї дипломної роботи була обрана мова програмування C# та фреймворк .NET Framework. .NET Framework - це програмна технологія [13], яка підходить для зручного та швидкого створення додатків для платформи Windows; має широкий інструментарій розробки, що значно спрощує роботу. Інтегроване середовище розробки, що підходить для обраних технологій – це Microsoft Visual Studio 2019.

Використані наступні бібліотеки:

- 1) System.Collections.Generic – містить у собі інтерфейси та класи для створення строго типізованих колекцій, кращу безпеку даних та вищу продуктивність, ніж у інших типізованих колекцій;
- 2) System.Data – для ефективного управління даними з декількох джерел;
- 3) System.Drawing – доступ до графічних функцій;

- 4) System.Linq – підтримує запити з використанням LINQ ;
- 5) System.Windows.Forms – має в собі класи, що дозволяють створювати користувацький інтерфейс для операційної системи Microsoft Windows;
- 6) System.IO – функції для роботи з файлами;
- 7) System.Windows.Forms.DataVisualization.Charting – містить у собі усі потрібні елементи для управління об'єктом Chart .

Описані інструменти для розробки покривають усі поставлені задачі.

### 3.3.2 Вибір формату збереження графічного представлення фракталів

Для зручної обробки збереженого зображення найбільше підходять типи файлів без стискання. Також важливою вимогою, в контексті поставленої задачі, є можливість зберігання широкоформатних зображень. Одним з простих та зрозумілих типів є .BMP. Цей тип дозволяє нам зберігати цілісну картинку дуже великого розміру та надає зручний доступ до даних про колір пікселів. Максимальний розмір піксельного поля для збереження становить  $15000 \times 15000$  пікселів. Враховуючи потреби та оптимальні можливості за часом обробки таких зображень, вирішено обмежити піксельне поле до  $10000 \times 10000$  пікселів.

### 3.3.3 Ієрархія та взаємодія класів системи

Щоб відобразити взаємодію класів в системі, подано діаграму класів на рисунку 3.3.

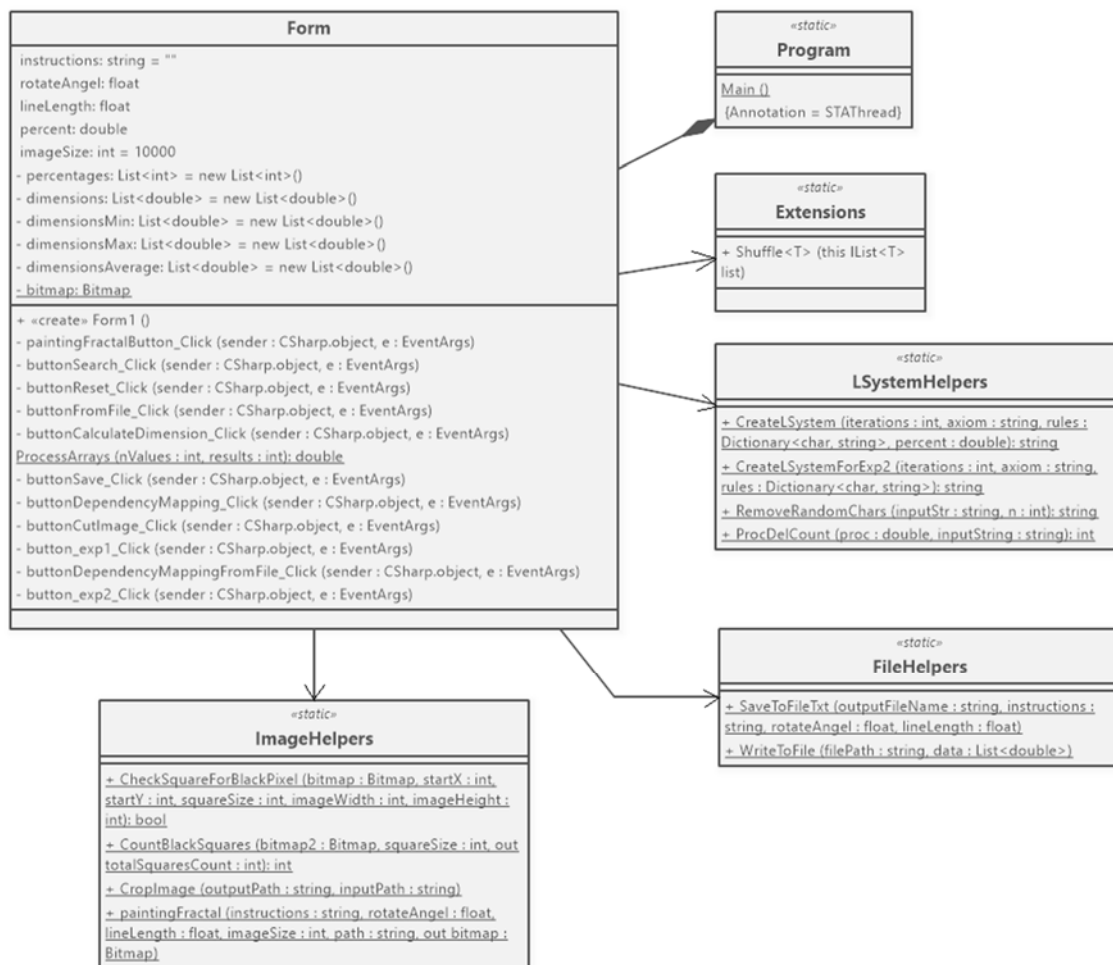


Рисунок 3.3 – Діаграма класів проекту

### 3.3.4 Використані принципи проектування

Принципи SOLID та KISS є основними принципами проектування програмного забезпечення, спрямованими на створення коду, який є ефективним, легким у розумінні, масштабованим та легко збереженим.

#### Принципи SOLID:

- 1) принцип єдиної відповідальності (Single Responsibility Principle - SRP): кожен клас повинен мати лише одну причину для зміни. Це означає, що клас повинен мати лише одну відповідальність;
- 2) принцип відкритості-закритості (Open-Closed Principle - OCP): програмні сутності, такі як класи, модулі чи функції, повинні бути відкритими для розширення, але закритими для модифікацій. Це сприяє внесенню змін без зміни вже наявного коду;

- 3) принцип підстановки Лісков (Liskov Substitution Principle - LSP): об'єкти базового класу повинні бути замінюваними об'єктами похідного класу без зміни правильності програми;
- 4) принцип інтерфейсу розділення (Interface Segregation Principle - ISP): клієнти не повинні залежати від інтерфейсів, які вони не використовують. Це означає, що краще створити багато малих, спеціалізованих інтерфейсів, ніж один великий загальний;
- 5) принцип інверсії залежностей (Dependency Inversion Principle - DIP): модулі високого рівня не повинні залежати від модулів низького рівня. Обидва повинні залежати від абстракцій. Абстракції не повинні залежати від деталей, деталі повинні залежати від абстракцій.

Принцип простоти (Keep It Simple, Stupid - KISS): цей принцип вказує на те, що програмне забезпечення повинно бути якомога простішим у розумінні та реалізації. Зайві складнощі і надмірна складність можуть призвести до помилок, ускладнення тестування та утруднення обслуговування.

Ці принципи допомагають розробникам створювати програмне забезпечення, яке є гнучким, легким у розумінні та обслуговуванні, а також може легко пристосовуватися до змін. Вони є важливими у побудові стійкого та якісного програмного продукту.

### 3.3.5 Використані шаблони проектування

Під час розробки програми використані такі шаблони:

- 1) MVC (Модель-Представлення-Контролер) – розділення архітектури додатку на три частини для полегшення проектування логіки, обробки подій та роботи з інтерфейсом.
- 2) Iterator (Ітератор) – визначає механізм доступу до елементів колекції без розкриття внутрішньої структури цієї колекції; дозволяє послідовно обходити елементи колекції, надаючи уніфікований інтерфейс для доступу.

3) Singleton (Одиночка) – визначає, що клас може мати тільки один екземпляр.

### 3.4 Розробка інтерфейсу користувача

Інтерфейс програми для користувача представлений одним вікном з меню, що являє собою дві вкладки ComboBox.

Перша вкладка «Дослідження» (рисунок 3.4) надає можливості для роботи з генерації часткових фракталів та розрахунку їх розмірностей.

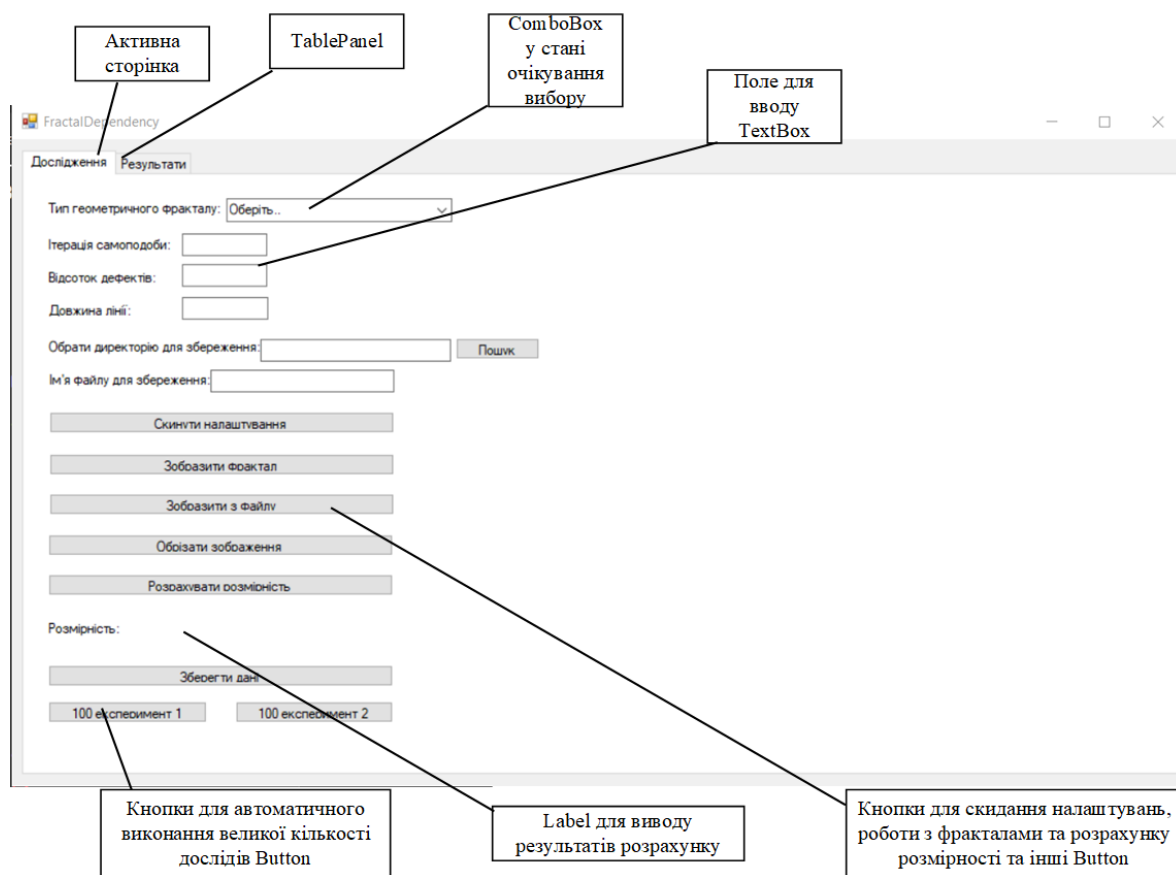


Рисунок 3.4 – Візуальне представлення вкладки «Дослідження»

Друга вкладка «Результати» (рисунок 3.5) надає можливості для перегляду графіку виконаного дослідження.

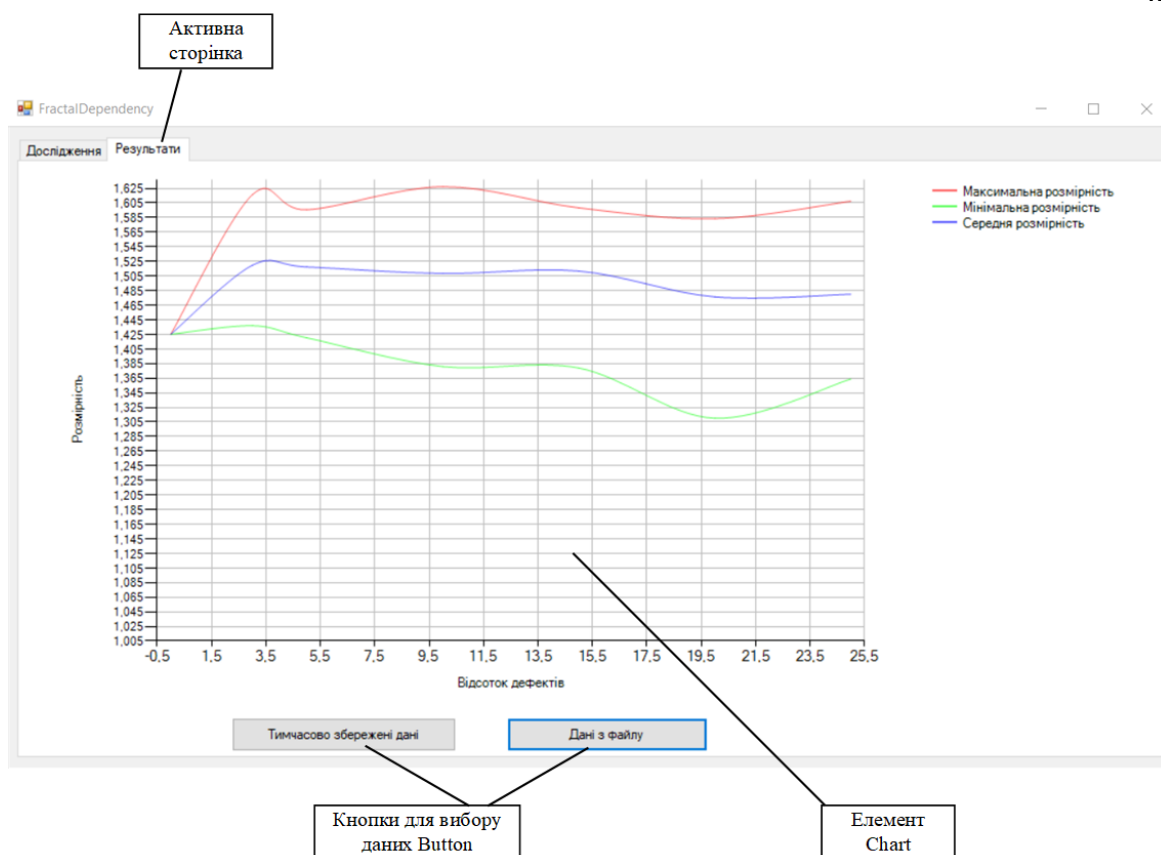


Рисунок 3.5 – Візуальне представлення вкладки «Результати»

### 3.5 Тестування та налагодження програми

На етапі тестування важливо виявити слабкі місця, де вірогідніше виникнуть ускладнення. Оскільки методи дослідження використовують обробку зображень великого розміру, то важливо зрозуміти максимальні можливості такої обробки. Для такої перевірки підходить метод тестування граничними умовами. Тестування граничними умовами (Boundary Value Testing) - це метод тестування програмного забезпечення, в якому тестові дані вибираються так, щоб включати граничні, екстремальні або крайні значення.

Перевірка тестових сценаріїв в таблиці 3.1.

Налагодження програми виконано методом Step-by-Step Debugging. Виявлення та виправлення помилок в програмному кодї за цим методом відбувається шляхом виконання програми поетапно (крок за кроком). Цей метод дозволяє програмістові слідкувати за виконанням коду, аналізувати стан змінних та

визначати місце та причину виникнення помилок. Більшість інтегрованих середовищ розробки (IDE) надають зручні інструменти для використання цього методу.

Таблиця 3.1 – Тестові сценарії та результати

| Тестовий сценарій                                     | Вхідні дані       | Очікуваний результат                                       | Результат  | Коментарі   |
|---|-------------------|--|--|---|
| Збереження зображення розміром 15001 × 15001 пікселів | imageSize = 15001 | Повідомлення про неможливість збереження такого зображення | Повідомлення про неможливість збереження такого зображення | Граничним доступним значенням розміру сторін в пікселях для збереження зображення типу BMP є 15000. |

Основні кроки методу налагодження Step-by-Step Debugging можуть бути описані наступним чином:

- 1) програміст визначає місця в коді, де він хоче зупинити виконання програми для аналізу;
- 2) програма виконується в режимі налагодження, який дозволяє програмістові контролювати кожен крок виконання;
- 3) покрокове виконання дозволяє програмістові аналізувати стан змінних та перевіряти, чи відбувається виконання коду так, як очікувалося;
- 4) після виявлення помилок програміст може вносити зміни в код та перевіряти їх ефективність через режим налагодження;
- 5) після виправлення помилок програміст може видаляти точки зупинки та завершувати режим налагодження.

Налагодження програми є обов'язковим етапом у розробці, оскільки саме це допомагає виявити не тільки саму помилку, а й причину її виникнення, що не завжди можливо виконати тестуванням.

### Висновки до третього розділу

У розділі описані етапи проектування та розробки програми, що є інструментом для виконання дослідження.

Приведені діаграми прецедентів, які відображають роботу користувача з дослідженнями та результатами.

Описано патерн Model-View-Controller (MVC), за яким вибудовувалась архітектура додатку, обрані мова програмування, середовище та платформа для розробки. Приведені специфічні бібліотеки, що мають бути використані для вирішення завдань дипломної роботи. Також аргументовано вибір формату BMP для збереження графічного представлення часткових фракталів.

Розроблений інтерфейс користувача, враховуючи виконаний в першому розділі огляд програмних рішень.

Виконано тестування граничними умовами та налаштування «крок-за-кроком» написаної програми.

## 4 ДОСЛІДЖЕННЯ РОЗМІРНОСТІ ЧАСТКОВИХ ФРАКТАЛІВ

### 4.1 Підготовка до експериментів

Для дослідження розмірності були обрані декілька геометричних фракталів:

- 1) сніжинка Коха;
- 2) квадратний острів Коха;
- 3) крива Серпінського;
- 4) крива Дракона;
- 5) крива Леві.

Вхідні дані для виконання дослідження кожного з обраних типів фракталів подано в таблиці 4.1.

Таблиця 4.1 – Вхідні дані для виконання досліджень

| Фрактал                | Ітерація | Довжина лінії (в пікселях) | Розмірність при 0% дефектів |
|------------------------|----------|----------------------------|-----------------------------|
| Сніжинка Коха          | 3        | 120                        | 1,216                       |
| Квадратний острів Коха | 2        | 40                         | 1,3613                      |
| Крива Серпінського     | 3        | 90                         | 1,4257                      |
| Крива Дракона          | 10       | 40                         | 1,5988                      |
| Крива Леві             | 10       | 50                         | 1,4251                      |

Усі представлені дані підібрані з урахуванням особливостей конкретних фракталів та технічних можливостей.

### 4.2 Дослідження розмірності часткових фракталів сформованих першим методом

Відсотки дефектів при дослідженні для кожного часткового фракталу, сформованого першим методом, взяті однакові, а саме: 1%, 3%, 5%, 8%, 10%, 15%, 20%, 25%, 30%, 35%, 40%. Одиничний дослід не показав чітких результатів, тому розрахунок розмірності для кожного відсотку пошкоджень виконується сто разів, на графік виносяться мінімальне, максимальне та середнє значення.

#### 4.2.1 Вплив ступеня пошкодження першим методом фракталів сніжинки Коха на фрактальну розмірність

Результат ста експериментів розрахунку розмірності для сніжинки Коха при різних значеннях відсотку внесених дефектів представлено в таблиці 4.2. Вікно налаштувань програми на рисунку 4.1.

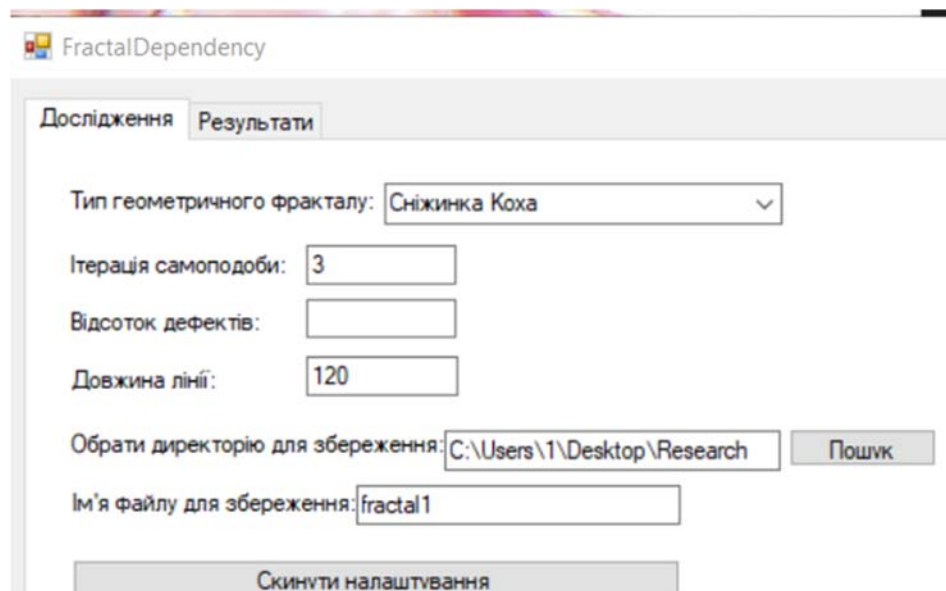


Рисунок 4.1 – Налаштування для дослідів

Таблиця 4.2 – Мінімальні, максимальні і середні значення розмірностей отримані в ході експерименту

| Відсоток дефектів | Мінімальна розмірність | Середня розмірність | Максимальна розмірність |
|-------------------|------------------------|---------------------|-------------------------|
| 0                 | 1,216                  | 1,216               | 1,216                   |
| 1                 | 1,144                  | 1,209               | 1,2767                  |
| 3                 | 1,0961                 | 1,2164              | 1,3348                  |
| 5                 | 1,0778                 | 1,2155              | 1,3254                  |
| 8                 | 1,1014                 | 1,2394              | 1,3436                  |
| 10                | 1,1029                 | 1,2279              | 1,3746                  |
| 15                | 1,1349                 | 1,237               | 1,3488                  |
| 20                | 1,1398                 | 1,2316              | 1,316                   |
| 25                | 1,1286                 | 1,2338              | 1,3287                  |
| 30                | 1,0917                 | 1,2097              | 1,3488                  |
| 35                | 1,1134                 | 1,2198              | 1,3463                  |
| 40                | 1,114                  | 1,2048              | 1,3182                  |

Графік отриманої в ході експерименту залежності між відсотком дефектів та значенням розмірності на рисунку 4.2.

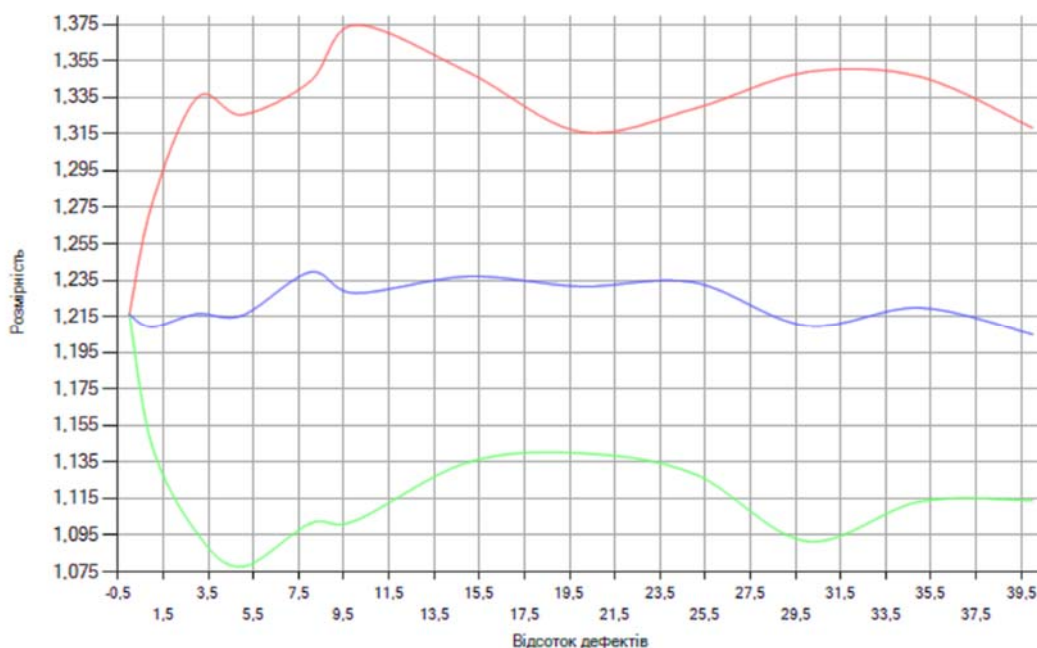


Рисунок 4.2 – Графік залежності між відсотком дефектів та значенням розмірності для сніжинки Коха. Червоним позначено максимальну розмірність, синім – середню, зеленим – мінімальну

#### 4.2.2 Вплив ступеня пошкодження першим методом фракталів квадратного острова на фрактальну розмірність

Результат ста експериментів розрахунку розмірності для квадратного острова Коха при різних значеннях відсотку внесених дефектів представлено в таблиці 4.3. Вікно налаштувань програми на рисунку 4.3.

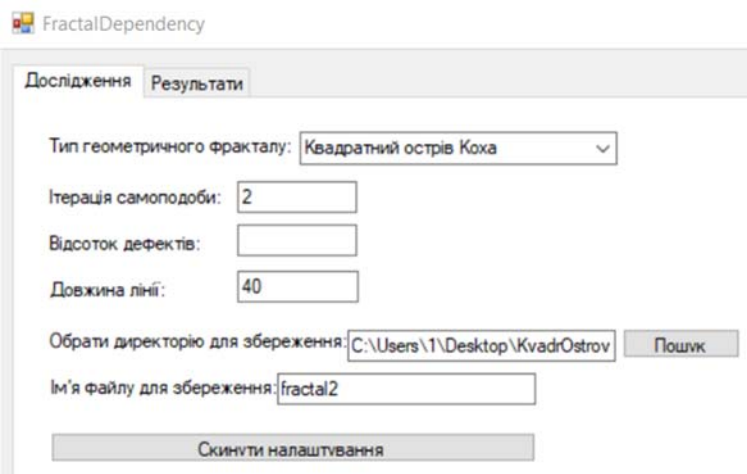


Рисунок 4.3 – Налаштування для дослідження

Таблиця 4.3 – Мінімальні, максимальні і середні значення розмірностей отримані в ході експерименту

| Відсоток дефектів | Мінімальна розмірність | Середня розмірність | Максимальна розмірність |
|-------------------|------------------------|---------------------|-------------------------|
| 0                 | 1,3613                 | 1,3613              | 1,3613                  |
| 1                 | 1,2402                 | 1,3391              | 1,395                   |
| 3                 | 1,1777                 | 1,3245              | 1,4031                  |
| 5                 | 1,234                  | 1,3096              | 1,4021                  |
| 8                 | 1,1324                 | 1,2894              | 1,3858                  |
| 10                | 1,1947                 | 1,288               | 1,39                    |
| 15                | 1,1578                 | 1,2816              | 1,365                   |
| 20                | 1,1783                 | 1,2675              | 1,3499                  |
| 25                | 1,1328                 | 1,2378              | 1,3192                  |
| 30                | 1,0958                 | 1,2356              | 1,3318                  |
| 35                | 1,119                  | 1,2076              | 1,3009                  |
| 40                | 1,1488                 | 1,2118              | 1,3168                  |

Графік отриманої в ході експерименту залежності між відсотком дефектів та значенням розмірності на рисунку 4.4.

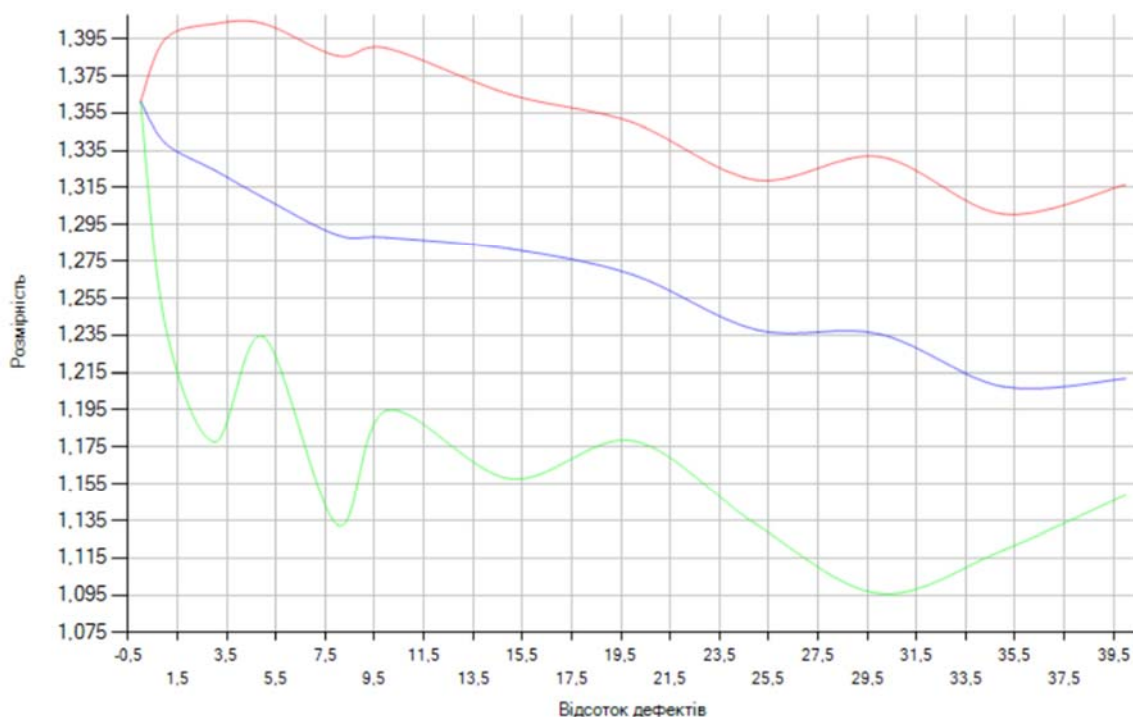


Рисунок 4.4 – Графік залежності між відсотком дефектів та значенням розмірності для квадратного острова Коха. Червоним позначено максимальну розмірність, синім – середню, зеленим – мінімальну

### 4.2.3 Вплив ступеня пошкодження першим методом фракталів кривої Серпінського на фрактальну розмірність

Результат ста експериментів розрахунку розмірності для кривої Серпінського при різних значеннях відсотку внесених дефектів представлено в таблиці 4.4. Вікно налаштувань програми на рисунку 4.5.

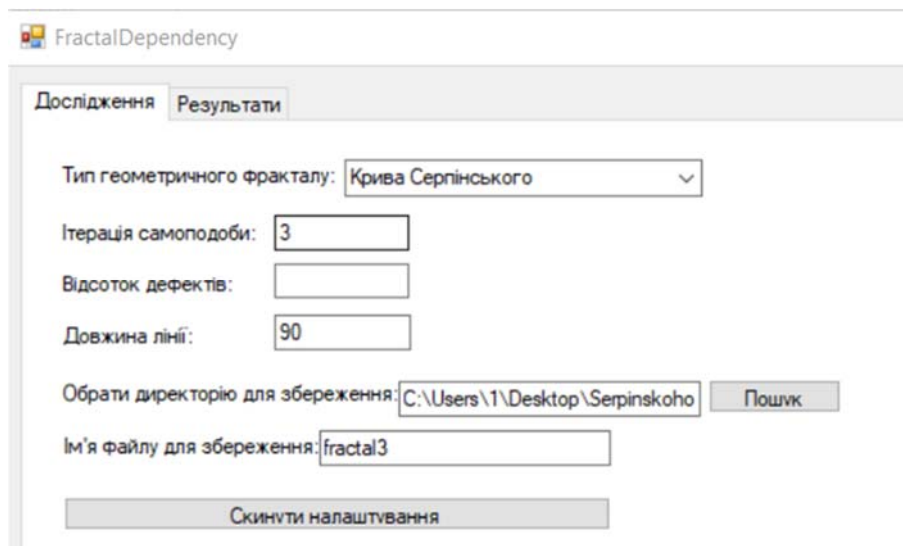


Рисунок 4.5 – Налаштування для дослідю

Таблиця 4.4 – Мінімальні, максимальні і середні значення розмірностей отримані в ході експерименту

| Відсоток дефектів | Мінімальна розмірність | Середня розмірність | Максимальна розмірність |
|-------------------|------------------------|---------------------|-------------------------|
| 0                 | 1,4257                 | 1,4257              | 1,4257                  |
| 1                 | 1,2726                 | 1,3819              | 1,4634                  |
| 3                 | 1,2263                 | 1,348               | 1,426                   |
| 5                 | 1,2474                 | 1,34                | 1,4333                  |
| 8                 | 1,2392                 | 1,3229              | 1,4071                  |
| 10                | 1,2071                 | 1,3266              | 1,3944                  |
| 15                | 1,2128                 | 1,3018              | 1,396                   |
| 20                | 1,2051                 | 1,2887              | 1,365                   |
| 25                | 1,1291                 | 1,2685              | 1,334                   |
| 30                | 1,1374                 | 1,2523              | 1,3354                  |
| 35                | 1,176                  | 1,2544              | 1,3232                  |
| 40                | 1,1448                 | 1,2247              | 1,316                   |

Графік отриманої в ході експерименту залежності між відсотком дефектів та значенням розмірності на рисунку 4.6.

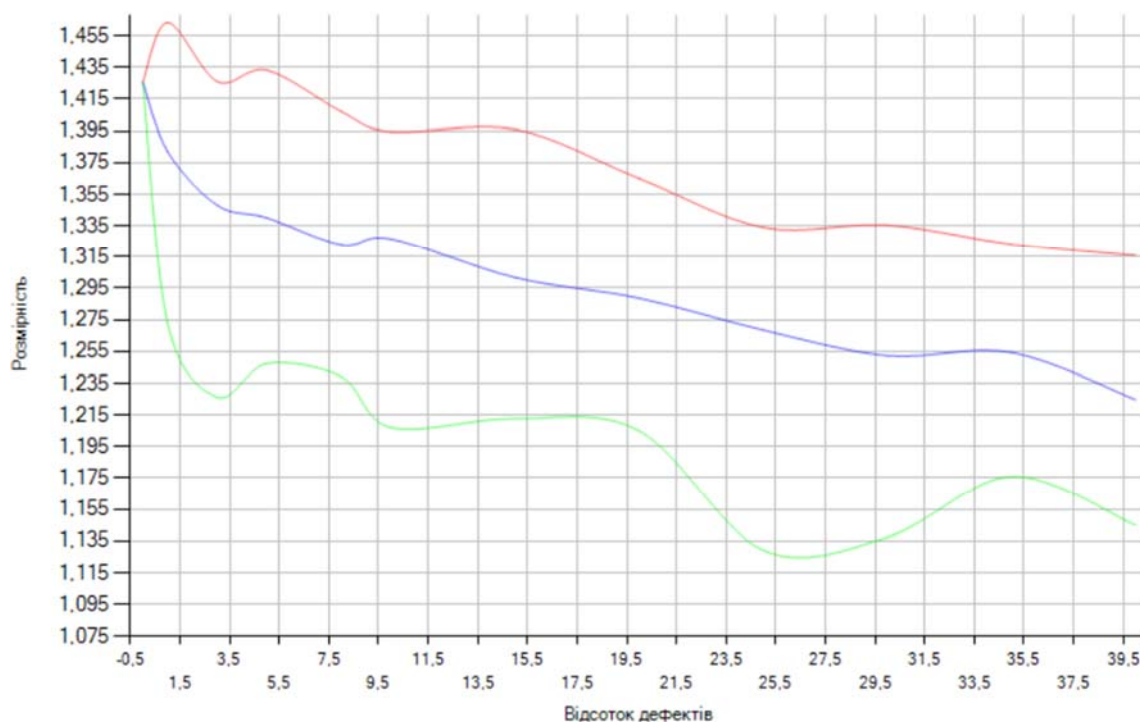


Рисунок 4.6 – Графік залежності між відсотком дефектів та значенням розмірності для кривої Серпінського. Червоним позначено максимальну розмірність, синім – середню, зеленим – мінімальну

#### 4.2.4 Вплив ступеня пошкодження першим методом фракталів кривої дракона на фрактальну розмірність

Результат ста експериментів розрахунку розмірності для кривої дракона при різних значеннях відсотку внесених дефектів представлено в таблиці 4.5. Вікно налаштувань програми на рисунку 4.7.

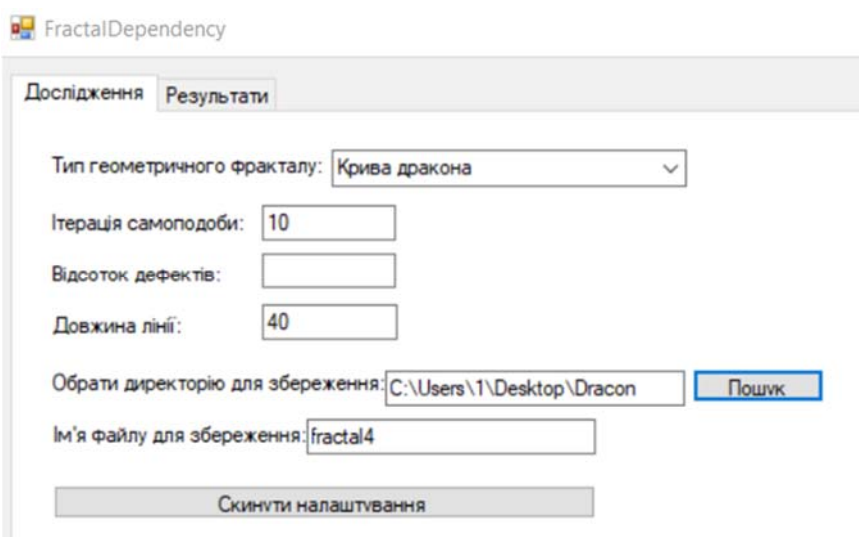


Рисунок 4.7 – Налаштування для дослідження

Таблиця 4.5 – Мінімальні, максимальні і середні значення розмірностей отримані в ході експерименту

| Відсоток дефектів | Мінімальна розмірність | Середня розмірність | Максимальна розмірність |
|-------------------|------------------------|---------------------|-------------------------|
| 0                 | 1,5988                 | 1,5988              | 1,5988                  |
| 1                 | 1,4705                 | 1,5663              | 1,6397                  |
| 3                 | 1,4095                 | 1,5239              | 1,5831                  |
| 5                 | 1,4316                 | 1,4983              | 1,5572                  |
| 8                 | 1,329                  | 1,4808              | 1,5613                  |
| 10                | 1,355                  | 1,4699              | 1,5627                  |
| 15                | 1,3554                 | 1,4586              | 1,5239                  |
| 20                | 1,341                  | 1,4355              | 1,5082                  |
| 25                | 1,3244                 | 1,4288              | 1,5024                  |
| 30                | 1,3065                 | 1,4066              | 1,5149                  |
| 35                | 1,2783                 | 1,3936              | 1,5048                  |
| 40                | 1,2842                 | 1,3801              | 1,4579                  |

Графік отриманої в ході експерименту залежності між відсотком дефектів та значенням розмірності на рисунку 4.8.

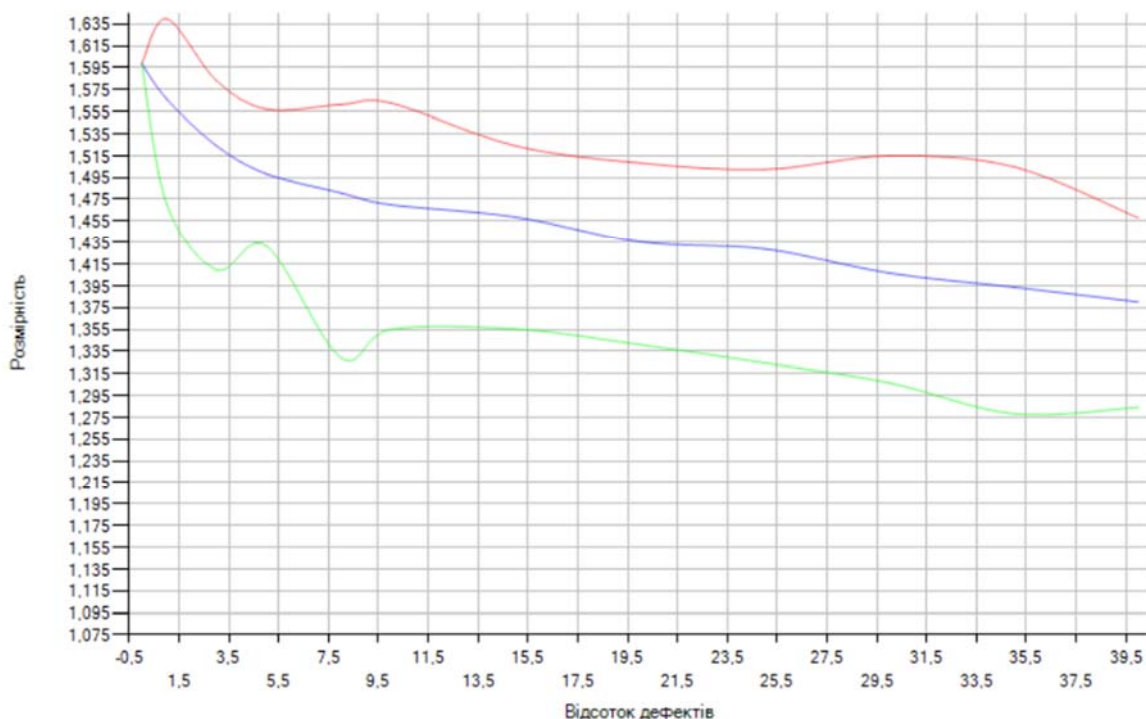


Рисунок 4.8 – Графік залежності між відсотком дефектів та значенням розмірності для кривої дракона. Червоним позначено максимальну розмірність, синім – середню, зеленим – мінімальну

#### 4.2.5 Вплив ступеня пошкодження першим методом фракталів кривої Леві на фрактальну розмірність

Результат ста експериментів розрахунку розмірності для кривої Леві при різних значеннях відсотку внесених дефектів представлено в таблиці 4.6. Вікно налаштувань програми на рисунку 4.9.

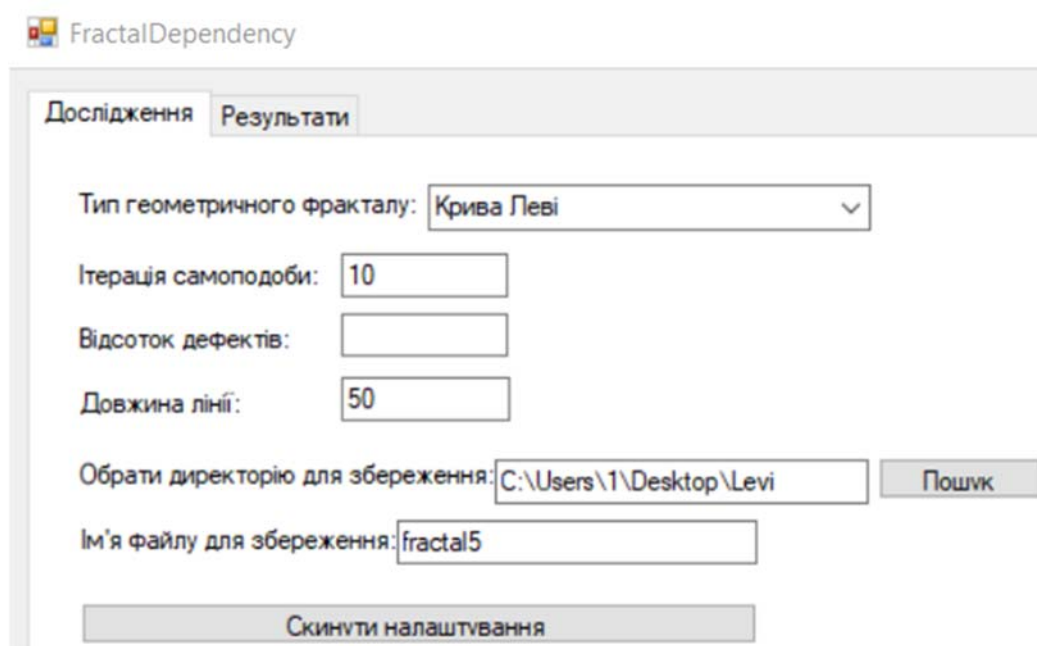


Рисунок 4.9 – Налаштування для дослідів

Таблиця 4.6 – Мінімальні, максимальні і середні значення розмірностей отримані в ході експерименту

| Відсоток дефектів | Мінімальна розмірність | Середня розмірність | Максимальна розмірність |
|-------------------|------------------------|---------------------|-------------------------|
| 0                 | 1,4251                 | 1,4251              | 1,4251                  |
| 1                 | 1,4017                 | 1,5276              | 1,6545                  |
| 3                 | 1,3519                 | 1,5169              | 1,6071                  |
| 5                 | 1,3927                 | 1,5068              | 1,6184                  |
| 8                 | 1,3542                 | 1,5071              | 1,6375                  |
| 10                | 1,3821                 | 1,4918              | 1,6076                  |
| 15                | 1,3805                 | 1,4844              | 1,6468                  |
| 20                | 1,3779                 | 1,4768              | 1,5992                  |
| 25                | 1,3607                 | 1,4684              | 1,5992                  |
| 30                | 1,3434                 | 1,4504              | 1,5379                  |
| 35                | 1,3329                 | 1,4391              | 1,5615                  |
| 40                | 1,3118                 | 1,44                | 1,5485                  |

Графік отриманої в ході експерименту залежності між відсотком дефектів та значенням розмірності на рисунку 4.10.

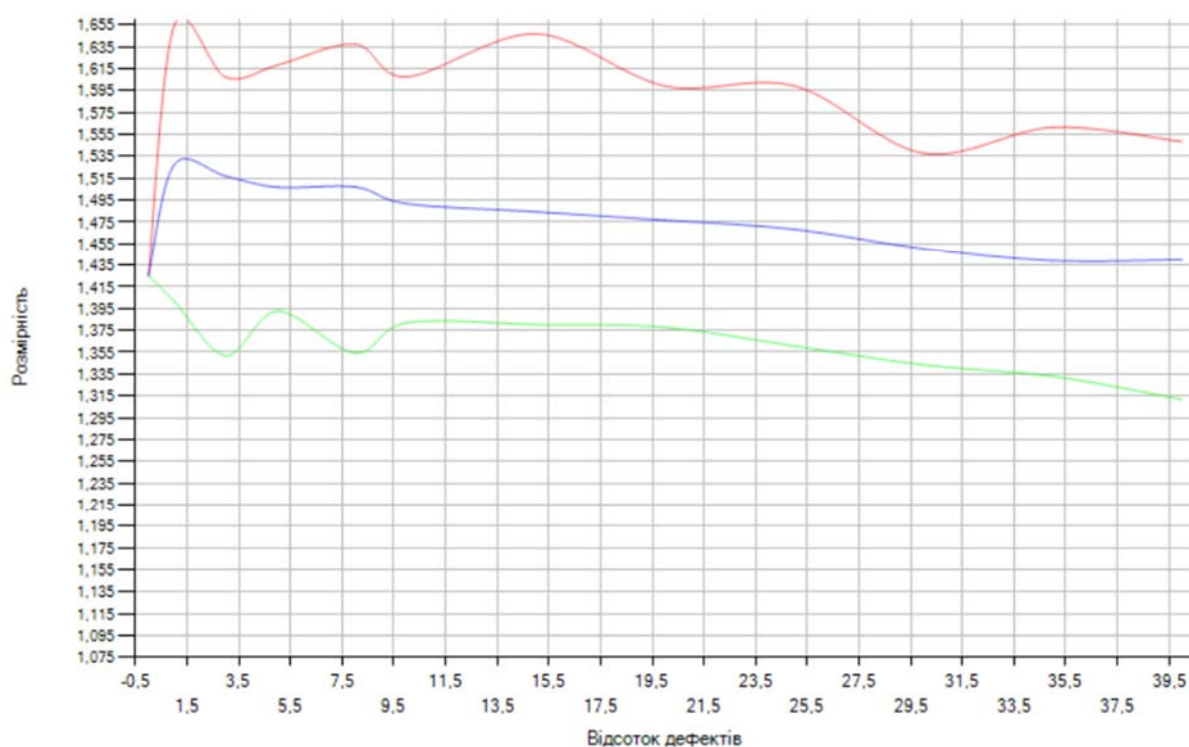


Рисунок 4.10 – Графік залежності між відсотком дефектів та значенням розмірності для кривої Леві. Червоним позначено максимальну розмірність, синім – середню, зеленим – мінімальну

#### 4.3 Дослідження розмірності часткових фракталів сформованих другим методом

Відсотки дефектів при дослідженні для кожного часткового фракталу, сформованого другим методом, взяті однакові, а саме: 3%, 5%, 10%, 15%, 20%, 25%.

##### 4.3.1 Залежність фрактальної розмірності від ступеня пошкодження другим методом фракталів сніжинки Коха

Результат п'ятидесяти експериментів розрахунку розмірності для сніжинки Коха при різних значеннях відсотку внесених дефектів представлено в таблиці 4.7. Вікно налаштувань програми на рисунку 4.11.

Дослідження    **Результати**

Тип геометричного фракталу: Сніжинка Коха

Ітерація самоподоби: 3

Відсоток дефектів:

Довжина лінії: 120

Обрати директорію для збереження: C:\Users\1\Desktop\Exp2    **Пошук**

Ім'я файлу для збереження: fractal

Рисунок 4.11 – Налаштування для дослідів

Таблиця 4.7 – Мінімальні, максимальні і середні значення розмірностей отримані в ході експерименту

| Відсоток дефектів | Мінімальна розмірність | Середня розмірність | Максимальна розмірність |
|-------------------|------------------------|---------------------|-------------------------|
| 0                 | 1,216                  | 1,216               | 1,216                   |
| 3                 | 1,1303                 | 1,197876            | 1,2666                  |
| 5                 | 1,1016                 | 1,189284            | 1,2565                  |
| 10                | 1,1116                 | 1,18871             | 1,314                   |
| 15                | 1,0881                 | 1,186852            | 1,3093                  |
| 20                | 1,1036                 | 1,175954            | 1,2454                  |
| 25                | 1,0757                 | 1,174136            | 1,2625                  |

Графік отриманої в ході експерименту залежності між відсотком дефектів та значенням розмірності на рисунку 4.12.

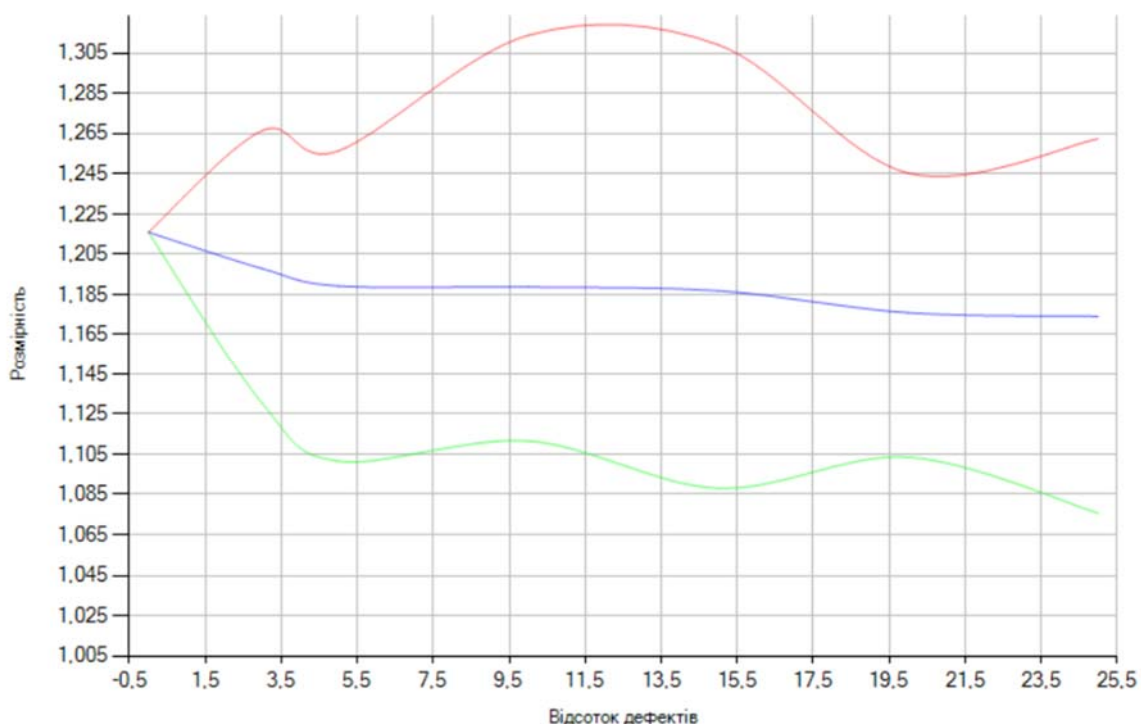


Рисунок 4.12 – Графік залежності між відсотком дефектів та значенням розмірності для сніжинки Коха. Червоним позначено максимальну розмірність, синім – середню, зеленим – мінімальну

#### 4.3.2 Залежність фрактальної розмірності від ступеня пошкодження другим методом фракталів квадратного острову Коха

Результат п'ятидесяти експериментів розрахунку розмірності для квадратного острову Коха при різних значеннях відсотку внесених дефектів представлено в таблиці 4.8. Вікно налаштувань програми на рисунку 4.13.

Вікно налаштувань програми з наступними параметрами:

- Тип геометричного фракталу: Квадратний острів Коха
- Ітерація самоподоби: 2
- Відсоток дефектів: (порожнє)
- Довжина лінії: 40
- Обрати директорію для збереження: C:\Users\1\Desktop\Exp2\_rvadra
- Ім'я файлу для збереження: fractal

Рисунок 4.13 – Налаштування для дослідження

Таблиця 4.8 – Мінімальні, максимальні і середні значення розмірностей отримані в ході експерименту

| Відсоток дефектів | Мінімальна розмірність | Середня розмірність | Максимальна розмірність |
|-------------------|------------------------|---------------------|-------------------------|
| 0                 | 1,3613                 | 1,3613              | 1,3613                  |
| 3                 | 1,3038                 | 1,3355              | 1,376                   |
| 5                 | 1,249                  | 1,3288              | 1,3939                  |
| 10                | 1,199                  | 1,3                 | 1,3571                  |
| 15                | 1,1725                 | 1,2828              | 1,3874                  |
| 20                | 1,1807                 | 1,2732              | 1,3385                  |
| 25                | 1,1313                 | 1,2522              | 1,3384                  |

Графік отриманої в ході експерименту залежності між відсотком дефектів та значенням розмірності на рисунку 4.14.

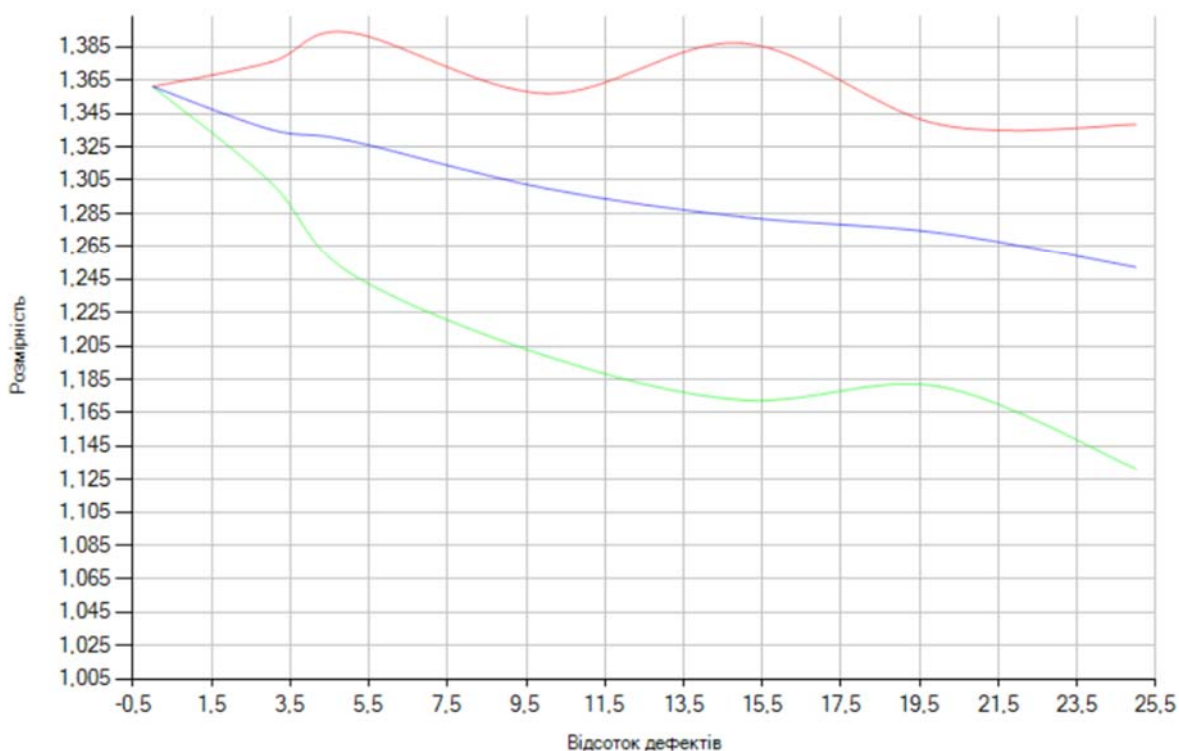


Рисунок 4.14 – Графік залежності між відсотком дефектів та значенням розмірності для квадратного острова Коха. Червоним позначено максимальну розмірність, синім – середню, зеленим – мінімальну

### 4.3.3 Залежність фрактальної розмірності від ступеня пошкодження другим методом фракталів кривої Серпінського

Результат п'ятидесяти експериментів розрахунку розмірності для кривої Серпінського при різних значеннях відсотку внесених дефектів представлено в таблиці 4.9. Вікно налаштувань програми на рисунку 4.15.

Рисунок 4.15 – Налаштування для дослідів

Таблиця 4.9 – Мінімальні, максимальні і середні значення розмірностей отримані в ході експерименту

| Відсоток дефектів | Мінімальна розмірність | Середня розмірність | Максимальна розмірність |
|-------------------|------------------------|---------------------|-------------------------|
| 0                 | 1,4257                 | 1,4257              | 1,4257                  |
| 3                 | 1,178                  | 1,3707              | 1,4389                  |
| 5                 | 1,2323                 | 1,3506              | 1,4336                  |
| 10                | 1,1774                 | 1,3297              | 1,4293                  |
| 15                | 1,1998                 | 1,3003              | 1,3829                  |
| 20                | 1,1862                 | 1,2917              | 1,3709                  |
| 25                | 1,1012                 | 1,2588              | 1,3936                  |

Графік отриманої в ході експерименту залежності між відсотком дефектів та значенням розмірності на рисунку 4.16.

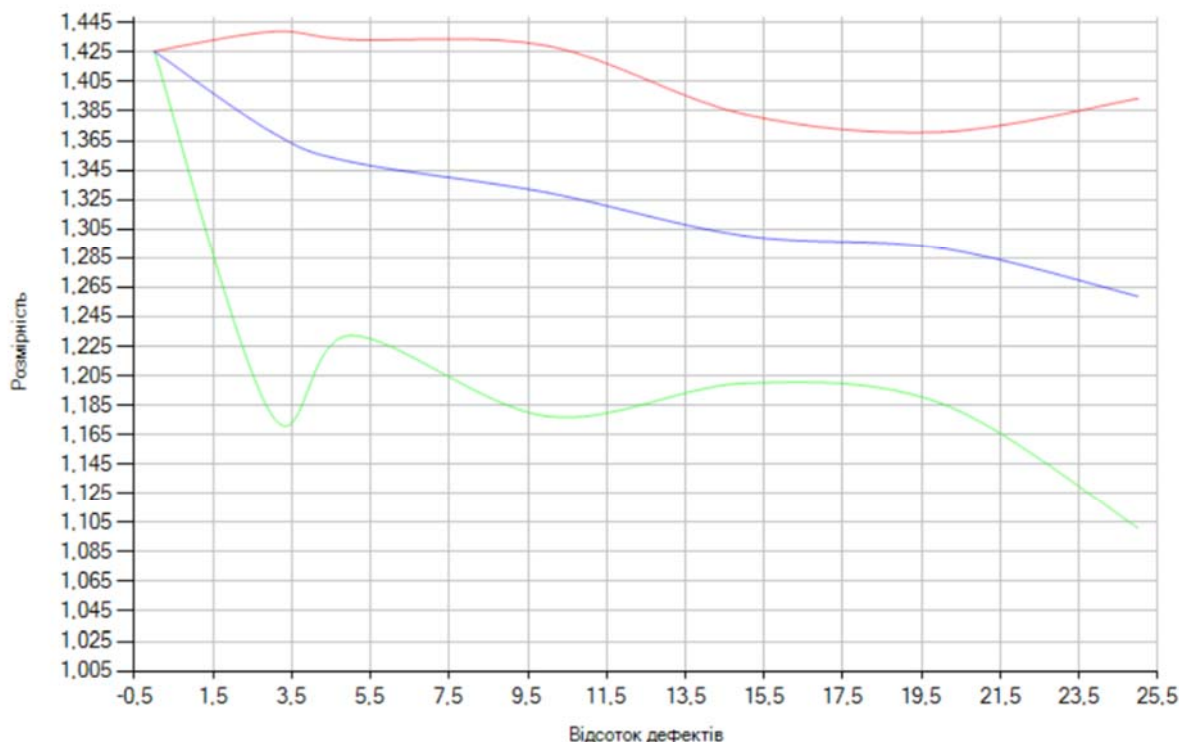


Рисунок 4.16 – Графік залежності між відсотком дефектів та значенням розмірності для кривої Серпінського. Червоним позначено максимальну розмірність, синім – середню, зеленим – мінімальну

#### 4.3.4 Залежність фрактальної розмірності від ступеня пошкодження другим методом фракталів кривої дракона

Результат п'ятидесяти експериментів розрахунку розмірності для кривої дракона при різних значеннях відсотку внесених дефектів представлено в таблиці 4.10. Вікно налаштувань програми на рисунку 4.17.

Дослідження    Результати

Тип геометричного фракталу:

Ітерація самоподоби:

Відсоток дефектів:

Довжина лінії:

Обрати директорію для збереження:

Ім'я файлу для збереження:

Рисунок 4.17 – Налаштування для дослідження

Таблиця 4.10 – Мінімальні, максимальні і середні значення розмірностей отримані в ході експерименту

| Відсоток дефектів | Мінімальна розмірність | Середня розмірність | Максимальна розмірність |
|-------------------|------------------------|---------------------|-------------------------|
| 0                 | 1,5988                 | 1,5988              | 1,5988                  |
| 3                 | 1,4651                 | 1,5366              | 1,6402                  |
| 5                 | 1,4353                 | 1,5164              | 1,5829                  |
| 10                | 1,3633                 | 1,4801              | 1,5699                  |
| 15                | 1,37                   | 1,481               | 1,5745                  |
| 20                | 1,3618                 | 1,4493              | 1,5557                  |
| 25                | 1,3531                 | 1,4363              | 1,5174                  |

Графік отриманої в ході експерименту залежності між відсотком дефектів та значенням розмірності на рисунку 4.18.

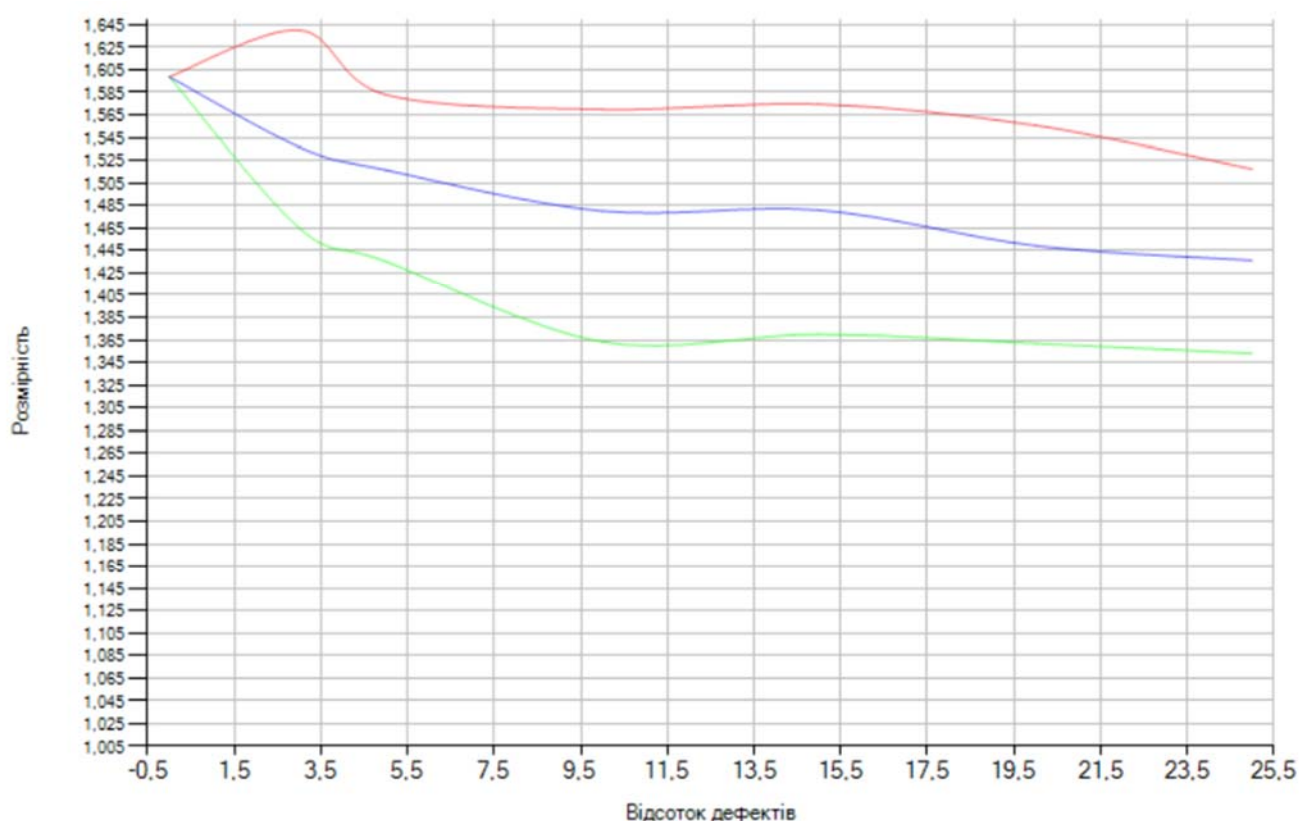


Рисунок 4.18 – Графік залежності між відсотком дефектів та значенням розмірності для кривої дракона. Червоним позначено максимальну розмірність, синім – середню, зеленим – мінімальну

### 4.3.5 Залежність фрактальної розмірності від ступеня пошкодження другим методом фракталів кривої Леві

Результат п'ятидесяти експериментів розрахунку розмірності для кривої Леві при різних значеннях відсотку внесених дефектів представлено в таблиці 4.11. Вікно налаштувань програми на рисунку 4.19.

Рисунок 4.19 – Налаштування для дослідів

Таблиця 4.11 – Мінімальні, максимальні і середні значення розмірностей отримані в ході експерименту

| Відсоток дефектів | Мінімальна розмірність | Середня розмірність | Максимальна розмірність |
|-------------------|------------------------|---------------------|-------------------------|
| 0                 | 1,4251                 | 1,4251              | 1,4251                  |
| 3                 | 1,4369                 | 1,5196              | 1,6163                  |
| 5                 | 1,4203                 | 1,5173              | 1,5957                  |
| 10                | 1,3809                 | 1,5084              | 1,6273                  |
| 15                | 1,3789                 | 1,5117              | 1,5983                  |
| 20                | 1,3101                 | 1,4763              | 1,5835                  |
| 25                | 1,3639                 | 1,48                | 1,6074                  |

Графік отриманої в ході експерименту залежності між відсотком дефектів та значенням розмірності на рисунку 4.20.

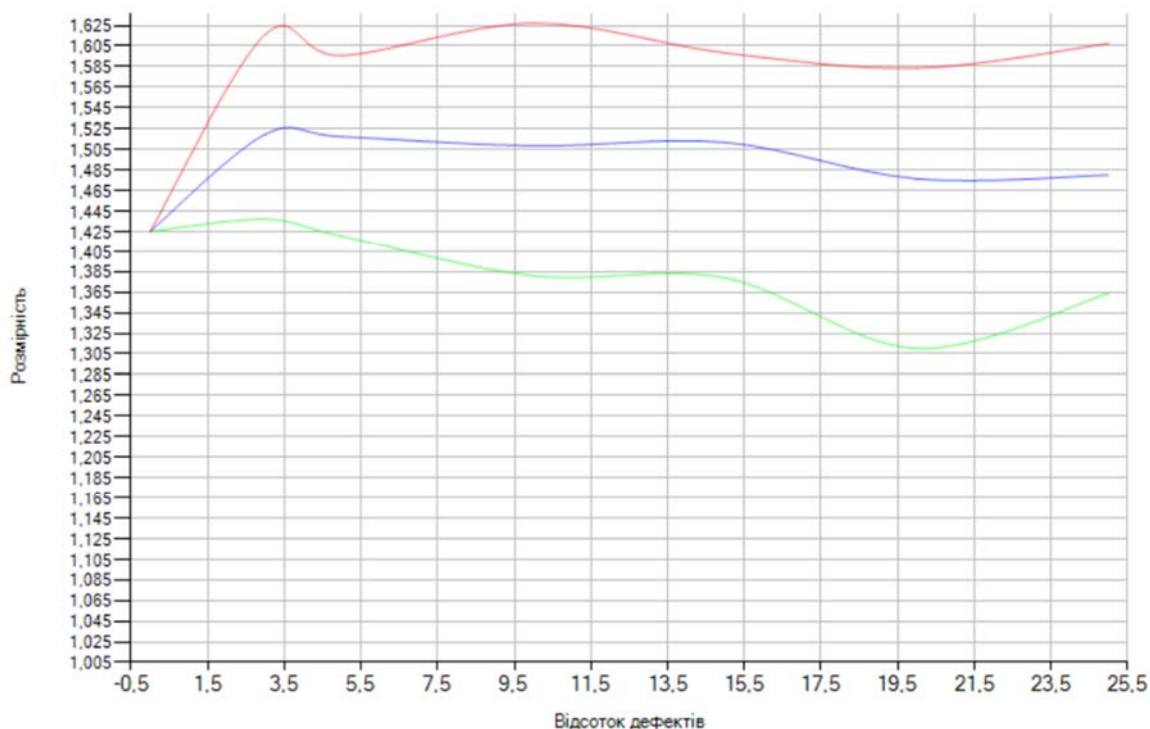


Рисунок 4.20 – Графік залежності між відсотком дефектів та значенням розмірності для кривої Леві. Червоним позначено максимальну розмірність, синім – середню, зеленим – мінімальну

#### 4.4 Результати експериментів

Аналізуючи усі утворені в ході експериментів графіки поведінки розмірності часткових фракталів, можна побачити, що зі збільшенням відсотку дефектів зменшується значення розмірності, яке поступово наближається до одиниці. Ця тенденція стосується обох описаних варіантів формування фракталів.

Для часткових фракталів сніжинки Коха результат обох експериментів показує незначне зменшення розмірності, а для кривої Леві експерименти показали, що з появою дефектів графік залежності розмірності стрибає вгору, а лише потім починає спадати.

Оскільки розмірність це один з важливих показників фрактальної структури фігури, то можна підсумувати, що зі збільшенням дефектів, внесених у регулярний фрактал, фігура втрачає свою фрактальність.

### Висновки до четвертого розділу

Для кожного обраного типу геометричного фракталу виконано по два досліді. За першим дослідом дефекти вводились в рядок після формування інструкції потрібної ітерації. За другим – дефекти вводились на ітерацію менше.

Представлені таблиці з розрахованими даними та графіки поведінки розмірності відносно кількості дефектів.

Аналізуючи отримані результати дослідження – зі збільшенням відсотку внесених дефектів до регулярного геометричного фракталу, значення його розмірності зменшується.

## ВИСНОВКИ

Розроблений метод дослідження та аналізу часткових фракталів та їхніх розмірностей має безпосереднє практичне застосування в різних галузях. Основні аспекти практичного застосування включають:

- 1) розробка методів формування часткових фракталів та введення дефектів може бути використана для створення нових моделей та дослідження їхнього впливу на складність та нерегулярні форми;
- 2) дослідження впливу різних типів та кількостей дефектів на структуру фракталів може бути корисним у виробництві матеріалів, де точність та надійність структур визначають їхні властивості;
- 3) визначення змін у фрактальній розмірності при різних рівнях пошкодженості регулярних фракталів може мати широке практичне застосування;
- 4) результати досліджень вносять нові аспекти в область вимірювання та аналізу фрактальних структур;
- 5) розроблений інтерфейс користувача та програмне забезпечення для генерації та аналізу фракталів представляють собою інноваційний інструмент для вивчення та дослідження фрактальних структур, розширюючи можливості аналізу їхньої геометричної складності.

Дана робота не лише пропонує новий погляд на розуміння фрактальних структур, але також надає практичні інструменти для їхнього вивчення та застосування в різних галузях науки та технології.

## БІБЛІОГРАФІЧНИЙ СПИСОК

1. Mandelbrot B. B., The fractal geometry of nature [Virtual Resource] / B. B. Mandelbrot // – 1982. – 460 p. – Access Mode: [https://archive.org/details/fractalgeometryo0000mand\\_i0s3/page/n7/mode/2up](https://archive.org/details/fractalgeometryo0000mand_i0s3/page/n7/mode/2up). – Title from Screen. – Date of Access: 2 December 2023.
2. Федік Л.Ю., Фрактальна графіка [Virtual Resource] / Л.Ю. Федік // – 2011. 1 p. – Access Mode: [https://elib.lntu.edu.ua/sites/default/files/elib\\_upload/page12.html](https://elib.lntu.edu.ua/sites/default/files/elib_upload/page12.html). – Title from Screen. – Date of Access: 15 December 2023.
3. Кветний Р. Н., Комп'ютерне моделювання систем та процесів. Методи обчислень. [Virtual Resource] / Р. Н. Кветний // – 2020. 514 p. – Access Mode: [https://web.posibnyky.vntu.edu.ua/fksa/2kvetnyj\\_komp'yuterne\\_modelyuvannya\\_system\\_procesiv/t2/217..htm](https://web.posibnyky.vntu.edu.ua/fksa/2kvetnyj_komp'yuterne_modelyuvannya_system_procesiv/t2/217..htm). – Title from Screen. – Date of Access: 15 December 2023.
4. Адашевська І. Ю., Конструктивні фрактали як результуюче стискуюче відображення подібності [Virtual Resource] / І. Ю. Адашевська // CORE. – 2019. – 1 January. – 8 p. – Access Mode: <https://core.ac.uk/download/pdf/286927257.pdf>. – Title from Screen. – Date of Access: 15 December 2023.
5. Кветний Р. Н., Комп'ютерне моделювання систем та процесів. Методи обчислень. Частина 2 [Virtual Resource] / Р. Н. Кветний // – 2020. 514 p. – Access Mode: [https://web.posibnyky.vntu.edu.ua/fksa/2kvetnyj\\_komp'yuterne\\_modelyuvannya\\_system\\_procesiv/t2/221..htm](https://web.posibnyky.vntu.edu.ua/fksa/2kvetnyj_komp'yuterne_modelyuvannya_system_procesiv/t2/221..htm) – Title from Screen. – Date of Access: 15 December 2023.
6. Рудик О. Основи фрактальної геометрії [Virtual Resource] / Рудик О. // - 2020. 1 p. – Access Mode: <http://www.kievoi.ippo.kubg.edu.ua/kievoi/dynsys/fractals.html> – Title from Screen. – Date of Access: 15 December 2023.
7. Jonsson F. BoxCount [Virtual Resource] / F. Jonsson // – 2024. 1 p. – Access Mode: <http://jonsson.eu/programs/cweb/boxcount/>. – Title from Screen. –

Date of Access: 15 December 2023.

8. Glare Technologies Limited. Chaotica [Virtual Resource] – 2024. 1 p. – Access Mode: <https://www.chaoticafractals.com/>. – Title from Screen. – Date of Access: 15 December 2023.
9. Trowbridge D. Fyre: News. [Virtual Resource] / D. Trowbridge // – 2005. 1 p. – Access Mode: <https://web.archive.org/web/20201213234427/http://fyre.navi.cx/>. – Title from Screen. – Date of Access: 15 December 2023.
10. Slijkerman F. Ultra Fractal 6 [Virtual Resource] / F. Slijkerman // – 2022. 1 p. – Access Mode: <https://www.ultrafractal.com/>. – Title from Screen. – Date of Access: 15 December 2023.
11. Fractal Explorer [Virtual Resource] – 2022. 1 p. – Access Mode: <http://www.fractal-explorer.com/>. – Title from Screen. – Date of Access: 15 December 2023.
12. Kleinbock. D. FalconerCh3 [Virtual Resource] / D. Kleinbock // – 2017. 1 p. – Access Mode: <https://people.brandeis.edu/~kleinboc/211a/FalconerCh3.pdf>. – Title from Screen. – Date of Access: 15 December 2023.
13. Microsoft. Build. Test. Deploy. [Virtual Resource] – 2023. 1 p. – Access Mode: <https://dotnet.microsoft.com/en-us/>. – Title from Screen. – Date of Access: 15 December 2023.
14. Маслюк В. О., Шинкаренко В. І. Визначення фрактальних розмірностей часткових геометричних фракталів [Virtual Resource] / В. О. Маслюк // Сучасні інформаційні та комунікаційні технології на транспорті, в промисловості і освіті: Тези XVII Міжнародної науково-практичної конференції (Дніпро, 13-14 грудня 2023 р.). – Д.: УДУНТ, 2023. – 1 с.

## ДОДАТКИ

## ДОДАТОК А

**Тези XVII Міжнародної науково-практичної конференції «СУЧАСНІ ІН-  
ФОРМАЦІЙНІ ТА КОМУНІКАЦІЙНІ  
ТЕХНОЛОГІЇ НА ТРАНСПОРТІ, В ПРОМИСЛОВОСТІ ТА ОСВІТІ»**

Міністерство освіти і науки України

Міністерство інфраструктури України

Український державний університет науки та технологій

Східний науковий центр транспортної академії наук



**TEMPUS: CITISET & SEREIN & CRENG**

## ТЕЗИ

**XVII Міжнародної науково-практичної конференції  
«СУЧАСНІ ІНФОРМАЦІЙНІ ТА КОМУНІКАЦІЙНІ  
ТЕХНОЛОГІЇ НА ТРАНСПОРТІ, В ПРОМИСЛОВОСТІ ТА ОСВІТІ»**

## ABSTRACTS

**of the XVII International Conference  
«MODERN INFORMATION AND COMMUNICATION TECHNOLOGIES  
ON A TRANSPORT, IN INDUSTRY AND EDUCATION»**

**13.12.2023 – 14.12.2023**

**Дніпро  
2023**

|   |    |
|---|----|
| Використання методики побудови рівнянь регресії до обробки результатів математичного моделювання в задачах геотехнічної механіки.....         | 74 |
| Ларіонов Г.І., Земляна Ю.В., Хворостян В.О., Головка С.А., Інститут геотехнічної механіки ім. М.С. Полякова НАН України                       |    |
| Розробка апаратної частини системи моніторингу пасіки.....  | 75 |
| Ларченко С. В., Дзюба В. В., Український державний університет науки і технологій, Україна  |    |
| Засоби формування графових 3D фракталів.....  | 76 |
| Летучий О. І., Шинкаренко В. І. Український державний університет науки і технологій, Україна   |    |
| Напрямки удосконалення технології роботи метрополітену.....   | 77 |
| Ломотько Д.В., Шаповалов А.М., Український державний університет науки і технологій, Україна  |    |
| Сучасні ефективні алгоритми сортування.....   | 78 |
| Макаров О. В., Шинкаренко В. І., Український державний університет науки і технологій, Україна  |    |
| Визначення фрактальних розмірностей часткових геометричних фракталів.....   | 79 |
| Маслюк В. О., Шинкаренко В. І., Український державний університет науки і технологій, Україна   |    |
| Алгоритм формування начерків зображення та сигналу у колірному просторі на засадах парадигми граничних узагальнень.....                       | 80 |
| Мельник В. В., Придніпровська державна академія будівництва і архітектури, Україна  |    |
| Особливості систем та алгоритмів керування безпілотними транспортними засобами.....   | 81 |
| Остапещ Я. Д., Дзюба В.В., Український державний університет науки і технологій, Україна  |    |
| Застосування інформаційних технологій в управлінні ризиками та оптимізації технічного обслуговування тягового рухомого складу.....            | 82 |
| Очкасов О.Б., Бобирь Д.В., Боднар Є.Б., Грипечкіна Т.С., Український державний університет науки і технологій, Україна                        |    |
| Перспективи розвитку систем моніторингу технічного стану дизельних тепловозів.....  | 83 |
| Очкасов О.Б., Студенко О.І., Український державний університет науки і технологій, Україна  |    |
| Організація передачі керуючих повідомлень в інформаційно-телекомунікаційній системі залізничного транспорту засобами нейронної мережі.....    | 84 |
| Пахомова В. М., Набоков Д. О., Український державний університет науки і технологій, Україна  |    |
| Дослідження можливості мурашиного методу для визначення топології бездротової локальної мережі на сортувальній станції.....                   | 85 |
| Пахомова В. М., Салогуб М. В., Український державний університет науки і технологій   |    |
| Прогнозування затримки на маршрутизаторі в інформаційно-телекомунікаційній системі залізничного транспорту засобами нейронечіткої мережі..... | 86 |
| Пахомова В. М., Хрестян А. В., Український державний університет науки і технологій   |    |

### Визначення фрактальних розмірностей часткових геометричних фракталів

Маслюк В. О., Шинкаренко В. І., Український державний університет науки і технологій,  
Україна

Вивчення фракталів набуває все більшої популярності. Фрактальні моделі та їх аналіз насправді дуже поширені не тільки в природних системах, а й у різних сферах науки, таких як економіка, медицина, хімія, логістика та інші. Серед популярних напрямів у комп'ютерних науках, де знайшли своє застосування фрактали – це графіка для ігор та програми для обробки зображень. Дослідження фракталів для цих сфер є важливим, оскільки вони допоможуть значно оптимізувати цифрову графіку та роботу з зображеннями, в тому числі і 3D моделями.

Фрактали це математичні структури, що виявляються у вигляді самоподібних геометричних образів на будь-якому масштабі. Оскільки у складних структурах може не бути ідеальної самоподібності, то постає питання вивчення часткових фракталів та їх властивостей. Часткові фрактали - це концепція, яка виникає в контексті фрактальної геометрії та використовується для опису певних фрактальних структур, які виявляють самоподібні властивості.

Серед властивостей фракталів виділяють самоподібність, ітеративність, дробові розмірності, заповненість простору та інші.

Робота присвячена вивченню розмірностей геометричних фракталів, які мають деяку самоподібність, але не є типовими фракталами у повному розумінні цього терміну.

За основу у дослідженні розмірності часткових геометричних фракталів взяті типові геометричні фрактали. Формування фракталу обрано через метод L-систем, що базуються на використанні набору правил для рекурсивного заміщення символів у строках. Для того, щоб отримана фігура мала тільки часткову самоподібність, в неї вносяться дефекти шляхом видалення деякого відсотка символів з сформованої строки випадковим чином.

Серед методів розрахунку розмірності фракталів виділяють: розмірність Хаусдорфа (Hausdorff Dimension), бокс-зліплений метод (Box-Counting Method), геодезичні методи та інші. Аналіз розмірності для часткових фракталів складніший та потребує більш специфічних та адаптованих методів, у порівнянні з правильними геометричними фракталами. Усі ці методи через складність у дослідженні є наближеними.

Основна ідея, обраного для даної роботи методу, полягає в розбитті фрактальної структури на квадрати різного розміру та визначенні кількості таких елементів, необхідних для покриття відповідної геометричної фігури.

Для визначення поведінки значень розмірності для різної ступені спотворення фракталу було згенеровано декілька геометричних фракталів з різним відсотком внесених дефектів. Обробка їх графічного вигляду виконується через збереження у файл типу .BMP з високим розширенням, оскільки треба уникнути небажаних дефектів через пікселізацію зображення. Кожен з цих часткових фракталів десять разів покривається квадратною сіткою з деякою змінною довжиною сторони квадрата. Наступним кроком підраховується кількість квадратів, які покривають фрактал, і на основі отриманих даних формується логарифмічна залежність для кількості квадратів та довжин сторін. Розрахована розмірність кожного згенерованого часткового фракталу та його відсоток деформацій відповідно переноситься на графік залежності для подальшого дослідження.

Для перевірки правильності отриманих розрахунків, було проведено дослідження на не зміненому фракталі сніжинки Коха, для якого вже відома розмірність, що становить приблизно 1,26. За цим дослідженням отримана розмірність сніжинки Коха має достатньо малі розбіжності з відомим значенням, що доводить правильність розрахунку представленим методом і для часткових фракталів.

ДОДАТОК Б  
Технічне завдання

ЗАТВЕРДЖУЮ  
Проректор  
Українського державного  
університету науки і технології  
Анатолій РАДКЕВИЧ

ПРОГРАМА З ВИЗНАЧЕННЯ ТА  
ДОСЛІДЖЕННЯ ФРАКТАЛЬНИХ  
РОЗМІРНОСТЕЙ ЧАСТКОВИХ  
ГЕОМЕТРИЧНИХ ФРАКТАЛІВ

Технічне завдання  
ЛИСТ ЗАТВЕРДЖЕННЯ  
44165850.1350-01-ЛЗ

Завідувач кафедри КІТ  
\_\_\_\_\_Вадим ГОРЯЧКІН

Керівник розробки  
\_\_\_\_\_Віктор ШИНКАРЕНКО

Виконавець  
\_\_\_\_\_Віра МАСЛЮК

Нормоконтролер  
\_\_\_\_\_Світлана ВОЛКОВА

ЗАТВЕРДЖЕНО  
44165850.1350-01-ЛЗ

ПРОГРАМА З ВИЗНАЧЕННЯ ТА  
ДОСЛІДЖЕННЯ ФРАКТАЛЬНИХ  
РОЗМІРНОСТЕЙ ЧАСТКОВИХ  
ГЕОМЕТРИЧНИХ ФРАКТАЛІВ

Технічне завдання

Листів 13

44165850.95210-01

2

АНОТАЦІЯ

Документ 44165850.1350-01 «ПРОГРАМА З ВИЗНАЧЕННЯ ТА ДОСЛІДЖЕННЯ ФРАКТАЛЬНИХ РОЗМІРНОСТЕЙ ЧАСТКОВИХ ГЕОМЕТРИЧНИХ ФРАКТАЛІВ. Технічне завдання».

У даному документі представлені підстави для розробки, призначення та область застосування, вимоги до виконання програмного продукту та документації.

## ЗМІСТ

|   |    |
|---|----|
| ВСТУП .....   | 3  |
| 1. ПІДСТАВИ ДЛЯ РОЗРОБКИ .....                            | 4  |
| 2. ПРИЗНАЧЕННЯ РОЗРОБКИ .....                             | 5  |
| 3. ВИМОГИ ДО ПРОГРАМНОГО ПРОДУКТУ .....                   | 6  |
| 4.1.Вимоги до функціональних характеристик.....           | 6  |
| 4.2.Умови експлуатації .....                              | 8  |
| 4.3.Вимоги до інформаційної та програмної сумісності..... | 8  |
| 4. ВИМОГИ ДО ПРОГРМАНОЇ ДОКУМЕНТАЦІЇ.....                 | 9  |
| 5. ПОРЯДОК КОНТРОЛЮ І ПРИЙМАННЯ.....                      | 10 |
| 6. БІБЛІОГРАФІЧНИЙ СПИСОК.....                            | 11 |

## ВСТУП

Програмний продукт «ПРОГРАМА З ВИЗНАЧЕННЯ ТА ДОСЛІДЖЕННЯ ФРАКТАЛЬНИХ РОЗМІРНОСТЕЙ ЧАСТКОВИХ ГЕОМЕТРИЧНИХ ФРАКТАЛІВ», що розробляється, призначений для визначення та дослідження фрактальних розмірностей часткових фракталів.

Фрактали - це об'єкти, які мають складну структуру та самоподібність на різних масштабах. Вони знаходять застосування в багатьох науках, включаючи фізику, біологію, економіку та інші.

Фрактальні розмірності - це параметри, які описують ступінь деталізації та складності фракталів. Зазвичай, для визначення фрактальних розмірностей використовуються різні методи, такі як метод обмежувальної кривої та метод ящика з комахами.

Дослідження фракталів зосереджені на розумінні і моделюванні складних систем зі складною геометрією, які мають фрактальну структуру. Використання фрактальних аналізів дозволяє досліджувати структуру та динаміку складних систем і виявляти характерні регулярності, що можуть бути невидимі для традиційних методів дослідження.

Причина розробки: недостатність готових програмних рішень з дослідження розмірностей для часткових фракталів.

Область застосувань: програмний продукт призначений для індивідуального використання.

## 1. ПІДСТАВИ ДЛЯ РОЗРОБКИ

Підставою для розробки є наказ від 05.12.2022 р. «Про призначення керівників та затвердження магістерських робіт», затверджений ректором Українського державного університету науки і технологій проф. Пшіньком О. М.

Тема роботи – «ВИЗНАЧЕННЯ ФРАКТАЛЬНИХ РОЗМІРНОСТЕЙ ЧАСТКОВИХ ГЕОМЕТРИЧНИХ ФРАКТАЛІВ»

## 2. ПРИЗНАЧЕННЯ РОЗРОБКИ

Функціональне призначення – програмний продукт призначений для виконання функцій відображення часткового фракталу, розрахунку розмірності отриманого фракталу та дослідження результатів впливу відсотку пошкоженості на значення розмірності.

Експлуатаційне призначення – програмний продукт може знайти застосування в технологічних процесах виробництва, наприклад, для удосконалення методів обробки матеріалів або створення структур з певними властивостями. Також може бути застосована у технологіях обробки зображень, наприклад, для створення реалістичних та високоякісних комп'ютерних зображень чи відео. Виконані у програмі дослідження можуть допомогти виявити вплив дефектів на фрактальні структури в природних системах. Це може мати важливе значення для розуміння екосистем, геологічних процесів, чи інших природних явищ.

### 3. ВИМОГИ ДО ПРОГРАМНОГО ПРОДУКТУ

#### 4.1. Вимоги до функціональних характеристик

Вимоги до функціональних характеристик наступні:

- програма повинна забезпечити можливість генерації часткових геометричних фракталів, враховуючи відсоток дефектів, ітерацію самоподоби, вид фракталу (сніжинка Коха, крива Леві, крива дракона, крива Серпінського, квадратний острів Коха) та довжину ліній фракталу. Описані вхідні дані мають вводиться користувачем;
- забезпечити можливість зберігання графічного зображення фракталу у достатньо високому розширенні, щоб пікселізація не заважала якості розрахунків;
- передбачити функції обрізання картинки до розмірів, які фактично займає зображення фракталу;
- можливість збереження отриманого фрактала в текстовий файл у вигляді рядка L-системи, даних про розмір ліній та кут повороту, відповідний до обраного фракталу;
- забезпечити можливість накладання сітки з різним розміром клітинок на лінії фракталу для наступних розрахунків;
- виконати підрахунок квадратів сітки, що містять в собі лінію фракталу;
- забезпечити можливість отримання розрахункових даних про розмірність фракталу;
- можливість використати вже згенерований фрактал для розрахунків знову;
- відображення залежностей фрактальної розмірності конкретного геометричного фракталу від відсотку дефектів;
- забезпечити можливість виконувати дослідження автоматично та відображати їх результати в вигляді графіку залежності.

Вхідні дані:

- тип фракталу;
- ітерація самоподоби;
- відсоток дефектів;
- розмір ліній фрактала;
- шлях до директорії для збереження файлів;
- ім'я для файлів, що будуть зберігатись.

Вихідні дані:

- текстовий файл з даними про згенерований фрактал;
- зображення типу BMP з графічним зображенням часткового фракталу в високому розширенні;
- обрізане зображення типу BMP з графічним зображенням часткового фракталу;
- значення розмірності фракталу;
- графік поведінки розмірності часткових фракталів відносно кількості дефектів, отриманий одиничним дослідом;
- графік поведінки розмірності часткових фракталів відносно кількості дефектів, отриманий автоматичним виконанням сотні дослідів, результати яких були збережені до текстових файлів.

Дані вводяться з використанням миші та клавіатури.

#### 4.2. Умови експлуатації

Програмний продукт повинен використовуватись у приміщеннях, які відповідають умовам роботи ЕОМ, а саме мають такі кліматичні, санітарні та гігієнічні умови, які відповідають ДНАОП 0.00-1.31-99 (див. табл. 1).

Таблиця 1 – Кліматичні умови

| Пора року | Категорія робіт згідно з ГОСТ 12.1-005-88 | Температура повітря, град.С | Відносна вологість повітря, % | Швидкість руху повітря, м/с |
|-----------|---|-----------------------------|-------------------------------|-----------------------------|
|           |   | оптимальна                  | оптимальна                    | оптимальна                  |
| Холодна   | легка-1 а                                 | 22 - 24                     | 40 - 60                       | 0,1                         |
|           | легка-1 б                                 | 21 - 23                     | 40 - 60                       | 0,1                         |
| Тепла     | легка-1 а                                 | 23 - 25                     | 40 - 60                       | 0,1                         |
|           | легка-1 б                                 | 22 - 24                     | 40 - 60                       | 0,2                         |

Працювати з програмою може людина, що має навички роботи з персональним комп'ютером та ознайомена з керівництвом користувача.

#### 4.3. Вимоги до інформаційної та програмної сумісності

Програмний продукт розробляється для операційної системи Microsoft Windows 10 Professional. Для розробки обрано мову програмування С# та фреймворк .NET Framework. Рекомендоване середовище розробки - Microsoft Visual Studio 2019. Всі використовувані сторонні бібліотеки та компоненти повинні бути сумісні з операційною системою та середовищем розробки.

#### 4. ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ

До складу програмної документації мають входити:

- текст програми;
- керівництво користувача.

Вся документація до програмного додатку повинна задовольняти вимоги до програмної документації [2].

## 5. ПОРЯДОК КОНТРОЛЮ І ПРИЙМАННЯ

Контроль за виконанням роботи здійснює керівник розробки проф. Шинкаренко В. І.

Приєм здійснюється комісією.

## 6. БІБЛІОГРАФІЧНИЙ СПИСОК

1. Mandelbrot, Benoit B. (1983). The Fractal Geometry of Nature. San Francisco: W.H. Freeman. ISBN 978-0-7167-1186-5.
2. Основи стандартизації програмних систем [Текст]: методичні вказівки до дипломного проектування та лабораторних робіт / уклад.: Ю.М.Івченко, В.І.Шинкаренко, В.Г.Івченко; Дніпропетр. нац. ун-т залізн. трансп. ім. акад. В.Лазаряна. – Д.: Вид-во Дніпропетр. нац. ун-ту залізн. трансп. ім. акад. В.Лазаряна, 2009. – 38 с.

ДОДАТОК Г  
Текст програми

ЗАТВЕРДЖУЮ  
Проректор  
Українського державного  
університету науки і технології  
Анатолій РАДКЕВИЧ

ПРОГРАМА З ВИЗНАЧЕННЯ ТА  
ДОСЛІДЖЕННЯ ФРАКТАЛЬНИХ  
РОЗМІРНОСТЕЙ ЧАСТКОВИХ  
ГЕОМЕТРИЧНИХ ФРАКТАЛІВ

Текст програми  
ЛИСТ ЗАТВЕРДЖЕННЯ  
44165850.1350-01 12 01-ЛЗ

Завідувач кафедри КІТ  
\_\_\_\_\_Вадим ГОРЯЧКІН

Керівник розробки  
\_\_\_\_\_Віктор ШИНКАРЕНКО

Виконавець  
\_\_\_\_\_Віра МАСЛЮК

Нормоконтролер  
\_\_\_\_\_Світлана ВОЛКОВА

ЗАТВЕРДЖЕНО  
44165850.1350-01 12 01-ЛЗ

ПРОГРАМА З ВИЗНАЧЕННЯ ТА  
ДОСЛІДЖЕННЯ ФРАКТАЛЬНИХ  
РОЗМІРНОСТЕЙ ЧАСТКОВИХ  
ГЕОМЕТРИЧНИХ ФРАКТАЛІВ

Текст програми

Листів 18

44165850.1350-01 12 01

## АНОТАЦІЯ

Документ 44165850.1350-01 12 01 «ПРОГРАМА З ВИЗНАЧЕННЯ ТА ДОСЛІДЖЕННЯ ФРАКТАЛЬНИХ РОЗМІРНОСТЕЙ ЧАСТКОВИХ ГЕОМЕТРИЧНИХ ФРАКТАЛІВ. Текст програми» входить до складу програмної документації на програму, що реалізує роботу з дослідження поведінки розмірності часткових геометричних фракталів відносно ступеня їх пошкодженості.

У даному документі представлений текст програми. Програма написана на мові C#. Об'єм пам'яті, що займають програми системи з урахування бібліотек складає від 1 ГБ. Конфігурація комп'ютер. Комплекс функціонує в середовищі операційних систем сімейства Windows починаючи з версії Windows 10.

## ЗМІСТ

|     |                                   |    |
|-----|-----------------------------------|----|
| 1   | СХЕМА ВЗАЄМОДІЇ МОДУЛІВ .....     | 4  |
| 2   | ТЕКСТ ПРОГРАМИ .....              | 5  |
| 2.1 | Текст модуля Form .....           | 5  |
| 2.2 | Текст модуля Program .....        | 12 |
| 2.3 | Текст модуля Extensions .....     | 12 |
| 2.4 | Текст модуля LSystemHelpers ..... | 13 |
| 2.5 | Текст модуля ImageHelpers .....   | 14 |
| 2.6 | Текст модуля FileHelpers .....    | 16 |

44165850.1350-01 12 01

## 1 СХЕМА ВЗАЄМОДІЇ МОДУЛІВ

Для демонстрації взаємодії частин програми, приведена діаграма класів. Класи в даному випадку і є модулями з яких складається програма.

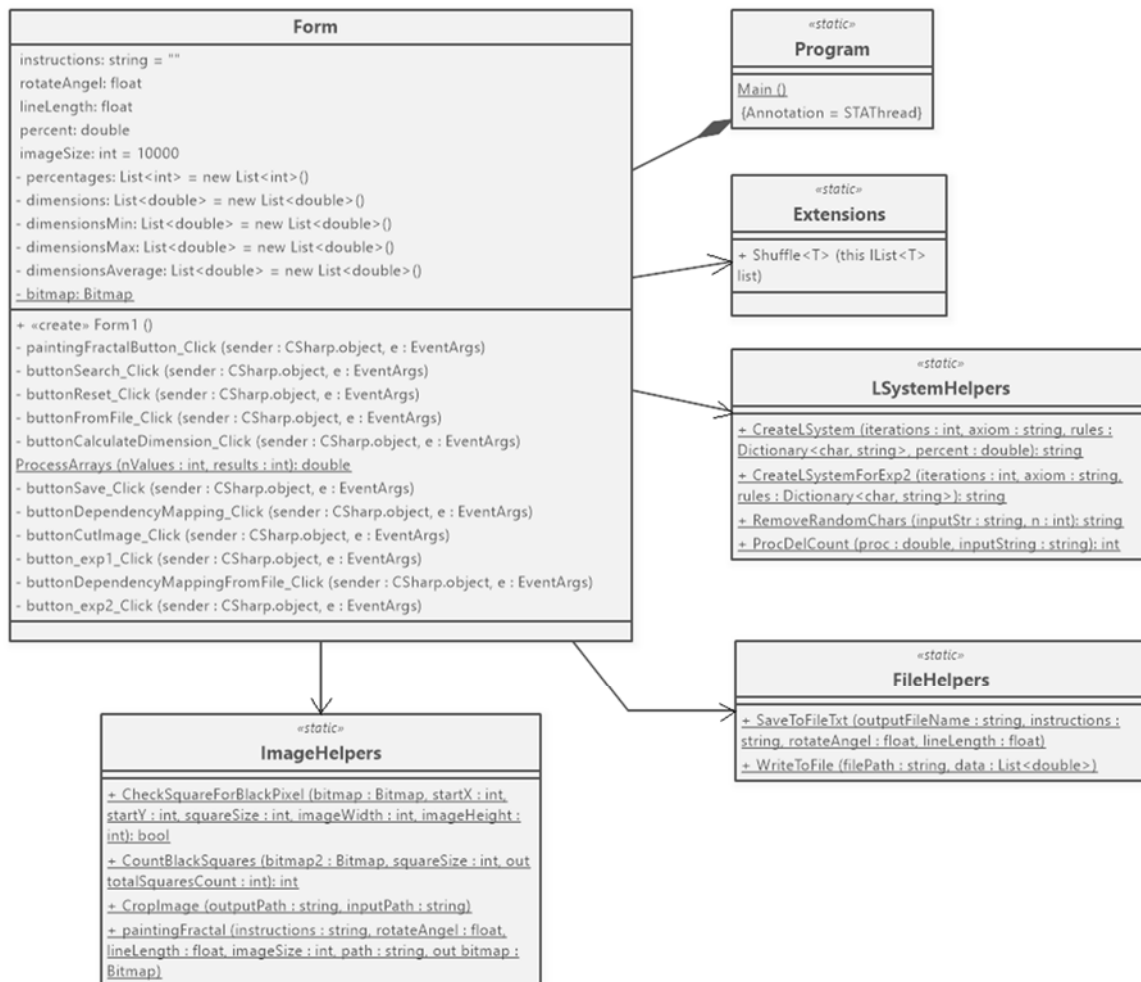


Рисунок 1.1 – Схема взаємодії модулів програми

## 2 ТЕКСТ ПРОГРАМИ

## 2.1 Текст модуля Form

```

using System;
using System.Collections.Generic;
using System.Data;
using System.Drawing;
using System.IO;
using System.Linq;
using System.Windows.Forms;
using System.Windows.Forms.DataVisualization.Charting;

namespace FractalApp
{
    public partial class Form1 : Form
    {
        string instructions = "";
        float rotateAngel;
        float lineLength;
        double percent;
        int imageSize = 10000;
        private List<int> percentages = new List<int>();
        private List<double> dimensions = new List<double>();
        private List<double> dimensionsMin = new List<double>();
        private List<double> dimensionsMax = new List<double>();
        private List<double> dimensionsAverage = new List<double>();
        private static Bitmap bitmap;
        public Form1()
        {
            InitializeComponent();
        }

        private void paintingFractalButton_Click(object sender, EventArgs e)
        {
            labelProcessPaintingFractal.Text = "";
            string path = textBoxDirectory.Text + "\\\" + textBoxNameFile.Text + ".bmp";
            lineLength = Convert.ToInt16(textBoxLineLength.Text);
            int iteration = Convert.ToInt16(textBoxIteration.Text);
            int percentint = Convert.ToInt32(textBoxPercent.Text);
            percent = percentint;
            percent = percent / 100;
            string axiom="";
            Dictionary<char, string> rules = new Dictionary<char, string>();
            int choise = comboBoxTypeFractal.SelectedIndex;
            switch (choise)
            {
                case 0:
                    axiom = "F--F--F";
                    rules = new Dictionary<char, string>
                    {
                        {'F', "F+F--F+F"}
                    };
                    rotateAngel = 60;
                    break;
                case 1:
                    axiom = "FX";
                    rules = new Dictionary<char, string>
                    {
                        {'X', "X+YF+"},
                        {'Y', "-FX-Y"}
                    };
                    rotateAngel = 90;
                    break;
                case 2:
                    axiom = "F+XF+F+XF";
                    rules = new Dictionary<char, string>
                    {

```

44165850.1350-01 12 01

```

        {'X',"XF-F+F-XF+F+XF-F+F-X"}
    };
    rotateAngel = 90;
    break;
case 3:
    axiom = "F";
    rules = new Dictionary<char, string>
    {
        {'F',"+F--F+"}
    };
    rotateAngel = 45;
    break;
case 4:
    axiom = "F+F+F+F";
    rules = new Dictionary<char, string>
    {
        {'F',"F-F+F+FFF-F-F+F"}
    };
    rotateAngel = 90;
    break;
}

instructions = LSystemHelpers.CreateLSystem(iteration, axiom, rules, percent);
string pathForSaveToTxt = textBoxDirectory.Text + "\\\" + textBoxNameFile.Text +
".txt";
FileHelpers.SaveToFileTxt(pathForSaveToTxt, instructions, rotateAngel,
lineLength);
bitmap = new Bitmap(imageSize, imageSize);
ImageHelpers.paintingFractal(instructions, rotateAngel, lineLength, imageSize,
path, ref bitmap);
labelProcessPaintingFractal.Text = "Готово!";
}

private void buttonSearch_Click(object sender, EventArgs e)
{
    FolderBrowserDialog ofd = new FolderBrowserDialog();

    if (ofd.ShowDialog() == DialogResult.OK)
    {
        textBoxDirectory.Text = ofd.SelectedPath;
    }
}

private void buttonReset_Click(object sender, EventArgs e)
{
    textBoxIteration.Text = "";
    textBoxPercent.Text = "";
    textBoxLineLength.Text = "";
    textBoxDirectory.Text = "";
    textBoxNameFile.Text = "";
}

private void buttonFromFile_Click(object sender, EventArgs e)
{
    labelProcessFromFile.Text = "";
    string path = textBoxDirectory.Text + "\\\" + textBoxNameFile.Text + ".bmp";
    OpenFileDialog instrfile = new OpenFileDialog();
    instrfile.Filter = "Txt Files (*.TXT)|*.TXT|All files (*.*)|*.*";

    if (instrfile.ShowDialog() == DialogResult.OK)
    {
        try
        {
            StreamReader sr = new StreamReader(instrfile.FileName);
            instructions = sr.ReadLine();
            rotateAngel = Convert.ToInt16(sr.ReadLine());
            lineLength = Convert.ToInt16(sr.ReadLine());
        }
        catch

```

44165850.1350-01 12 01

```

        {
            MessageBox.Show("Неможливо відкрити обраний файл", "Помилка",
                MessageBoxButtons.OK, MessageBoxIcon.Error);
        }

    }
    bitmap = new Bitmap(imageSize, imageSize);
    ImageHelpers.paintingFractal(instructions, rotateAngel, lineLength, imageSize,
        path, ref bitmap);
    labelProcessFromFile.Text = "Готово!";
}

private void buttonCalculateDimension_Click(object sender, EventArgs e)
{
    string fileName = textBoxDirectory.Text + "\\\" + textBoxNameFile.Text +
        "_обрізано" + ".bmp";
    labelDimensionResult.Text = "";
    Bitmap bitmap2 = new Bitmap(fileName);
    int imageWidth = bitmap2.Width;
    int imageHeight = bitmap2.Height;
    int minL = Math.Min(imageWidth, imageHeight);
    int maxN = (int)Math.Floor(minL * 0.2);
    int[] ns = new int[10];

    for (int i = 0, n = maxN; i < 10; i++)
    {
        ns[i] = n;
        n = (int)Math.Floor(n / 1.3);
    }

    int[] countResults = new int[10];

    // Выполнение функции подсчета для разных значений n
    for (int i = 0; i < 10; i++)
    {
        countResults[i] = ImageHelpers.CountBlackSquares(bitmap2, ns[i], out _);
    }

    var S = ProcessArrays(ns, countResults);

    // Форматування S з чотирма знаками після коми
    string formatted_S = $"{S:F4}";

    labelDimensionResult.Text = formatted_S;
}

static double ProcessArrays(int[] nValues, int[] results)
{
    double[] z = results.Select(r => Math.Log(r)).ToArray();
    double[] x = nValues.Select(n => Math.Log(n)).ToArray();

    double sumz = z.Sum();
    double sumx = x.Sum();
    double sumzx = z.Zip(x, (zVal, xVal) => zVal * xVal).Sum();
    double sumx2 = x.Select(xVal => xVal * xVal).Sum();

    double a = (sumzx - ((sumz * sumx) / nValues.Length)) / (-(sumx * sumx) /
        nValues.Length + sumx2);
    double S = -a;

    return S;
}

private void buttonSave_Click(object sender, EventArgs e)
{
    labelSave.Text = "";
    int percentageValue = Convert.ToInt16(textBoxPercent.Text);
    double dimensionValue = Convert.ToDouble(labelDimensionResult.Text);
}

```

## 44165850.1350-01 12 01

```

percentages.Add(percentageValue);
dimensions.Add(dimensionValue);
labelSave.Text = "Готово!";
}

private void buttonDependencyMapping_Click(object sender, EventArgs e)
{
    // Очистка попереднього графіка (якщо потрібно)
    chartDependence.Series.Clear();

    // Створення нового ряду для графіка
    Series series = new Series("Dependency");
    series.ChartType = SeriesChartType.Spline;

    // Додавання точок до ряду на основі зібраних значень
    for (int i = 0; i < percentages.Count; i++)
    {
        series.Points.AddXY(percentages[i], dimensions[i]);
    }

    // Додавання ряду до графіка
    chartDependence.Series.Add(series);

    Series pointSeries = new Series("Points");
    pointSeries.ChartType = SeriesChartType.Point;
    for (int i = 0; i < percentages.Count; i++)
    {
        pointSeries.Points.AddXY(percentages[i], dimensions[i]);
    }
    chartDependence.Series.Add(pointSeries);

    // Налаштування вигляду графіка (необов'язково)
    chartDependence.ChartAreas[0].AxisX.Title = "Percentage";
    chartDependence.ChartAreas[0].AxisY.Title = "Dimension";

    chartDependence.ChartAreas[0].AxisX.Minimum = percentages.Min() - 2;
    chartDependence.ChartAreas[0].AxisX.Maximum = percentages.Max() + 2;
    chartDependence.ChartAreas[0].AxisY.Minimum = dimensions.Min() - 0.05;
    chartDependence.ChartAreas[0].AxisY.Maximum = dimensions.Max() + 0.05;

    chartDependence.ChartAreas[0].AxisX.Interval = 2;
    chartDependence.ChartAreas[0].AxisX.IntervalType = DateTimeIntervalType.Number;
    chartDependence.ChartAreas[0].AxisY.Interval = 0.02;
    chartDependence.ChartAreas[0].AxisY.IntervalType = DateTimeIntervalType.Number;

    // Оновлення графіка
    chartDependence.Update();
}

private void buttonCutImage_Click(object sender, EventArgs e)
{
    labelCutImage.Text = "";

    string inputPath = textBoxDirectory.Text + "\\\" + textBoxNameFile.Text + ".bmp";
    string outputPath = textBoxDirectory.Text + "\\\" + textBoxNameFile.Text +
    "_обрізано" + ".bmp";
    ImageHelpers.CropImage(outputPath, inputPath);
    labelCutImage.Text = "Готово!";
}

private void button_expl_Click(object sender, EventArgs e)
{
}

private void buttonDependencyMappingFromFile_Click(object sender, EventArgs e)
{
    // Очистка попереднього графіка (якщо потрібно)
    chartDependence.Series.Clear();
    percentages = new List<int> { 0, 3, 5, 10, 15, 20, 25 };
}

```

## 44165850.1350-01 12 01

```

//percentages = new List<int> { 0,1, 3, 5, 8, 10, 15, 20, 25, 30, 35, 40 };
// Створення нового ряду для графіка
Series series1 = new Series("Максимальна розмірність");
series1.ChartType = SeriesChartType.Spline;
Series series2 = new Series("Мінімальна розмірність");
series2.ChartType = SeriesChartType.Spline;
Series series3 = new Series("Середня розмірність");
series3.ChartType = SeriesChartType.Spline;

string path1 = "Max.txt";
string path2 = "Min.txt";
string path3 = "Average.txt";

try
{
    StreamReader sr = new StreamReader(path1);
    for (int i=0; i<7; i++) //13
    {
        dimensionsMax.Add(Convert.ToDouble(sr.ReadLine()));
    }
}
catch
{
    MessageBox.Show("Неможливо відкрити обраний файл", "Помилка",
    MessageBoxButtons.OK, MessageBoxIcon.Error);
}
try
{
    StreamReader sr = new StreamReader(path2);
    for (int i = 0; i < 7; i++) //13
    {
        dimensionsMin.Add(Convert.ToDouble(sr.ReadLine()));
    }
}
catch
{
    MessageBox.Show("Неможливо відкрити обраний файл", "Помилка",
    MessageBoxButtons.OK, MessageBoxIcon.Error);
}
try
{
    StreamReader sr = new StreamReader(path3);
    for (int i = 0; i < 7; i++) //13
    {
        dimensionsAverage.Add(Convert.ToDouble(sr.ReadLine()));
    }
}
catch
{
    MessageBox.Show("Неможливо відкрити обраний файл", "Помилка",
    MessageBoxButtons.OK, MessageBoxIcon.Error);
}

// Додавання точок до ряду на основі зібраних значень
for (int i = 0; i < percentages.Count; i++)
{
    series1.Points.AddXY(percentages[i], dimensionsMax[i]);
}
for (int i = 0; i < percentages.Count; i++)
{
    series2.Points.AddXY(percentages[i], dimensionsMin[i]);
}
for (int i = 0; i < percentages.Count; i++)
{
    series3.Points.AddXY(percentages[i], dimensionsAverage[i]);
}

// Додавання ряду до графіка

```

44165850.1350-01 12 01

```

chartDependence.Series.Add(series1);
chartDependence.Series.Add(series2);
chartDependence.Series.Add(series3);

//Series pointSeries = new Series("Points");
//pointSeries.ChartType = SeriesChartType.Point;
//for (int i = 0; i < percentages.Count; i++)
//{
//    pointSeries.Points.AddXY(percentages[i], dimensions[i]);
//}
//chartDependence.Series.Add(pointSeries);

// Налаштування вигляду графіка (необов'язково)
chartDependence.ChartAreas[0].AxisX.Title = "Відсоток дефектів";
chartDependence.ChartAreas[0].AxisY.Title = "Розмірність";

chartDependence.ChartAreas[0].AxisX.Minimum = percentages.Min() - 0.5;
chartDependence.ChartAreas[0].AxisX.Maximum = percentages.Max() + 0.5;
chartDependence.ChartAreas[0].AxisY.Minimum = 1.005;
chartDependence.ChartAreas[0].AxisY.Maximum = dimensionsMax.Max() + 0.01;

chartDependence.ChartAreas[0].AxisX.Interval = 2;
chartDependence.ChartAreas[0].AxisX.IntervalType = DateTimeIntervalType.Number;
chartDependence.ChartAreas[0].AxisY.Interval = 0.02;
chartDependence.ChartAreas[0].AxisY.IntervalType = DateTimeIntervalType.Number;

// Оновлення графіка
chartDependence.Update();
}

private void button_exp2_Click(object sender, EventArgs e)
{
    percentages = new List<int> { 3,5,10,15,20,25 };
    for (int k = 0; k < 6; k++)
    {
        int percentint = percentages[k];
        for (int i = 0; i < 100; i++)
        {
            string path = textBoxDirectory.Text + "\\\" + textBoxNameFile.Text + "_" +
i + k + ".bmp";
            lineLength = Convert.ToInt16(textBoxLineLength.Text);
            int iteration = Convert.ToInt16(textBoxIteration.Text);
            iteration = iteration - 1;
            percent = percentint;
            percent = percent / 100;
            string axiom = "";
            Dictionary<char, string> rules = new Dictionary<char, string>();
            int choise = comboBoxTypeFractal.SelectedIndex;
            switch (choise)
            {
                case 0:
                    axiom = "F--F--F";
                    rules = new Dictionary<char, string>
                    {
                        { 'F', "F+F--F+F" }
                    };
                    rotateAngel = 60;
                    break;
                case 1:
                    axiom = "FX";
                    rules = new Dictionary<char, string>
                    {
                        { 'X', "X+YF+" },
                        { 'Y', "-FX-Y" }
                    };
                    rotateAngel = 90;
                    break;
            }
        }
    }
}

```

## 44165850.1350-01 12 01

```

        case 2:
            axiom = "F+XF+F+XF";
            rules = new Dictionary<char, string>
            {
                {'X', "XF-F+F-XF+F+XF-F+F-X"}
            };
            rotateAngel = 90;
            break;
        case 3:
            axiom = "F";
            rules = new Dictionary<char, string>
            {
                {'F', "+F--F+"}
            };
            rotateAngel = 45;
            break;
        case 4:
            axiom = "F+F+F+F";
            rules = new Dictionary<char, string>
            {
                {'F', "F-F+F+FFF-F-F+F"}
            };
            rotateAngel = 90;
            break;
    }

    instructions = LSystemHelpers.CreateLSystem(iteration, axiom, rules,
percent);
    instructions = LSystemHelpers.CreateLSystemForExp2(1, instructions,
rules);
    string pathForSaveToTxt = textBoxDirectory.Text + "\\\" +
textBoxNameFile.Text + "_" + i + k + ".txt";
    FileHelpers.SaveToFileTxt(pathForSaveToTxt, instructions, rotateAngel,
lineLength);
    bitmap = new Bitmap(imageSize, imageSize);
    ImageHelpers.paintingFractal(instructions, rotateAngel, lineLength,
imageSize, path, ref bitmap);
    labelProcessPaintingFractal.Text = "Готово!" + (char)i;
    }
    for (int j = 0; j < 100; j++)
    {
        string inputPath = textBoxDirectory.Text + "\\\" + textBoxNameFile.Text +
        "_\" + j + k + ".bmp";
        string outputPath = textBoxDirectory.Text + "\\\" + textBoxNameFile.Text +
        "_обрізано\" + "_" + j + k + ".bmp";
        ImageHelpers.CropImage(outputPath, inputPath);
        labelCutImage.Text = "Готово!" + (char)j;
    }
    }
    for (int k = 0; k < 6; k++)
    {
        List<double> dimensions1 = new List<double>();
        for (int j = 0; j < 100; j++)
        {
            string fileName = textBoxDirectory.Text + "\\\" + textBoxNameFile.Text +
            "_обрізано\" + "_" + j + k + ".bmp";
            Bitmap bitmap2 = new Bitmap(fileName);
            int imageWidth = bitmap2.Width;
            int imageHeight = bitmap2.Height;
            int minL = Math.Min(imageWidth, imageHeight);
            int maxN = (int)Math.Floor(minL * 0.2);
            int[] ns = new int[10];

            // Инициализация массива ns начиная с 20 и уменьшая на 2 10 раз
            for (int i = 0, n = maxN; i < 10; i++)
            {

```

44165850.1350-01 12 01

```

        ns[i] = n;
        n = (int)Math.Floor(n / 1.3);
    }

    int[] countResults = new int[10];

    // Выполнение функции подсчета для разных значений n
    for (int i = 0; i < 10; i++)
    {
        countResults[i] = ImageHelpers.CountBlackSquares(bitmap2, ns[i], out
_);
    }

    var S = ProcessArrays(ns, countResults);

    // Форматирования S з чотирма знаками після коми
    string formatted_S = $"{S:F4}";
    S = Convert.ToDouble(formatted_S);
    dimensions1.Add(S);
}

double minValue = dimensions1.Min();
double maxValue = dimensions1.Max();
double averageValue = dimensions1.Average();
dimensionsMin.Add(minValue);
dimensionsMax.Add(maxValue);
dimensionsAverage.Add(averageValue);
}

string filePath1 = "Min.txt";
string filePath2 = "Max.txt";
string filePath3 = "Average.txt";
FileHelpers.WriteToFile(filePath1, dimensionsMin);
FileHelpers.WriteToFile(filePath2, dimensionsMax);
FileHelpers.WriteToFile(filePath3, dimensionsAverage);
}
}
}

```

## 2.2 Текст модуля Program

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace FractalApp
{
    internal static class Program
    {
        /// <summary>
        /// Главная точка входа для приложения.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new Form1());
        }
    }
}

```

## 2.3 Текст модуля Extensions

```

using System;
using System.Collections.Generic;

namespace FractalApp
{
    static class Extensions
    {
        public static void Shuffle<T>(this IList<T> list)
        {
            Random random = new Random();
            int n = list.Count;
            while (n > 1)
            {
                n--;
                int k = random.Next(n + 1);
                T value = list[k];
                list[k] = list[n];
                list[n] = value;
            }
        }
    }
}

```

## 2.4 Текст модуля LSystemHelpers

```

using System;
using System.Collections.Generic;
using System.Linq;

namespace FractalApp
{
    static class LSystemHelpers
    {
        public static string CreateLSystem(int iterations, string axiom, Dictionary<char,
string> rules, double percent)
        {
            string startString = axiom;
            if (iterations == 0)
            {
                return axiom;
            }

            string endString = "";
            for (int i = 0; i < iterations; i++)
            {
                endString = "";
                foreach (char c in startString)
                {
                    if (rules.ContainsKey(c))
                    {
                        endString += rules[c];
                    }
                    else
                    {
                        endString += c;
                    }
                }
                startString = endString;
            }
            int n = ProcDelCount(percent, endString);
            string modifiedString = RemoveRandomChars(endString, n);

            endString = modifiedString;
            return endString;
        }

        public static string CreateLSystemForExp2(int iterations, string axiom,

```

44165850.1350-01 12 01

```

Dictionary<char, string> rules)
{
    string startString = axiom;

    string endString = "";
    for (int i = 0; i < iterations; i++)
    {
        endString = "";
        foreach (char c in startString)
        {
            if (rules.ContainsKey(c))
            {
                endString += rules[c];
            }
            else
            {
                endString += c;
            }
        }
        startString = endString;
    }

    return endString;
}

public static string RemoveRandomChars(string inputStr, int n)
{
    if (n >= inputStr.Length)
    {
        return "";
    }

    List<int> indices = new List<int>(Enumerable.Range(0, inputStr.Length));
    indices.Shuffle();

    List<string> result = new List<string>();
    int prevIndex = 0;

    foreach (int index in indices.Take(n).OrderBy(i => i))
    {
        result.Add((string)(object)inputStr.Substring(prevIndex, index - prevIndex));
        prevIndex = index + 1;
    }

    result.Add(inputStr.Substring(prevIndex));

    return string.Join("", result);
}

public static int ProcDelCount(double proc, string inputString)
{
    int stringLen = inputString.Length;
    return (int)Math.Floor(stringLen * proc);
}
}
}

```

## 2.5 Текст модуля ImageHelpers

```

using System;
using System.Drawing;
using System.Windows.Forms;

namespace FractalApp
{
    static class ImageHelpers
    {
        public static bool CheckSquareForBlackPixel(Bitmap bitmap, int startX, int startY, int

```

## 44165850.1350-01 12 01

```

squareSize, int imageWidth, int imageHeight)
{
    for (int y = startY; y < startY + squareSize && y < imageHeight; y++)
    {
        for (int x = startX; x < startX + squareSize && x < imageWidth; x++)
        {
            System.Drawing.Color pixelColor = bitmap.GetPixel(x, y);

            // Проверяем, является ли пиксель черным
            if (pixelColor.R == 0 && pixelColor.G == 0 && pixelColor.B == 0)
            {
                return true; // Найден черный пиксель в квадрате
            }
        }
    }

    return false; // В квадрате нет черных пикселей
}

public static int CountBlackSquares(Bitmap bitmap2, int squareSize, out int
totalSquaresCount)
{
    int imageWidth = bitmap2.Width;
    int imageHeight = bitmap2.Height;

    int blackSquaresCount = 0;
    totalSquaresCount = 0;

    for (int y = 0; y < imageHeight; y += squareSize)
    {
        for (int x = 0; x < imageWidth; x += squareSize)
        {
            totalSquaresCount++;

            bool containsBlackPixel = CheckSquareForBlackPixel(bitmap2, x, y,
squareSize, imageWidth, imageHeight);
            if (containsBlackPixel)
            {
                blackSquaresCount++;
            }
        }
    }

    return blackSquaresCount;
}

public static void CropImage(string outputPath, string inputPath)
{
    try
    {
        using (Bitmap image = new Bitmap(inputPath))
        {
            int width = image.Width;
            int height = image.Height;

            // Находим самый верхний, нижний, правый и левый черные пиксели
            int left = width, top = height, right = 0, bottom = 0;

            for (int x = 0; x < width; x++)
            {
                for (int y = 0; y < height; y++)
                {
                    Color pixel = image.GetPixel(x, y);
                    if (pixel.R < 30 && pixel.G < 30 && pixel.B < 30) // Проверка на
черный цвет
                    {
                        left = Math.Min(left, x);
                        top = Math.Min(top, y);
                        right = Math.Max(right, x + 1);
                    }
                }
            }
        }
    }
}

```



44165850.1350-01 12 01

```
using System.Collections.Generic;
using System.IO;
using System.Windows.Forms;

namespace FractalApp
{
    static class FileHelpers
    {
        public static void SaveToFileTxt(string outputFileName, string instructions, float
rotateAngel, float lineLength)
        {
            try
            {
                File.WriteAllText(outputFileName,
${instructions}\n{rotateAngel}\n{lineLength}");
            }
            catch
            {
                MessageBox.Show("Невозможно зберегти файл", "Помилка", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }

        public static void WriteToFile(string filePath, List<double> data)
        {
            try
            {
                using (StreamWriter writer = File.AppendText(filePath))
                {
                    foreach (var item in data)
                    {
                        writer.WriteLine(item);
                    }
                }
            }
            catch
            {
                MessageBox.Show("Невозможно зберегти файл", "Помилка", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }
    }
}
```

ДОДАТОК В  
Керівництво користувача

ЗАТВЕРДЖУЮ  
Проректор  
Українського державного  
університету науки і технології  
Анатолій РАДКЕВИЧ

ПРОГРАМА З ВИЗНАЧЕННЯ ТА  
ДОСЛІДЖЕННЯ ФРАКТАЛЬНИХ  
РОЗМІРНОСТЕЙ ЧАСТКОВИХ  
ГЕОМЕТРИЧНИХ ФРАКТАЛІВ

Керівництво користувача  
ЛИСТ ЗАТВЕРДЖЕННЯ  
44165850.1350-01 ІЗ 01-ЛЗ

Завідувач кафедри КІТ  
\_\_\_\_\_Вадим ГОРЯЧКІН

Керівник розробки  
\_\_\_\_\_Віктор ШИНКАРЕНКО

Виконавець  
\_\_\_\_\_Віра МАСЛЮК

Нормоконтролер  
\_\_\_\_\_Світлана ВОЛКОВА

ЗАТВЕРДЖЕНО  
44165850.1350-01 ІЗ 01-ЛЗ

ПРОГРАМА З ВИЗНАЧЕННЯ ТА  
ДОСЛІДЖЕННЯ ФРАКТАЛЬНИХ  
РОЗМІРНОСТЕЙ ЧАСТКОВИХ  
ГЕОМЕТРИЧНИХ ФРАКТАЛІВ

Керівництво користувача

Листів 11

44165850.1350-01 ІЗ 01

### АНОТАЦІЯ

Документ 44165850.1350-01 ІЗ 01 «ПРОГРАМА З ВИЗНАЧЕННЯ ТА ДОСЛІДЖЕННЯ ФРАКТАЛЬНИХ РОЗМІРНОСТЕЙ ЧАСТКОВИХ ГЕОМЕТРИЧНИХ ФРАКТАЛІВ. Керівництво користувача».

У даному документі описані рекомендації з користування програмою. Програма написана на мові С#. Об'єм пам'яті, що займає програма складає від 1 ГБ. Конфігурація комп'ютер. Комплекс функціонує в середовищі операційних систем сімейства Windows починаючи з версії Windows 10.

## ЗМІСТ

|     |  |    |
|-----|--|----|
| 1   | ВСТУП .....  | 4  |
| 2   | ПРИЗНАЧЕННЯ ТА УМОВИ ЗАСТОСУВАННЯ.....   | 5  |
| 3   | ОПИС ОПЕРАЦІЙ.....   | 6  |
| 3.1 | Інструкція роботи з програмою при виконанні одиничного дослідження вручну.       | 6  |
| 3.2 | Інструкція роботи з програмою для дослідження з багатьма експериментами<br>..... | 7  |
| 3.3 | Робота з вкладкою «Результати».....  | 8  |
| 4   | АВАРІЙНІ СИТУАЦІЇ .....  | 9  |
|     | СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ .....   | 10 |

## 1 ВСТУП

Сучасна наука активно виявляє нові сфери застосування для математичних структур з фрактальними рисами. Однією з ключових властивостей для вивчення фракталів є їхній фрактальний розмір, який трактується неоднозначно, і існує кілька методів його розрахунку. Кожен із цих підходів може допомагати описувати досліджувані об'єкти з різних перспектив. Актуальність проведення досліджень обумовлена недостатнім розумінням розмірностей фігур, які виникають в результаті змін у регулярних фракталах.

## 2 ПРИЗНАЧЕННЯ ТА УМОВИ ЗАСТОСУВАННЯ

Функціональне призначення – програмний продукт призначений для виконання функцій відображення часткового фракталу, розрахунку розмірності отриманого фракталу та дослідження результатів впливу відсотку пошкодженості на значення розмірності.

Експлуатаційне призначення – програмний продукт може знайти застосування в технологічних процесах виробництва, наприклад, для удосконалення методів обробки матеріалів або створення структур з певними властивостями. Також може бути застосована у технологіях обробки зображень, наприклад, для створення реалістичних та високоякісних комп'ютерних зображень чи відео. Виконані у програмі дослідження можуть допомогти виявити вплив дефектів на фрактальні структури в природних системах. Це може мати важливе значення для розуміння екосистем, геологічних процесів, чи інших природних явищ.

Дослідження фракталів зосереджені на розумінні і моделюванні складних систем зі складною геометрією, які мають фрактальну структуру. Використання фрактальних аналізів дозволяє досліджувати структуру та динаміку складних систем і виявляти характерні регулярності, що можуть бути невидимі для традиційних методів дослідження.

Для використання програми потрібне наступне обладнання:

- мінімум 8 гігабайт оперативної пам'яті;
- процесор: 64 розрядний процесор з тактовою частотою 2.8-4.1 ГГц або вище;
- мінімум 10 гігабайт простору на жорсткому диску;
- засоби контролю, такі як миша, віддалений робочий стіл, клавіатура;
- монітор.

### 3 ОПИС ОПЕРАЦІЙ

#### 3.1 Інструкція роботи з програмою при виконанні одиничного дослідження вручну

Для виконання одиничного дослідження за введеними параметрами слід дотриматись наступних кроків:

- 1) Оберіть з випадаючого меню «Тип геометричного фракталу»;
- 2) Введіть необхідні дані в числовому вигляді у поля «Ітерація самоподоби», «Відсоток дефектів» та «Довжина лінії»;
- 3) Введіть у поле «Обрати директорію для збереження» або оберіть через кнопку «Пошук» директорію, де будуть збережені файли в ході роботи;
- 4) Задайте ім'я для файлів, що зберігаються, у полі «Ім'я файлу для збереження»;
- 5) Далі для того щоб згенерувати зображення фракталу за введеними параметрами натисніть кнопку «Зобразити фрактал». Зображення, яке буде збережене як результат натиснення цієї кнопки – це файл «\*.bmp» розміру  $10000 \times 10000$  пікселів та файл «\*.txt»;
- 6) Перед тим, як виконати розрахунок розмірності треба обов'язково обрізати зображення по розмірам фігури, тобто наступний крок – натиснення на кнопку «Обрізати зображення». Результатом натиснення цієї кнопки буде файл «\*\_обрізано.bmp»;
- 7) Далі для отримання значення розмірності згенерованої фігури, натискаємо на кнопку «Розрахувати розмірність». Через деякий час після обробки нижче цієї кнопки буде виведено отримане значення розмірності;
- 8) Якщо потрібно виконати цей дослід з різними вхідними значеннями «Відсотку дефектів» та відобразити графік залежності розмірності від кількості дефектів – натисніть після кожного виконаного розрахунку кнопку «Зберегти дані». Тобто виконуйте інструкцію описану вище змінюючи тільки значення відсотку дефектів і зберігайте розрахунки.

Для виконання одиничного дослідження використовуючи раніше збережені дані фракталу слід дотриматись наступних кроків:

- 1) Введіть у поле «Обрати директорію для збереження» або оберіть через кнопку «Пошук» директорію, де будуть збережені файли в ході роботи;
- 2) Задайте ім'я для файлів, що зберігаються, у полі «Ім'я файлу для збереження»;
- 3) Натисніть на кнопку «Зобразити з файлу» та оберіть файл «\*.txt» з готовими інструкціями для фракталу;
- 4) Перед тим, як виконати розрахунок розмірності треба обов'язково обрізати зображення по розмірам фігури, тобто наступний крок – натиснення на кнопку «Обрізати зображення». Результатом натиснення цієї кнопки буде файл «\*\_обрізано.bmp»;
- 5) Далі для отримання значення розмірності згенерованої фігури, натискаємо на кнопку «Розрахувати розмірність». Через деякий час після обробки нижче цієї кнопки буде виведено отримане значення розмірності.

### 3.2 Інструкція роботи з програмою для дослідження з багатьма експериментами

Для виконання автоматизованих дослідів виконайте наступні кроки:

- 1) Оберіть з випадаючого меню «Тип геометричного фракталу»;
- 2) Введіть необхідні дані в числовому вигляді у поля «Ітерація самоподоби», «Відсоток дефектів» та «Довжина лінії»;
- 3) Введіть у поле «Обрати директорію для збереження» або оберіть через кнопку «Пошук» директорію, де будуть збережені файли в ході роботи;
- 4) Задайте ім'я для файлів, що зберігаються, у полі «Ім'я файлу для збереження»;
- 5) В папці з програмою зайдіть три текстові файли з назвами: «Average.txt», «Max.txt» та «Min.txt». Перевірте, щоб в них не було даних попередніх досліджень,

якщо ці файли не пусті – очистіть їх;

б) В залежності від методу побудови фрактального зображення оберіть між кнопками «100 експеримент 1» та «100 експеримент 2» та натисніть;

7) Обробка буде тривати деякий час, все залежить від технічних параметрів комп'ютера, на якому виконується програма;

8) Результати обчислень будуть збережені до файлів «Average.txt», «Max.txt» та «Min.txt».

### 3.3 Робота з вкладкою «Результати»

Вкладка результати пропонує розглянути або графік побудований з тимчасових даних або обрати для демонстрації залежності дані з файлу.

Натисніть на кнопку «Тимчасово збережені дані», якщо виконували та зберігали результати одиничних дослідів.

Перед тим, як натиснути кнопку «Дані з файлу», перевірте наявність даних в файлах «Average.txt», «Max.txt» та «Min.txt», що знаходяться в одній папці з програмою. Дані в файли можна вписувати самостійно, враховуючи попередні виконані дослідження, або дані там з'являться автоматично, після закінчення виконання автоматичного дослідів.

#### 4 АВАРІЙНІ СИТУАЦІЇ

Оскільки програма обробляє дуже велику кількість даних, може виникнути помилка переповнення стеку. Програмно передбачити момент появи такої помилки доволі складно. Щоб уникнути переривання робочого процесу через таку помилку, рекомендується перезавантажувати програму після великої кількості виконаних дослідів.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Mandelbrot, Benoît B. (1983). The Fractal Geometry of Nature. San Francisco: W.H. Freeman. ISBN 978-0-7167-1186-5.
2. Основи стандартизації програмних систем [Текст]: методичні вказівки до дипломного проектування та лабораторних робіт / уклад.: Ю.М.Івченко, В.І.Шинкаренко, В.Г.Івченко; Дніпропетр. нац. ун-т залізн. трансп. ім. акад. В.Лазаряна. – Д.: Вид-во Дніпропетр. нац. ун-ту залізн. трансп. ім. акад. В.Лазаряна, 2009. – 38 с.