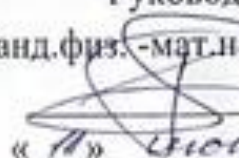


Министерство образования и науки Российской Федерации
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)
Механико-математический факультет
Кафедра вычислительной математики и компьютерного моделирования

Допустить к защите в ГЭК
Руководитель ООП
канд. физ.-мат. наук, доцент
Л.В. Гензе

« 11 » июня 2021 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА

ВИЗУАЛИЗАЦИЯ ГЕОМЕТРИЧЕСКИХ ПРЕОБРАЗОВАНИЙ С
ПОМОЩЬЮ WOLFRAM MATHEMATICA
по направлению подготовки 02.03.01 – Математика и компьютерные науки
Направленность (профиль) «Основы научно-исследовательской деятельности
в области Математики и компьютерных наук»

Бер Кристина Андреевна

Руководитель ВКР
канд. физ.-мат. наук

В.М. Зюзьков
« 11 » июня 2021 г.

Автор работы
студент группы №04702

К.А. Бер

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

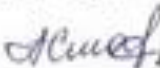
Механико-математический факультет

Кафедра вычислительной математики и компьютерного моделирования

УТВЕРЖДАЮ

Зав. кафедрой КВМ и КМ

д-р. физ.-мат. наук профессор

 А.В. Старченко

" 15 " сентября 2020г.

ЗАДАНИЕ

по бакалаврской работе студенту Бер Кристине Андреевне

Тема работы: «Визуализация геометрических преобразований с помощью *Wolfram Mathematica*»

Сроки защиты: июнь 2021 г

1. Содержание работы (вопросы, подлежащие разработке)

- а) Изучение литературы по темам: аффинные преобразования (сжатие, поворот, параллельный перенос, преобразование подобия, масштабирование, инверсия), геометрические преобразования на плоскости в языке Wolfram.
- б) Программирование: визуализация простых видов аффинных преобразований плоскости и их композиций.
- в) Программирование: визуализация инверсии.
- г) Изучение литературы по темам: классические фракталы, размерность Хаусдорфа, функции языка Wolfram для построения фракталов.
- д) Программирование фракталов на плоскости: кривая Серпинского; снежинка Коха; кривые, заполняющих плоскость.
- е) Изучение литературы по теме «Детерминированные системы итерированных функций (ДСИМ)».
- ж) Программирование фракталов с помощью ДСИМ».
- з) Изучение алгоритма рандомизированных системы итерированных функций и построение с его помощью хаотических фракталов.
- и) Написание бакалаврской работы и подготовка к защите.

2. Сроки выполнения основных этапов работы:

- а) и б) с 15 сентября по 15 октября 2020г;
- в) с 15 октября по 1 ноября 2020г;
- г) и д) с 1 ноября по 20 декабря 2020г;
- е) февраль – март 2021г;
- ж) март – апрель 2021г;
- з) апрель – май 2021г;
- и) май 2021г.

Студент обязан являться к руководителю курсовой работы с отчетом о проделанной работе и для получения консультаций не реже 1 раза в неделю.

3. Дата выдачи задания 15.09.2020

Руководитель ВКР бакалавра

Зюзьков Валентин Михайлович, к.ф.-м.н., старший научный сотрудник,
доцент ТГУ

Подпись руководителя 

Задание принял к выполнению 15.09.2020

Подпись студента 

АННОТАЦИЯ

Важную роль в математике играют геометрические преобразования. Наглядные изображения геометрических преобразований позволяют лучше понять математику предметной области.

Объектом и предметом исследования в данной ВКР являются: визуализация двумерных геометрических преобразований разных видов (аффинные преобразования, инверсия, фрактальные преобразования).

Цель исследования: алгоритмизация и создание программ на языке Wolfram Mathematica, с помощью которых можно наглядно показать геометрические преобразования. В частности, большое внимание уделяется:

- 1) построению фракталов различного типа;
- 2) созданию демонстраций - интерактивных вычислимых документов (CDF).

Задачи:

- 1) Изучение литературы по темам: аффинные преобразования (поворот, параллельный перенос, преобразование подобия, масштабирование), инверсия, геометрические преобразования на плоскости в языке Wolfram;
- 2) Программирование: визуализация простых видов аффинных преобразований плоскости и их композиций;
- 3) Программирование: визуализация инверсии;
- 4) Изучение литературы по темам: классические фракталы, размерность Хаусдорфа, функции языка Wolfram для построения фракталов;
- 5) Программирование фракталов на плоскости: салфетка и ковер Серпинского; кривая Коха; кривые, заполняющие плоскость;
- 6) Изучение литературы по теме «Детерминированные системы итерированных функций (ДСИМ)»;
- 7) Программирование фракталов с помощью ДСИМ;
- 8) Изучение алгоритма рандомизированных систем итерированных функций и построение с его помощью хаотических фракталов.

Методы: математические, большей частью геометрические; использование языка Wolfram для программирования в системе Mathematica.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	7
1 Геометрические преобразования	10
1.1 Параллельный перенос	10
1.2 Поворот	13
1.3 Отображение подобия	16
1.4 Масштабирование	17
1.5 Композиция преобразований	20
1.6 Аффинные преобразования	21
1.7 Инверсия	24
2 Фракталы	28
2.1 Фракталы: развитие и характеристики	28
2.2 Береговая линия	29
2.3 Фрактальная размерность	30
2.4 Канторовское множество	32
2.5 Примеры известных фракталов	35
2.5.1 Снежинка Коха	35
2.5.2 Салфетка и ковер Серпинского	36
2.5.3 Дракон Хартера-Хейтуэя	39
3 Методы построения фракталов	41
3.1 Система итерируемых функций (СИФ)	41
3.1.1 Детерминированный метод	41
3.1.2 Метод случайных итераций, или игра в хаос	44
3.1.3 Игры с поворотами	47
3.2 Лист папоротника	50
4 Построение фракталов с помощью Wolfram Mathematica	58
4.1 Описание основных функций	58

4.2 Кривая Коха.....	59
4.3 Салфетка и ковер Серпинского	60
4.4 Модификации салфетки Серпинского	64
4.5 Лист папоротника	68
4.6 Дракон Хартера-Хейтуэя	70
4.7 Примеры известных фракталов	73
ЗАКЛЮЧЕНИЕ.....	81
ЛИТЕРАТУРА	82
ПРИЛОЖЕНИЕ А	83
ПРИЛОЖЕНИЕ Б	87

ВВЕДЕНИЕ

«Вся наука записана в этой великой книге, - я имею в виду Вселенную, - которая всегда открыта для нас, но которую нельзя понять, не научившись понимать язык, на котором она написана. А написана она на языке математики, и её буквами являются треугольники, окружности и другие геометрические фигуры, без которых человеку невозможно разобрать ни одного её слова; без них он подобен блуждающему во тьме» - Галилео Галилей, 1623 г. [1, с. 5].

Результаты исследования направлены на создание математических и программных инструментов, с помощью которых можно уменьшить трудности в понимании и изучении геометрических преобразований и фракталов.

Объектом и предметом исследования в данной ВКР являются: визуализация двумерных геометрических преобразований разных видов (аффинные преобразования, инверсия, фрактальные преобразования).

Цель исследования: используя возможности языка Wolfram Mathematica, создать программы, реализующие наглядное представление геометрических преобразований. В частности, большое внимание уделяется:

1. построению фракталов различного типа;
2. созданию демонстраций - интерактивных вычислимых документов (CDF).

Формат вычисляемых документов (Computable Document Format, или CDF) – это электронный формат документов, созданный с целью облегчения создания динамически сгенерированного интерактивного контента. Файлы CDF создаются в системе Mathematica [2, с.18].

Задачи:

1. Изучение литературы по темам: аффинные преобразования (поворот, параллельный перенос, преобразование подобия, масштабирование), инверсия, геометрические преобразования на плоскости в языке Wolfram;
2. Программирование: визуализация простых видов аффинных преобразований плоскости и их композиций;
3. Программирование: визуализация инверсии;
4. Изучение литературы по темам: классические фракталы, размерность Хаусдорфа, функции языка Wolfram для построения фракталов;
5. Программирование фракталов на плоскости: салфетка и ковер Серпинского; кривая Коха и другие известные фракталы;

6. Изучение литературы по теме «Детерминированные системы итерированных функций (ДСИМ)»;

7. Программирование фракталов с помощью ДСИМ;

8. Изучение алгоритма рандомизированных систем итерированных функций и построение с его помощью хаотических фракталов.

Методы: математические, большей частью геометрические; использование языка Wolfram для программирования в системе Mathematica.

Wolfram Mathematica - это современная техническая вычислительная система, охватывающая технические, научные, инженерные, математические и вычислительные области. Система была задумана Стивеном Вольфрамом и в 1988г. была разработана первая версия в компании Wolfram Research (Шампейн, штат Иллинойс, США).

Mathematica представляет собой модульную систему программного обеспечения, в которой ядро, фактически выполняющее вычисления, отделено от интерфейса, отвечающего за взаимодействие с пользователем.

Язык Wolfram является интерпретирующим языком. Команды выполняются в реальном времени без необходимости компилировать их в программы. Напечатав выражение, вы нажимаете комбинацию Shift+Enter, Mathematica присваивает введенному выражению номер и после этого ядро вычисляет значение выражения. Результат вычислений располагается под вводом и так же нумеруется.

Основные возможности [2, с.16]:

- Библиотека традиционных и специальных математических функций;
- Поддержка комплексных чисел, рациональных и вещественных чисел произвольной точности, интервальной арифметики и символьных вычислений;
- Поддержка 2D и 3D-данных и функций визуализации и анимации;
- Решение систем уравнений, диофантовых уравнений, обыкновенных дифференциальных уравнений, уравнений в частных производных, дифференциальных алгебраических уравнений, дифференциальных уравнений с задержкой аргумента, стохастических дифференциальных уравнений и рекуррентных соотношений;
- Библиотеки многомерной статистики, включая интерполяцию и аппроксимацию данных, проверку гипотез и расчёты вероятности и математического ожидания для более 100 различных распределений;
- Расчеты и моделирование случайных процессов и очередей;
- Локальная и глобальная оптимизации с ограничениями и без них;

– Язык Wolfram поддерживает различные парадигмы: процедурное, функциональное и объектно-ориентированное программирование.

Пример:

Simplify[$x^2 + 2x + 1$]
 $(1 + x)^2$

Большинство различных природных конструкций изначально описывалось примитивными геометрическими фигурами: прямыми, многоугольниками, окружностям, многогранниками и сферами. Но не для всех конструкций подходили эти фигуры. Для характеристики более сложных объектов, таких как очертание береговых линий, разряд молнии в воздухе, поле скоростей в турбулентном потоке жидкости, контуры дерева, кровеносно-сосудистая система человека, этот набор становится плохо применимым. Поэтому в математике возникает новое понятие - фрактал, которое изменило традиционное представление о геометрии. Оно было введено французским математиком польского происхождения Бенуа Мандельбротом в 1975 года, хотя это только дата возникновения понятия. Сами же конструкции возникли несколько десятков лет назад.

В данной работе большое внимание уделяется изучению и построению фракталов различных видов: салфетка и ковер Серпинского, кривая Коха, дракон Хартера-Хейтуэя, лист папоротника, кристалл и др.

Структура работы:

Глава 1 – теория по геометрическим преобразованиям; построение каждого преобразования в Wolfram Mathematica в отдельности и в композиции.

Глава 2 – история появления и развития фракталов; вычисление фрактальной размерности; знакомство с известными фракталами и их построением.

Глава 3 – изучение системы итерируемых функций, а именно подходы: детерминированный и метод случайных итераций; на основе изученного материала рассматривается последовательность построения листа папоротника.

Глава 4 – построение фракталов на основе методов СИФ и их визуальное представление в Wolfram Mathematica.

Описание использованных источников:

[4]-[6] – изучение геометрических преобразований;

[2], [8] – изучение языка Wolfram Mathematica, его возможности, ознакомление с готовыми примерами;

[1], [7], [9] – изучение истории развития и появления первых фракталов, разбор построения известных фракталов, изучение метода систем итерируемых функций и его подходов.

1 Геометрические преобразования

Определение 1. Ортогональным отображением плоскости π на плоскость π' называется такое отображение, при котором каждый отрезок плоскости π переходит в равный ему отрезок плоскости π' [3, с.19].

Ортогональное отображение плоскости π на плоскость π' взаимнооднозначно:

- двум различным точкам M_1 и M_2 плоскости π всегда соответствуют две различные же точки M'_1 и M'_2 плоскости π' ;
- каждая точка M' плоскости π' имеет прообраз M в плоскости π .

Определение 2. Ортогональное отображение плоскости на себя называется ортогональным преобразованием плоскости (или движением) [3, с.19].

1.1 Параллельный перенос

Определение 3. Пусть дан вектор a , лежащий в плоскости π . Поставим в соответствие каждой точке M плоскости π точку M' той же плоскости, такую, что $\overrightarrow{MM'} = a$ (Рисунок 1.1). Это соответствие есть преобразование плоскости, называемое переносом. Таким образом, при переносе каждая точка плоскости перемещается в одном и том же направлении на одно и то же расстояние [3, с.32].

Перенос есть ортогональное преобразование. В самом деле: пусть M_1 и M_2 – две произвольные точки плоскости, а M'_1 и M'_2 – их образы при указанном переносе (Рисунок 1.1). По определению переноса $\overrightarrow{M_1M'_1} = \overrightarrow{M_2M'_2} = a$. Прибавляя к обеим частям вектор $\overrightarrow{M'_1M_2}$, получим $\overrightarrow{M_1M'_1} + \overrightarrow{M'_1M_2} = \overrightarrow{M'_1M_2} + \overrightarrow{M_2M'_2}$, или $\overrightarrow{M_1M_2} = \overrightarrow{M'_1M'_2}$, значит отрезки M_1M_2 и $M'_1M'_2$ равны.

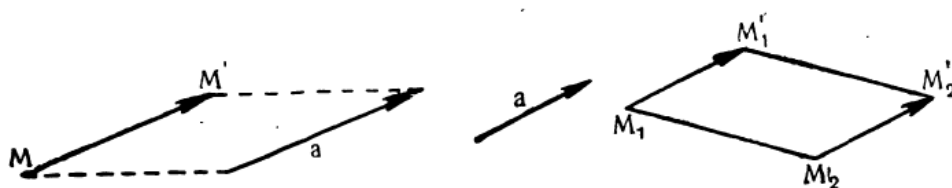


Рисунок 1.1 – Перенос. Слева - перенос точки M на заданный вектор a , справа – образы M'_1 и M'_2 произвольных точек M_1 и M_2 , соответственно, при переносе на вектор a [3, с. 34]

Свойства:

1. При параллельном переносе любая прямая переходит в параллельную прямую;
2. Композиция параллельных переносов на векторы $\vec{m}$ и $\vec{n}$ есть параллельный перенос на вектор $\vec{m} + \vec{n}$;

3. Преобразование, обратное к параллельному переносу на вектор $\vec{n}$ есть параллельный перенос на вектор $-\vec{n}$.

Реализация параллельного переноса с помощью Wolfram Mathematica (ПРИЛОЖЕНИЕ А)

Зададим по точкам ломанную:

```
ld = Line[{{0,0},{1,0.25},{2,0.5},{3,0},{3.5,0.5},{4,1},{4,1.5},{4,2},{3,2},{2,2},{1,2},
{0.5,1},{0,0}}];
```

Сделаем графическое представление этой ломанной (Рисунок 1.2):

```
Graphics[{Blue,ld},PlotRange→8,Axes→True,Frame→True,
GridLines→{Table[k,{k,-8,8,2}],Table[k,{k,-8,8,2}]},
GridLinesStyle→Directive[Orange,Dashed]]
```

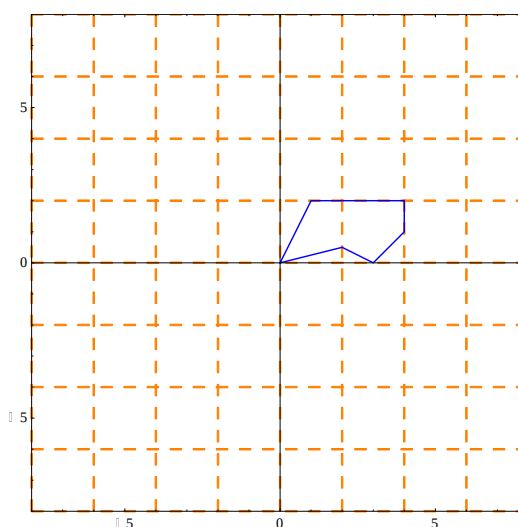


Рисунок 1.2 – Изображение ломанной ld

Графическое изображение переноса синей ломанной на вектор $\{-2, -1.5\}$.

Результатом переноса является красная ломанная (Рисунок 1.3).

```
Graphics[{{Blue,ld},{GeometricTransformation[{Red,ld},
TranslationTransform[{-2,-1.5}]}]},PlotRange→8,Axes→True,Frame→True,
GridLines→{Table[k,{k,-8,8,2}],Table[k,{k,-8,8,2}]},
GridLinesStyle→Directive[Orange,Dashed]]
```

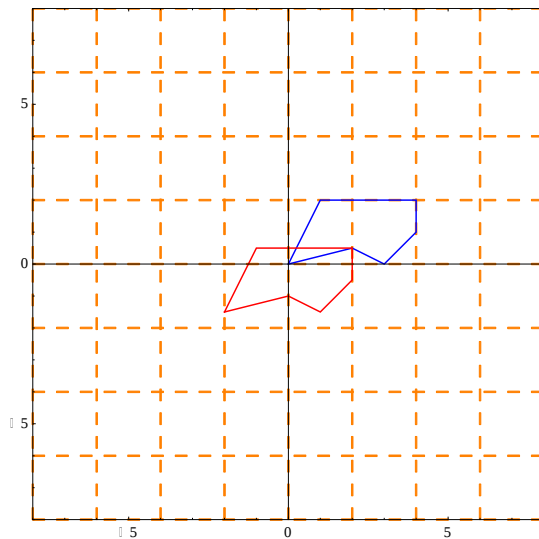


Рисунок 1.3 – Перенос ломанной ld на вектор $\{-2, -1.5\}$

Графическое представление движущегося переноса, путём изменения координат вектора переноса (Рисунок 1.4).

```
Manipulate[Graphics[{{Blue, ld}, {GeometricTransformation[{Red, ld},
TranslationTransform[{x, y}]}]}, PlotRange → 8, Axes → True, Frame → True,
GridLines → {Table[k, {k, -8, 8, 2}], Table[k, {k, -8, 8, 2}]},
GridLinesStyle → Directive[Orange, Dashed]}, {x, -4, 2, .2}, {y, -4, 3, .2},
SaveDefinitions → True]
```

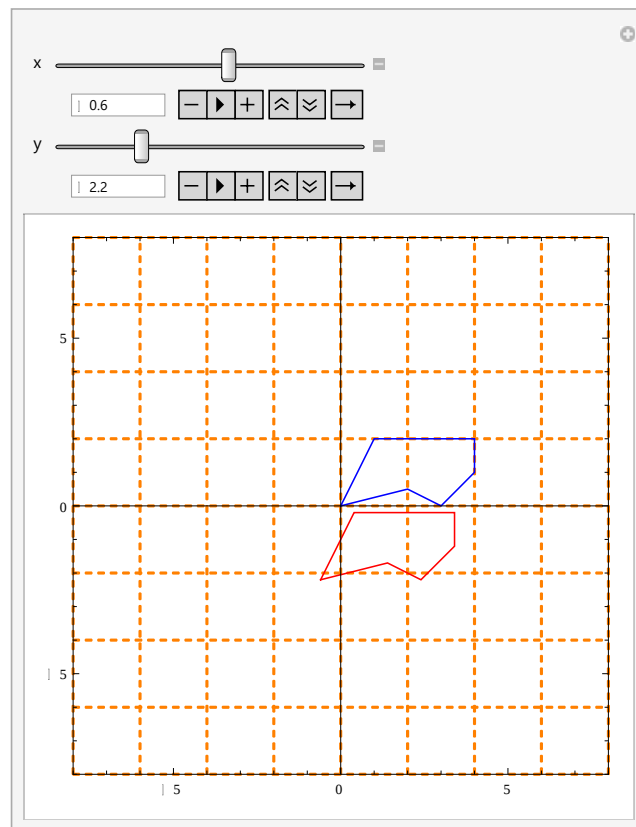


Рисунок 1.4 – Перенос на вектор $\{x, y\}$. Координаты x, y выбираются за счет движка

Графическое представление переноса с помощью локатора (Рисунок 1.5).

```
Manipulate[Graphics[{{Blue, ld}, {GeometricTransformation[{Red, ld},  
TranslationTransform[v]]}}, PlotRange → 8, Axes → True, Frame → True,  
GridLines → {Table[k, {k, -8, 8, 2}], Table[k, {k, -8, 8, 2}]},  
GridLinesStyle → Directive[Orange, Dashed]], {v, {-4, -4}, {2, 3}},  
ControlType → Slider2D, ControlPlacement → Left, SaveDefinitions → True]
```

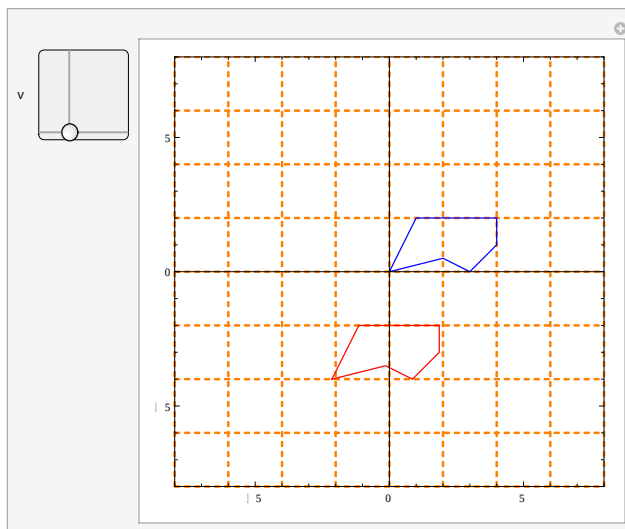


Рисунок 1.5 – Перенос на вектор $\{x, y\}$. Координаты x, y выбираются за счет локатора

1.2 Поворот

Определение 4. Поворотом вокруг точки O на угол α называется преобразование плоскости, переводящее каждую точку A в такую точку A' , что $OA = OA'$ и угол между лучами OA и OA' (т.е. угол, отсчитываемый против часовой стрелки от луча OA к лучу OA') равен α (Рисунок 1.6) [4, с.6].

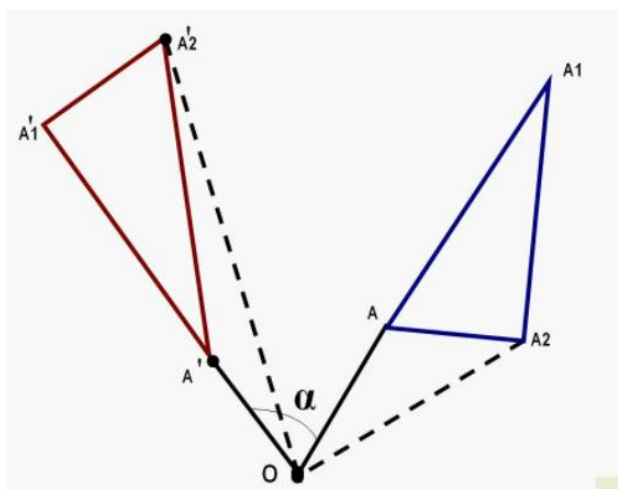


Рисунок 1.6 – Графическое представление определения поворота

Утверждение [4, с.7].

Поворот является движением.

Доказательство:

Пусть точки A, B переходят соответственно в точки A', B' . Так как $\angle AOA' = \angle BOB'$, мы заключаем, что $\angle AOB = \angle A'OB'$. Поэтому треугольники AOB и $A'OB'$ равны по двум сторонам и углу между ними, и, значит, $AB = A'B'$. Очевидно, что поворот на угол 2π является тождественным преобразованием. Поэтому можно считать, что $0 \leq \varphi < 2\pi$.

Свойства [4, с.7]:

1. Если прямая a при повороте на угол φ переходит в прямую a' , то один из углов между a и a' равен φ при $0 \leq \varphi < \pi$ и $\varphi - \pi$ при $\pi \leq \varphi < 2\pi$;

Доказательство:

Если прямая a проходит через центр поворота O , утверждение очевидно. В противном случае опустим на a перпендикуляр OA и построим образ A' точки A . Так как движения сохраняют величины углов, образом прямой a будет прямая a' , проходящая через A' и перпендикулярная OA' . Из свойств углов с перпендикулярными сторонами вытекает искомое свойство (Рисунок 1.7).



Рисунок 1.7 – Графическое представление свойства 1 [4, с.7]

2. Композиция поворотов вокруг точки O на углы α и β есть поворот вокруг O на угол $\alpha + \beta$ (или $\alpha + \beta - 2\pi$);

3. Преобразование, обратное к повороту вокруг точки O на угол φ , есть поворот вокруг O на угол $2\pi - \varphi$.

Реализация поворота с помощью Wolfram Mathematica (ПРИЛОЖЕНИЕ А)

Пример поворота будем производить, используя ломанную ld . Для хранения координат точек ломанной вводится обозначение $ld[[1]]$.

$rPointO[\theta, p_] := RotationMatrix[\theta].p$ – функция, отвечающая за вращение точки $p = \{x, y\}$ вокруг начала координат против часовой стрелки на угол θ .

Графическое изображение поворота на угол $\frac{\pi}{2}$ (Рисунок 1.8).

```
Graphics[Line[rPointO[ $\frac{\pi}{2}$ , #]&/@ld[[1]]], PlotRange  $\rightarrow$  8, Axes  $\rightarrow$  True,
```

```
Frame → True, GridLines → {Table[k, {k, -8, 8, 2}], Table[k, {k, -8, 8, 2}]},
GridLinesStyle → Directive[Orange, Dashed]]
```

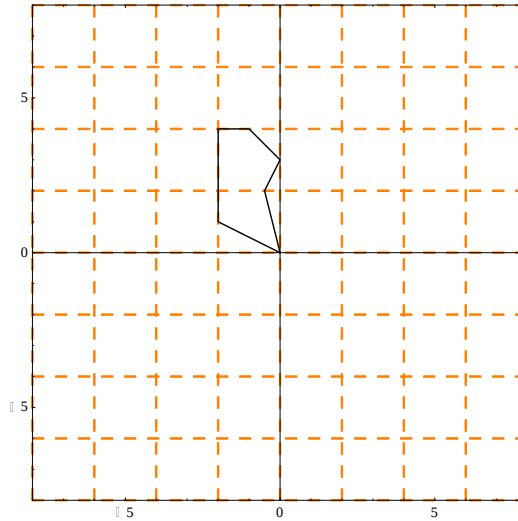


Рисунок 1.8 – Поворот ломанной ld на угол $\frac{\pi}{2}$

Графическое изображение движения фигуры при повороте на угол a , который меняется в диапазоне $\{a, 0, 2\pi, \frac{2\pi}{20}\}$. Ломанная ld , окрашенная в синий цвет, является исходной. Результатом преобразования является ломанная красного цвета (Рисунок 1.9).

```
Manipulate[Graphics[{Blue, ld}, {GeometricTransformation[{Red, ld},
RotationMatrix[a]]}], PlotRange → 8, Axes → True, Frame → True,
GridLines → {Table[k, {k, -8, 8, 2}], Table[k, {k, -8, 8, 2}]},
GridLinesStyle → Directive[Orange, Dashed]], {a, 0, 2 π,  $\frac{2\pi}{20}$ },
SaveDefinitions → True]
```

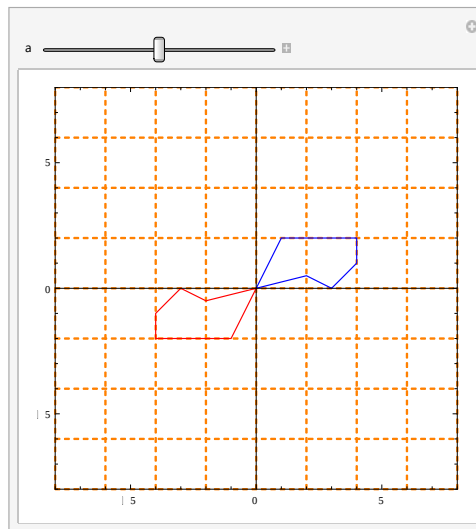


Рисунок 1.9 – Поворот ломанной ld на угол a , который меняется в диапазоне $\{a, 0, 2\pi, \frac{2\pi}{20}\}$

1.3 Отображение подобия

Определение 5. Отображение α плоскости π на плоскость π' называется отображением подобия с коэффициентом $k > 0$, если оно обладает следующим свойством: если A и B – две любые точки плоскости π , а A' и B' – их образы при отображении α , то $A'B' = kAB$. Если $k = 1$, то отображение подобия является движением [3, с.69].

Отображение α^{-1} плоскости π' на плоскость π , обратное отображению α подобия с коэффициентом k плоскости π на плоскость π' , есть отображение подобия плоскости π' на плоскость π с коэффициентом $\frac{1}{k}$.

Определение 6. Отображение подобия плоскости π на себя называется преобразованием подобия, или подобным преобразованием [3, с.69].

Произведение двух любых подобных преобразований (с коэффициентами k_1 и k_2) есть подобное преобразование (с коэффициентом k_1k_2); единичное преобразование плоскости можно рассматривать как подобное преобразование (с коэффициентом 1).

Свойства подобных преобразований [3, с.70]:

1. При отображении подобия три точки A, B и C , лежащие на одной прямой, переходят в три точки A', B', C' , также лежащие на одной прямой с сохранением порядка;

2. При отображении подобия отношение любых двух отрезков равно отношению образов этих отрезков;

3. В отображении подобия параллельные прямые переходят в параллельные (в силу взаимной однозначности отображения);

4. При подобных отображениях сохраняются углы. Из этого свойства, в частности, следует, что при подобном отображении взаимно-перпендикулярные прямые переходят во взаимно-перпендикулярные прямые;

5. Если на плоскости π заданы три точки A, B и C , не лежащие на одной прямой, а A', B', C' – три точки плоскости π' , такие, что $A'B' = kAB$, $B'C' = kBC$, $C'A' = kCA$, то существует и притом только одно отображение подобия плоскости π на плоскость π' , при котором точки A, B и C переходят соответственно в точки A', B' и C' ;

6. Пусть M – произвольная точка плоскости π , а M^* – ее образ на плоскости π' при отображении β . Рассмотрим преобразование γ плоскости π' , которое точке M^* плоскости π' ставит в соответствие точку M' , лежащую на луче $A'M^*$ и такую, что $A'M' = kA'M^*$. Поставим теперь в соответствие точке M плоскости π точку M' плоскости π' . Это

соответствие и будет отображением подобия, которое точкам А, В и С ставит в соответствие точки A' , B' и C' .

Реализация подобия с помощью Wolfram Mathematica (ПРИЛОЖЕНИЕ А)

Зададим ломанную.

```
fig = Line[{{1,1},{3,2},{4,6},{1,1}}];
```

Построим и изобразим заданную ломанную и ей подобную (Рисунок 1.10).

```
Graphics[{Blue, fig, Green, GeometricTransformation[fig, ScalingTransform[{2,2}]]},  
Frame → True, Axes → True, GridLines → Automatic, PlotRange → {{-1,14},{-1,14}}]
```

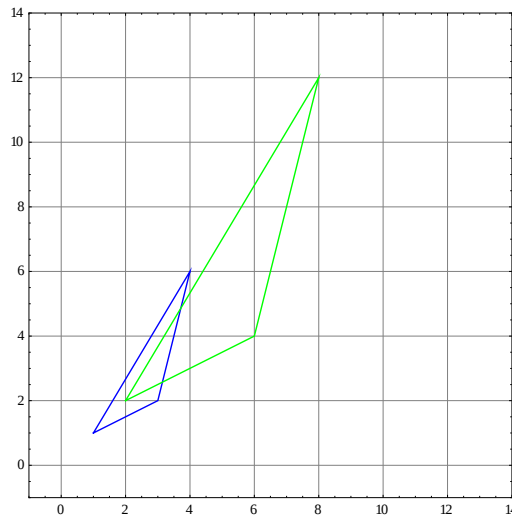


Рисунок 1.10 – Подобие двух треугольников. Синим цветом изображен исходный треугольник, а зеленым – подобный ему с $k = 2$

1.4 Масштабирование

Масштабирование объекта можно реализовать путём умножения координат всех его точек на некоторое число.

Пусть имеются точки с координатами (x_1, y_1) и (x_2, y_2) , над которыми выполняется такое преобразование. Результатом будут новые точки с координатами $(S_x x_1, S_y y_1)$ и $(S_x x_2, S_y y_2)$. Если $S_x = S_y = S_0$, то несложно доказать, что обе точки переместятся вдоль прямых, проходящих через саму точку и начало координат, т.е. в направлении своего же радиус-вектора (Рисунок 1.11).

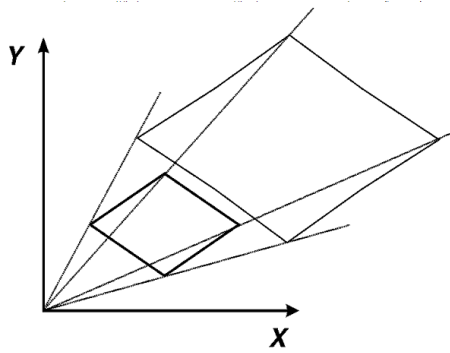


Рисунок 1.11 – Изменение масштаба вдоль горизонтальной и вертикальной осей на одно и то же число [5]

При этом расстояние между новыми точками будет в S_0 раз отличаться от прежнего, но углы между отрезками сохранятся. Ясно, что если коэффициент масштабирования S_0 больше единицы, соответствующий отрезок растягивается, а если меньше, то сжимается. Кроме того, при таком преобразовании объект смещается.

В случае $S_x \neq S_y$, то растяжения в горизонтальном и вертикальном направлениях будут различными. Углы между отрезками также не сохранятся (Рисунок 1.12).

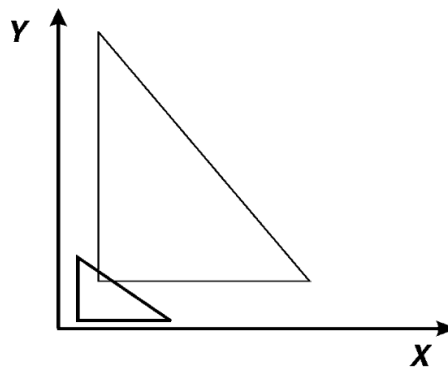


Рисунок 1.12 – Изменение масштаба в горизонтальном и вертикальном направлениях различны [5]

Реализация масштабирования с помощью Wolfram Mathematica (ПРИЛОЖЕНИЕ А)

Зададим ломанную:

```
| line4 = Line[{{1,0},{2,1},{3,0.5},{4,2}}];
```

Зададим фигуру из ломанной, диска и окружности:

```
| fig = {line4, Circle[{1,0},1/3], Disk[{4,2},1/4]};
```

Увеличение фигуры в t раз по направлению вектора v (Рисунок 1.13).

```
| Manipulate[Graphics[{fig, {Blue, GeometricTransformation[fig,
ScalingTransformation[t, v]]}, {Red, Line[{0,0}, v], Point[{0,0}]}],
PlotRange -> {{-10,10}, {-8,6}}, Axes -> True, Frame -> True, GridLines -> Automatic,
GridLinesStyle -> Directive[Orange, Dashed], {v, {-1, -1}, {1,1}}, {t, -2,2},
SaveDefinitions -> True]
```

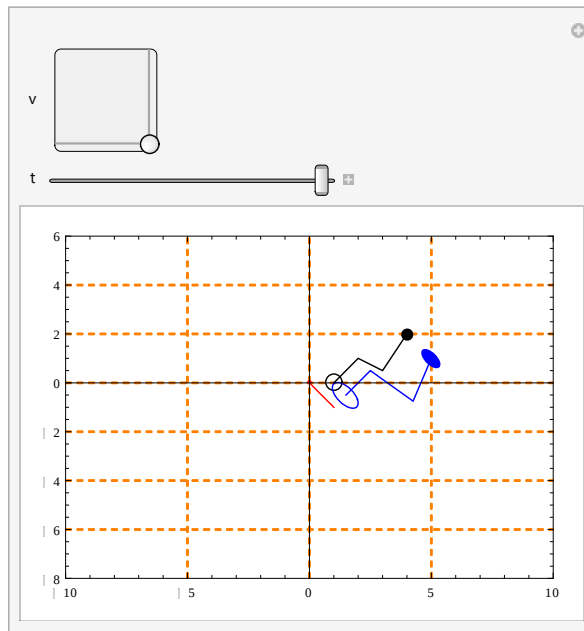


Рисунок 1.13 - Увеличение фигуры в t раз по направлению вектора v

Увеличение в t раз по направлению вектора v с центром в точке c (Рисунок 1.14).

```
Manipulate[Graphics[{fig, {Blue, GeometricTransformation[fig,
ScalingTransform[t, v, c]]}, {Red, Line[{c, v}], Point[c]}},
PlotRange  $\rightarrow$  {{-10, 10}, {-11, 8}}, Axes  $\rightarrow$  True, Frame  $\rightarrow$  True,
GridLines  $\rightarrow$  Automatic, GridLinesStyle  $\rightarrow$  Directive[Orange, Dashed]],
{c, {-1, -1}, {1, 1}}, {t, -2, 2}, {v, {-1, -1}, {1, 1}}, SaveDefinitions  $\rightarrow$  True]
```

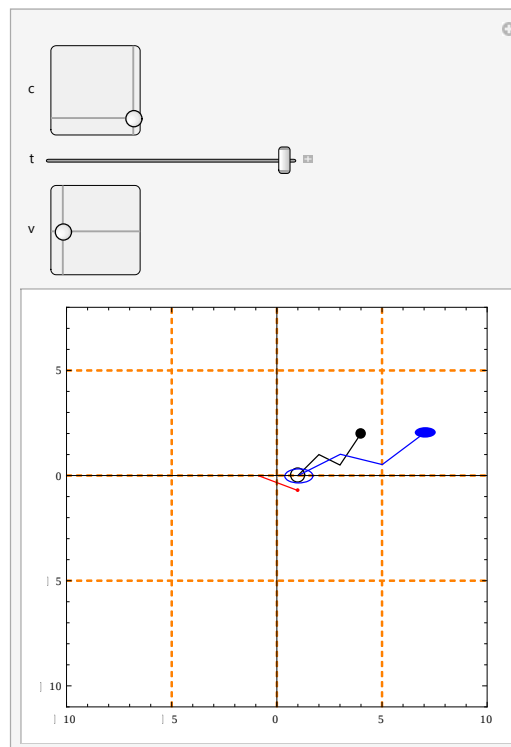


Рисунок 1.14 - Увеличение в t раз по направлению вектора v с центром в точке c

1.5 Композиция преобразований

Композиция сдвига и поворота.

Для реализации композиции геометрических преобразований будем использовать ломанную ld .

Изображение движущейся композиции сдвига и поворота. Исходная ломанная изображена синим цветом. Красным цветом изображена ломанная, являющаяся результатом композиции. Перемещение по вектору $\{v, \{-4, -4\}, \{3, 3\}\}$ с использованием локатора. Поворот на угол $\{a, 0, 2\pi, \frac{2\pi}{30}\}$ (Рисунок 1.15).

```
Manipulate[Graphics[{Blue, ld}, {GeometricTransformation[{Red, ld},
Composition[TranslationTransform[v], RotationTransform[a]]}],
PlotRange -> 8, Axes -> True, Frame -> true, GridLines -> {Table[k, {k, -8, 8, 2}],
Table[k, {k, -8, 8, 2}]}, GridLinesStyle -> Directive[Orange, Dashed]],
{v, {-4, -4}, {2, 3}}, {a, 0, N[2 π], N[ $\frac{2\pi}{20}$ ]}, SaveDefinitions -> True]
```

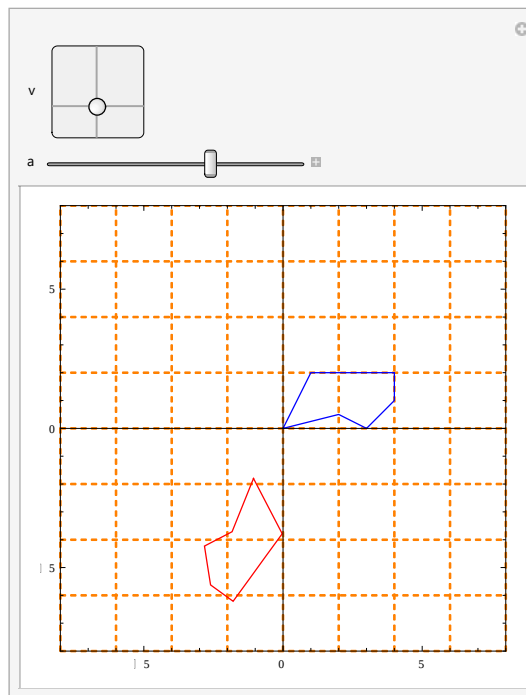


Рисунок 1.15 – Композиция сдвига и поворота

Перенос и масштабирование.

Зададим фигуру:

```
bird = Line[{{3,4}, {3,6}, {2,6}, {4,8}, {4,10}, {6,10}, {6,9}, {8,8}, {6,8}, {6,6}, {4.2,6},
{4.2,4}, {4,4}, {4,6}, {3.2,6}, {3.2,4}, {3,4}}];
```

Изображение движущейся композиции (Рисунок 1.16) переноса и масштабирования.

```

Manipulate[Graphics[{Blue, bird}, {Green, GeometricTransformation[bird,
Composition[ScalingTransform[t, {0,1}], TranslationTransform[{x,y}]]}],
PlotRange → 15, Axes → True, Frame → True,
GridLines → {Table[k, {k, -15, 15, 2}], Table[k, {k, -15, 15, 2}]},
GridLinesStyle → Directive[Orange, Dashed]], {x, -15, 2, .2}, {y, -15, 3, .2},
{t, -2, 2}, SaveDefinitions → True]

```

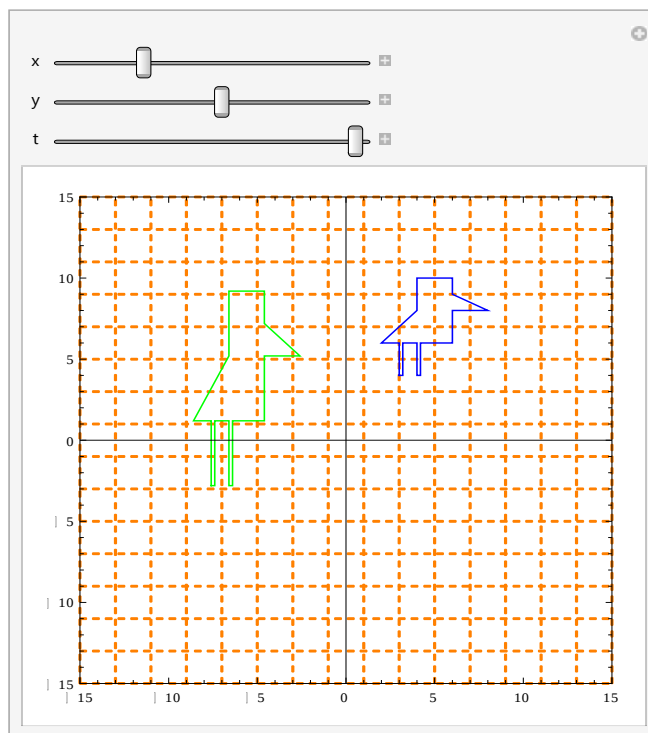


Рисунок 1.16 – Композиция переноса и масштабирования

1.6 Аффинные преобразования

Определение 7. Взаимнооднозначное отображение плоскости π на плоскость π' называется аффинным, если при этом отображении всякие три точки, расположенные на одной прямой, переходят в три точки, также расположенные на одной прямой [3, с.87].

Замечание: Ю.М. Смирнов заметил, что требование взаимной однозначности в определении аффинного отображения плоскости π на плоскость π' является излишним. Это вытекает из того, что плоскость π отображается на всю плоскость π' , и из сохранения прямолинейного расположения трех точек.

Определение 8. [6, с.130] Преобразование плоскости называется аффинным, если оно задается следующей формулой:

$$F: \begin{pmatrix} x \\ y \end{pmatrix} \rightarrow \begin{pmatrix} ax + by + e \\ cx + dy + f \end{pmatrix} \quad (1.1)$$

Аффинное преобразование называется неособым, если определить его матрицы отличен от нуля, т.е. если $a_1b_2 - a_2b_1 \neq 0$. Если определить положителен, то неособое аффинное преобразование называется собственным. Иначе оно называется неособственным. Примерами собственных движений являются поворот и параллельный перенос, а неособственных движений - осевая и скользящая симметрии.

Представим преобразование плоскости в матричной форме

$$\begin{pmatrix} x_{n+1} \\ y_{n+1} \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x_n \\ y_n \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix} \quad (1.2)$$

Аффинное преобразование на плоскости задается шестью независимыми действительными числами. Числа e и f отвечают за перенос, а числа a, b, c, d задают произвольное линейное преобразование при неизменном положении начала координат $(0,0)$. Рассмотрим за какое преобразование отвечает каждое число (Рисунок 1.17).

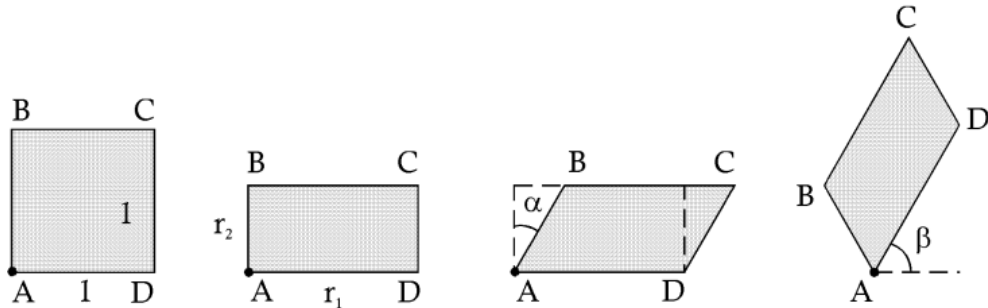


Рисунок 1.17 - Аффинное преобразование единичного квадрата $ABCD$ [1, с.50]

Если при аффинном преобразовании не меняется направление обхода произвольного контура, то его можно представить как произведение 3 преобразований:

$$\hat{A} \equiv \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \hat{T}_3 \hat{T}_2 \hat{T}_1 \quad (1.3)$$

Первое переводит единичный квадрат в произвольный прямоугольник со сторонами r_1 и r_2 и описывается диагональной матрицей:

$$\hat{T}_1 = \begin{pmatrix} r_1 & 0 \\ 0 & r_2 \end{pmatrix} \quad (1.4)$$

Это соответствует простому масштабированию двух координатных осей.

Второе преобразование описывает скос этого прямоугольника в горизонтальном направлении на угол α и превращение его в параллелограмм со сторонами $r_1, r_2 / \cos \alpha$ и углом $\frac{\pi}{2} - \alpha$ между ними:

$$\hat{T}_2 = \begin{pmatrix} 1 & tg \alpha \\ 0 & 1 \end{pmatrix} \quad (1.5)$$

И наконец, третье преобразование есть простой поворот получившегося параллелограмма на угол β вокруг начала координат:

$$\widehat{T}_3 = \begin{pmatrix} \cos \beta & -\sin \beta \\ \sin \beta & \cos \beta \end{pmatrix} \quad (1.6)$$

Поскольку определитель произведения матриц равен произведению определителей, то нетрудно видеть, что $\det \widehat{A} = r_1 r_2$ и положителен. Это является признаком того, что преобразование не меняет направления обхода контура. Действительно, во всех четырёх случаях направление обхода параллелограмма $ABCD$ все время происходит по часовой стрелке.

Если же определитель аффинного преобразования отрицателен, то это значит, что преобразование включает в себя еще и операцию отражения в вертикальной ($\widehat{T}_v$) или в горизонтальной плоскости ($\widehat{T}_h$):

$$\widehat{T}_v = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}, \widehat{T}_h = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad (1.7)$$

В этом случае, как видно из рисунка 1.18, направление обхода контура $ABCD$ меняется на противоположное.

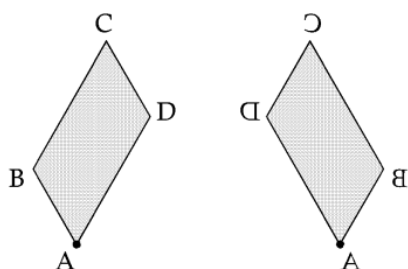


Рисунок 1.18 - Отражение в вертикальной плоскости [1, с.51]

Таким образом, в зависимости от знаков $\det \widehat{A}$ четыре коэффициента аффинного преобразования a, b, c, d можно всегда выразить через два масштабных фактора r_1, r_2 и два угла α и β . Они характеризуют форму, размеры, ориентацию и направление обхода получившегося параллелограмма. Фактически аффинное преобразование описывает переход от прямоугольной декартовой к произвольной косоугольной (правой или левой) системе координат.

Свойства аффинных отображений:

Теорема 1. При аффинном отображении α плоскости π на плоскость π' образом всякой прямой l плоскости π служит прямая l' плоскости π' .

Теорема 2. При аффинном отображении α плоскости π на плоскость π' параллельные прямые плоскости π переходят в параллельные прямые плоскости π' . Пересекающиеся прямые плоскости π переходят в пересекающиеся прямые плоскости π' . При этом точка пресечения прямых плоскости π переходит в точку пересечения образов этих прямых.

Теорема 3. При аффинном отображении середина отрезка, соединяющего точки A и B , переходит в середину отрезка, соединяющего их образы A' и B' .

Следствие. Если точки $D_1, D_2, \dots, D_{n-1}$ делят отрезок AB на n равных частей, то их образы $D'_1, D'_2, \dots, D'_{n-1}$ при аффинном отображении делят отрезок $A'B'$ (A' и B' - образы точек A и B) также на n равных частей.

Лемма Дарбу. При аффинном отображении всякая точка C , лежащая на отрезке AB (т.е. между точками A и B), переходит в некоторую точку C' , лежащую на отрезке $A'B'$ (т.е. между образами A' и B' точек A и B).

Следствие 1. При аффинном отображении образом отрезка является отрезок.

Следствие 2. При аффинном отображении внутренние точки треугольника (или параллелограмма) переходят во внутренние точки образа этого треугольника (соответственно параллелограмма).

Доказательство свойств приводится в [3, с.100].

1.7 Инверсия

Определение 9. Степенью точки O относительно окружности C называется число $\sigma = \overrightarrow{OA} \cdot \overrightarrow{OB}$, где A и B – точки пересечения окружности C с любой прямой, проходящей через точку O (и пересекающей окружность C) [3, с. 218].

В зависимости от расположения точки меняется и знак степени точки. Если точка O лежит вне окружности C (тогда векторы $\overrightarrow{OA}$ и $\overrightarrow{OB}$ имеют одинаковое направление), то степень точки O относительно окружности C положительна. Если точка O лежит внутри окружности C (в этом случае векторы $\overrightarrow{OA}$ и $\overrightarrow{OB}$ имеют противоположное направление), то степень точки O относительно окружности C отрицательна. Если точка O лежит на окружности C , то степень точки O относительно окружности C равна нулю.

Заметим, что степень точки O относительно окружности C в том случае, когда точка O лежит вне окружности C , равна квадрату длины отрезка касательной, проведённой из точки O к окружности C (произведение секущей на внешнюю её часть равно квадрату касательной).

Так же степень точки O относительно окружности C может быть определена соотношением $\sigma = d^2 - R^2$, где d – расстояние от точки O до центра окружности C , а R – радиус этой окружности. Покажем достоверность данного утверждения.

Если точка O лежит вне окружности C , то её степень относительно окружности C равна квадрату длины отрезка касательной, проведённой из точки O к окружности C ; квадрат же указанного отрезка касательной равен $d^2 - R^2$.

Если точка O лежит внутри окружности C , то проведём через неё и через центр окружности C прямую; пусть эта прямая пересечёт окружность C в точках A и B . Тогда, обозначая через S центр окружности C , будем иметь

$$\sigma = \overrightarrow{OA} \cdot \overrightarrow{OB} = (\overrightarrow{OS} + \overrightarrow{SA})(\overrightarrow{OS} + \overrightarrow{SB}) = (\overrightarrow{OS} - \overrightarrow{AS})(\overrightarrow{OS} + \overrightarrow{SB}) = (\overrightarrow{OS} - \overrightarrow{AS})(\overrightarrow{OS} + \overrightarrow{AS}) = OS^2 - AS^2 = d^2 - R^2$$

(этот вывод верен и для того случая, когда точка O лежит вне окружности C).

Наконец, если точка O лежит на окружности C , то $d=R$, значит $d^2 - R^2 = 0$, но в этом случае и $\sigma = 0$.

Определение 10. Вернёмся от вводных понятий к определению инверсии. Зафиксируем на плоскости точку O и фиксируем число k ($k \neq 0$). Поставим в соответствие каждой точке M плоскости (отличной от O) точку M' , лежащую на прямой OM и такую, что $\overrightarrow{OM} \cdot \overrightarrow{OM'} = k$. Это преобразование называется инверсией. Точка O называется полюсом, или центром, инверсии, k – степенью инверсии. Если $k > 0$, инверсия называется положительной, если $k < 0$, – отрицательной. Инверсию I с полюсом O и степенью k будем обозначать так: (O, k) [3, с.219].

Инверсия есть инволюционное преобразование: квадрат инверсии есть тождественное преобразование: $I^2 = E$. Другими словами, если в инверсии I точке M соответствует точка M' , то точке M' соответствует точка M .

Если инверсия (O, k) положительная ($k > 0$), то все точки окружности K радиуса $\sqrt{k}$ с центром O неподвижны, так как если M – любая точка этой окружности, то $\overrightarrow{OM} \cdot \overrightarrow{OM} = (\sqrt{k})^2 = k$.

Образ M' точки M в положительной инверсии (O, k) строится так:

1. если точка M лежит вне окружности с центром O и радиусом $\sqrt{k}$, то проводим из точки M касательную MT к этой окружности (T – точка прикосновения); проекция M' точка T на OM и есть образ точки M ;

2. если точка M лежит внутри окружности с центром O и радиусом $\sqrt{k}$, то через точку M проводим прямую, перпендикулярную OM ; пусть T – одна из точек пересечения этого перпендикуляра с окружностью K ; тогда касательная к окружности K в точке T пересечет OM в точке M' , являющейся образом M в инверсии (O, k) .

В обоих случаях: $\overrightarrow{OM} \cdot \overrightarrow{OM'} = (\sqrt{k})^2 = k$. Из этого можно сделать следующий вывод: все точки, лежащие вне окружности с центром O и радиусом $\sqrt{k}$ перейдут при инверсии (O, k) внутрь этой окружности, а все точки, лежащие внутри указанной

окружности, перейдут во внешние ее точки (точки самой окружности будут неподвижными).

Свойства инверсии:

Теорема 1: При инверсии всякие четыре точки, лежащие на окружности, не проходящей через полюс инверсии, переходят в четыре точки, также лежащие на окружности, не проходящей через полюс инверсии.

Следствие 1: При инверсии всякая окружность S , не проходящая через центр инверсии, отображается взаимнооднозначно на некоторую окружность S' , также не проходящую через центр инверсии.

Следствие 2: Если R – радиус окружности S , R' – радиус окружности S' при инверсии (O, k) , а σ – степень полюса O инверсии относительно окружности S , то $\frac{R'}{R} = \frac{k}{\sigma}$.

Теорема 2: При инверсии всякие четыре точки, лежащие на окружности, проходящей через полюс инверсии, переходят в четыре точки, лежащие на одной прямой, не проходящей через полюс инверсии.

Следствие 1: Всякая окружность S , проходящая через полюс инверсии, отображается взаимнооднозначно на прямую, не проходящую через полюс инверсии.

Следствие 2: При инверсии любая прямая, не проходящая через полюс инверсии, отображается взаимнооднозначно на некоторую окружность, проходящую через полюс инверсии.

Теорема 3: При инверсии любые четыре точки, лежащие на одной прямой, проходящей через полюс инверсии, переходят в четыре точки, лежащие на той же прямой.

Теорема 4: Угол между двумя пересекающимися окружностями, угол между окружностью и пересекающей ее прямой и угол между двумя пересекающимися прямыми сохраняется при преобразовании инверсии.

Замечание к теореме 4: Углом между двумя пересекающимися окружностями называется угол между касательными к этим окружностям в их общей точке. Углом между окружностью и пересекающей ее прямой называется угол между этой прямой и касательной к окружности в точке их пересечения.

С доказательством свойств можно ознакомиться в [3, с.220].

Реализация инверсии с помощью Wolfram Mathematica (ПРИЛОЖЕНИЕ А)

Зададим несколько функций, которые будут использоваться для создания инверсии. Первая из них, это функция, которая создает из координат точек евклидовой плоскости точку комплексной плоскости.

$dToZ[\{x_, y_\}]: = x + Iy;$

Функция, которая выделяет у комплексной точки мнимую и действительную часть.

```
zToD[z_] := {Re[z], Im[z]};
```

Функция, которая создает координаты точек, полученные после инверсии.

```
inversion[z_, r_, a_] := a + r2/(z - a);
```

Перейдем к построению инверсии. Строим внутри окружности точки и к этим точка применяет инверсию. На рисунке 1.19 справа видим результат.

```
pointsD = Flatten[Table[{i, j}, {i, 3, 6, 0.5}, {j, -1, 1, 0.2}], 1];
```

```
pointsZ = dToZ/@pointsD;
```

```
invsZ = inversion[#, 7, 0] & /@ pointsZ;
```

```
invsD = zToD/@invsZ;
```

```
Graphics[{Blue, Point[{0, 0}], Circle[{0, 0}, 7], Red, Point[pointsD], Point[invsD]},
```

```
PlotRange -> {{-1, 24}, {-10, 10}}]
```

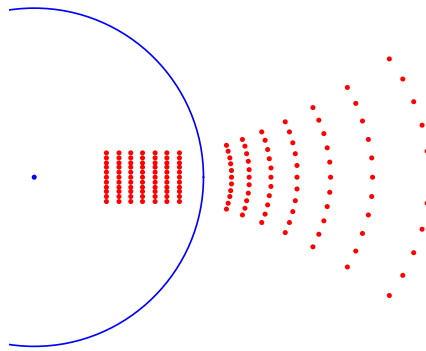


Рисунок 1.19 – Пример действия инверсии

2 Фракталы

2.1 Фракталы: развитие и характеристики

Развитие фрактальной геометрии тесно связано с именами таких известных математиков, как Вейштрасс, Кантор, Пеано, Хаусдорф, Кох, Серпинский и др.

Появление первых фракталов в математике, не обрадовало ученых. Эти странные объекты были восприняты с неприязнью. Один известный математик, Шарль Эрмит, даже назвал их монстрами. По его мнению, они не заслуживали внимания, и заниматься их изучением могли только псевдоученные. Но со временем отношение к этим монстрам изменилось, благодаря упорству и силе французского математика Бенуа Мандельброта.

Вся геометрия основывалась только на гладких кривых, т.е. геометрии Евклида. Время показало, что эта геометрия проигрывает большинству природных объектов из-за отсутствия в ней некоторой нерегулярности, беспорядка и непредсказуемости. Так на смену непрерывным кривым, обладающим всеми своими производными, пришли ломанные и очень изрезанные кривые. Именно Вейерштрасс впервые ввел в обращение непрерывную, но нигде не дифференцируемую функцию. Ярким примером такой кривой является траектория броуновской частицы.

Хаусдорф в 1919 г. ввел понятие о дробной размерности множеств и привел первые примеры этих множеств. Среди них были канторовское множество, кривая Коха и другие удивительные объекты, мало в то время известные. На основе теории фрактальной размерности Хаусдорфа, Бенуа Мандельброт ввел в том же году понятие фрактала:

«Фракталом называется структура, состоящая из частей, которые в каком-то смысле подобны целому» [7, с.11].

В переводе с латинского слово «фрактал» означает дробный, изрезанный. Фракталами являются геометрические объекты: линии, поверхности, пространственные тела, имеющие значительно изрезанную форму и обладающие свойством самоподобия.

Самоподобие, как основная характеристика фрактала означает, что при увеличении маленькие фрагменты фрактала получаются очень похожими на большие. В идеальном случае такое самоподобие приводит к тому, что фрактальный объект оказывается инвариантным относительно растяжений, т.е. ему, как говорят, присуща дилатационная симметрия. Она предполагает неизменность основных геометрических особенностей фрактала при изменении масштаба.

Для реального природного фрактала существует некоторый минимальный масштаб длины l_{min} , такой, что на расстояниях $l \approx l_{min}$ его основное свойство – самоподобие –

пропадает. Кроме того, на достаточно больших масштабах длин $l > l_{max}$, где l_{max} – характерный геометрический размер объектов, это свойство самоподобия также нарушается. Поэтому свойства природных фракталов рассматриваются лишь на масштабах l , удовлетворяющих соотношению $l_{min} \ll l \ll l_{max}$.

В основу построения многих фракталов лег универсальный фрактал, который Мандельброт назвал пылью. Это было классическое множество Кантора, полученное в 1875 или ранее годах. Оно настолько разрежено, что не содержит интервалов, но, тем не менее, имеет столько же точек, сколько интервал. Фрактальная пыль того или иного рода появляется в многочисленных ситуациях. Любой фрактал, - аттрактор системы итерированных функций - представляет собой либо фрактальную пыль, либо ее проекцию на пространство с более низкой размерностью. Различные древовидные фракталы применялись для моделирования деревьев - растений, бронхиального дерева, работы почек, кровеносной системы и др. Б. Мандельброт использовал такую «пыль» для моделирования стационарного шума в телефонии.

Огромный вклад во фрактальную геометрию внесли знаменитые работы французских математиков Г. Жюлиа и П. Фату, которые в начале 20 века занимались теорией рациональных отображений в комплексной плоскости. Их деятельность получила развитие в начале 80-х годов, когда с помощью компьютеров математикам удалось получить картины этих отображений. Так началась эра фрактальной геометрии.

2.2 Береговая линия

Понятие «фрактал» в физике возникло в связи с задачей об определении длины береговой линии. При ее вычислении по имеющейся карте местности была обнаружена интересная деталь. Оказалось, что чем более крупномасштабнее берется карта, тем более длинной оказывается эта береговая линия (Рисунок 2.1).



Рисунок 2.1 – Изображение береговой линии [1, с.13]

Предположим расстояние по прямой между расположенными на береговой линии точками А и В равно R . Для измерения длины береговой линии между этими точками, расставим по берегу жестко связанные друг с другом метки так, что расстояние между

соседними метками равнялось бы, например, $\delta = 10$ км. Длину береговой линии в километрах между точками А и В мы примем тогда равной числу меток минус одна, помноженному на 10. Следующие измерение этой длины мы произведем подобным же образом, но расстояние между соседними метками сделаем уже равным $\delta = 1$ км.

Получим, что результаты этих измерений будут отличаться. При уменьшении масштаба δ будут получаться все большие и большие значения длины. Линия морского побережья чаще всего оказывается настолько изрезанной, что с уменьшением длины звена δ величина L – длина береговой линии – не стремится к конечному пределу, а увеличивается по степенному закону

$$L \approx \delta \left(\frac{R}{\delta} \right)^D, \quad (2.1)$$

где $D > 1$ – некоторый показатель степени, который называется фрактальной размерностью береговой линии. С увеличением величины D , береговая линия становится более изрезанной.

Зависимость (2.1) можно объяснить следующим образом. Чем меньше масштаб мы используем, тем меньше детали побережья будут учтены и дадут вклад в измеряемую длину. С другой стороны, увеличивая масштаб, мы «спрямляем» побережье, уменьшая длину L .

Таким образом, для определения длины береговой линии L с помощью жёсткого масштаба δ , необходимо сделать $N = \frac{L}{\delta}$ шагов, причём величина L меняется с δ так, что N зависит от δ по закону:

$$N \approx \left(\frac{R}{\delta} \right)^D. \quad (2.2)$$

В результате с уменьшением масштаба длина береговой линии неограниченно возрастает. Это и отличает фрактальную кривую от обычной гладкой кривой, для которой предел длины аппроксимирующей ломанной L при стремлении к нулю длины ее звена δ конечен.

2.3 Фрактальная размерность

Основной чертой фракталов является изрезанность, извилистость их геометрических образов. Естественно возникает вопрос: можно ли указать какую-нибудь характеристику степени сложности фрактального объекта. Ответ на этот вопрос предложил Бенуа Мандельброт. В качестве такой характеристики он решил взять определение размерности, введенное в 1919 г. немецким математиком Ф. Хаусдорфом.

Рассмотрим обычную Евклидову размерность пространства, в котором будет находиться наш фрактальный объект. Обозначим ее через d и вспомним уже известные примеры. При $d = 1$ - линия, при $d = 2$ - плоскость, при $d = 3$ - обычное трехмерное пространство. Покроем фрактальный объект целиком d -мерными кубиками, причем величина ребра любого кубика δ_i не превышает некоторое значение $\delta > 0$, т.е. $\delta_i < \delta$. Введем зависящую от параметров D и δ сумму по всем элементам покрытия

$$l_D(\delta) = \sum \delta_i^D \quad (2.3)$$

и определим нижнюю грань этой суммы по всевозможным покрытиям, таким что $\delta_i < \delta$:

$$L_D(\delta) = \inf \sum \delta_i^D \quad (2.4)$$

Теперь устремим δ к нулю. Тогда, если D велико, то, очевидно, $L_D(\delta) \rightarrow 0$ при $\delta \rightarrow 0$. Если же D мало (близко к нулю), то $L_D(\delta) \rightarrow \infty$ при $\delta \rightarrow 0$. Существует промежуточное, критическое значение d_x , такое, что при уменьшении δ значение $L_D(\delta)$ стремится к нулю для $D > d_x$ и к бесконечности для $D < d_x$, т.е.

$$\lim_{\delta \rightarrow 0} L_D(\delta) = \begin{cases} 0, D > d_x \\ \infty, D < d_x \end{cases} \quad (2.5)$$

Число d_x называется размерностью Хаусдорфа-Безиковича.

Предположим, что для покрытия потребовалось не менее, чем $N(\delta)$ кубиков со стороной δ . Тогда, если при достаточно малых δ величина $N(\delta)$ меняется с δ по степенному закону

$$N(\delta) \sim \frac{1}{\delta^D} \quad (2.6)$$

то D – называется хаусдорфовой или фрактальной размерностью этого объекта. Эта формула эквивалентна соотношению (2.2), использованному для определения длины береговой линии.

Логарифмируя левую и правую части (2.6), (приблизительно) получаем

$$\log N(\delta) = \log 1 - D \log \delta$$

откуда, учитывая (2.5), получаем

$$D = - \lim_{\delta \rightarrow 0} \frac{\log N(\delta)}{\log \delta} \quad (2.7).$$

Это и является общим определением фрактальной размерности D . В соответствии с ним величина D является локальной характеристикой данного объекта.

Рассмотрим данное определение на примере хорошо известных множеств и проверим, действительно ли мы получим целочисленные значения. Для множества, состоящего из конечного числа изолированных точек, N , минимальное число d -мерных кубиков, с помощью которых можно покрыть это множество. При достаточно малом

размере кубиков оно совпадает с количеством точек, т.е. $N(\delta) = N$ и не зависит от размера грани этих кубиков δ . Следовательно согласно формуле (2.7), фрактальная размерность этого множества $D = 0$. Она совпадает с обычной Евклидовой размерностью изолированной точки $d = 0$ (точка – нульмерный объект).

Если рассмотрим отрезок прямой линии длиной L (состоящего из бесконечного числа точек), то минимальное число $N(l)$ кубиков с размером грани δ , с помощью которых можно покрыть данный отрезок целиком, равно, очевидно $N(\delta) = L/\delta$. Тогда согласно формуле (2.7), фрактальная размерность $D = 1$, т.е. совпадает с Евклидовой размерностью отрезка прямой $d = 1$. Для области площадью S гладкой двумерной поверхности число необходимых для ее покрытия квадратиков $N(\delta) = \frac{S}{\delta^2}$ (при достаточно малых δ), поэтому фрактальная размерность гладкой поверхности $D = 2$. И наконец, для покрытия некоторого конечного объема V необходимо $N(\delta) = \frac{V}{\delta^3}$ кубиков с ребром δ . Следовательно, фрактальная размерность этого множества $D = 3$.

Разберем теперь некоторые классические примеры регулярных фракталов, которые обладают свойством идеального самоподобия. Их покрытие можно осуществлять элементами, из которых состоит данный фрактал. В этом случае имеет место упрощенный вариант формулы (2.7) для определения фрактальной размерности.

Пусть на некотором этапе покрытия фрактала нам пришлось использовать, как минимум, $N(\delta)$ таких элементов характерного размера δ , а на другом $N(\delta')$ элементов размера δ' . Тогда величина фрактальной размерности D может быть вычислена по формуле

$$D = -\frac{\ln\left(\frac{N(\delta)}{N(\delta')}\right)}{\ln\left(\frac{\delta}{\delta'}\right)} \quad (2.8)$$

Теперь когда мы ввели понятие фрактальной размерности, можем привести другое, более строгое определение фрактала, также сформулированное Бенуа Мандельбротом:

«Фракталом называется множество, размерность Хаусдорфа-Безиковича которого строго больше топологической размерности» [7, с.40]

2.4 Канторовское множество

Классическое множество Кантора, или пыль Кантора (Рисунок 2.2), названо по имени немецкого математика Георга Кантора, который описал его в 1833 г.

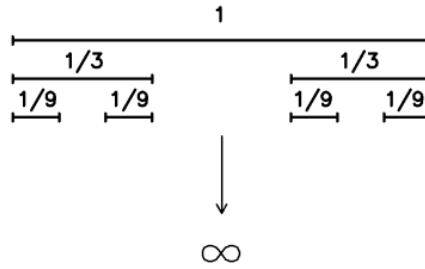


Рисунок 2.2 – Схема построения множества Кантора [1, с.17]

Схема построения данного множества:

Берётся отрезок прямой единичной длины (ему приписывают номер $n = 0$). Затем делим его на три равные части и вынимаем среднюю треть (не включая концы), т.е. отрезок находящийся между точками $\frac{1}{3}$ и $\frac{2}{3}$ (Рисунок 2.2). Это следующий шаг итерационной процедуры, обозначенный через $n = 1$. На $n = 2$ шаге подобной же процедуре деления на три части и последующего удаления середины подвергается каждый из двух оставшихся отрезков. Так продолжается до бесконечности, и мы получаем последовательность множеств:

$$C_0 = [0, 1]$$

$$C_1 = \left[0; \frac{1}{3}\right] \cup \left[\frac{2}{3}; 1\right]$$

$$C_2 = \left[0; \frac{1}{9}\right] \cup \left[\frac{2}{9}; \frac{1}{3}\right] \cup \left[\frac{2}{3}; \frac{7}{9}\right] \cup \left[\frac{8}{9}; 1\right]$$

... ..

Вычисли фрактальную размерность этого множества. На n -м шаге нашего построения мы имеем 2^n отрезков длиной $\frac{1}{3^n}$ каждый. Образует покрытие множества Кантора кубиками со стороной, равной длине отрезка на соответствующем шаге. Тогда в качестве $N(\delta)$ на n -м шаге мы можем взять значение 2^n , а в качестве δ – значение $\frac{1}{3^n}$. Предел $\delta \rightarrow 0$ соответствует пределу $n \rightarrow \infty$, поэтому фрактальная размерность

$$D = - \lim_{n \rightarrow \infty} \frac{\ln 2^n}{\ln \left(\frac{1}{3}\right)^n} = \frac{\ln 2}{\ln 3} \approx 0.6309$$

Она оказалась не целым числом и больше топологической размерности $d = 0$ элементов (точек) множества Кантора.

Свойства канторовой пыли:

1. Сумма длин интервалов, удалённых при построении множества C , в точности равна 1. Докажем это. Длина первого интервала, который мы выкинули, составляет $\frac{1}{3}$. Чтобы получить C_2 , мы выкинули два интервала, каждый длиной $\frac{1}{3^2}$. На следующем шаге

мы выбросили 2^2 интервалов, каждый длиной $\frac{1}{3^3}$, и т.д. Таким образом, сумма длин удаленных интервалов S составляет:

$$S = \frac{1}{3} + \frac{2}{3^2} + \frac{2^2}{3^3} + \dots + \frac{2^{n-1}}{3^n} + \dots$$

Но это выражение можно переписать в виде:

$$S = \left(\frac{1}{3}\right) \left(1 + \frac{2}{3} + \left(\frac{2}{3}\right)^2 + \left(\frac{2}{3}\right)^3 + \dots\right),$$

и с помощью формулы для суммы геометрической прогрессии, а именно,

$$\frac{1}{1-x} = 1 + x + x^2 + x^3 + \dots, \quad |x| < 1$$

мы получаем:

$$S = \left(\frac{1}{3}\right) \frac{1}{1-\frac{2}{3}} = 1.$$

Можно предположить, что если в C что-нибудь и осталось после удаления всех этих интервалов, то, наверное, не очень много. Однако это предположение отвергается следующим свойством.

2. Удивительный результат сравнения множества Кантора с интервалом состоит в том, что мощности этих множеств равны. Два множества обладают равной мощностью, если существует взаимно однозначное соответствие между точками этих множеств. В случае конечных множеств данное утверждение тривиально. Для бесконечных множеств, таких как интервал или множество Кантора, понятие мощности требует аккуратного обращения.

Теорема [9, с.42]:

Мощность множества Кантора C равна мощности континуума $[0,1]$.

Доказательство: Нам необходимо установить взаимно однозначное соответствие между точками из C и точками отрезка $[0,1]$. Для этого нам потребуется рассмотреть двоичное (по основанию 2), а также троичное (по основанию 3) представления точек отрезка $[0,1]$.

Для того чтобы избежать двусмысленности в случае, когда точка имеет два двоичных или троичных представления, мы будем всегда выбирать то представление, которое заканчивается всеми единицами в двоичном случае и всеми двойками в троичном.

Замечаем, что точка попадает в множество Кантора C тогда и только тогда, когда в ее троичном представлении отсутствуют единицы, то есть когда в нем присутствуют только нули и двойки. Тогда искомое соответствие точек из C с точками отрезка $[0,1]$ осуществляется заменой всех двоек в троичном представлении x на единицы. Полученное

таким образом двоичное представление определяет некоторое вещественное число y .
Например, если $x \in C$ есть:

$x = 0,202202002 \dots$ (в троичной системе),

то полагаем

$y = 0,101101001 \dots$ (в двоичной системе).

Описанная процедура определяет взаимно однозначное соответствие между $x \in C$ и $y \in [0,1]$. ■

Фрактальные свойства Кантора имеют огромное значение, так как многие известные фракталы подобны этому множеству.

2.5 Примеры известных фракталов

2.5.1 Снежинка Коха

Для построения снежинки Коха выполним следующие операции. Рассмотрим в качестве нулевой итерации равносторонний треугольник. Затем каждую из сторон этого треугольника разделим на три равные части, уберем среднюю часть и в середине построим равносторонний треугольник так, как изображено на рисунке 2.3. На следующем шаге такой же процедуре деления на три части и достраивания равностороннего треугольника подвергается каждая из сторон новой фигуры, и так до бесконечности. В результате возникает симметричная, похожая на снежинку, бесконечно изломанная кривая, которая представляет собой самоподобное множество, называемое снежинкой Коха.

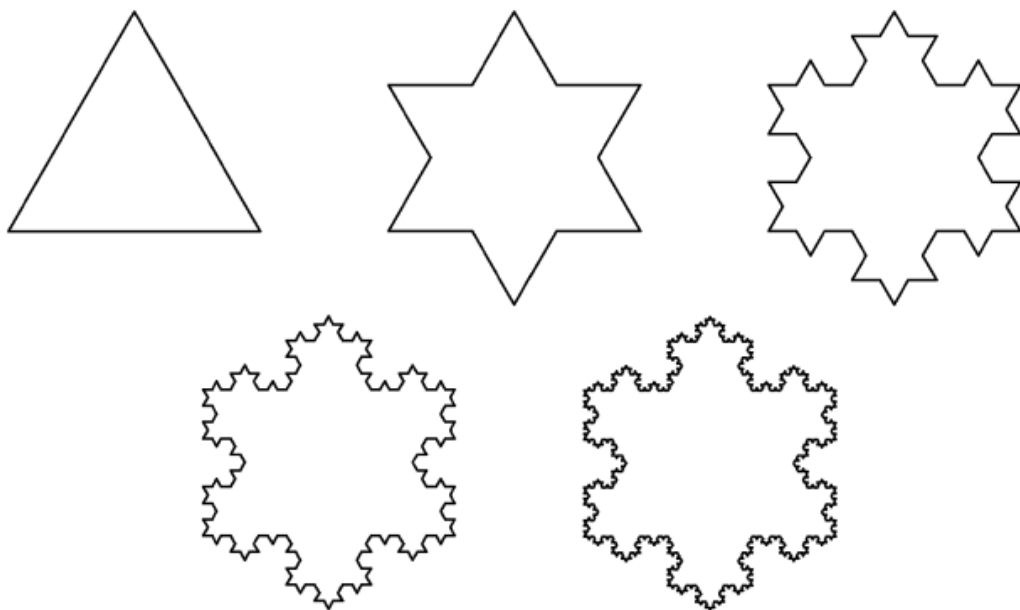


Рисунок 2.3 – Процесс построения снежинки Коха [1, с.18]

Она была так названа в честь шведского математика Helge von Koch, который впервые описал ее в 1904. Отличительной ее особенностью является то, что она, будучи замкнутой, тем не менее нигде себя не пересекает, поскольку достраиваемые треугольники каждый раз достаточно малы и никогда не «сталкиваются» друг с другом.

Посчитаем фрактальную размерность. В качестве длины исходного треугольника возьмем $l = 1$. Тогда число отрезков такой длины, которое покрывают снежинку Коха на $n = 0$ шаге, равно $N(l) = 3$. На следующем шаге имеем $l' = 1/3$, а число отрезков $N(l') = 12$. Тогда фрактальная размерность снежинки Коха равна

$$D = -\frac{\ln\left(\frac{3}{12}\right)}{\ln(3)} = \frac{\ln 4}{\ln 3} = 1.2618.$$

Эта величина больше единицы (топологической размерности линии), но меньше Евклидовой размерности плоскости, $d = 2$, на которой расположена кривая. Таким образом, снежинка Коха представляет собой линию бесконечной длины, ограничивающую конечную площадь. На n -м шаге ее длина $L = 3\left(\frac{4}{3}\right)^n$.

2.5.2 Салфетка и ковер Серпинского

Самоподобный фрактал, называемый салфеткой Серпинского, создан польским математиком В. Серпинским (1882-1969) в 1915 г.

Схема построения салфетки Серпинского:

На $n = 0$ шаге будет равносторонний треугольник с длиной стороны $l = 1$. Разобьем его на четыре области, соединив середины сторон исходного треугольника отрезками. Теперь удалим внутренность маленькой центральной области. Это и будет $n = 1$ шагом. Затем этот же процесс продолжается для трех оставшихся треугольников (Рисунок 2.4).



Рисунок 2.4 – Начальные шаги $n = 0, 1, 2$ построения салфетки Серпинского [1, с.20]

Продолжая данный процесс до бесконечности, мы придем к салфетке Серпинского (Рисунок 2.5).

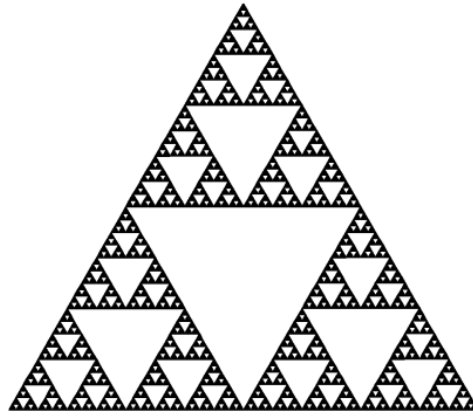


Рисунок 2.5 – Салфетка Серпинского [1, с.21]

Вычислим фрактальную размерность. Длина стороны исходного треугольника равна $l = 1$. Количество треугольников $N(l) = 1$. Сторона треугольников после $n=1$ шага станет равной $l' = 1/2$, а по количеству их станет $N(l') = 3$. Таким образом, фрактальная размерность салфетки Серпинского равна:

$$D = -\frac{\ln(1/3)}{\ln(2)} = \frac{\ln 3}{\ln 2} = 1.5849$$

Посчитаем суммарную площадь частей, исключённых при построении. На первом шаге мы выбросили $1/4$ часть площади исходного треугольника, образовав фигуру из трёх треугольников с площадью (каждого), равной $1/4$ площади исходного. На втором шаге у каждого из трёх треугольников удаляется $1/4$ часть его площади. Такие рассуждения позволяют определить полную удалённую площадь:

$$\begin{aligned} S &= \frac{1}{4} + 3 \left(\frac{1}{4}\right)^2 + 3^2 \left(\frac{1}{4}\right)^3 + \dots + 3^{n-1} \left(\frac{1}{4}\right)^n + \dots = \frac{1}{4} \left[1 + \frac{3}{4} + \left(\frac{3}{4}\right)^2 + \dots + \left(\frac{3}{4}\right)^n + \dots \right] = \\ &= \frac{1}{4} \cdot \frac{1}{1 - \frac{3}{4}} = 1. \end{aligned}$$

Салфетка имеет нулевую площадь, поскольку в процессе ее построения была исключена площадь, в точности равная площади исходного треугольника.

Ковер Серпинского

Аналогично салфетке Серпинского можно построить ковер Серпинского (Рисунок 2.6), который является двумерным аналогом канторовского множества исключенных средних третей.

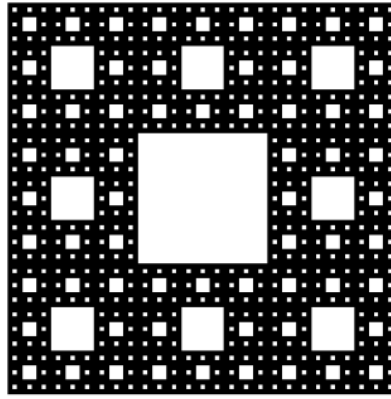


Рисунок 2.6 – Ковер Серпинского [1, с.23]

Схема построения:

Нулевой этап построения начинается с единичного квадрата. Затем каждая из сторон квадрата делится на три равные части, а весь квадрат, соответственно, на девять одинаковых квадратиков со стороной, равной $\frac{1}{3}$. Из полученной фигуры вырезается центральный квадрат. Затем такой же процедуре подвергается каждый из 8 оставшихся квадратиков и т.д. (Рисунок 2.7).

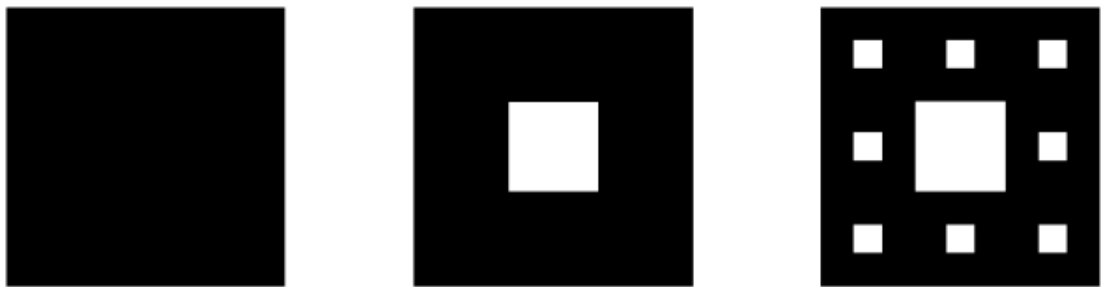


Рисунок 2.7 - Начальные шаги $n = 0, 1, 2$ построения ковра Серпинского [1, с.23]

В результате получается дырявый квадратный ковер Серпинского со значением фрактальной размерности

$$D = \frac{\ln 8}{\ln 3} = 1.8928.$$

Он также представляет собой пример идеального самоподобного фрактала. Его фрактальная размерность, однако, больше, чем у салфетки Серпинского, т.е. он является в каком-то смысле менее дырявым.

Суммарная площадь частей, исключенных при построении, будет равна:

$$\begin{aligned} S &= \frac{1}{9} + 8 \left(\frac{1}{9}\right)^2 + 8^2 \left(\frac{1}{9}\right)^3 + \dots + 8^{n-1} \left(\frac{1}{9}\right)^n + \dots = \frac{1}{9} \left[1 + \frac{8}{9} + \left(\frac{8}{9}\right)^2 + \dots + \left(\frac{8}{9}\right)^n + \dots \right] = \\ &= \frac{1}{9} \cdot \frac{1}{1 - \frac{8}{9}} = 1. \end{aligned}$$

2.5.3 Дракон Хартера-Хейтуэя

Это пример кривой Пеано, для которой область, которую она заполняет на плоскости, имеет весьма причудливую форму (Рисунок 2.8). Кривые Пеано - это фракталы, которые плотно заполняют пространство, в котором они находятся, так что их фрактальная размерность $D=d$.

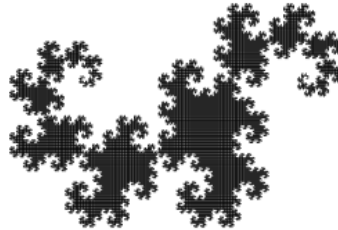


Рисунок 2.8 – Дракон Хартера-Хейтуэя [1, с.29]

Рассмотрим построение дракона Хартера-Хейтуэя. Первые 4 шага его построения изображены на рисунке 2.9.

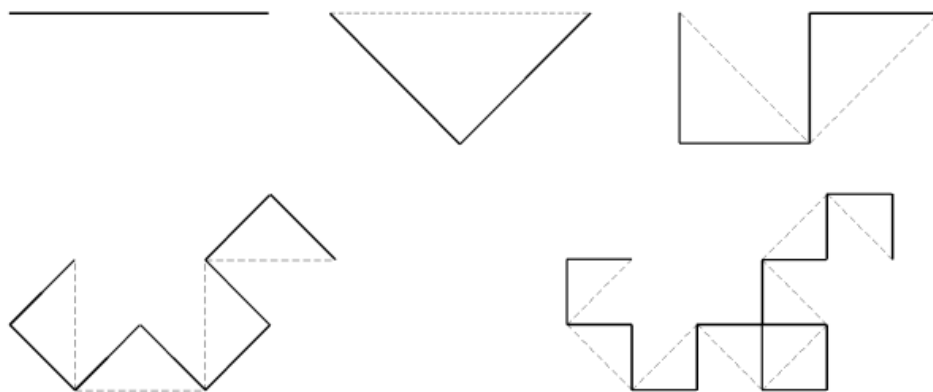


Рисунок 2.9 – Схема построения дракона Хартера-Хейтуэя [1, с.28]

Начнем с отрезка прямой единичной длины. На следующем шаге отрезок прямой заменяется на два отрезка, образующих боковые стороны равнобедренного прямоугольного треугольника, для которого исходный отрезок являлся бы гипотенузой. В результате отрезок как бы прогибается под прямым углом. Направление прогиба чередуется. Первый отрезок прогибается вправо (по ходу движения слева направо), второй – влево, третий – опять вправо и т.д. Для удобства восприятия на рисунке 2.9 пунктиром показана конфигурация предыдущего шага. Таким образом, после каждого шага число имеющихся отрезков удваивается, а длина каждого соответственно уменьшается в $\sqrt{2}$ раз.

Вычислим фрактальную размерность дракона Харетера-Хейтуэя. Длина стороны исходного отрезка равна $l = 1$. Количество отрезков $N(l) = 1$. Сторона отрезка после $n=1$ шага станет равной $l' = 1/\sqrt{2}$, а по количеству их станет $N(l') = 2$. Таким образом, фрактальная размерность равна:

$$D = -\frac{\ln\left(\frac{1}{2}\right)}{\ln\left(\frac{1}{\sqrt{2}}\right)} = \frac{\ln 2}{\ln \sqrt{2}} = \frac{2 \ln \sqrt{2}}{\ln \sqrt{2}} = 2.$$

Дракон представляет собой своеобразную гирлянду в форме двусторонней правой спирали, состоящую из подобных друг другу спиралевидных звеньев, непрерывно уменьшающихся в размерах от центра к периферии.

3 Методы построения фракталов

3.1 Система итерируемых функций (СИФ)

Создание многих регулярных фракталов основано на бесконечном повторении нескольких простых операций, например, замены одного элемента некоторой комбинацией других, ему подобных. Тогда возникает вопрос, можно ли перевести такую процедуру на язык математических формул.

Ответом послужил появившийся в середине 80-х годов метод Систем Итерируемых Функций - СИФ (*Iterated Function System*), придуманный американским математиком М.Барнсли. Результат применения системы итерированных функций называется аттрактором. Это может быть любой компакт, будь то интервал, квадрат или фрактал.

Метод СИФ можно реализовать с помощью двух подходов: детерминированного и метода случайных итераций (или игра в хаос).

3.1.1 Детерминированный метод

Сущность данного метода рассмотрим на примере салфетки Серпинского.

Поместим исходный равносторонний треугольник с длиной стороны, для определенности равной единице, на комплексную плоскость $[z]$. Теперь необходимо определить, с помощью какого линейного преобразования t_1 на комплексной плоскости (Рисунок 3.1) исходный треугольник перейдет в равносторонний треугольник в два раза меньшего размера?

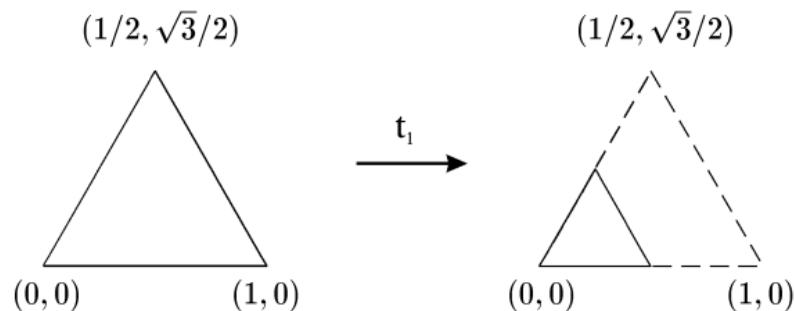


Рисунок 3.1 – Преобразование t_1 для равностороннего треугольника [1, с.33]

Так как левое основание обоих треугольников лежит в начале координат $z=0$, то функция $f_1(z)$, осуществляющая данное преобразование, определяется выражением

$$t_1: f_1(z) = \frac{1}{2}z$$

Преобразование t_2 будет соответствовать преобразованию t_1 , смещенному по горизонтали вправо на величину равную $\frac{1}{2}$ (Рисунок 3.2). Функция $f_2(z)$, соответствующая этому преобразованию, равна

$$t_2: f_2(z) = \frac{1}{2}z + \frac{1}{2}$$

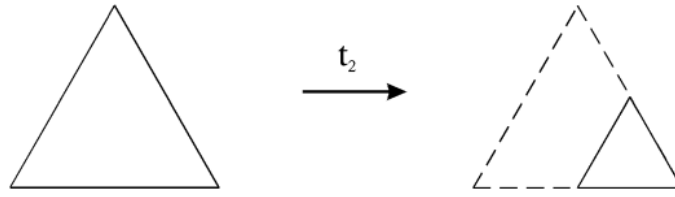


Рисунок 3.2 – Преобразование t_2 для равностороннего треугольника [1, с.34]

Последний, третий, маленький треугольник получается с помощью преобразования t_3 (Рисунок 3.3). Отвечающая ему функция $f_3(z)$ получается из $f_1(z)$ трансляцией на комплексный вектор $\frac{1}{4} + \frac{i\sqrt{3}}{4}$

$$t_3: f_3(z) = \frac{1}{2}z + \frac{1}{4} + \frac{i\sqrt{3}}{4}$$

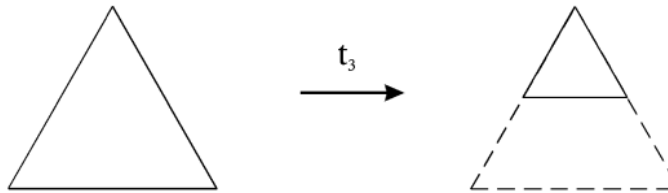


Рисунок 3.3 – Преобразование t_3 для равностороннего треугольника [1, с.34]

В итоге три вышеназванные линейные функции $f_1(z)$, $f_2(z)$ и $f_3(z)$ осуществляют искомое преобразование исходного треугольника в три треугольника в два раза меньшего размера. Если теперь каждый из этих трех маленьких треугольников подвергнуть этим трем преобразованиям (Рисунок 3.4), то возникнет уже 9 треугольников с размером в 4 раза меньше исходного.

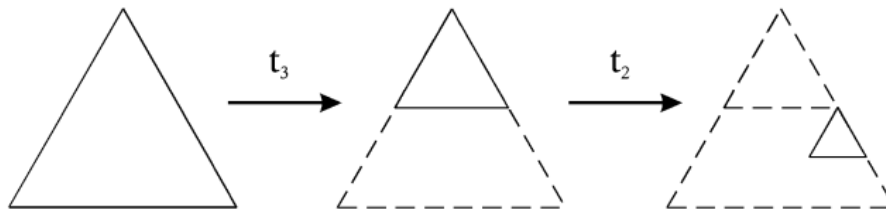


Рисунок 3.4 – Изображение двух последовательных преобразований [1, с.35]

Например, выполняя сначала преобразование t_3 , а затем преобразование t_2 , мы в итоге получаем треугольник со стороной $1/4$. На рисунке 3.5 слева показан первый шаг итерационной процедуры, а справа изображены все треугольники с обозначением результирующего преобразования – генеалогического кода, при помощи которого они были получены из исходного треугольника.

На каждом шаге уменьшаясь в размерах в два раза и утраиваясь в количестве, эти фигуры в конце концов превратились бы в неразличимые глазом бесформенные точки, образующие фрактал – салфетку Серпинского. Это является преимуществом детерминированного подхода. Недостатком же является то, что приходится хранить предыдущие вычисления для получения новых.

3.1.2 Метод случайных итераций, или игра в хаос

Рассмотрим рандомизированный метод, который М. Барнсли назвал игрой в хаос (chaos game). Для этого возьмём знакомый равносторонний треугольник с вершинами в точках А, В и С. Выберем внутри этого треугольника произвольным образом начальную точку (Рисунок 3.6). Бросим теперь игральную кость, представляющую собой кубик, на 6 гранях которого проставлены буквы А, В и С. Пусть каждая буква присутствует на двух из них, тогда вероятность выпадения любой буквы одинакова и равна $\frac{1}{3}$.

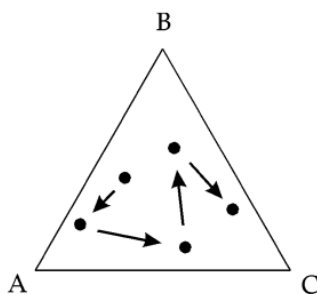


Рисунок 3.6 – Метод случайных итераций [1, с.38]

Предположим, что в результате первого броска выпала буква А. Соединим нашу начальную точку с вершиной треугольника А отрезком прямой и на его середине поставим точку. Пусть теперь она будет играть роль начальной. После чего повторим вышеописанную процедуру с бросанием кубика и проставлением точки в середине соответствующего отрезка. Допустим, на втором шаге выпала буква С, потом В, затем опять С и т.д. В результате на каждом шаге мы будем получать все новые и новые точки. И с каждым увеличением все явственнее будет проступать структура треугольника Серпинского (Рисунок 3.7).

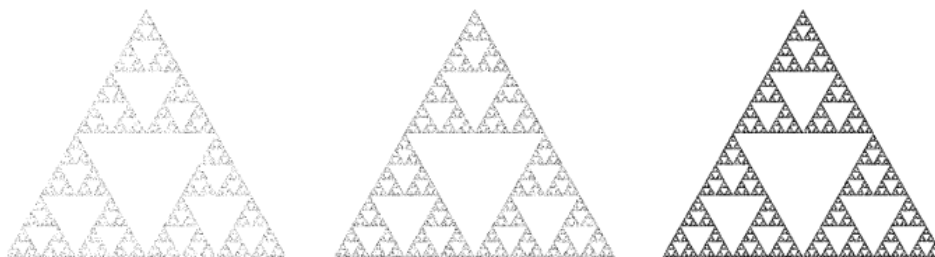


Рисунок 3.7 – Результаты игры в хаос [1, с.38]

Связь игры в хаос с системой итерируемых функций прослеживается в следующем. На каждом шаге к начальной точке z применяется (случайным образом) одно из трех вышеописанных линейных преобразований $f_1(z)$, $f_2(z)$, или $f_3(z)$. Если обозначить координаты вершин треугольника A, B, C на комплексной плоскости через z_a, z_b и z_c соответственно, то поскольку $z_a = 0, z_b = \frac{1}{2} + \frac{i\sqrt{3}}{2}$ и $z_c = 1$, видно, что

$$f_1(z) = \frac{z + z_a}{2} = \frac{1}{2}z$$

$$f_2(z) = \frac{z + z_c}{2} = \frac{1}{2}z + \frac{1}{2}$$

$$f_3(z) = \frac{z + z_b}{2} = \frac{1}{2}z + \frac{1}{4} + \frac{i\sqrt{3}}{4}$$

Таким образом, треугольник Серпинского, являясь аттрактором для этой системы итерируемых функций, возникает и при чисто случайном выборе последовательности преобразований $t_i t_j t_k \dots$

Приведем алгоритм действия метода случайных итераций:

z_0 — начальная точка (произвольная)

$z_1 = t_1(z_0)$ или $t_2(z_0)$ или $t_3(z_0)$,

$z_2 = t_1(z_1)$ или $t_2(z_1)$ или $t_3(z_1)$,

.....

$z_n = t_1(z_{n-1})$ или $t_2(z_{n-1})$ или $t_3(z_{n-1})$,

Покажем, что изображающая точка в результате бесконечной цепочки случайных итераций сколь угодно близко подойдет к каждой точке этого фрактального множества (Рисунок 3.8).

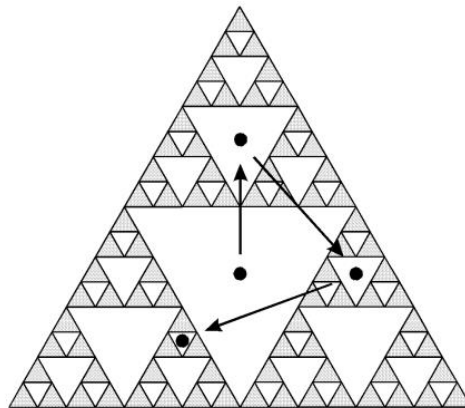


Рисунок 3.8 - Изображающая точка в результате бесконечной цепочки случайных итераций [1, с.39]

Возьмем начальную точку в центре самого большого исключенного треугольника. На следующем шаге изображающая точка оказывается в центре одного из трех

треугольников поменьше. Эти треугольники представляют собой геометрическое место точек, которые находятся на половине расстояния до соответствующих вершин от точек большого центрального треугольника. При следующей итерации точка попадает в центр еще меньшего исключенного треугольника и т.д. В конце концов после небольшого числа итераций точка попадает в исключенный треугольник столь малого размера, что его можно для всех практических целей считать точечным.

Для этой игры совершенно необязательно чтобы исходный треугольник был равносторонним. На рисунке 3.9 приведен пример применения игры для треугольника произвольной формы.

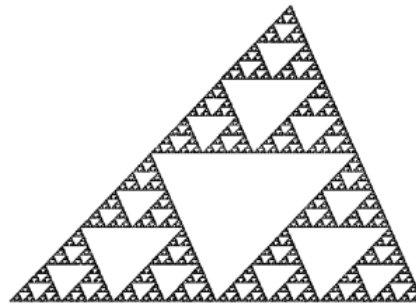


Рисунок 3.9 – Салфетка Серпинского для произвольного треугольника [1, с.40]

Если мы изменим правила игры и, например, будем проставлять точку не на середине отрезка, а на расстоянии в $\frac{1}{3}$ от соответствующей вершины, то мы получим множество точек (Рисунок 3.10), которое можно назвать двумерным аналогом канторовского множества исключенных средних третей.



Рисунок 3.10 – Салфетка Серпинского при сжатии в 3 раза [1, с.41]

А если возьмем другую фигуру, скажем правильный шестиугольник, и поставим точку на расстоянии в $\frac{1}{3}$ от соответствующей вершины, то эти точки в процессе итераций образуют множество, которое условно можно назвать шестиугольником Серпинского (Рисунок 3.11).

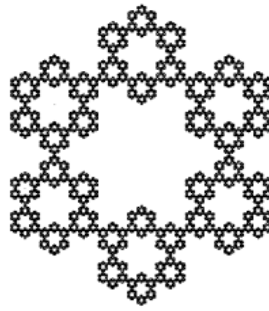


Рисунок 3.11 – Шестиугольник Серпинского [1, с.41]

Именно в этом случае игра в хаос будет настоящей игрой в кости, так как теперь на шести гранях игрального кубика можно поставить цифры от одного до шести, соответствующие каждой из вершин шестиугольника. Можно заметить, что внутренняя граница этой фигуры представляет собой уже известный фрактал – снежинку Коха.

Преимущество рандомизированного алгоритма состоит в том, что нет необходимости хранить большие массивы данных в памяти. С другой стороны, для того чтобы получить более четкое изображение, требуются тысячи точек, что в свою очередь делает подход затратным.

3.1.3 Игры с поворотами

В детерминированном и рандомизированном подходах использовались линейные преобразования, которые включали в себя сжатие в m раз и параллельный перенос на некоторый комплексный вектор. Добавим теперь еще одно линейное преобразование - поворот. Разберемся на примере кривой Коха.

Инициатором для кривой Коха является отрезок единичной длины, который на следующем шаге преобразуется в генератор (Рисунок 3.12).

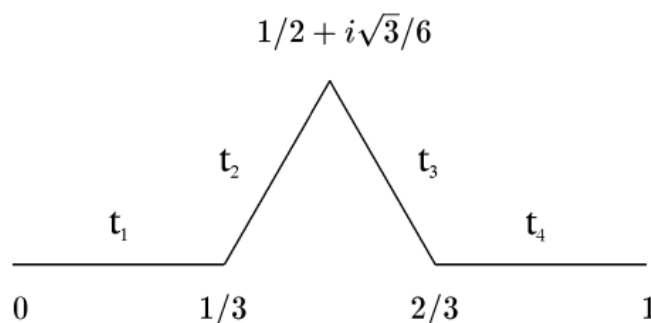


Рисунок 3.12 – Генератор кривой Коха [1, с.42]

Числа показывают координаты соответствующих вершин на комплексной плоскости. Данное преобразование осуществляет следующие функции:

$f_1(z) = \frac{1}{3}z$ - функция, осуществляющее преобразование t_1 , которое отвечает за сжатие в 3 раза исходного отрезка (0,1).

$f_2(z) = \frac{1}{3}e^{\frac{i\pi}{3}}z + \frac{1}{3}$ - функция, осуществляющая преобразование t_2 - сжатие в 3 раза исходного отрезка (0,1), поворот против часовой стрелки вокруг начала координат на угол 60° и последующий сдвиг по горизонтали вправо на $\frac{1}{3}$.

$f_3(z) = \frac{1}{3}e^{-\frac{i\pi}{3}}z + \frac{3+i\sqrt{3}}{6}$ - функция, осуществляющая преобразование t_3 - сжатие в 3 раза исходного отрезка (0,1), поворот на угол 60° по часовой стрелке и последующий сдвиг на комплексный вектор $\frac{3+i\sqrt{3}}{6}$.

$f_4(z) = \frac{1}{3}z + \frac{2}{3}$ - функция, осуществляющая преобразование t_4 - сжимает отрезок в 3 раза и смещает его по горизонтали вправо на $\frac{2}{3}$.

В данном случае порядок выполняемых операций имеет значение. Если теперь подвергнуть генератор этим четырем преобразованиям, то получим конструкцию изображенную на рисунке 3.13. Она состоит из 16 отрезков длиной уже $\frac{1}{9}$, и для каждого из них показана последовательность операций, с помощью которых он был получен из исходного единичного отрезка. Кривая Коха получится, если повторять этот процесс до бесконечности.

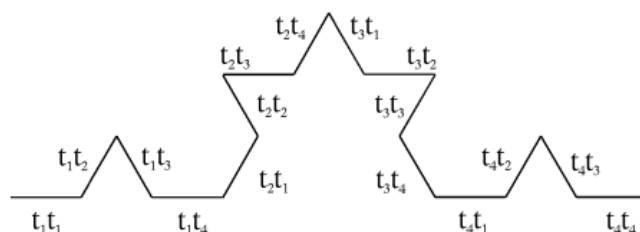


Рисунок 3.13 – Результат действия преобразований на генератор кривой Коха [1,с.43]

Рассмотрим другой пример. Пусть на плоскости находятся две точки А и В. Случайным образом выберем точку z_0 и одну из этих двух точек. Предположим, что на первом шаге «выпала» точка А. Мысленно соединяем ее с начальной отрезком прямой линии, длина которого пусть будет равна l , и перемещаемся вдоль этого отрезка в точку, отстоящую от А на расстоянии $l/\sqrt{2}$. Затем поворачиваемся вокруг А по часовой стрелке на угол в 45° и ставим в этом месте новую точку, которая теперь будет играть роль начальной. После чего опять случайным образом выбираем одну из точек А или В и повторяем весь процесс сначала.

Результат данного процесса изображен на рисунке 3.14. Своими очертаниями он похож на дракона Хартера-Хейтуэя. Единственное отличие - это отсутствие характерных перемычек, которые соединяют разные части дракона. Это связано с тем, что на самом деле на рисунке 3.14 изображено два совершенно одинаковых дракона Хартера-Хейтуэя, один из которых повернут на 180° относительно другого. Их стыковка настолько идеальна, что между ними не существует никаких пробелов. Поэтому эту фигуру и называют двойным драконом. Одним из интересных геометрических свойств двойного дракона является то, что его можно покрыть четырьмя уменьшенными копиями его самого. Фрактальная размерность данного аттрактора равна 2.

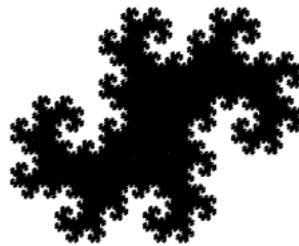


Рисунок 3.14 – Двойной дракон Хартера Хейтуэя [1, с.44]

Запишем этот фрактал через систему итерируемых функций. Пусть координаты точек А и В на комплексной плоскости соответственно равны: $z_1 = i - 1$ и $z_2 = i + 1$. Тогда

$$f_1(z) = \frac{1}{\sqrt{2}}(z - z_1)e^{-\frac{i\pi}{4}} + z_1 = \frac{1}{\sqrt{2}}ze^{-\frac{i\pi}{4}} - 1$$

$$f_2(z) = \frac{1}{\sqrt{2}}(z - z_2)e^{-\frac{i\pi}{4}} + z_2 = \frac{1}{\sqrt{2}}ze^{-\frac{i\pi}{4}} + i$$

Найдем систему итерируемых функций для самого дракона Хартера-Хейтуэя. Иницирующим элементом является равнобедренный прямоугольный треугольник ABC (изображенный на рисунке 3.15 пунктиром). Затем он заменяется на два ему подобных треугольника ADB и BEC.

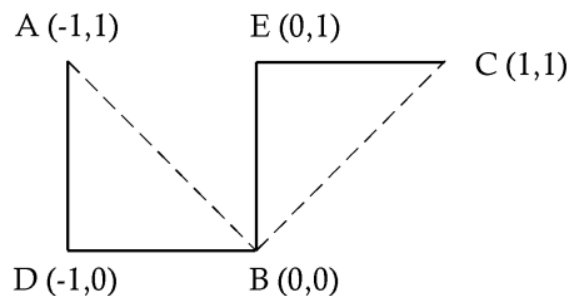


Рисунок 3.15 – Начальный этап построения дракона Хартера-Хейтуэя [1, с.47]

Первое преобразование переводит треугольник ABC в треугольник ADB, причем вершины треугольников переходят друг в друга по правилу $A \rightarrow A, B \rightarrow D$ и $C \rightarrow B$.

$f_1(z) = \frac{1}{\sqrt{2}}ze^{-\frac{i\pi}{4}} - 1$ - поворот треугольника ABC вокруг начала координат (т.е. точки B) по часовой стрелке на угол 45° , сжатие вдоль осей X и Y в $\sqrt{2}$ раз и перенос на единицу влево.

Второе преобразование переводит треугольник ABC в CEB, причем вершины треугольников переходят друг в друга по правилу $A \rightarrow C, B \rightarrow E$ и $C \rightarrow B$.

$f_2(z) = \frac{1}{\sqrt{2}}ze^{-\frac{i3\pi}{4}} + i$ - поворот треугольника ABC вокруг точки B на угол в 135° по часовой стрелке, сжатие вдоль осей X и Y в $\sqrt{2}$ раз и перенос на единицу вверх в положительном направлении оси Y.

Эти два линейных комплексных преобразования и составляют систему итерируемых функций, аттрактором для которой является дракон Хартера-Хейтуэя (Рисунок 3.16).

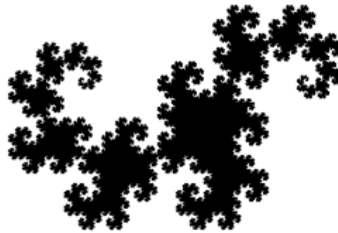


Рисунок 3.16 – Дракон Хартера-Хейтуэя [1, с.48]

3.2 Лист папоротника

Лист папоротника является одним из интересных примеров системы итерируемых функций. Он был открыт М.Барнсли. Лист папоротника - это аттрактор для системы из четырех сжимающих аффинных преобразований. Аффинные преобразования называются сжимающими, если длина произвольного отрезка при аффинном сжатии уменьшается.

В таблице 1 приведены коэффициенты аффинных преобразований для системы. Результат действия этой системы функций на некоторую точку для разного количества итераций приведен на рисунке 3.17.

Таблица 1 – Коэффициенты аффинных преобразований для листа папоротника [1, с.55]

a	b	c	d	e	f
0.00	0.00	0.00	0.16	0.00	0.00
0.85	0.04	-0.04	0.85	0.00	1.60
0.20	-0.26	0.23	0.22	0.00	1.60

Окончание таблицы 1

a	b	c	d	e	f
-0.15	0.28	0.26	0.24	0.00	0.44

С увеличением количества итераций проявляется более четкое изображение существующего в природе растения.



Рисунок 3.17 – Результат действия СИФ для листа папоротника [1,с.55]

Как и полагается всякому фракталу данное множество точек бесконечно самоподобно. Увеличивая малые фрагменты этого изображения можно увидеть их подобие целому (Рисунок 3.18).



Рисунок 3.18 – Увеличенное изображение листа папоротника [1, с.56]

Но откуда взялись эти коэффициенты? Разберемся в этом и попробуем дать геометрическую интерпретацию каждому числу. В качестве иницилирующего элемента возьмем квадрат ABCD (изображен на рисунке 3.19 пунктиром), повернутый на 45° так, что одна из его диагоналей совпадает с вертикальной осью Y, а другая параллельна горизонтальной оси X. Начало координат совпадает с точкой A.

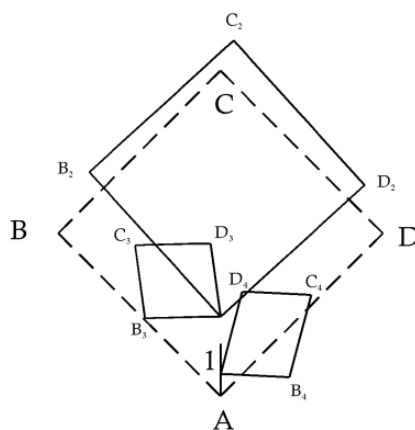


Рисунок 3.19 – Схема построения листа папоротника [1, с.57]

Первая строка в таблице соответствует сжатию исходного квадрата в вертикальный отрезок прямой, длина которого составляет 16% от диагонали квадрата. Этот отрезок обозначим цифрой 1, и он будет служить в качестве «стебля» для листьев папоротника.

Второе преобразование изменяет квадрат $ABCD$ в квадрат $A_2B_2C_2D_2$, который имеет размер в 85% от исходного. Затем этот квадрат повернули по часовой стрелке относительно вершины A на угол в 2.7° , и сместили по вертикали вверх на расстояние в 1.6 см (см - в качестве условной единицы измерения).

Третье - сжимает сторону квадрата AB в 3.07 раз и поворачивает вокруг точки A на угол в 46.25° против часовой стрелки. Сторона же квадрата AD этим преобразованием сжимается примерно в 3.12 раза и поворачивается в том же направлении на угол 52.6° . После чего образовавшийся параллелограмм сдвигается вертикально на 1.6 см. Таким образом, квадрат $ABCD$ преобразовывается в параллелограмм $A_3B_3C_3D_3$.

Последнее преобразование переводит исходный квадрат в параллелограмм $A_4B_4C_4D_4$. В этом случае сторона AB сжимается в 3.29 раза и поворачивается по часовой стрелке вокруг начала координат на угол примерно в 137.7° . Сторона же AD сжимается в 2.74 раза и поворачивается на угол в 30.4° , но против часовой стрелки. Затем получившийся параллелограмм смещается на 0.44 см по вертикали вверх.

В результате четвертого преобразования направление обхода вершин параллелограмма $A_4B_4C_4D_4$ меняется на противоположный по сравнению с исходным квадратом. Для дальнейшего разбора поработаем немного с таблицей 1.

Для начала исключим первую строчку из таблицы 1 и тогда получим таблицу 2.

Таблица 2 – Коэффициенты аффинных преобразований для листа папоротника без первого преобразования [1, с.58]

a	b	c	d	e	f
0.85	0.04	-0.04	0.85	0.00	1.60

Продолжение таблицы 2

a	b	c	d	e	f
0.20	-0.26	0.23	0.22	0.00	1.60
-0.15	0.28	0.26	0.24	0.00	0.44

Исключив всего лишь одно преобразование, изображение листа папоротника сменилось другим (Рисунок 3.20 слева).

Преобразование 1 и 2 отвечает за формирование стебля. Если мы оставим только их (Таблица 3), то получится стебель, изображенный на рисунке 3.20 справа.

Таблица 3 – Коэффициенты аффинных преобразований, отвечающие за формирование стебля [1, с.58]

a	b	c	d	e	f
0.00	0.00	0.00	0.16	0.00	0.00
0.85	0.04	-0.04	0.85	0.00	1.60



Рисунок 3.20 – Слева изображен лист папоротника, полученный при исключении первой строчки из СИФ. Справа изображен стебель папоротника [1, с.58]

За изгиб листа папоротника вправо отвечает поворот на угол в 2.7° , присутствующий во втором преобразовании. Исключим его и получим еще одну систему функций (Таблица 4).

Таблица 4 – Коэффициенты аффинных преобразований без изгиба вправо для листа папоротника [1, с.59]

a	b	c	d	e	f
0.00	0.00	0.00	0.16	0.00	0.00
0.85	0.00	0.00	0.85	0.00	1.60
0.20	-0.26	0.23	0.22	0.00	1.60
-0.15	0.28	0.26	0.24	0.00	0.44

Данной системе функций соответствует рисунок 3.21 слева. Лист папоротника на нем получился прямым.

Если изменить в четвертом преобразовании перенос, т.е. поменять коэффициент на число соответствующее переносу в третьем преобразовании (Таблица 5). Получится

изображение на рисунке 3.21 справа. Можно заметить, что третье и четвертое преобразования отвечают за расположение листьев папоротника.

Таблица 5 – Коэффициенты аффинных преобразований для листа папоротника. Замена коэффициента, отвечающего за перенос в четвертом преобразовании, на коэффициент, отвечающий за перенос в третьем преобразовании [1, с.59]

a	b	c	d	e	f
0.00	0.00	0.00	0.16	0.00	0.00
0.85	0.00	0.00	0.85	0.00	1.60
0.20	-0.26	0.23	0.22	0.00	1.60
-0.15	0.28	0.26	0.24	0.00	1.60



Рисунок 3.21 – Слева лист папоротника без изгиба. Справа лист папоротника при замене значений коэффициентов переноса [1, с.59]

Таким образом, можно сделать вывод, что 1 и 2 преобразование формируют стебель, 2 и 3 - листья с одной стороны, 2 и 4 - листья с другой стороны. Осталось понять, зачем в 4 преобразовании присутствует отражение. Для этого рассмотрим для 3 и 4 преобразования следующие матрицы:

$$\tilde{A}_3 = \frac{1}{3} \begin{pmatrix} \cos 60^\circ & -\sin 60^\circ \\ \sin 60^\circ & \cos 60^\circ \end{pmatrix}, \quad \tilde{A}_4 = \frac{1}{3} \begin{pmatrix} -\cos 60^\circ & \sin 60^\circ \\ \sin 60^\circ & \cos 60^\circ \end{pmatrix}$$

Матрица $\tilde{A}_3$ соответствует сжатию в 3 раза по обеим осям и повороту против часовой стрелки на угол 60° . Матрица $\tilde{A}_4$ соответствует матрице $\tilde{A}_3$ с дополнительной операцией отражения в вертикальной плоскости, т.е. замена $x \rightarrow -x$ (Таблица 6). Результат такого преобразования показан на рисунке 3.22 слева.

Таблица 6 – Коэффициенты аффинных преобразований для листа папоротника. Третья строчка записана через матрицу $\tilde{A}_3$, а четвертая строка с использованием матрицы $\tilde{A}_4$ [1, с.60]

a	b	c	d	e	f
0.0000	0.0000	0.0000	0.1600	0.0000	0.0000
0.8500	0.0400	-0.0400	0.8500	0.0000	1.6000

Продолжение таблицы 6

a	b	c	d	e	f
0.1667	-0.2887	0.2887	0.1667	0.0000	1.6000
-0.1667	0.2887	0.2887	0.1667	0.0000	0.4400

Если теперь для преобразования 4 вместо поворота на $+60^\circ$ и последующего отражения в вертикальной плоскости используем поворот на -60° (т.е. по часовой стрелке), то получим систему функций, указанных в таблице 7.

Таблица 7 – Коэффициенты аффинных преобразований для листа папоротника. В преобразовании 4 заменили поворот на 60° по часовой стрелке, вместо 60° против часовой [1, с.61]

a	b	c	d	e	f
0.0000	0.0000	0.0000	0.1600	0.0000	0.0000
0.8500	0.0400	-0.0400	0.8500	0.0000	1.6000
0.1667	-0.2887	0.2887	0.1667	0.0000	1.6000
-0.1667	0.2887	-0.2887	0.1667	0.0000	0.4400

Тогда результатом будет изображение на рисунке 3.22 справа.



Рисунок 3.22 – Листы папоротника. В левом присутствует преобразование – поворот на 60° против часовой стрелки, в правом - поворот на 60° по часовой стрелке [1, с.61]

Если сравнить эти изображения, то мы увидим, что отражение приводит к правильному изгибу тех листьев, которые расположены с правой стороны от стебля.

На основе примера листа папоротника можно создать различные его модификации. Например, хорошо известное и вечно зеленое дерево – елка (Рисунок 3.23). При построении надо учесть, что ветки елки должны смотреть вниз, а не вверх как у папоротника.



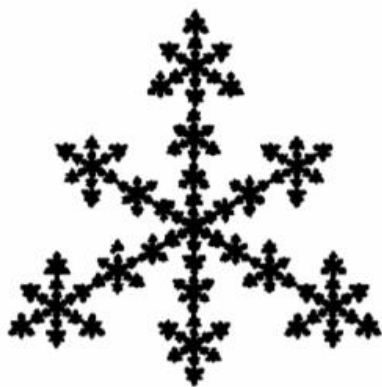
Рисунок 3.23 – Елка [1, с.62]

Коэффициенты для ее построения представлены в Таблице 8.

Таблица 8 – Коэффициенты аффинных преобразований для елки [1, с.62]

a	b	c	d	e	f
0.1000	0.0000	0.0000	0.1600	0.0000	0.0000
0.8500	0.0000	0.0000	0.8500	0.0000	1.6000
-0.1667	-0.2887	0.2887	-0.1667	0.0000	1.6000
-0.1667	0.2887	-0.2887	-0.1667	0.0000	1.6000

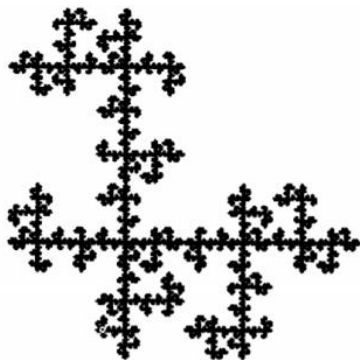
Приведем ряд других примеров систем итерируемых функций (Рисунок 3.24) на $[0,1] \times [0,1]$, соответствующие таблицы для их построения находятся в ПРИЛОЖЕНИИ Б.



а)



б)



в)



г)



д)



е)

Рисунок 3.24 – Примеры систем итерируемых функций [, с.]: а) кристалл; б) лист папоротника; в) ковер А; г) ковер В; д) лист; е) дерево

4 Построение фракталов с помощью Wolfram Mathematica

4.1 Описание основных функций

Перед тем как перейдем к построению фракталов, зададим основные функции, реализующие подходы системы итерируемых функций. Функции, используемые в выражениях, детально разобраны в ПРИЛОЖЕНИИ А [8].

1. Данная функция к каждому элементу списка pl применяет преобразование T . Затем результаты этого списка объединяются.

$$IFS[\{T_TransformationFunction\}][pl_List] := Join@@Through[\{T\}[pl]] \quad (4.1)$$

2. Функция отвечает за графическое изображение результатов СИФ. Изображение получается в виде крошечных точек. Они получаются из списка точек pts , к которым n раз применили функцию f .

$$IFSPlot[f_IFS, pts_n_Integer] := Graphics[\{PointSize[Tiny], Point[Nest[f, pts, n]]\}] \quad (4.2)$$

3. Из списка точек pl произвольным образом будут составляться список из m точек. Затем к этому списку применяется преобразование T , а результаты объединяться. Данная функция предназначена для реализации метода «игра в хаос».

$$RandomIFS[IFS[\{T_TransformationFunction\}], m_][pl_List] := Apply[Join, Through[\{T\}[RandomChoice[pl, m]]]] \quad (4.3)$$

4. При построении изображения, полученного рандомизированным методом, необходимо учесть, как будет задаваться теперь функция. Вместо f_IFS нужно указать $f_RandomIFS$.

$$IFSPlot[f_RandomIFS, pts_n_Integer] := Graphics[\{pointSize[Tiny], Point[Nest[f, pts, n]]\}] \quad (4.4)$$

5. Функция для реализации детерминированного метода. Задаются локальные переменные: $prim$ - выбор первого элемента, h - определение действующей функции, t - генерирует список функции преобразования $l[[i]]$, применённых к $prim$. Эти переменные используются для вычисления функции $h[t]$.

$$TransformIFS[g_IFS[l_List]] := Module[\{prim = First[g], h = Head[g]\}, t = Table[GeometricTransformation[prim, l[[i]]], \{i, Length[l]\}]; h[t]]; \quad (4.5)$$

Принцип работы этой функции в общем виде:

$$\begin{aligned} TransformIFS[g_ifs_IFS, 0] &:= g; \\ TransformIFS[g_ifs_IFS, 1] &:= TransformIFS[g, ifs]; \end{aligned}$$

$$\begin{aligned} & TransformIFS[g_, ifs_IFS, n_Integer? Positive] := \\ & TransformIFS[TransformISF[g, ifs], ifs, n - 1]; \end{aligned} \quad (4.6)$$

4.2 Кривая Коха

Для построения будем использовать игру с поворотами. Зададим систему итерируемых функций, которые будут описывать данный фрактал.

$AffineTransform[\{D\}]$ - сжатие в 3 раза.

$AffineTransform[\{RotationMatrix[60Degree]/3, \{1/3, 0\}\}]$ - сжатие в 3 раза, поворот против часовой стрелки вокруг начала координат на угол 60° и последующий сдвиг по горизонтали вправо на $\frac{1}{3}$.

$AffineTransform[\{RotationMatrix[-60Degree]/3, \{1/2, Sqrt[3]/6\}\}]$ - сжатие в 3 раза, поворот на угол 60° по часовой стрелке и последующий сдвиг на комплексный вектор $\frac{3+i\sqrt{3}}{6}$.

$AffineTrasform[\{D, \{2/3, 0\}\}]$ - сжатие в 3 раза и смещение по горизонтали вправо на $\frac{2}{3}$.

СИФ для кривой Коха:

$$\begin{aligned} ShowCoha = & With[\{D = DiagonalMatrix[\{1, 1\}/3]\}, IFS[\{AffineTransform[\{D\}], \\ & AffineTransform[\{RotationMatrix[60Degree]/3, \{1/3, 0\}\}], \\ & AffineTransform[\{RotationMatrix[-60Degree]/3, \{1/2, Sqrt[3]/6\}\}], \\ & AffineTrasform[\{D, \{2/3, 0\}\}]\}]; \end{aligned} \quad (4.7)$$

а) Построение кривой Коха рандомизированным методом (Рисунок 4.1), используя функции (4.1), (4.2). Случайным образом выбираются 100 точек в двумерном пространстве в диапазоне от 0 до 1. К этим точкам применяется СИФ (4.7). Результатом будут точки, переведенные с помощью функции N в числовое значение. Затем эти значения изображаются в виде крошечных точек. Такая процедура повторяется n раз.

$$Table[IFSPlot[N@SnowCoha, RandomReal[1, \{100, 2\}], n], \{n, 0, 6\}]$$

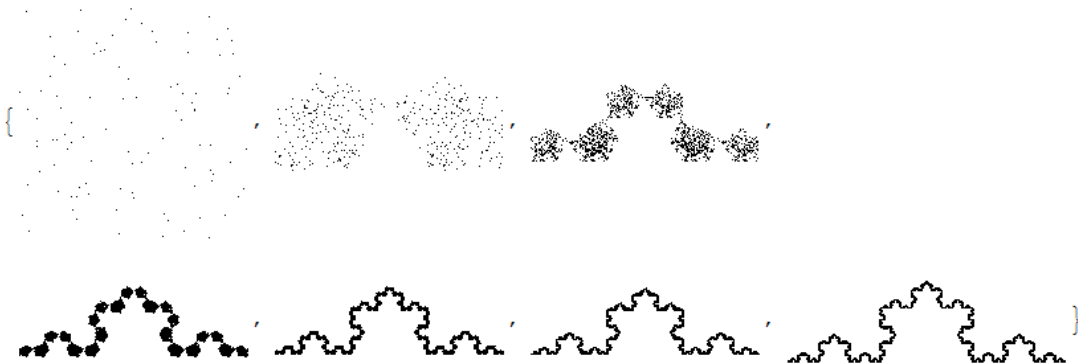


Рисунок 4.1 – Кривая Коха, построенная методом случайных итераций

б) Построим кривую Коха тем же методом, но немного другим способом, используя функции (4.3), (4.4). Он заключается в увеличении количества точек в списке и в применении СИФ (4.7) не к этому списку, а к его подмножеству. После чего результат данного алгоритма строится и затем повторяется n раз (Рисунок 4.2).

`Table[IFSPlot[RandomIFS[N@SierpinskiGasket, 103], RandomReal[1, {103, 2}], n], {n, 0, 5}]`

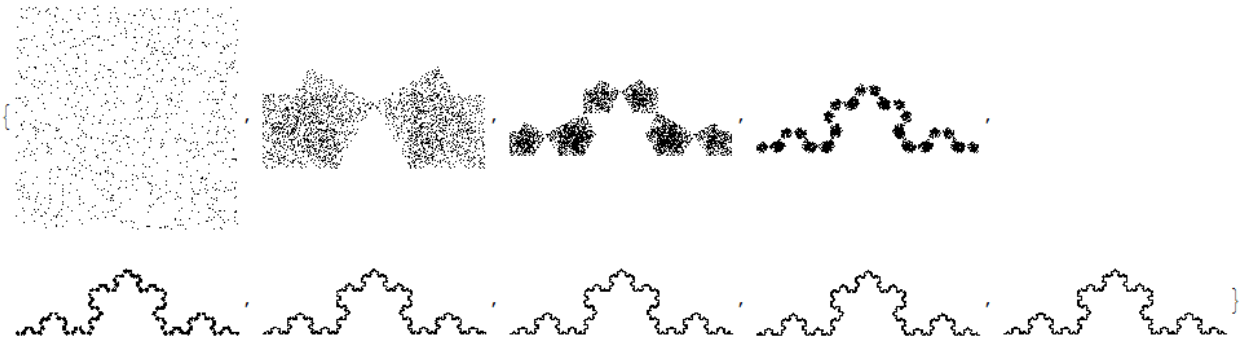


Рисунок 4.2 - Кривая Коха, построенная методом случайных итерация с увеличением количества точек

в) Построение кривой Коха, используя детерминированный метод, используя функции (4.5), (4.6). Разберемся с локальными переменными: `prim = First[g]=Rectangle[]; h = Head[g]=Graphics`. Получаем, что к прямоугольнику применяется СИФ (4.7), а результаты изображаются графически (Рисунок 4.3). Данный алгоритм повторяется n раз, причем для получения следующего изображения, используется результат предыдущего.

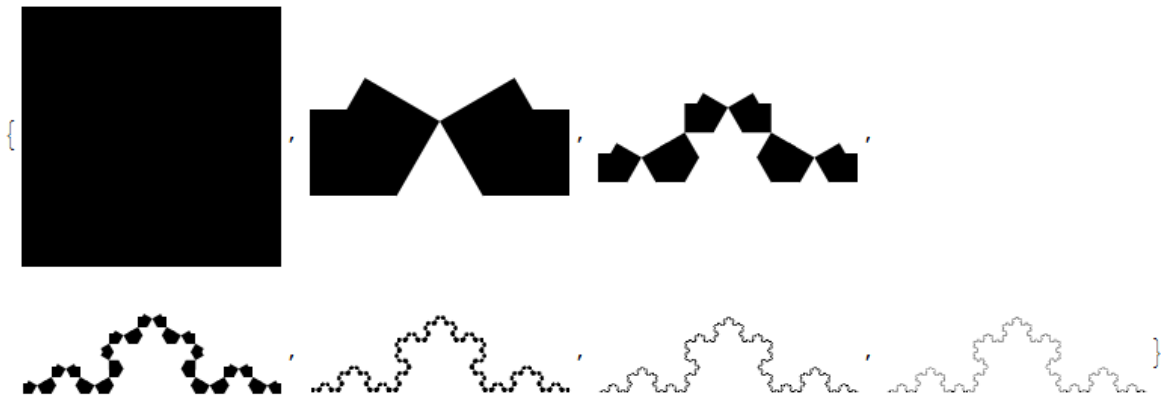


Рисунок 4.3 - Кривая Коха, построенная детерминированным методом

4.3 Салфетка и ковер Серпинского

Зададим систему итерируемых функций для салфетки Серпинского:

`AffineTransform[{D}]` - сжатие в 2 раза.

$AffineTransform[\{D, \{1/2, 0\}\}]$ - сжатие в 2 раза и сдвиг по горизонтали на $\frac{1}{2}$.

$AffineTransform[\{D, \{0.25, 0.433\}\}]$ - сжатие в 2 раза и перенос на вектор $\frac{1+\sqrt{3}i}{4}$.

$SierpinskiGasket = With[\{D = DiagonalMatrix[\{1, 1\}/2\}], IFS[\{AffineTransform[\{D\}], AffineTransform[\{D, \{1/2, 0\}\}], AffineTransform[\{D, \{0.25, 0.433\}\}]\}];$ (4.8)

а) Построим салфетку Серпинского рандомизированным методом (Рисунок 4.4), используя функции (4.1), (4.2), (4.8).

$Table[IFSPlot[N@SierpinskiGasket, RandomReal[1, \{100, 2\}], n], \{n, 0, 5\}]$

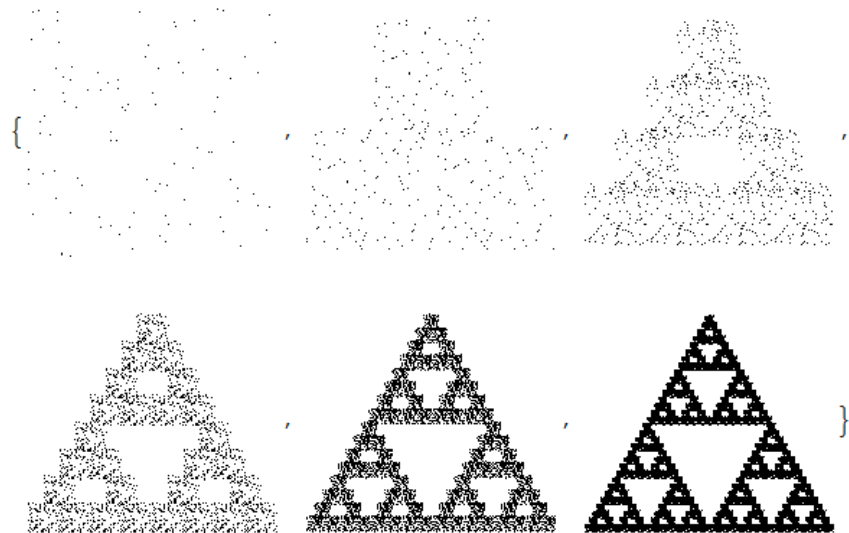


Рисунок 4.4 - Салфетка Серпинского, построенная рандомизированным методом

б) Построение салфетки Серпинского (Рисунок 4.5), используя функции (4.3), (4.4) и (4.8).

$Table[IFSPlot[RandomIFS[N@SierpinskiGasket, 10^3], RandomReal[1, \{10^3, 2\}], n], \{n, 0, 5\}]$

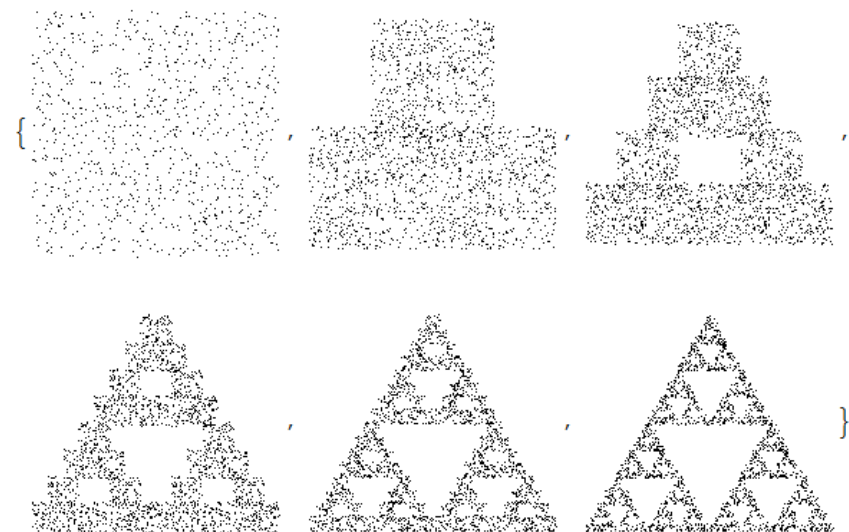


Рисунок 4.5 - Салфетка Серпинского, построенная рандомизированным методом с увеличением количества точек

в) Построение салфетки Серпинского (Рисунок 4.6), используя функции детерминированного метода (4.5), (4.6) и (4.8).

```
Table[TransformIFS[Graphics[Rectangle[]], SierpinskiGasket, n], {n, 0, 6}]
```

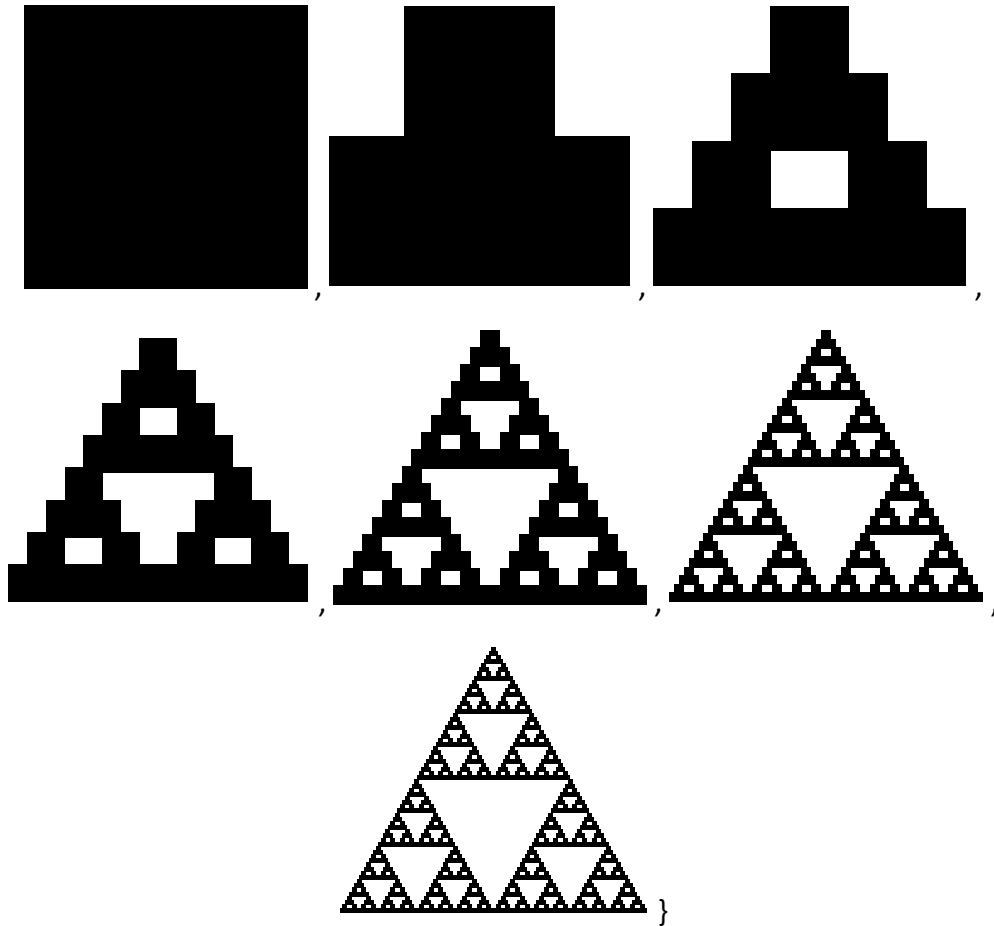


Рисунок 4.6 - Салфетка Серпинского, построенная детерминированным методом

Зададим систему итерируемых функций для ковра Серпинского:

$AffineTransform[\{D\}]$ - сжатие в 3 раза.

$AffineTransform[\{D, \{0, 1/3\}\}]$ - сжатие в 3 раза, перенос по вертикальной оси на $\frac{1}{3}$.

$AffineTransform[\{D, \{0, 2/3\}\}]$ - сжатие в 3 раза, перенос по вертикальной оси на $\frac{2}{3}$.

$AffineTransform[\{D, \{1/3, 0\}\}]$ - сжатие в 3 раза, перенос вдоль горизонтальной оси на $\frac{1}{3}$.

$AffineTransform[\{D, \{1/3, 2/3\}\}]$ - сжатие в 3 раза, перенос на вектор $\frac{1+2i}{3}$.

$AffineTransform[\{D, \{2/3, 0\}\}]$ - сжатие в 3 раза, перенос вдоль горизонтальной оси на $\frac{2}{3}$.

$AffineTransform[\{D, \{2/3, 1/3\}\}]$ - сжатие в 3 раза, перенос на вектор $\frac{2+i}{3}$.

$AffineTransform[\{D, \{2/3, 2/3\}\}]$ - сжатие в 3 раза, перенос на вектор $\frac{2+2i}{3}$.

```
SierpinskiCarpet = With[{D = DiagonalMatrix[{1, 1}/3]},
```

```
IFS[{AffineTransform[{D}], AffineTransform[{D, {0, 1/3}}],
```

$AffineTransform[\{D, \{0, 2/3\}\}], AffineTransform[\{D, \{1/3, 0\}\},$ (4.9)
 $AffineTransform[\{D, \{1/3, 2/3\}\}], AffineTransform[\{D, \{2/3, 0\}\},$
 $AffineTransform[\{D, \{2/3, 1/3\}\}], AffineTransform[\{D, \{2/3, 2/3\}\}]]];$

а) Ковер Серпинского, построенный рандомизированным методом с помощью функций (4.1), (4.2) и (4.9). Результат изображен на Рисунке 4.7.

$Table[IFSPlot[N@SierpinskiCarpet, randomReal[1, \{100, 2\}], n], \{n, 0, 5\}]$

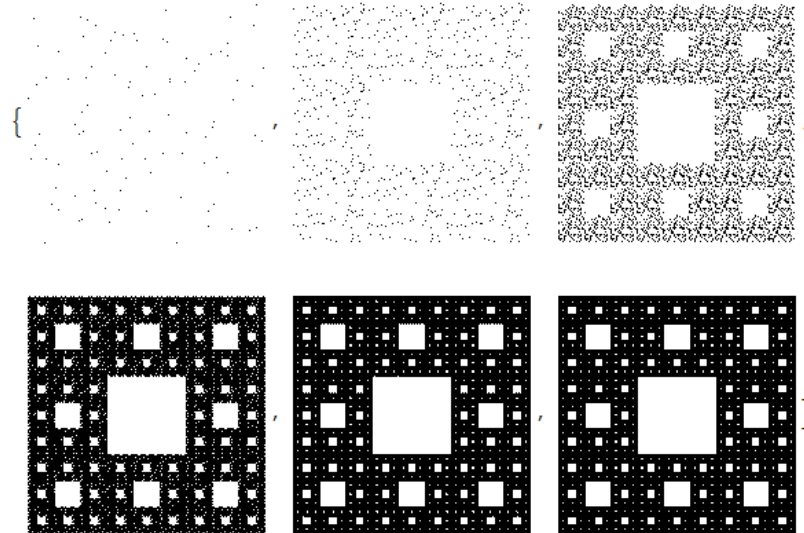


Рисунок 4.7 - Ковер Серпинского, построенный рандомизированным методом

б) Построение ковра Серпинского методом случайных итераций (Рисунок 4.8). Используются функции (4.3), (4.4) и (4.9).

$Table[IFSPlot[RandomIFS[N@SierpinskiCarpet, 10^3], RandomReal[1, \{1000, 2\}], n], \{n, 0, 7\}]$

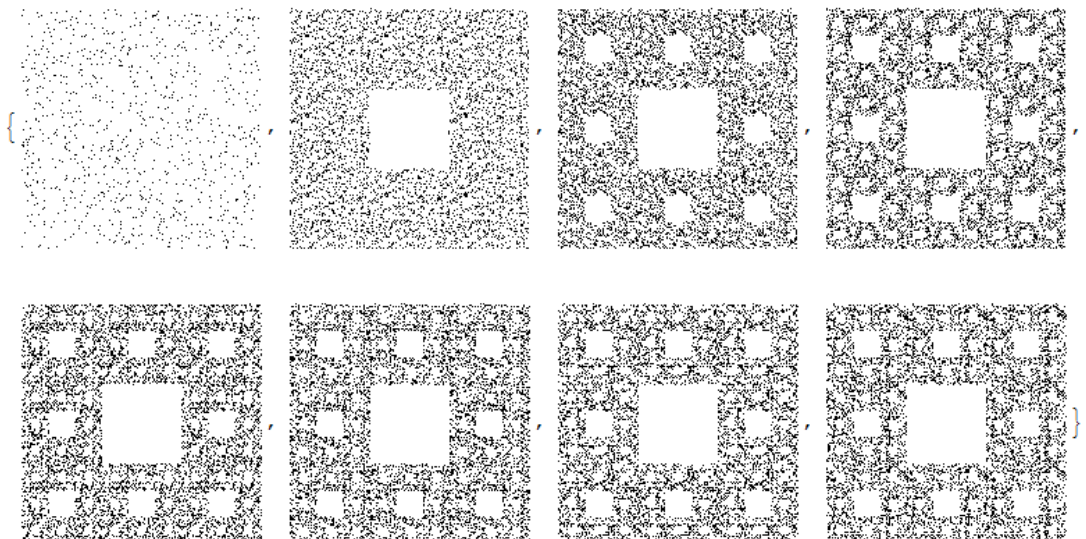


Рисунок 4.8 - Ковер Серпинского методом случайных итераций с увеличенным количеством точек

в) Построение ковра Серпинского, используя детерминированный метод (Рисунок 4.9), а именно функции (4.5), (4.6) и (4.9).

```
Table[TransformIFS[Graphics[Rectangle[]], SierpinskiCarpet, n], {n, 0, 4}]
```

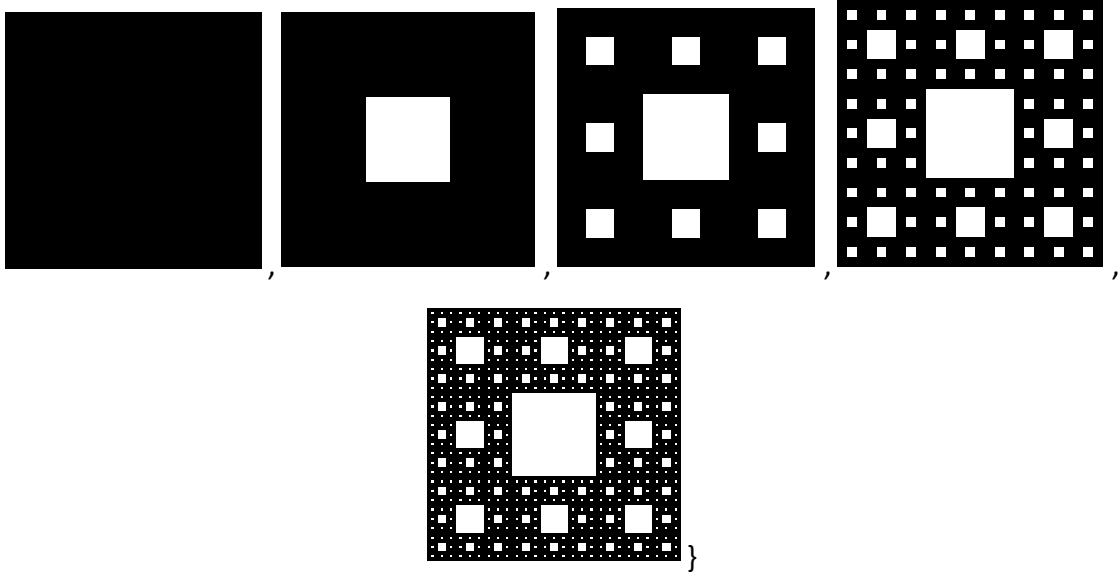


Рисунок 4.9 - Ковер Серпинского, построенный детерминированным методом

4.4 Модификации салфетки Серпинского

Пример1.

Зададим систему итерируемых функций следующим образом.

$AffineTransform[\{D\}]$ - изменение масштаба в $\frac{2}{3}$ раза.

$AffineTransform[\{D, \{1/2, 0\}\}]$ - изменение масштаба в $\frac{2}{3}$ раза, перенос вдоль горизонтальной оси на $1/2$.

$AffineTransform[\{D, \{0.25, 0.433\}\}]$ - изменение масштаба в $\frac{2}{3}$ раза, перенос на вектор $\frac{1+\sqrt{3}i}{4}$.

```
SierpinskiGasket2 = With[{D = DiagonalMatrix[{2, 2}/3]},  
IFS[{AffineTransform[{D}], AffineTransform[{D, {1/2, 0}]},  
AffineTransform[{D, {0.25, 0.433}}]}]]];
```

(4.10)

Для построения зададим правильный треугольник и воспользуемся детерминированным методом. Изобразим результат действия описанной системы итерируемых функций (Рисунок 4.10).

```
Table[TransformIFS[Graphics[Polygon[{{0, 0}, {1/2, Sqrt[3]/2}, {1, 0}}]],  
SierpinskiGasket2, n], {n, 0, 7}]
```

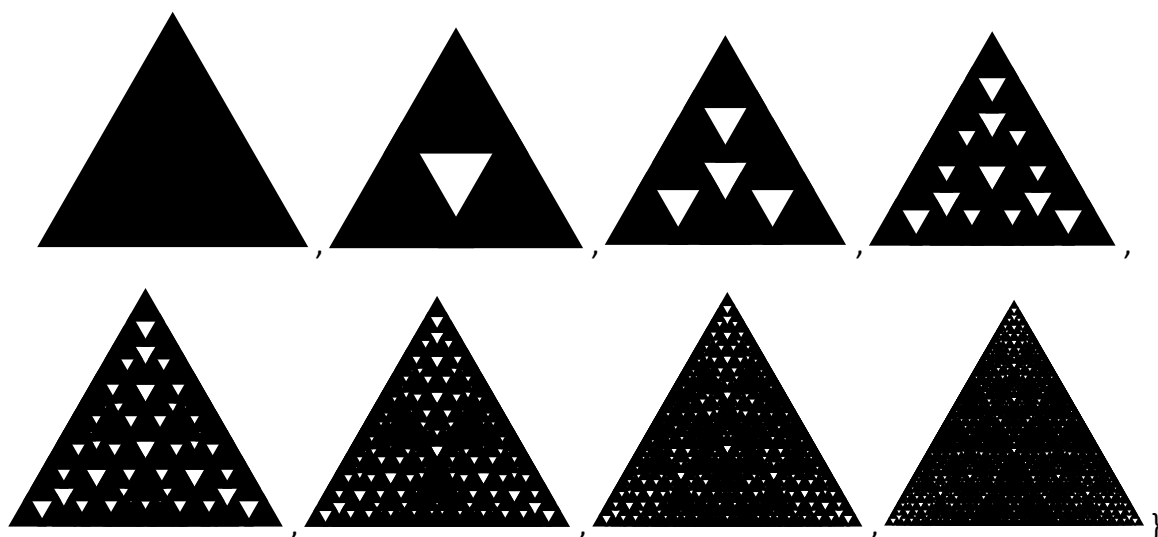


Рисунок 4.10 - Изображение примера 1

Пример 2.

Зададим систему итерируемых функций для построения салфетки Серпинского, у которой сжатие происходит не в 2 раза, а в 3 (Рисунок 4.11)

$AffineTransform[\{D\}]$ - сжатие в 3 раза.

$AffineTransform[\{D, \{1/2, 0\}\}]$ - сжатие в 3 раза и перенос вдоль горизонтальной оси на $1/2$.

$AffineTransform[\{D, \{0.25, 0.433\}\}]$ - сжатие в 3 раза и перенос на вектор $\frac{1+\sqrt{3}i}{4}$.

$SierpinskiGasket3 = With[\{D = DiagonalMatrix[\{1, 1\}/3\},$
 $IFS[\{AffineTransform[\{D\}], AffineTransform[\{D, \{1/2, 0\}\}],$
 $AffineTransform[\{D, \{0.25, 0.433\}\}]\};$ (4.11)

Для построения воспользуемся «игрой в хаос», а именно функциями (4.3) и (4.4).

$Table[IFSPlot[RandomIFS[N@SierpinskiGasket3, 10^3], RandomReal[1, \{10^3, 2\}], n], \{n, 0, 5\}]$

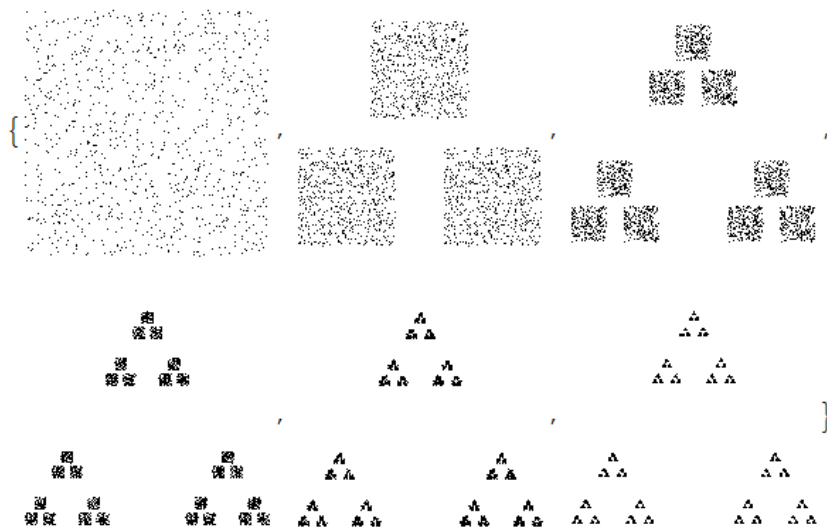


Рисунок 4.11 - Изображение примера 2

Пример 3.

Применим систему итерированных функций (4.7) для произвольного треугольника (Рисунок 4.13). Зададим этот треугольник (Рисунок 4.12).

```
Graphics [Polygon[{0,0},{0.5,2},{2.5,0}]]
```

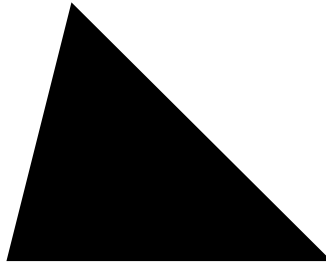


Рисунок 4.12 - Произвольный треугольник

Система итерируемых функций для данного случая подкорректируем в соответствии координатам точек вершин.

$AffineTransform[\{D\}]$ - сжатие в 2 раза.

$AffineTransform[\{D, \{5/4, 0\}\}]$ - сжатие в 2 раза и перенос вдоль горизонтальной оси на 5/4.

$AffineTransform[\{D, \{1/2, 3/4\}\}]$ - сжатие в 2 раза, перенос на вектор $\frac{2+3i}{4}$.

```
SierpinskiGasket = With[{D = DiagonalMatrix[{1,1}/2]}, IFS[{AffineTransform[{D}],  
AffineTransform[{D, {5/4, 0}}], AffineTransform[{D, {1/2, 3/4}}]}];
```

 (4.12)

Для построения воспользуемся функциями (4.3) и (4.4).

```
Table[IFSPLOT[RandomIFS[N@SierpinskiGasket, 103], RandomReal[1, {103, 2}], n], {n, 0, 6}]
```

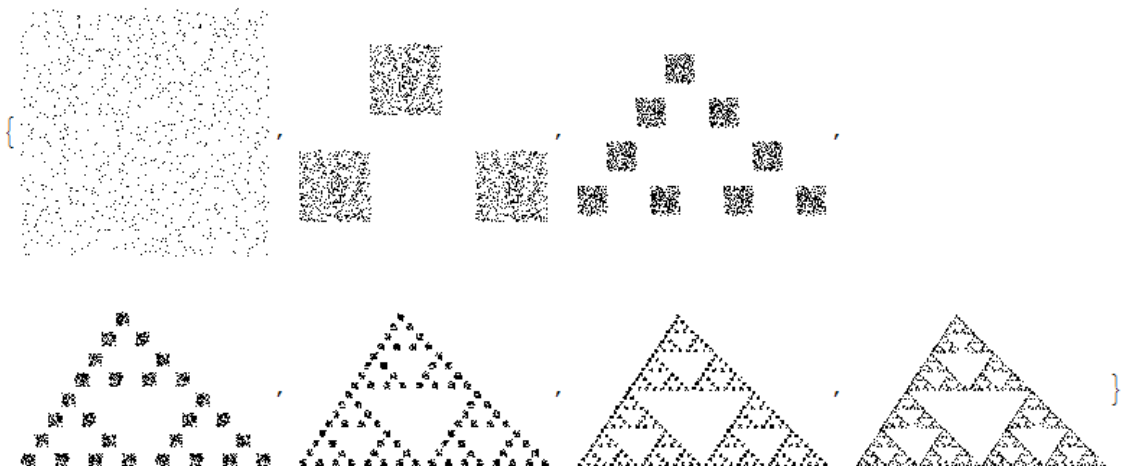


Рисунок 4.13 - Салфетка Серпинского для произвольного треугольника

Пример 4.

Построим СИФ для шестиугольника Серпинского (Рисунок 4.15). Для начала зададим правильный шестиугольник (Рисунок 4.14).

`Graphics[Polygon[{{0, 1/2}, {0, 3/2}, {Sqrt[3]/2, 2}, {Sqrt[3], 3/2}, {Sqrt[3], 1/2}, {Sqrt[3]/2, 0}]]]`

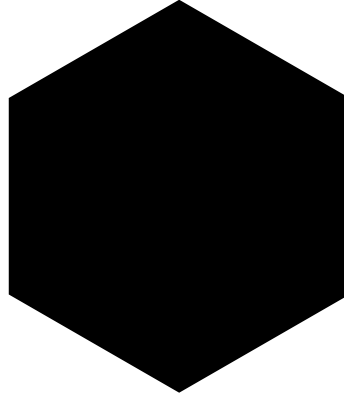


Рисунок 4.14 - Правильный шестиугольник

Система итерируемых функций для шестиугольника:

$AffineTransform[\{D, \{Sqrt[3]/6, 0\}\}]$ - сжатие в 3 раза, перенос вдоль горизонтальной оси на $\frac{\sqrt{3}}{6}$.

$AffineTransform[\{D, \{0, 1/6\}\}]$ - сжатие в 3 раза, перенос по вертикальной оси на $1/6$.

$AffineTransform[\{D, \{0, 1/2\}\}]$ - сжатие в 3 раза, перенос по вертикальной оси на $1/2$.

$AffineTransform[\{D, \{Sqrt[3]/6, 2/3\}\}]$ - сжатие в 3 раза, перенос на вектор $\frac{\sqrt{3}+4i}{6}$.

$AffineTransform[\{D, \{Sqrt[3]/3, 1/2\}\}]$ - сжатие в 3 раза, перенос на вектор $\frac{\sqrt{3}+3i}{6}$.

$AffineTransform[\{D, \{Sqrt[3]/3, 1/6\}\}]$ - сжатие в 3 раза, перенос на вектор $\frac{\sqrt{3}+i}{6}$.

$SierpinskiSix = With[\{D = DiagonalMatrix[\{1, 1\}/3\}],$

$IFS[\{AffineTransform[\{D, \{Sqrt[3]/6, 0\}\}], AffineTransform[\{D, \{0, 1/6\}\}],$ (4.13)

$AffineTransform[\{D, \{0, 1/2\}\}], AffineTransform[\{D, \{Sqrt[3]/6, 2/3\}\}],$

$AffineTransform[\{D, \{Sqrt[3]/3, 1/2\}\}], AffineTransform[\{D, \{Sqrt[3]/3, 1/6\}\}]]];$

Для получения изображения воспользуемся «игрой в хаос», т.е. функциями (4.3), (4.4).

`Table[IFSPLOT[RandomIFS[N@SierpinskiSix, 104], RandomReal[1, {103, 2}], n], {n, 0, 6}]`

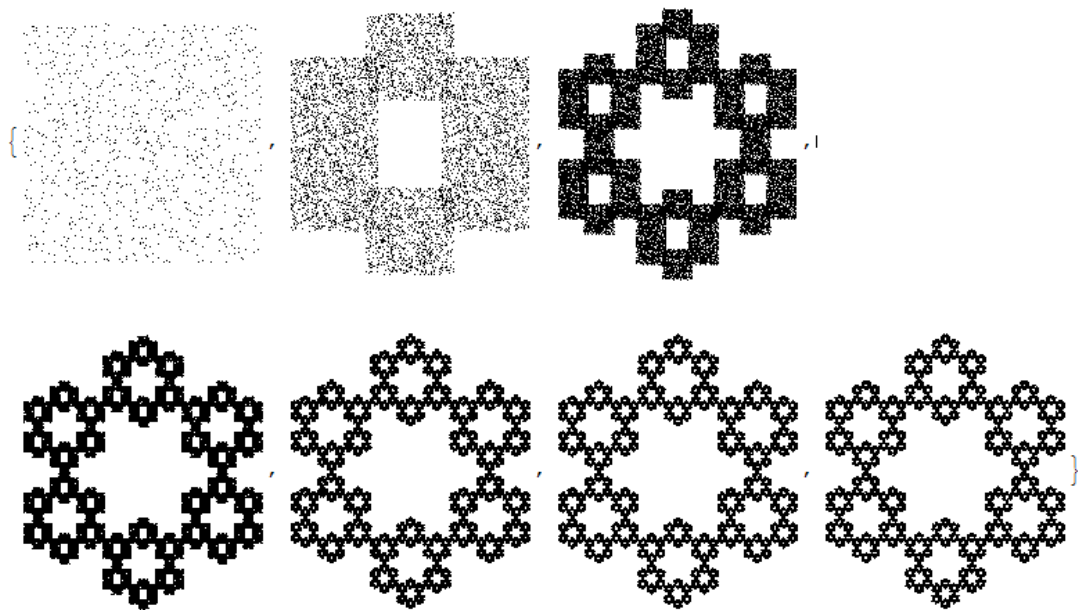


Рисунок 4.15 - Шестиугольник Серпинского

4.5 Лист папоротника

Составим систему итерируемых функций для листа папоротника (Рисунок 4.16) на основе Таблицы 1.

$$\begin{aligned}
 \text{PaporoTnic} = & \text{IFS}[\{\text{AffineTransform}[\{\{0,0\}, \{0,0.16\}\}], \\
 & \text{AffineTransform}[\{\{0.85,0.04\}, \{-0.04,0.85\}\}, \{0,1.6\}\}], \\
 & \text{AffineTransform}[\{\{0.2, -0.26\}, \{0.23,0.22\}\}, \{0,1.6\}\}], \\
 & \text{AffineTransform}[\{\{-0.15,0.28\}, \{0.26,0.24\}\}, \{0,0.44\}\}]\};
 \end{aligned} \tag{4.14}$$

а) Построение с помощью функций (4.1), (4.2).

```
Table[IFSPlot[N@PaporoTnic, RandomReal[1, {100,2}], n], {n, 0,7}]
```



Рисунок 4.16 - Лист папоротника, построенный рандомизированным методом

б) Построение листа папоротника детерминированным методом (Рисунок 4.17) с помощью функций (4.5), (4.6), (4.14).

```
Table[TransformIFS[Graphics[Polygon[\{\{0,0\}, \{1/2, Sqrt[3]/2\}, \{1,0\}\}],
PaporoTnic, n], {n, 0,8}]
```



Рисунок 4.17 - Лист папоротника детерминированным методом
Модификация 1 (Рисунок 4.18).

Система итерируемых функций соответствующая Таблице Б.2.

$$\begin{aligned}
 \text{Paprotnic1} = & \text{IFS}[\{\text{AffineTransform}[\{\{0.7, 0\}, \{0, 0.7\}\}, \{0.1496, 0.2962\}], \\
 & \text{AffineTransform}[\{\{0.1, -0.433\}, \{0.1732, 0.25\}\}, \{0.4478, 0.0014\}], \\
 & \text{AffineTransform}[\{\{0.1, 0.433\}, \{-0.1732, 0.25\}\}, \{0.4445, 0.1559\}], \\
 & \text{AffineTransform}[\{\{0, 0\}, \{0, 0.3\}\}, \{0.4987, 0.007\}]\};
 \end{aligned} \tag{4.15}$$

а) Построим первую модификацию рандомизированным методом, используя функции (4.1) и (4.2).

$$\text{Table}[\text{IFSPlot}[N@\text{Paprotnic1}, \text{RandomReal}[1, \{100, 2\}], n], \{n, 0, 7\}]$$

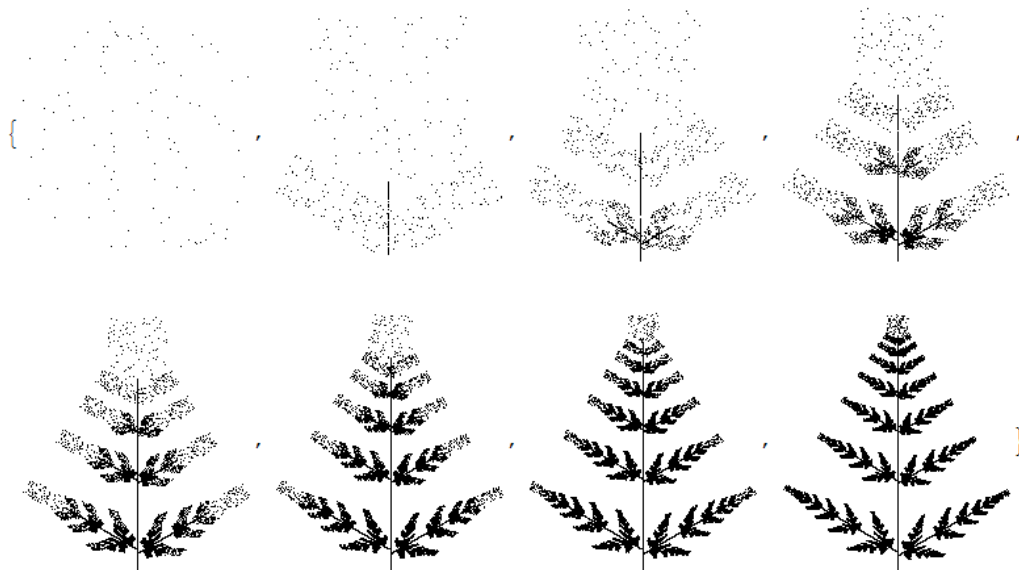


Рисунок 4.18 - Модификация 1 листа папоротника, построенная рандомизированным методом

б) Построение модификации 1 листа папоротника детерминированным методом (Рисунок 4.19) с помощью функций (4.5), (4.6).

$$\begin{aligned}
 & \text{Table}[\text{TransformIFS}[\text{Graphics}[\text{Polygon}[\{\{0, 0\}, \{1/2, \text{Sqrt}[3]/2\}, \{1, 0\}\}]], \\
 & \text{Paprotnic1}, n], \{n, 0, 8\}]
 \end{aligned}$$



Рисунок 4.19 - Модификация 1, построенная детерминированным методом
Модификация 2 (Рисунок 4.20).

Сделаем систему итерируемых функций для елки на основе Таблицы 8.

$$\begin{aligned}
 Tree = IFS[& \{AffineTransform[\{\{\{0.1,0\}, \{0,0.16\}\}, \{0,0\}\}], \\
 & AffineTransform[\{\{\{0.85,0\}, \{0,0.85\}\}, \{0,1.6\}\}], \\
 & AffineTransform[\{\{\{-0.1667, -0.2887\}, \{0.2887, -0.1667\}\}, \{0,1.6\}\}], \\
 & AffineTransform[\{\{\{-0.1667, 0.2887\}, \{-0.2887, -0.1667\}\}, \{0,1.6\}\}]\};
 \end{aligned} \tag{4.16}$$

Построим данную модификацию рандомизированным методом, используя функции (4.1) и (4.2).

$$Table[IFSPlot[N@Paporotnic1, RandomReal[1, \{100,2\}], n], \{n, 0,7\}]$$

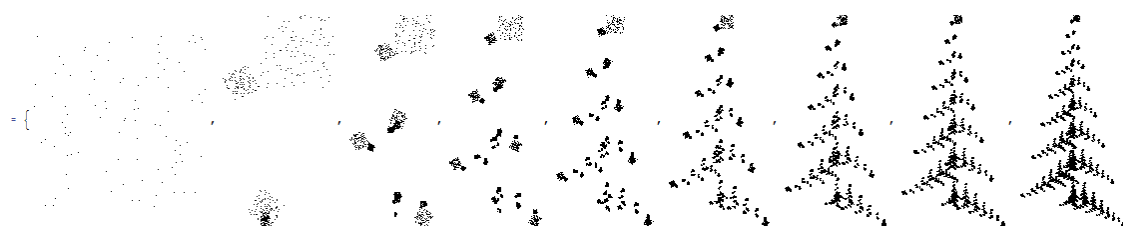


Рисунок 4.20 – Построение елки рандомизированным методом

4.6 Дракон Хартера-Хейтуэя

Зададим систему итерируемых функций для дракона Хартера-Хейтуэя.

$$\begin{aligned}
 HighwayDragon = IFS[& \{AffineTransform[\{RotationMatrix[-45Degree]/ \\
 & Sqrt[2], \{-1,0\}\}], \\
 & AffineTransform[\{RotationMatrix[-135Degree]/Sqrt[2], \{0,1\}\}]\};
 \end{aligned} \tag{4.17}$$

а) Построение рандомизированным методом (Рисунок 4.21), используя функции (4.1), (4.2).

`Table[IFSPlot[N@HeighwayDragon, RandomReal[1, {100, 2}], n], {n, 0, 15}]`

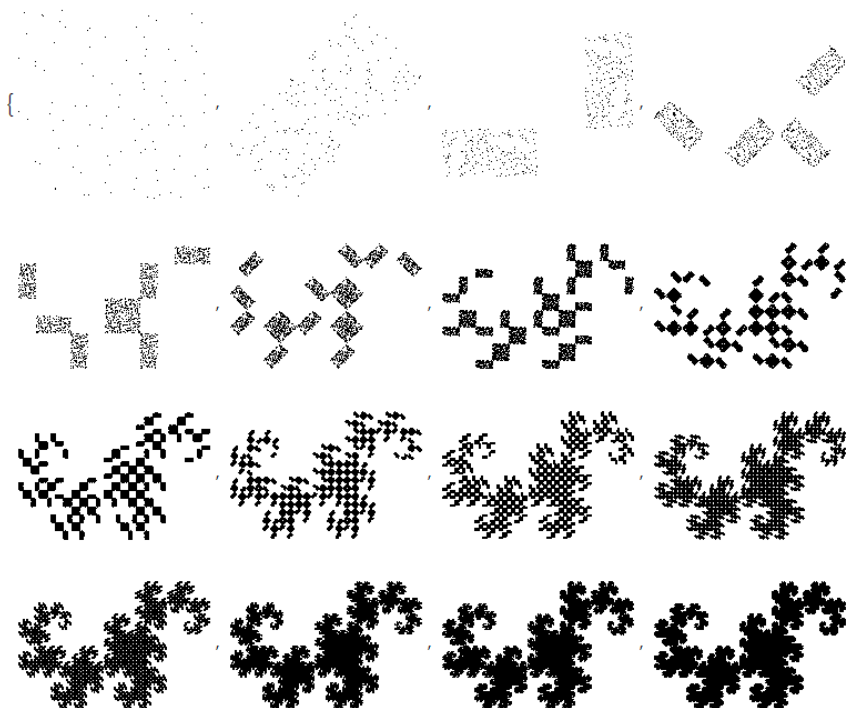


Рисунок 4.21 – Дракон Хартера-Хейтуэя рандомизированным методом

б) Для построения воспользуемся функциями (4.3) и (4.4). Результат изображен на рисунке 4.22.

`Table[IFSPlot[RandomIFS[N@HeighwayDragon, 103], RandomReal[1, {103, 2}], n], {n, 0, 15}]`

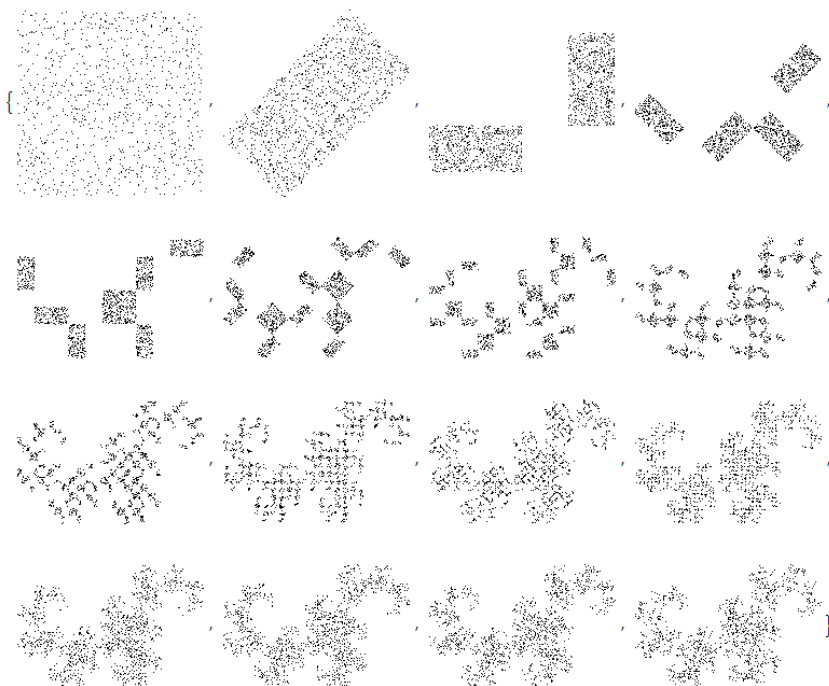


Рисунок 4.22 – Дракон Хартера-Хейтуэя, построенный методом случайных итераций с увеличением количества точек

в) Построение дракона детерминированным методом (Рисунок 4.23) с помощью функций (4.5), (4.6).

```
Table[TransformIFS[Graphics[Polygon[{{0,0},{1/2,Sqrt[3]/2},{1,0}}]],  
HeighwayDragon,n],{n,0,15}]
```

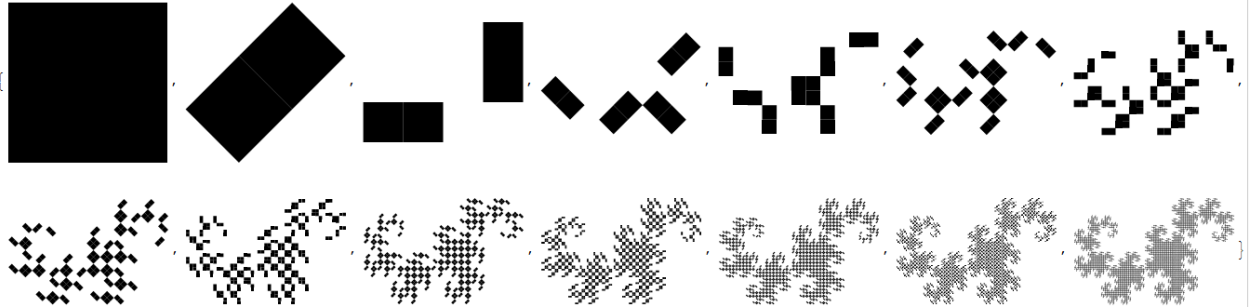


Рисунок 4.23 – Дракон Хартера-Хейтуэя, построенный детерминированным методом

Модификация 1.

Построим двойного дракона Хартера-Хейтуэя. Зададим для него систему итерируемых функций.

```
HeighwayDragon1 =  
IFS[{AffineTransform[{RotationMatrix[-45Degree]/Sqrt[2],{-1,0}}],  
AffineTransform[{RotationMatrix[-45Degree]/Sqrt[2],{0,1}}]}];
```

(4.18)

а) Построение рандомизированным методом (Рисунок 4.24), используя функции (4.1), (4.2), (4.18).

```
Table[IFSPlot[N@HeighwayDragon1,RandomReal[1,{100,2}],n],{n,0,15}]
```

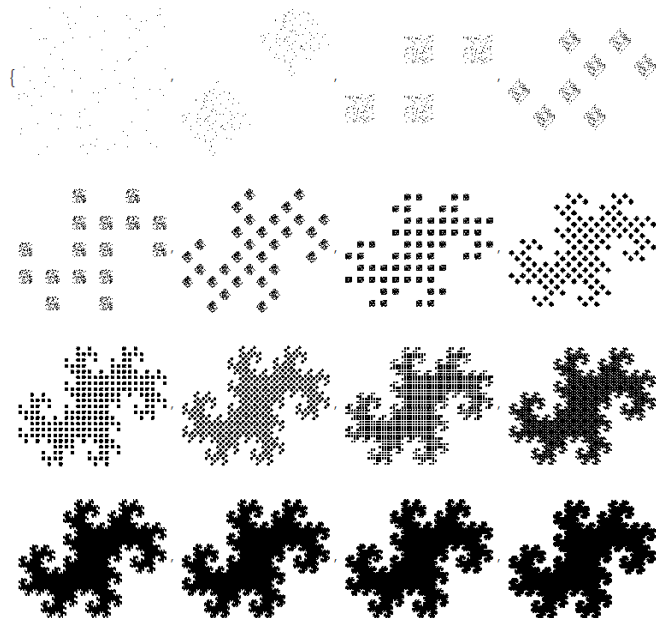


Рисунок 4.24 – Двойной дракон Хартера-Хейтуэя, построенный рандомизированным методом

б) Для построения воспользуемся функциями (4.3), (4.4) и (4.18). Результат изображен на Рисунке 4.25.

```
Table[IFSPlot[RandomIFS[N@HeighwayDragon1, 103], RandomReal[1, {103, 2}], n], {n, 0, 15}]
```

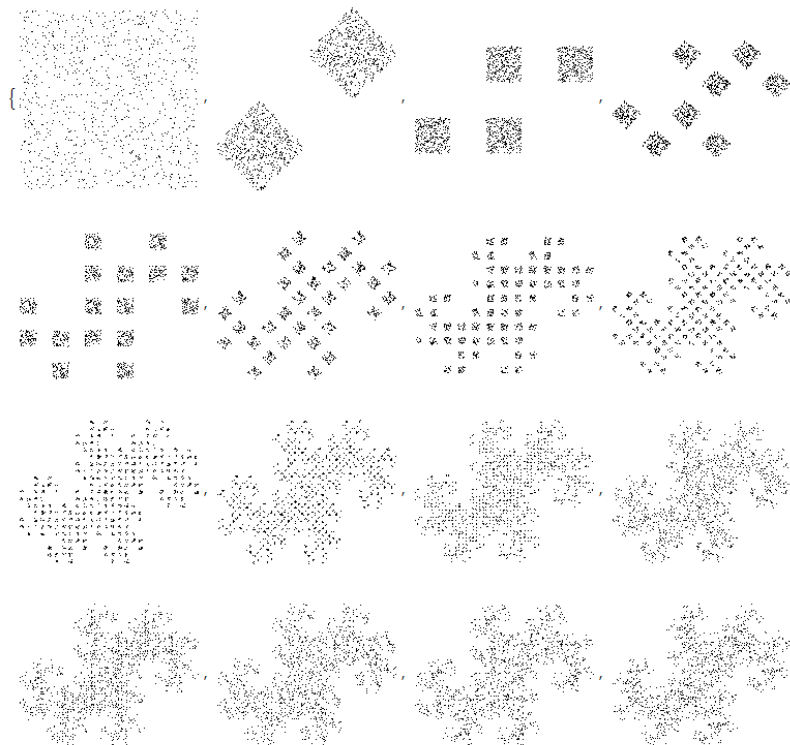


Рисунок 4.25 – Двойной дракон Хартера-Хейтуэя методом случайных итераций с увеличенным количеством точек

в) Построение двойного дракона детерминированным методом (Рисунок 4.26) с помощью функций (4.5), (4.6), (4.18).

```
Table[TransformIFS[Graphics[Polygon[{{0,0},{1/2,Sqrt[3]/2},{1,0}}]],  
HeighwayDragon1,n], {n, 0, 15}]
```

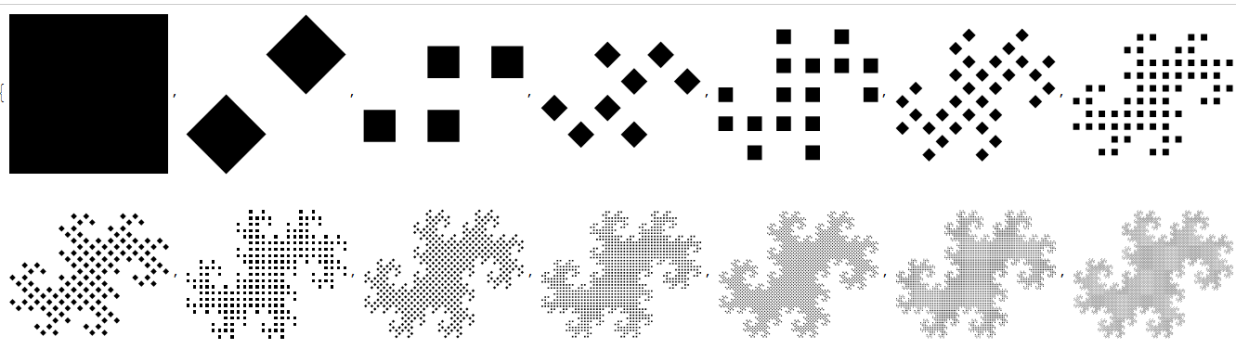


Рисунок 4.26 – Двойной дракон, полученный детерминированным методом

4.7 Примеры известных фракталов

Пример 1.

Зададим систему итерируемых функций для кристалла на основе таблицы Б.1.

$$\begin{aligned}
& Krystal = IFS[\{AffineTransform[\{\{0.255,0\}, \{0,0.255\}\}, \{0.3726,0.6714\}\}], \\
& AffineTransform[\{\{0.255,0\}, \{0,0.255\}\}, \{0.1146,0.2232\}\}], \\
& AffineTransform[\{\{0.255,0\}, \{0,0.255\}\}, \{0.6306,0.2232\}\}], \\
& AffineTransform[\{\{0.37, -0.642\}, \{0.642,0.37\}\}, \{0.6356, -0.0061\}\}]\};
\end{aligned} \tag{4.19}$$

а) Построение рандомизированным методом (Рисунок 4.27) на основе функций (4.1), (4.2), (4.19).

`Table[IFSPlot[N@Krystal, RandomReal[1, {100,2}], n], {n, 0,7}]`

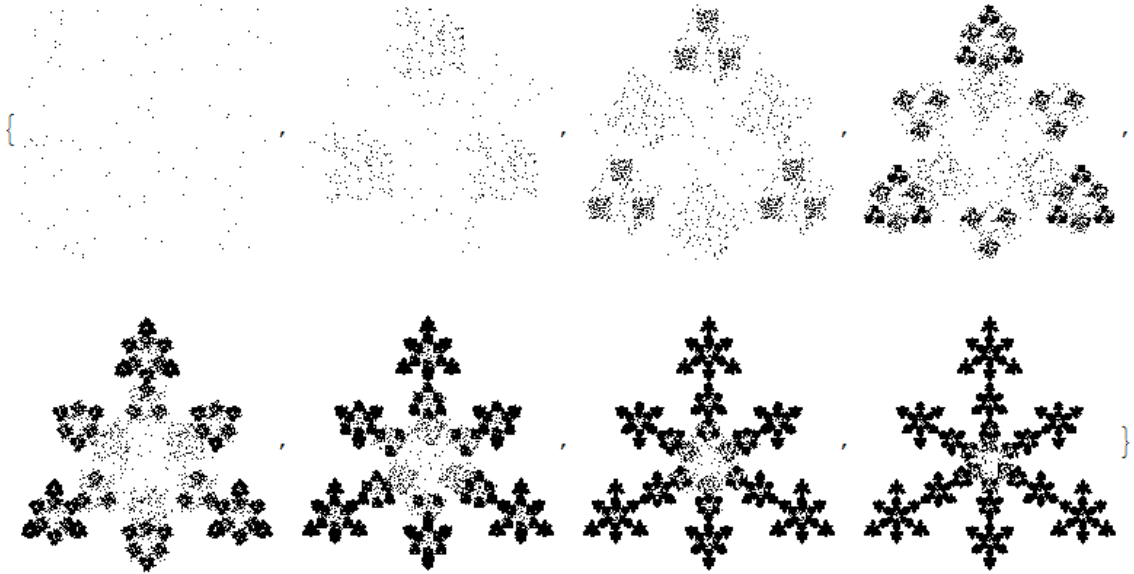


Рисунок 4.27 – Кристалл рандомизированным методом

б) Построение методом случайных итераций с увеличением количества точек, используя функции (4.3), (4.4) и (4.19). Результат изображен на Рисунке 4.28.

`Table[IFSPlot[RandomIFS[N@Krystal, 103], RandomReal[1, {103, 2}], n], {n, 0,8}]`

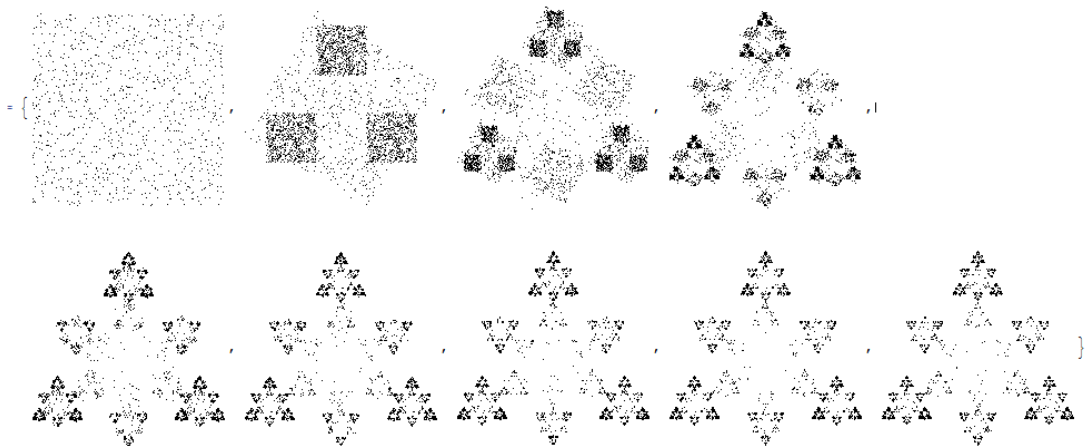


Рисунок 4.28 – Кристалл методом случайных итераций с увеличенным количеством точек

в) Построение кристалла детерминированным методом (Рисунок 4.29) с помощью функций (4.5), (4.6), (4.19).

```
Table[TransformIFS[Graphics[Polygon[{{0,0},{1/2,Sqrt[3]/2},{1,0}}]],  
Kristal,n],{n,0,8}]
```



Рисунок 4.29 – Кристалл детерминированным методом

Пример 2.

Зададим систему итерируемых функций для ковра А на основе таблицы Б.3.

```
KoverA = IFS[{AffineTransform[{{0.5,0},{0,-0.5}},{0.5,0.5}],  
AffineTransform[{{0,-0.5},{-0.5,0}},{0.5,0.5}],  
AffineTransform[{{-0.5,0},{0,-0.5}},{0.5,1}]}];
```

(4.20)

а) Построение рандомизированным методом (Рисунок 4.30) на основе функций (4.1), (4.2), (4.20).

```
Table[IFSPlot[N@KoverA,RandomReal[1,{100,2}],n],{n,0,6}]
```

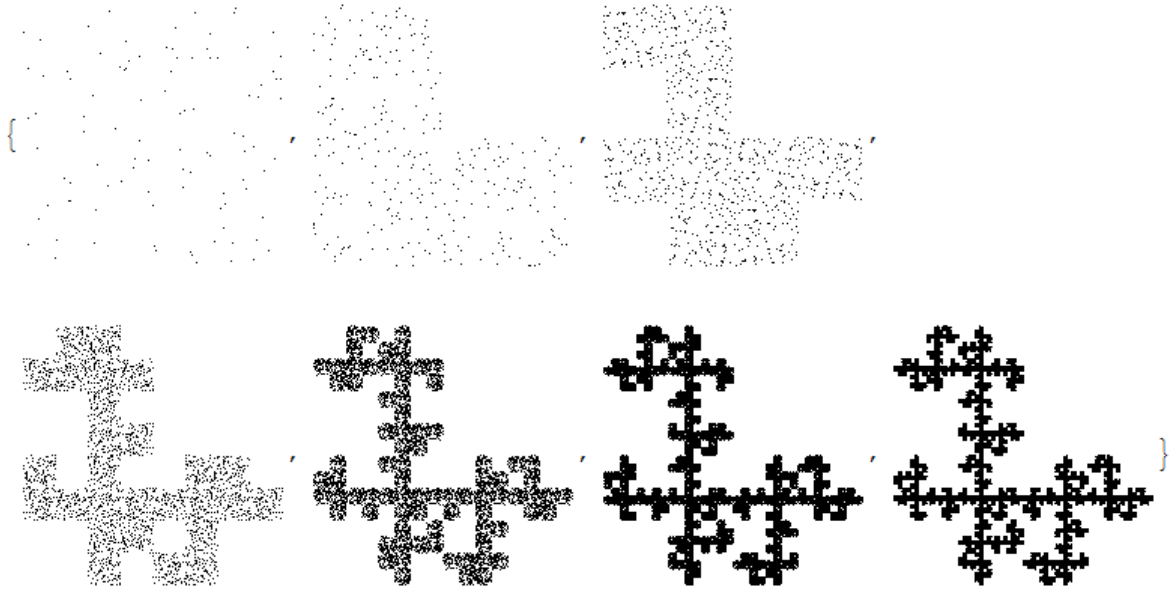


Рисунок 4.30 – Ковер А рандомизированным методом

б) Построение методом случайных итераций с увеличением количества точек, используя функции (4.3), (4.4) и (4.20). Результат изображен на Рисунке 4.31.

```
Table[IFSPlot[RandomIFS[N@KoverA,10^3],RandomReal[1,{10^3,2}],n],{n,0,10}]
```

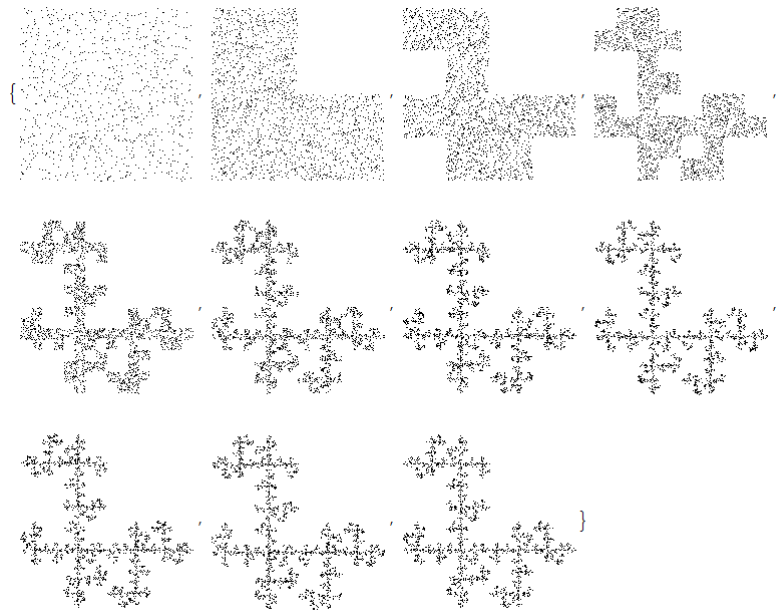


Рисунок 4.31 – Ковер А методом случайных итераций с увеличенным количеством точек

в) Построение ковра А детерминированным методом (Рисунок 4.32) с помощью функций (4.5), (4.6), (4.20).

```
Table[TransformIFS[Graphics[Polygon[{{0,0},{1/2,Sqrt[3]/2},{1,0}}]],
KoverA,n],{n,0,6}]
```

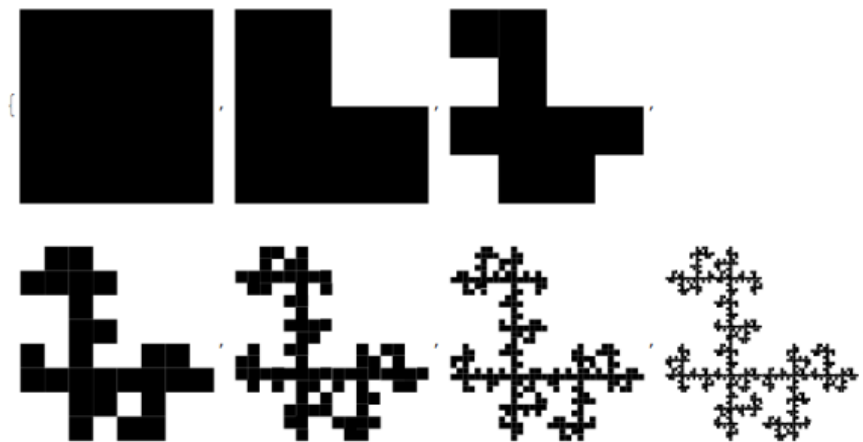


Рисунок 4.32 – Ковер А детерминированным методом

Пример 3.

Зададим систему итерируемых функций для ковра В на основе таблицы Б.4.

```
KoverB = IFS[{AffineTransform[{{0.5,0},{0,-0.5}},{0,1}],
AffineTransform[{{0,0.5},{0.5,0}},{0,0}],AffineTransform[{{0.5,0},{0,0.5}},{0.5,0}]}];
```

а) Построение рандомизированным методом (Рисунок 4.33) на основе функций (4.1), (4.2), (4.21).

```
Table[IFSPlot[N@KoverB,RandomReal[1,{100,2}],n],{n,0,6}]
```

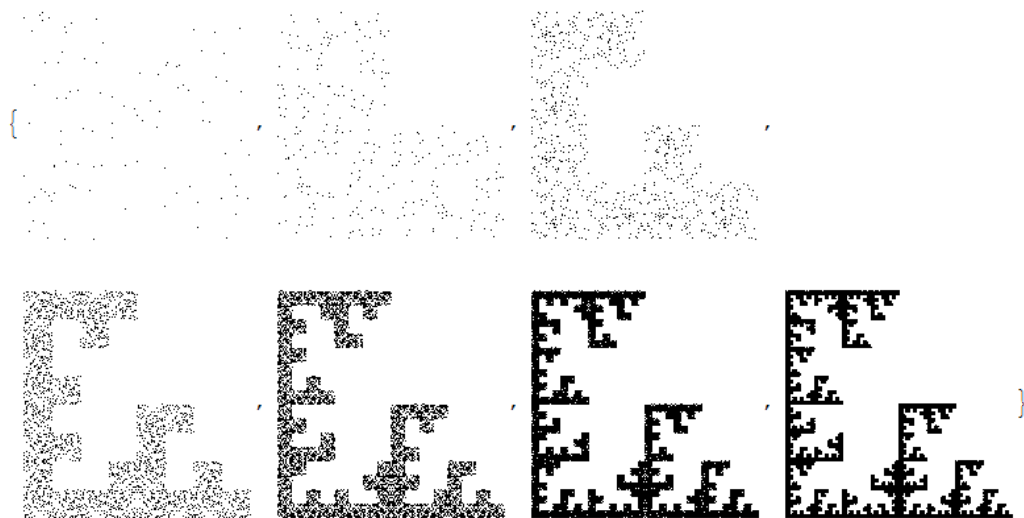


Рисунок 4.33 – Ковер В рандомизированным методом

б) Построение методом случайных итераций с увеличением количества точек, используя функции (4.3), (4.4), (4.21). Результат изображен на рисунке 4.34.

`Table[IFSPlot[RandomIFS[N@KoverB, 103], RandomReal[1, {103, 2}], n], {n, 0, 10}]`

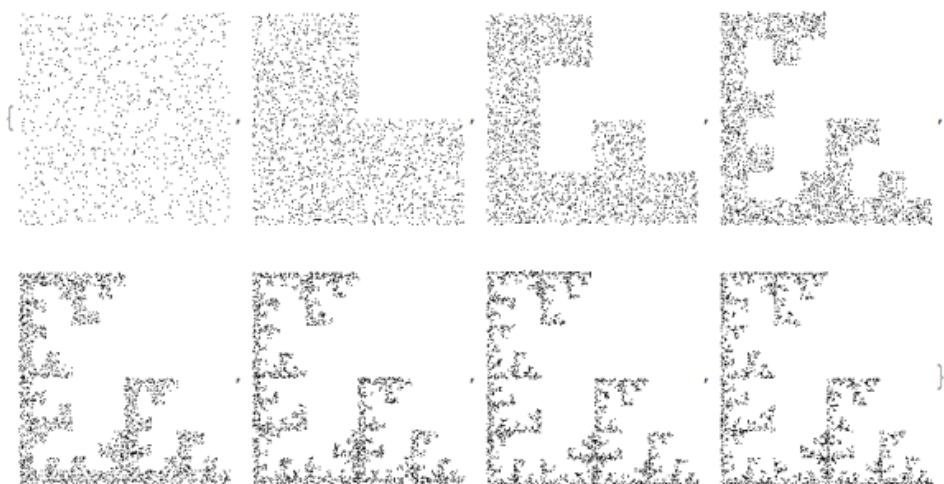


Рисунок 4.34 – Ковер В методом случайных итераций с увеличенным количеством точек

в) Построение ковра В детерминированным методом (Рисунок 4.35) с помощью функций (4.5), (4.6), (4.21).

`Table[TransformIFS[Graphics[Polygon[{{0,0},{1/2,Sqrt[3]/2},{1,0}}]],
KoverB, n], {n, 0, 6}]`

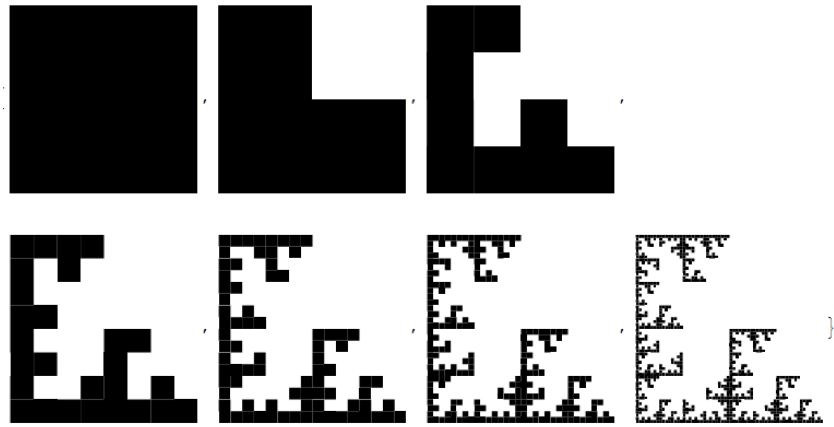


Рисунок 4.35 – Ковер В детерминированным методом

Пример 4.

Зададим систему итерируемых функций для дерева на основе таблицы Б.6.

$$\begin{aligned}
 Tree1 = IFS[& \{AffineTransform[\{\{0.195, -0.488\}, \{0.344, 0.443\}, \{0.4431, 0.2452\}\}], \\
 & AffineTransform[\{\{0.462, 0.414\}, \{-0.252, 0.361\}, \{0.2511, 0.5692\}\}], \\
 & AffineTransform[\{\{-0.058, -0.07\}, \{0.453, -0.111\}, \{0.5976, 0.0969\}\}], \\
 & AffineTransform[\{\{-0.035, 0.07\}, \{-0.469, 0.022\}, \{0.4884, 0.5069\}\}], \\
 & AffineTransform[\{\{-0.637, 0\}, \{0, 0.501\}, \{0.8562, 0.2513\}\}]\};
 \end{aligned} \tag{4.22}$$

Построение рандомизированным методом (Рисунок 4.36) на основе функций (4.1), (4.2), (4.22).

$$Table[IFSPlot[N@Tree1, RandomReal[1, \{100, 2\}], n], \{n, 0, 7\}]$$

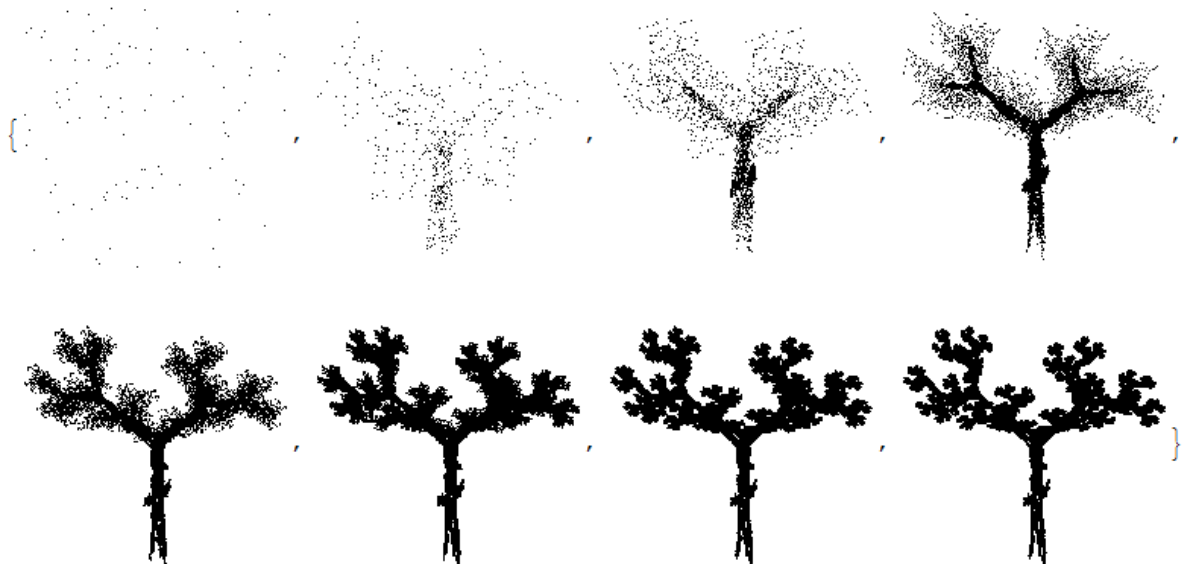


Рисунок 4.36 – Дерево рандомизированным методом

Пример 4.

Зададим систему итерируемых функций для листа на основе таблицы Б.5.

$$Lest = IFS[\{AffineTransform[\{\{0.4, -0.3733\}, \{0.06, 0.6\}\}, \{0.3533, 0\}], \\ AffineTransform[\{\{-0.8, -0.1867\}, \{0.1371, 0.8\}\}, \{1.1, 0.1\}]\}\}; \quad (4.23)$$

а) Построение рандомизированным методом (Рисунок 4.37) на основе функций (4.1), (4.2), (4.23).

$$Table[IFSPlot[N@Lest, RandomReal[1, \{100, 2\}], n], \{n, 0, 11\}]$$



Рисунок 4.37 – Лист рандомизированным методом

б) Построение методом случайных итераций с увеличением количества точек, используя функции (4.3), (4.4), (4.23). Результат изображен на Рисунке 4.38.

$$Table[IFSPlot[RandomIFS[N@Lest, 10^3], RandomReal[1, \{10^3, 2\}], n], \{n, 0, 10\}]$$

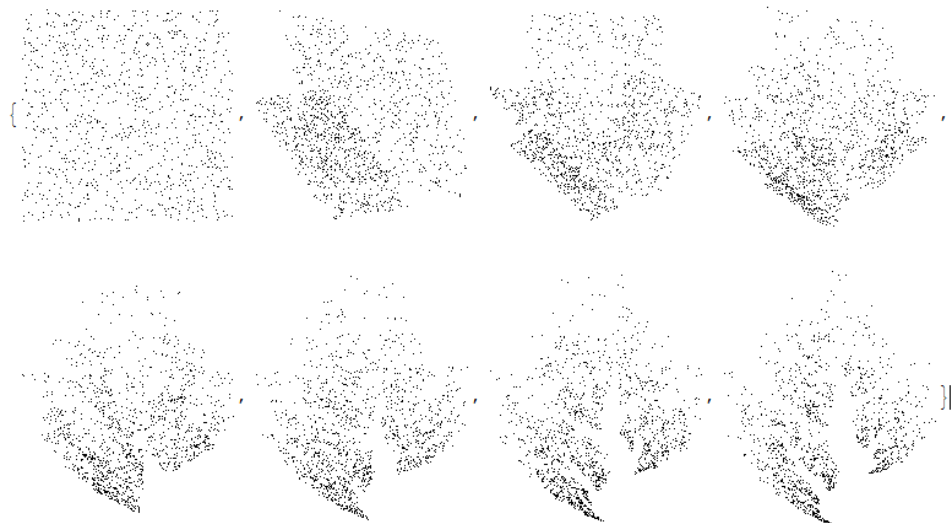


Рисунок 4.38 – Лист методом случайных итераций с увеличенным количеством точек
 в) Построение листа детерминированным методом (Рисунок 4.39) с помощью функций (4.5), (4.6), (4.23).

```
Table[TransformIFS[Graphics[Polygon[{{0,0},{1/2,Sqrt[3]/2},{1,0}}]],  
Lest,n],{n,0,8}]
```

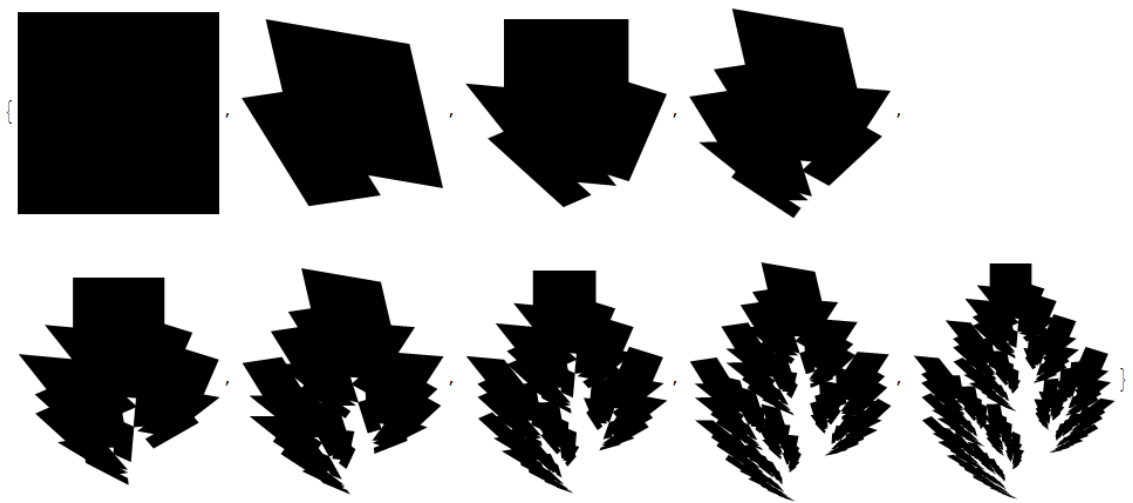


Рисунок 4.39 – Лист, построенный детерминированным методом

ЗАКЛЮЧЕНИЕ

В ходе проделанной работы, используя возможности языка Wolfram Mathematica, были созданы программы, реализующие наглядное представление геометрических преобразований. Полученные результаты облегчат понимание и изучение геометрических преобразований и фракталов.

По поставленным задачам можно сделать следующие выводы:

1. Изучена литература по темам: аффинные преобразования, инверсия, геометрические преобразования на плоскости в языке Wolfram;
2. Созданы программы, визуализирующие простые виды аффинных преобразований на плоскости, их композиции и инверсию;
3. Изучена литература по темам: классические фракталы, фрактальная размерность, функции языка Wolfram для построения фракталов;
4. Изучена литература по методам построения систем итерируемых функций;
5. Рассмотрены подходы построения системы итерируемых функций: детерминированный и метод случайных итераций, выявлены преимущества и недостатки этих методов;
6. С помощью программы Wolfram Mathematica удалось реализовать эти подходы и построить известные фракталы и их модификации.

ЛИТЕРАТУРА

1. Божокин С. В. Фракталы и мультифракталы : учебное пособие / С. В. Божокин, Д. А. Паршин. - [Б. м. : б. и., 200-?]. - .
URL: <http://sun.tsu.ru/limit/2016/000391940/000391940.pdf>
2. Зюзьков В. М. Эксперименты в теории чисел / В. М. Зюзьков ; Нац. исслед. Том. гос. ун-т. - Томск : Издательство научно-технической литературы, 2019.
URL: <http://vital.lib.tsu.ru/vital/access/manager/Repository/vtls:000658998>
3. Моденов П.С., Пархоменко А.С. Геометрические преобразования. М., Издательство МГУ, 1961. – 231 с.
4. Заславский А. А. Геометрические преобразования / А. А. Заславский. - [2-е изд., стер.]. - М. : МЦНМО, 2004. - 84, [2] с.: ил.
URL: <https://math.ru/lib/files/pdf/geometry/Zaslavsky.pdf>
5. НОУ ИНТУИТ| Лекция| Геометрические преобразования [Электронный ресурс] - URL: <https://intuit.ru/studies/courses/70/70/lecture/2096?page=6>
6. Иванов О. А. Элементарная математика для школьников, студентов и преподавателей. – М.: МЦНМО, 2009. – 385 с.: ил.
7. Гринченко В. Т. Введение в нелинейную динамику : хаос и фракталы / В. Т. Гринченко, В. Т. Мацыпура, А. А. Снарский. - Изд. стер.. - Москва : Издательство ЛКИ, 2014. - 280 с.: рис.
8. Wolfram Mathematica [Электронный ресурс] –
URL: <http://www.wolfram.com/mathematica>
9. Кроновер Р. М. Фракталы и хаос в динамических системах. Основы теории. Москва.:Постмаркет, 2000. – 352 с.

ПРИЛОЖЕНИЕ А

- *AffineTransform* - функция аффинного преобразования. Общее аффинное преобразование представляет собой:

$$t = \text{AffineTransform}[\{\{a_{1,1}, a_{1,2}\}, \{a_{2,1}, a_{2,2}\}\}, \{b_1, b_2\}]$$

$$: \text{TransformationFunction} \left[\left(\begin{array}{cc|c} a_{1,1} & a_{1,2} & b_1 \\ a_{2,1} & a_{2,2} & b_2 \\ \hline 0 & 0 & 1 \end{array} \right) \right]$$

Точки преобразования будут иметь вид:

$$t[\{x, y\}] \\ \{b_1 + xa_{1,1} + ya_{1,2}, b_2 + xa_{2,1} + ya_{2,2}\}$$

- *Apply[f, expr]* или *f@@expr* - применяет *f* к *expr*.

В качестве примера работы функции рассмотрим:

$$\text{Apply}[f, \{a, b, c, d\}]$$

Результат: $f\{a, b, c, d\}$

- *Axes* - опция для графических функций, которая указывает, должны ли быть нарисованы оси. Имеет два значения: True и False. Если True, то рисуются оси; если False, то нет.

- *Circle[\{x, y\}, r]* - представляет собой круг радиуса *r* с центром в точке $\{x, y\}$.

По умолчанию - единичный круг с центром в точке $\{0, 0\}$.

- *ControlPlacement* - это опция для объектов управления, таких как *Manipulate*, указывающая, где должны быть размещены элементы управления. Возможные варианты: *Bottom, Left, Top, Right*.

- *ControlType* - то опция для *Manipulate* и связанных функций, которая определяет, какой тип элементов управления должен отображаться. Возможные типы управления:

Animator, Checkbox, ColorSetter, ColorSlider, InputField, IntervalSlider, Manipulator, PopupMenu, RadioButton или *RadioButtonBar, Setter* или *SetterBar, Slider*.

- *Dashed* - это графическая директива, определяющая, что следующие линии должны быть нарисованы пунктирными линиями.

- *DiagonalMatrix[list]* - дает матрицу с элементами *list* на главной диагонали, а остальные элементы матрицы заполняются нулями.

Построение диагональной матрицы:

$$\text{DiagonalMatrix}[\{a, b, c\}]$$

Результат: $\begin{pmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{pmatrix}$

- $Disk[\{x, y\}, r]$ - представляет собой круг радиуса r с центром в точке $\{x, y\}$.
- $First[expr]$ - дает первый элемент в $expr$.

Принцип работы данной функции:

$First[\{a, b, c\}]$

Результат: a

- $Flatten[list, n]$ - выравнивает вложенный $list$ до заданного уровня n .

Сделаем выравнивание до уровня 1:

$Flatten[\{\{a, b\}, \{c, \{d\}, e\}, \{f\{g, h\}\}\}, 1]$

Результат: $\{a, b, c, \{d\}, e, f, \{g, h\}\}$

- $Frame$ - это опция для Graphics, Grid и других конструкций, которая указывает, следует ли включить рамку. Возможные значения: True, False, None, All
- $GeometricTransformation[g, tfun]$ - представляет результат применения функции преобразования $tfun$ к геометрическим объектам, соответствующим примитивам g .
- $Graphics[primitives, options]$ - представляет двумерное графическое изображение.
- $GridLines$ - опция для функций двумерной графики, определяющая линии сетки. Для $GridLines$ можно задать следующие настройки: *None* (линии сетки не нарисованы), *Automatic* (линии сетки размещаются автоматически) или указать линии сетки отдельно в каждом направлении $\{xgrid, ygrid\}$.
- $GridLinesStyle$ - это опция для функций двумерной графики, которая определяет, как должны отображаться линии сетки. С помощью этой функции можно задать толщину линий, их цвет и то, каким стилем будут отображаться, сплошной линией, пунктирной и другими.
- $Head[expr]$ - дает «главу» $expr$.

Рассмотрим несколько примеров, связанных с работой данной функции.

1. Определить руководителя функции: $Head[f[a, b]]$

Результат: f .

2. Глава списка: $Head[\{a, b, c\}]$

Результат: List

3. $Head[45]$

Результат: Integer

- $Join[list_1, list_2, \dots]$ - объединение списков с одинаковыми уровнями.

$Join[\{a, b, c\}, \{x, y\}, \{u, v, w\}]$

Результат: $\{a, b, c, x, y, u, v, w\}$

- $Length[expr]$ - дает количество элементов в $expr$.
- $Line[\{p_1, p_2, \dots\}]$ - представляет отрезки линии, соединяющие

последовательность точек p_i .

- $Manipulate[expr, \{u, u_{min}, u_{max}\}]$ - генерирует версию $expr$ с добавленными элементами управления, позволяющими интерактивно манипулировать значением u .

- $Module[\{x, y, \dots\}, expr]$ - указывает, что вхождение символов $x, y, \dots$ в $expr$ следует рассматривать как локальные.

- $N[expr]$ - дает числовое значение выражения.

- $Nest[f, expr, n]$ - дает выражение с f , примененным к $expr$ n раз.

Пример: $Nest[f, x, 3]$

Результат: $f[f[f[x]]]$

- $PlotRange$ - это опция для графических функций, которая определяет, какой диапазон координат, включать в график.

- $Point[p]$ - примитив графики и геометрии, представляющий точку в p .

- $PointSize[d]$ - это графическая директива, которая указывает, что следующие точки должны отображаться, если возможно, как круглые области с диаметром d . могут использоваться следующие символические формы для d : *Tiny, Small, Medium* и *Large*.

- $Polygon[\{p_1, p_2, \dots\}]$ - представляет собой закрашенный многоугольник с точками p_i .

- $RandomChoice[list, n]$ - дает список из n псевдослучайных вариантов $list$.

- $RandomReal[range, n]$ - дает список из n псевдослучайных чисел.

- $Rectangle[\{x_{min}, y_{min}\}, \{x_{max}, y_{max}\}]$ - представляет выровненный по оси закрашенный прямоугольник из $\{x_{min}, y_{min}\}$ до $\{x_{max}, y_{max}\}$.

- $RotationMatrix[\theta]$ - дает двумерную матрицу вращения, которая поворачивает двумерные векторы против часовой стрелки на θ радиан.

- $SaveDefinitions$ - функция, связанная с $Manipulate$, определяющую сохранять ли полученные значения или нет. Имеет значения: *True* и *False*.

- *ScalingTransform*[{ $s_x, s_y, \dots$ }] - дает *TransformationFunction*, которая представляет масштабирование с коэффициентом s_i вдоль каждой координатной оси от начала координат.
- *Slider2D*[{ x, y }] - представляет собой двумерный ползунок с настройками x и y .
- *Table*[*expr*, n] - генерирует список из n копий *expr*.
- *Through*[$p[f_1, f_2][x]$] - дает $p[f_1[x], f_2[x]]$.
- *Tiny* - это параметр стиля или параметр, указывающий, что объекты должны быть крошечными.
- *TranslationTransform*[v] - дает *TransformationFunction*, которая представляет перенос точек вектором v .
- *With*[{ $x = x_0, y = y_0, \dots$ }, *expr*] - указывает, что все параметры $x, y, \dots$, в *expr* должны иметь значения $x_0, y_0, \dots$, соответственно.

ПРИЛОЖЕНИЕ Б

Таблицы заимствованы из книги [9, с.111].

Таблица Б.1 – Коэффициенты аффинных преобразований для кристалла

a	b	c	d	e	f
0.2550	0.0000	0.0000	0.2550	0.3726	0.6714
0.2550	0.0000	0.0000	0.2550	0.1146	0.2232
0.2550	0.0000	0.0000	0.2550	0.6306	0.2232
0.3700	-0.6420	0.6420	0.3700	0.6356	-0.0061

Таблица Б.2 – Коэффициенты аффинных преобразований для листа папоротника

a	b	c	d	e	f
0.7000	0.0000	0.0000	0.7000	0.1496	0.2962
0.1000	-0.4330	0.1732	0.2500	0.4478	0.0014
0.1000	0.4330	-0.1732	0.2500	0.4445	0.1559
0.0000	0.0000	0.0000	0.3000	0.4987	0.0000

Таблица Б.3 – Коэффициенты аффинных преобразований для ковра А

a	b	c	d	e	f
0.5000	0.0000	0.0000	-0.5000	0.5000	0.5000
0.0000	-0.5000	-0.5000	0.0000	0.5000	0.5000
-0.5000	0.0000	0.0000	-0.5000	0.5000	1.0000

Таблица Б.4 – Коэффициенты аффинных преобразований для ковра В

a	b	c	d	e	f
0.5000	0.0000	0.0000	-0.5000	0.0000	1.0000
0.0000	0.5000	0.5000	0.0000	0.0000	0.0000
0.5000	0.0000	0.0000	0.5000	0.5000	0.0000

Таблица Б.5 – Коэффициенты аффинных преобразований для листа

a	b	c	d	e	f
0.4000	-0.3733	0.0600	0.6000	0.3533	0.0000
-0.8000	-0.1867	0.1371	0.8000	1.1000	0.1000

Таблица Б.6 – Коэффициенты аффинных преобразований для дерева

a	b	c	d	e	f
0.1950	-0.4880	0.3440	0.4430	0.4431	0.2452
0.4620	0.4140	-0.2520	0.3610	0.2511	0.5692

Продолжение таблицы Б.6

a	b	c	d	e	f
-0.0580	-0.0700	0.4530	-0.1110	0.5976	0.0969
-0.0350	0.0700	-0.4690	0.0220	0.4884	0.5069
-0.6370	0.0000	0.0000	0.5010	0.8562	0.2513

Отчет о проверке на заимствования №1



Автор: Бер Кристина Андреевна
Проверяющий: (kristina.ber.1999@mail.ru / ID: 9223472)
Отчет предоставлен сервисом «Антиплагиат» - [users.antiplagiat.ru](https://www.antiplagiat.ru)

ИНФОРМАЦИЯ О ДОКУМЕНТЕ

Имя документа: З
Начало загрузки: 11.06.2021 05:01:58
Длительность загрузки: 00:00:05
Имя исходного файла: Дипломная работа Бер.pdf
Название документа: Дипломная работа Бер
Размер текста: 97 кБ
Тип документа: Выпускная квалификационная работа
Символов в тексте: 99091
Слов в тексте: 10680
Число предложений: 635

ИНФОРМАЦИЯ ОБ ОТЧЕТЕ

Начало проверки: 11.06.2021 05:02:05
Длительность проверки: 00:00:09
Корректировка от 11.06.2021 05:16:04
Комментарии: не указано
Модуль поиска: Интернет



ЗАИМСТВОВАНИЯ	САМОЦИТИРОВАНИЯ	ЦИТИРОВАНИЯ	ОРИГИНАЛЬНОСТЬ
4,5%	0%	0%	91,65%

Заимствования — доля всех найденных текстовых пересечений, за исключением тех, которые система внесла в цитирования, по отношению к общему объему документа.
Самоцитирования — доля фрагментов текста проверенного документа, совпадающих или почти совпадающих с фрагментом текста источника, автором или соавтором которого является автор проверенного документа, по отношению к общему объему документа.

Цитирования — доля текстовых пересечений, которые не являются авторскими, но система посчитала их использование корректным, по отношению к общему объему документа. Сюда относятся оформленные по ГОСТу цитаты, общепотребительные выражения, фрагменты текста, найденные в источниках из коллекций нормативно-правовой документации.

Текстовое пересечение — фрагмент текста проверенного документа, совпадающий или почти совпадающий с фрагментом текста источника.

Источник — документ, проиндексированный в системе и содержащийся в модуле поиска, по которому проводится проверка.

Оригинальность — доля фрагментов текста проверенного документа, не обнаруженных ни в одном источнике, по которым шла проверка, по отношению к общему объему документа.

Заимствования, самоцитирования, цитирования и оригинальность являются отдельными показателями и в сумме дают 100%, что соответствует всему тексту проверенного документа.

Обращаем Ваше внимание, что система находит текстовые пересечения проверенного документа с проиндексированными в системе текстовыми источниками. При этом система является вспомогательным инструментом, определение корректности и прикрасности заимствований или цитирований, а также достоверности текстовых фрагментов проверенного документа остается в компетенции проверяющего.

№	Доля в отчете	Источник	Актуален на	Модуль поиска
[01]	4,5%	ТЕОРИЯ ИНФОРМАЦИОННЫХ ПРОЦЕССОВ И СИСТЕМ. http://ngs.ru	13 Сен 2017	Интернет
[02]	2,32%	550385.zip http://project.1september.ru	21 Июн 2018	Интернет
[03]	2,14%	ГЛАВА - Лекции по имитационному моделированию - Глава 1 Фракталы.doc http://gendocs.ru	29 Ноя 2016	Интернет

Научный руководитель
Зюзовов В. М.

15.06.21