Министерство образования и науки Российской Федерации

КАЗАНСКИЙ (ПРИВОЛЖСКИЙ) ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ

ИНСТИТУТ МАТЕМАТИКИ И МЕХАНИКИ
им. Н.И.ЛОБАЧЕВСКОГО

КАФЕДРА ГЕОМЕТРИИ

Специальность: 01.03.01 — Математика

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

(Бакалаврская работа)

ПОСТРОЕНИЕ L-СИСТЕМ В ПРОГРАММЕ WOLFRAM MATHEMATICA

Работа завершена:
"___"________2015 г. ________________________________(К.Р. Гимадеева)

Работа допущена к защите:
Научный руководитель
кандидат физ.-мат. наук, доцент
"___"___________2015 г. ______________________________(К.Б. Игудесман)

Заведующий кафедрой
док. физ.-мат. наук, профессор
"___"___________2015 г. ______________________________(В.В. Шурыгин)

Казань — 2015

Оглавление

Введение

Фракталом называется сложная геометрическая фигура, обладающая свойством самоподобия, т.е. составленная из нескольких частей, каждая из которых подобна целой фигуре. В более широком смысле под фракталами понимают множества точек в евклидовом пространстве, имеющие промежуточную (дробную) метрическую размерность (размерность Хаусдорфа) или метрическую размерность, отличную от топологической.

Термин фрактал был впервые введен в 1975 году Бенуа Мандельбротом, пионером в области фрактальной геометрии[1]. Почти все математические идеи появились задолго до этого, еще в XIX-м веке, в работах Георга кантора, Карла Вейерштрасса, Джузеппе Пеано и остальных. Понятие фрактальной (дробной) размерности возникло в 1919 году в работе Феликса Хаусдора. Однако именно Мандельброт объединил эти идеи и положил начало систематическому изучению фракталов и их приложений.

Фракталы бывают нескольких типов:

- *Геометрические фракталы* – самый наглядный класс фракталов. В двухмерном пространстве их получают с помощью некоторой ломаной (или поверхности в трехмерном пространстве), называемой генератором. За один шаг алгоритма каждый из отрезков, составляющих ломаную, заменяется на ломаную - генератор, в соответствующем масштабе. В результате бесконечного повторения этой процедуры, получается геометрический фрактал.
- *Алгебраические фракталы* – это самая большая группа фракталов. Получают их с помощью нелинейных процессов в n-мерных пространствах.

[1] Термин фрактал произведен от латинского глагола frangere – ломать и прилагательного fractus - дробный

Фазовое пространство системы разбивается на области притяжения аттракторов. Если фазовым является двухмерное пространство, то, окрашивая области притяжения различными цветами, можно получить цветовой фазовый портрет этой системы (итерационного процесса). Меняя алгоритм выбора цвета, можно получить сложные фрактальные картины с причудливыми многоцветными узорами.

- *Стохастические фракталы* являются ещё одним известным классом фракталов, которые получаются в том случае, если в итерационном процессе случайным образом менять какие-либо его параметры. При этом получаются объекты очень похожие на природные — несимметричные деревья, изрезанные береговые линии и т.д. Двумерные стохастические фракталы используются при моделировании рельефа местности и поверхности моря.

Кроме этих существует другая классификация фракталов:

- *Рукотворные фракталы* – фракталы, которые были придуманы учеными, при любом масштабе обладают фрактальными свойствами.
- *Природные фракталы* – накладывается ограничение на область существования – максимальный и минимальный размер, при которых у объекта наблюдаются фрактальные свойства.
- *Детерминированные фракталы* – образуются в процессе, называемом итерацией, которая применяет основной рисунок к инициатору, после чего применяет его к результату и так далее.
- *Недетерминированные (стохастические) фракталы* – получаются в том случае, если в итерационном процессе случайным образом менять какие-либо его параметры.

Понятие L-систем, тесно связанное с самоподобными фракталами, возникло лишь в 1968 году благодаря Аристиду Линденмайеру. Изначально L-системы были введены при изучении формальных языков, а также использовались в биологических моделях селекции. С их помощью можно строить многие известные самоподобные фракталы, включая снежинку Коха и ковер Серпинского. Некоторые другие классические построения, например кривые Пеано (работы Пеано, Гильберта, Серпинского), также укладываются в эту схему. И конечно, L-системы открывают путь к бесконечному разнообразию новых фракталов, что и послужило причиной их широкого применения в компьютерной графике для построения фрактальных деревьев и растений.

Целью дипломной работы является построение фракталов с помощью L-систем в программе Wolfram Mathematica и исследование свойств построенных фракталов.

Для достижения поставленной цели дипломной работы необходимо решить следующие задачи:

1) Исследовать самоподобные свойства фракталов, в частности вычислить их размерность подобия
2) Изучить в алгоритме построения фракталов на основе L-систем
3) Реализовать алгоритм в программе Wolfram Mathematica

Размерность подобия

Разделим отрезок прямой на N равных частей. Тогда каждую часть можно считать копией всего отрезка, уменьшенной в 1/r раз. Очевидно, N и r связаны соотношением Nr=1. Размерность d объекта появляется как степень r в соотношении между N, числом равных подобъектов, и коэффициентом подобия r. А именно

$$Nr^{d} = 1$$

Величину d называют фрактальной (дробной) размерностью или размерностью подобия. Явное выражение d через N и r находится логарифмированием обеих частей:

$$d = -\frac{\log N}{\log(1/r)}$$

Логарифм можно взять по любому положительному основанию, отличному от единицы. Например, по основанию 10 или по основанию $e \approx 2.7183$.

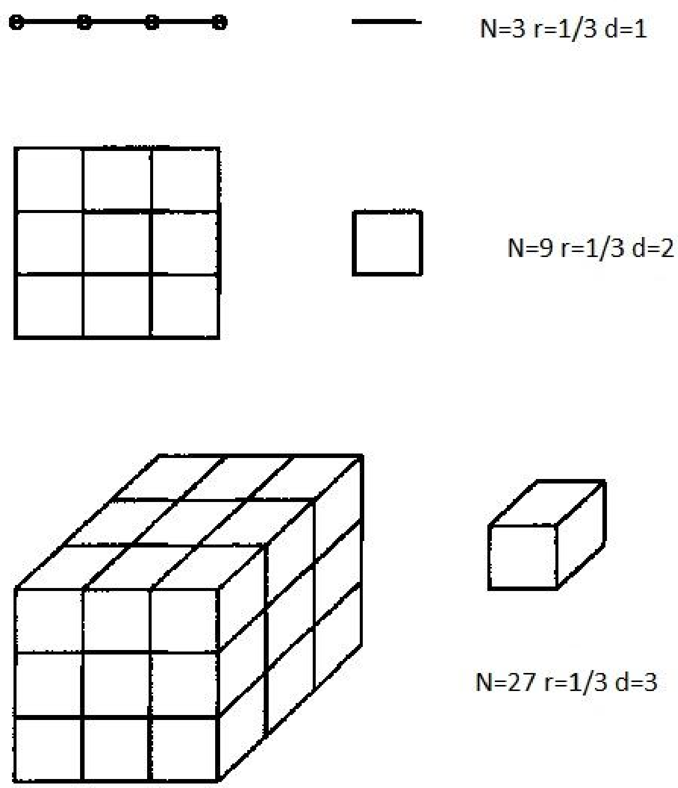


Рис.1. Связь размерности и коэффициента подобия

Чтобы найти N, считаем количество единичных отрезков после первой итерации. Оно равно количеству символов F в порождающем правиле (если порождающих правил несколько, то считаем количество символов F в каждом правиле и суммируем их).

r – расстояние между максимально удаленными точками.

L-системы

Детерминированные L-системы

Общее описание

Для графической реализации L-систем в качестве подсистемы вывода используется так называемая тертл - графика (turtle – черепаха). При этом точка (черепашка) движется по экрану дискретными шагами, как правило, прочерчивая свой след, но при необходимости может перемещаться без рисования. В нашем распоряжении имеются три параметра (x,y,α), где (x,y) – координаты черепашки, α – направление, в котором она смотрит. Черепашка обучена интерпретировать и выполнять последовательность команд, задаваемых кодовым словом, буквы которого читаются слева направо. Кодовое слово представляет собой результат работы L-системы и может включать следующие буквы:

F переместиться вперед на один шаг, прорисовывая след

b переместиться вперед на один шаг, не прорисовывая след

[открыть ветвь (подробнее см. ниже)

] закрыть ветвь (подробнее см. ниже)

+ увеличить угол α на величину θ.

- уменьшить угол α на величину θ

Размер шага и величина приращения по углу θ задаются заранее и остаются неизменными для всех перемещений черепашки. Если начальное направление движения α (угол, отсчитываемый от положительного направления оси X) не указано, то полагаем α равным нулю.

Несколько примеров иллюстрируют применение команд ветвления (обозначаются], [ ) и вспомогательных переменных (обозначаются X, Y и т.д.). команды ветвления используются для построения деревьев и растений, а вспомогательные переменные заметно облегчают построение некоторых L-систем.

Формально, детерминированная L-система состоит из алфавита, слова инициализации, называемого аксиомой или инициатором, и набора порождающих правил, указывающих, как следует преобразовывать слово при переходе от уровня к уровню (от итерации к итерации). К примеру, можно заменять букву F при помощи порождающего правила newF=F-F+F+FF-F-F+F, что соответствует L-системе для острова Коха, рассматриваемому ниже. Символы +, -,], [не обновляются, а просто остаются на тех местах, где они встретились. Обновление букв в данном слове предполагается одновременным, то есть все буквы слова одного уровня обновляются раньше любой буквы следующего уровня.

Простейшие L-системы

L-система для снежинки Коха (рис.2) задается следующим образом:

$\theta = \pi/3$

Аксиома: F++F++F

Порождающее правило: newF = F-F++F-F

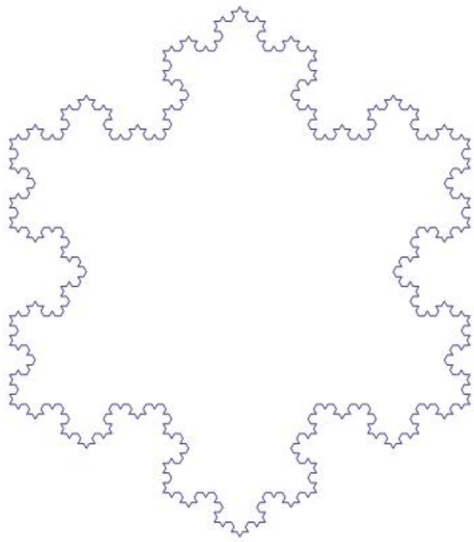
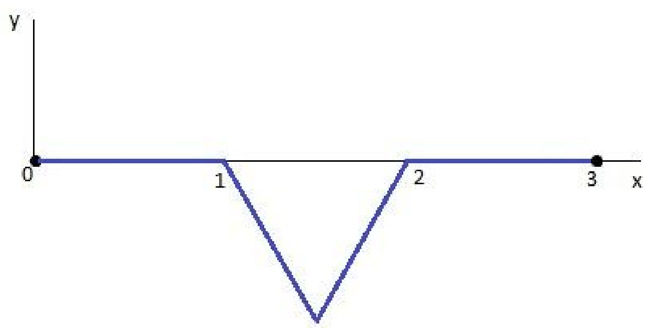


Рис.2. Снежинка Коха после 4 итераций

Найдем размерность снежинки Коха.



1) N=4

2) Построим систему координат так, чтобы одна из нужных нам точек совпадала с началом координат. Максимально удаленные друг от друга точки имеют координаты (0; 0) и (3; 0) (очевидно по рисунку). Следовательно, r = 3

$$d = -\frac{\ln N}{\ln(1/r)} = -\frac{\ln 4}{\ln(1/3)} = 1.26186$$

Графическое представление аксиомы F++F++F – равносторонний треугольник. Черепашка делает один шаг вперед, затем угол α увеличивается на 2π/3 и черепашка делает еще один шаг вперед, угол α снова увеличивается на 2π/3 и черепашка делает еще шаг.

На первом шаге каждая буква F в слове – инициаторе F++F++F заменяется на F-F++F-F:

(F-F++F-F)++(F-F++F-F)++(F-F++F-F).

Убирая скобки, получаем:

F-F++F-F++F-F++F-F++F-F++F-F.

Повторяя этот процесс, на втором шаге получим:

F-F++F-F-F-F++F-F++F-F++F-F-F-F++F-F++

F-F++F-F-F-F++F-F++F-F++F-F-F-F++F-F++

F-F++F-F-F-F++F-F++F-F++F-F-F-F++F-F

И т.д.

Псевдокод для итерирования порождающих правил в этом простейшем случае, когда используются только правило вида F=newF:

Назначение: реализует правила F=newF.

Вход:

axiom (слово инициализация)

newF (порождающее правило)

level (число итераций)

Выход:

word (слово – результат)

Инициализация:

W=axiom

T=“” (пустое множество)

Шаги:

```
For [j=1, j≤StringLength[W], l=StringTake[W,{j}];
If[l=="+", T=T<>"+"];
If[l=="-", T=T<>"-"];
If[l=="F", T=T<>newF];
j++];
W=T; i++]
W;
```

Замечание: l – j-ая буква в слове W.

axoim = F-F-F-F

newF = F-F+F+FF-F-F+F

$\theta = \pi/2$

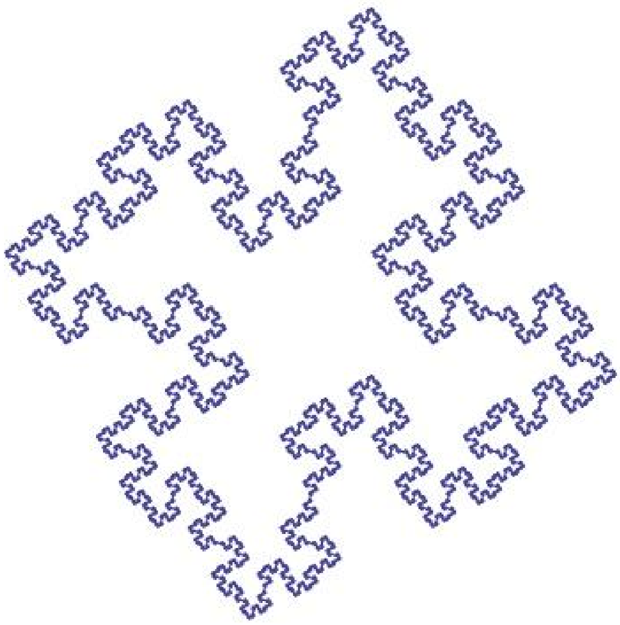
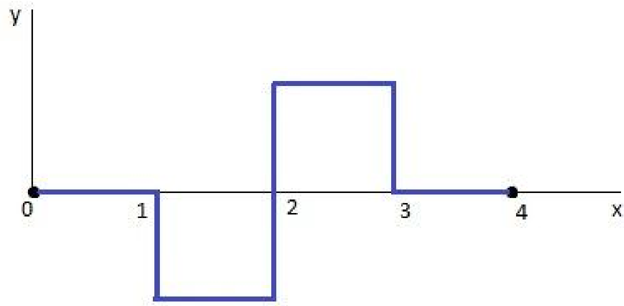


Рис.3. Остров Коха после 4 итераций

Найдем размерность острова Коха.



1) N=8

2) Построим систему координат так, чтобы одна из нужных нам точек совпадала с началом координат. Максимально удаленные друг от друга точки имеют координаты (0; 0) и (4; 0) (очевидно по рисунку). Следовательно, r = 4

$$d = -\frac{\ln N}{\ln(1/r)} = -\frac{\ln 8}{\ln(1/4)} = 1.5$$

axiom = F-F-F-F-F

newF = F-F-F++F+F-F

$\theta = 2\pi/5$

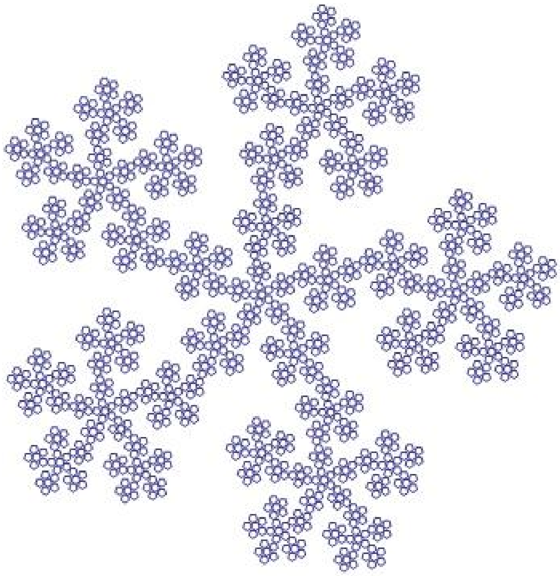
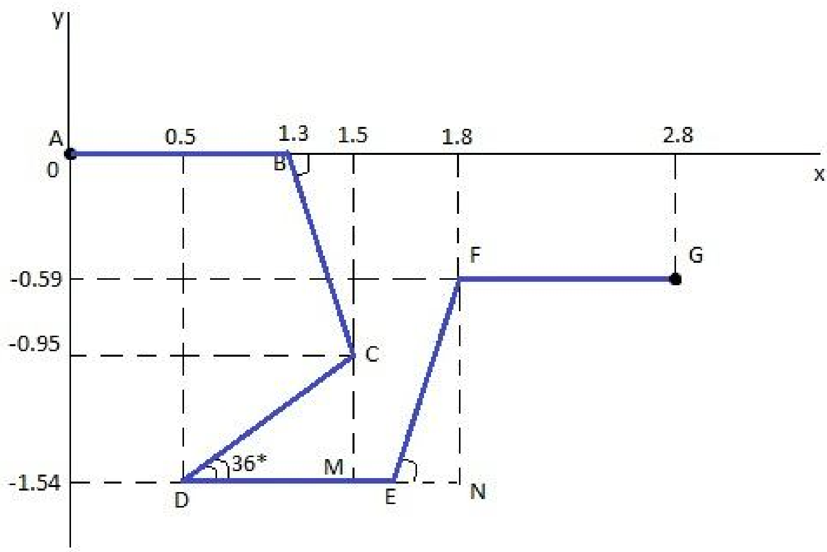


Рис.4. Снежинка после 4 итераций

Найдем размерность снежинки.



1) N = 6

2) Построим систему координат с началом в точке А.

т.А и т.G – максимально удаленные друг от друга точки.

A(0; 0), B(1; 0)

ΔBCK: $CK = \sin\theta = \sin\frac{2\pi}{5} = 0.95$

$BK = \cos\theta = \cos\frac{2\pi}{5} = 0,3$

=> C(1,3; -0,95)

ΔCDM: $CM = \sin 36° = 0.59$

$DM = \cos 36° = 0.8$

=> D(0,5; -1,54), E(1,5; -1,54)

ΔFEN: $EN = \cos\frac{2\pi}{5} = 0.3$

$FN = \sin\frac{2\pi}{5} = 0.95$

=> F(1.8; -0.59), G(2.8; -0.59)

Найдем расстояние между т.А и т.G по формуле:

$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} = \sqrt{2.8^2 + 0.59^2} = \sqrt{8.1881} = 2.861$

Следовательно, r = 2,861

$$d = -\frac{\ln N}{\ln(1/r)} = -\frac{\ln 6}{\ln(1/2.861)} = 1.70454$$

L-системы с разрывом

Разрывные графики можно получать, применяя в L-системе команду «b», то есть команду «переместиться на один шаг вперед без рисования». Служит для рисования фрактальных разрывных поверхностей. Может быть использовано при моделировании островов.

Псевдокод для итерирования порождающих правил, когда используются правила F=newF, b=newb:

Назначение: реализует правила F=newF, b=newb.

Вход:

axiom (слово инициализация)

newF (порождающее правило)

newb (порождающее правило)

level (число итераций)

Выход:

word (слово – результат)

Инициализация:

W=axiom

T=“” (пустое множество)

Шаги:

```
For [j=1, j≤StringLength[W], l=StringTake[W,{j}];
If[l=="+", T=T<>"+"];
If[l=="-", T=T<>"-"];
If[l=="F", T=T<>newF];
If[l=="b", T=T<>newb];
j++];
W=T; i++]
W;
```

Примерами могут служить следующие изображения.

axiom = F-F-F-F

newF = F-b+FF-F-FF-Fb-FF+b-FF+F+FF+Fb+FFF

newb = bbbbbb

$\theta = \pi/2$

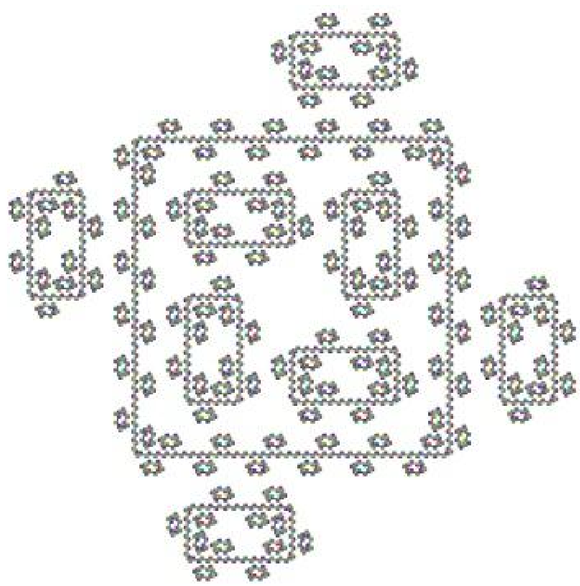
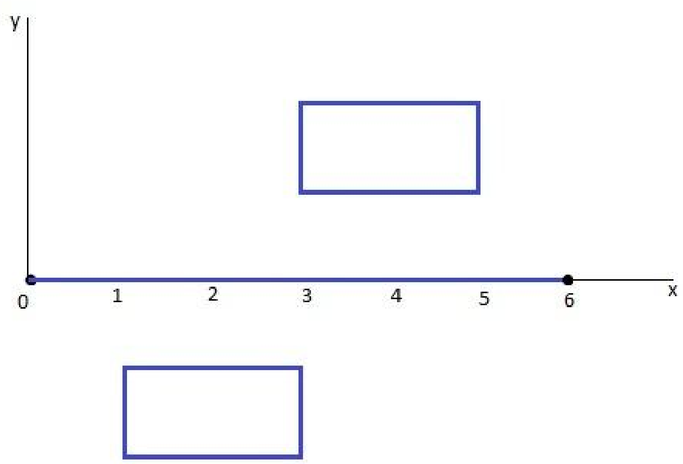


Рис.5. Мозаика после 3 итераций

Найдем размерность мозаики.



1) N = 18

2) Построим систему координат с началом в одной из нужных нам точек. Максимально удаленные друг от друга точки имеют координаты (0; 0) и (6; 0) (очевидно по рисунку).

Следовательно, r = 6

$$d = -\frac{\ln N}{\ln(1/r)} = -\frac{\ln 18}{\ln(1/6)} = 1.61315$$

axiom = F+F+F+F

newF = F+b-F-FFF+F+b-F

newb = bbb

$\theta = \pi/2$

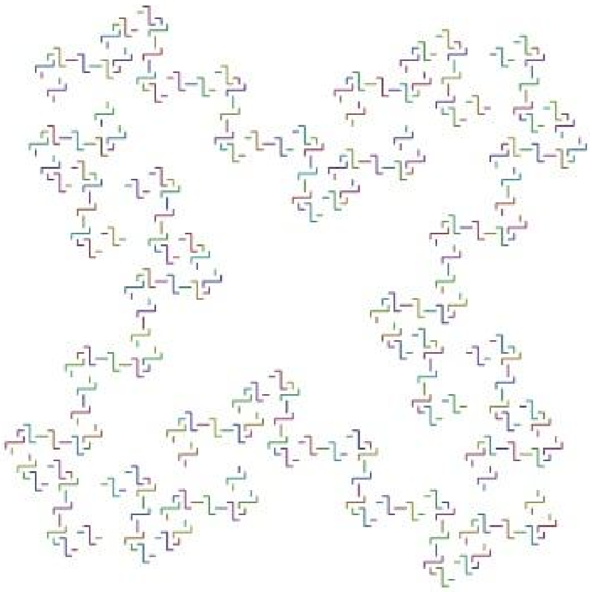
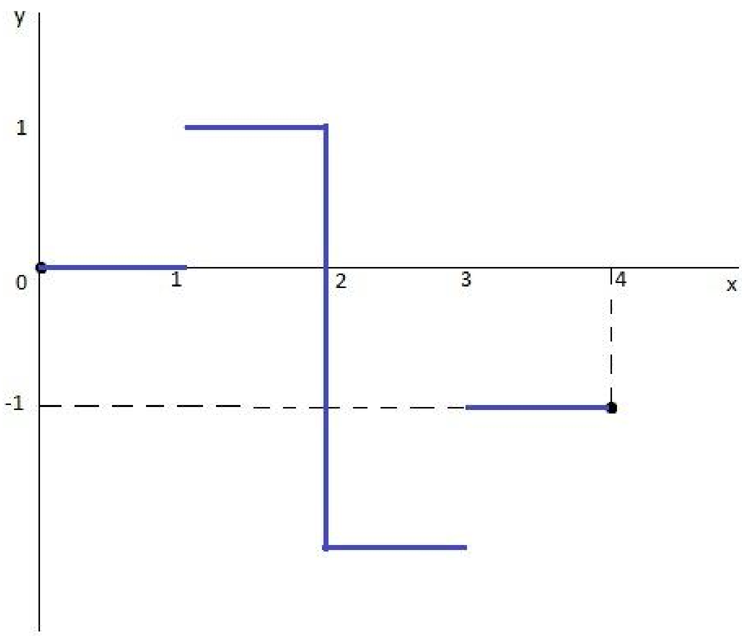


Рис.6. Цепочка после 3 итераций

Найдем размерность цепочки.



1) N = 7

2) Построим систему координат так, чтобы одна из нужных нам точек совпадала с началом координат. Найдем координаты точки максимально удаленной от раннее выбранной точки. По рисунку очевидно, что ее координаты (4; -1). Найдем расстояние между этими точками по формуле:

$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} = \sqrt{4^2 + 1^2} = \sqrt{17} = 4.1$

Следовательно, r = 4.1

$$d = -\frac{\ln N}{\ln(1/r)} = -\frac{\ln 7}{\ln(1/4.1)} = 1.37911$$

Введение новых переменных

При построении фракталов с использованием только одного порождающего правила возникает некоторое затруднение. Мы не можем изменить направление чтения правила на некоторых шагах, то есть читать его не слева направо, а справа налево. Например, для того чтобы построить фрактал под названием «дракон Хартера – Хайтвея», необходимо иметь возможность менять направление чтения порождающего правила. Порождающее правило в данном случае заключается в том, чтобы нарисовать инициатор сначала в прямом, а затем в обратном направлении. Эту проблему можно устранить, введя две различные команды для передвижения вперед, к примеру, X и Y. Будем считать, что черепашка интерпретирует X и Y одинаково, то есть как один шаг вперед. Благодаря этим буквам порождающее правило для дракона можно записать таким образом:

axiom = FX
newF = F
newX = X+YF+
newY = -FX-Y

Псевдокод для итерирования порождающих правил, когда используются правила F=newF, b=newb, X=newX, Y=newY:

Назначение: реализует правила F=newF, b=newb, X=newX, Y=newY.
Вход:
axiom (слово инициализация)
newF (порождающее правило)
newb (порождающее правило)

newX (порождающее правило)

newY (порождающее правило)

level (число итераций)

Выход:

word (слово – результат)

Инициализация:

W=axiom

T=“” (пустое множество)

Шаги:

```
For [j=1, j≤StringLength[W], l=StringTake[W,{j}];
If[l=="+", T=T<>"+"];
If[l=="-", T=T<>"-"];
If[l=="F", T=T<>newF];
If[l=="b", T=T<>newb];
If[l==”X”, T=T<>newX];
If[l==Y, T=T<>newY];
j++];
W=T; i++]
W;
```

axiom = FX

newF = F

newX = X+YF+

newY = -FX+Y

$\theta = \pi/2$

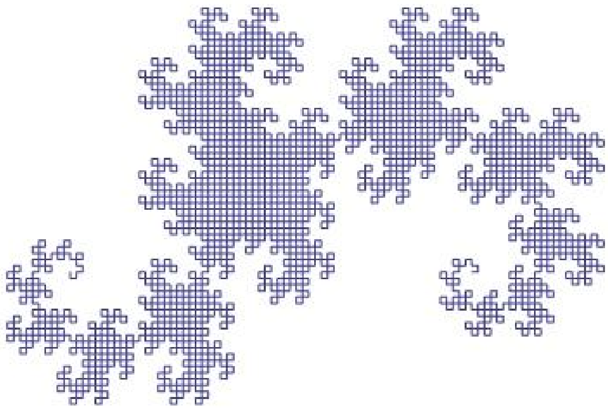
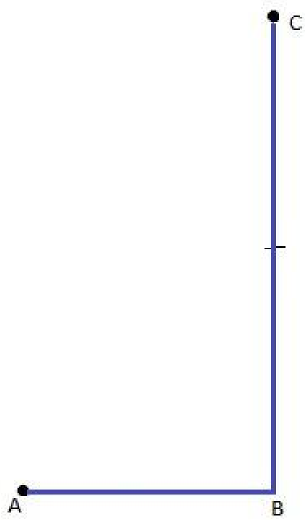


Рис.7. Дракон Хартера – Хайтвея после 12 итераций

Найдем размерность дракона Хартера – Хайтвея.



1) N = 3

2) т.А и т.С – максимально удаленные друг от друга точки. Найдем расстояние между ними по теореме Пифагора:

$$AC^2 = AB^2 + BC^2$$

$$AC = \sqrt{AB^2 + BC^2} = \sqrt{1+4} = \sqrt{5} = 2{,}23606798$$

Следовательно, r = 2,23606798

$$d = -\frac{\ln N}{\ln(1/r)} = -\frac{\ln 3}{\ln(1/2.23606798)} = 0.884074$$

axiom = XF

newF = F

newX = X+YF++YF-FX—FXFX-YF+

newY = -FX+YFYF++YF+FX—FX-Y

θ = π/3

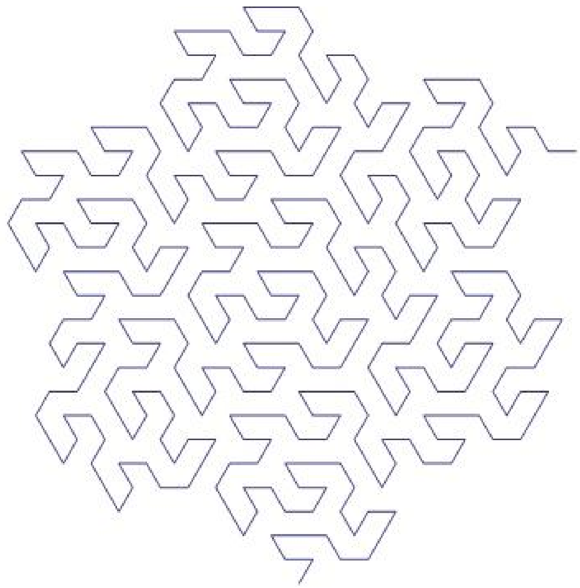
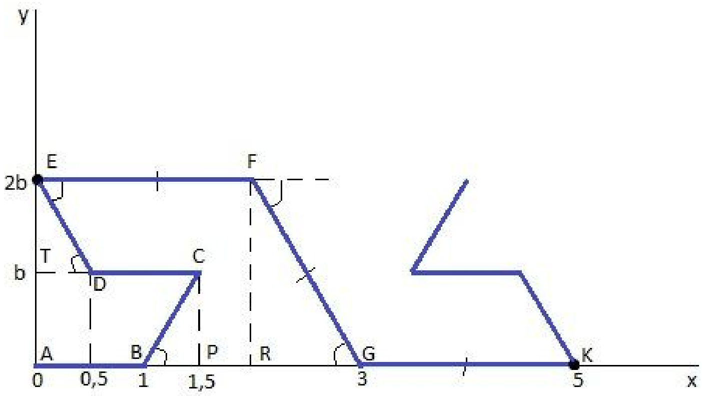


Рис.8. Кривая Госпера, заполняющая плоскость после 3 итераций

Найдем размерность кривой Госпера.



1) N = 13

2) т.Е и т.К – максимально удаленные друг от друга точки.

Построим систему координат с началом в т.А.

A(0; 0), B(1; 0)

$$b = \sin\theta = \sin\frac{\pi}{3} = 0.86602541$$

$$\cos\theta = \cos\frac{\pi}{3} = 0.5$$

=> C(1,5; 0,86602541), D(0,5; 0,86602541)

ΔETD: E(1,73205082; 0)

F(2; 1,73205082)

ΔFRG: $\frac{RG}{2} = \cos\frac{\pi}{3}$ => $RG = 2\cos\frac{\pi}{3} = 1$

=> G(3; 0), K(5; 0)

Расстояние между т.Е и т.К ищем по формуле:

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} = \sqrt{(5 - 1.73205082)^2} = 3.26794918$$

Слеовательно, r = 3,26794918

$$d = -\frac{\ln N}{\ln(1/r)} = -\frac{\ln 13}{\ln(1/3.26794918)} = 2.16604$$

Ветвящиеся L-системы

Когда встречается символ [(открыть ветвь), мы запоминаем положение и направление черепашки, то есть переменные (x,y,α), и возвращаемся к этим установкам позже. Для хранения триплетов (x,y,α) используется стек

| x1 y1 alpha1 |

| x2 y2 alpha2 |

| x3 y3 alpha3 |

| x4 y4 alpha4 |

причем новые данные записываются в конец стека. Когда ветвь закрывается, переменным (x,y,α) присваиваются значения, считанные из конца стека. Затем эти значения из стека удаляются.

Таким образом, ветвление задается двумя символами:

[Открыть ветвь. Сохранить (x,y,α) в конце стека.

] Закрыть ветвь. Присвоить переменным (x,y,α) значения, считанные из конца стека, после чего эти значения из стека удаляются.

Псевдокод для итерирования порождающих правил, когда используются правила F=newF, b=newb, X=newX, Y=newY:

Назначение: реализует правила F=newF, b=newb, X=newX, Y=newY.

Вход:

axiom (слово инициализация)

newF (порождающее правило)

newb (порождающее правило)

newX (порождающее правило)

newY (порождающее правило)

level (число итераций)

Выход:

word (слово – результат)

Инициализация:

W=axiom

T=“” (пустое множество)

Шаги:

```
For [j=1, j≤StringLength[W], l=StringTake[W,{j}];
If[l=="+", T=T<>"+"];
If[l=="-", T=T<>"-"];
If[l=="F", T=T<>newF];
If[l=="b", T=T<>newb];
If[l==”X”, T=T<>newX];
If[l==Y, T=T<>newY];
If[l==”[“, T=T<>”[“];
If[l==”]”, T=T<>”]”];
j++];
W=T; i++]
W;
```

На рис.9, рис.10, рис.11 и рис.12 изображены фракталы, построенные с помощью операции ветвления.

axiom = F

newF = F[+F]F[-F]F

$\alpha = \pi/2$

$\theta = \pi/7$

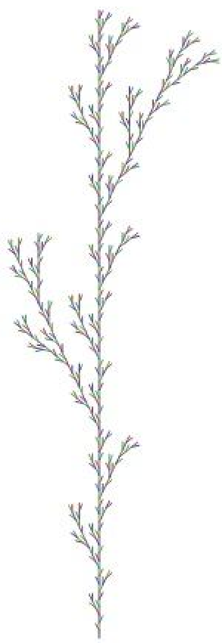
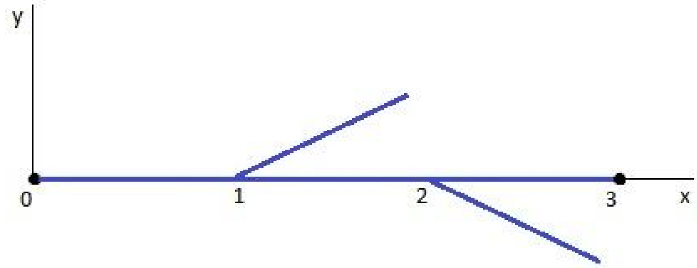


Рис.9. Сорняк после 4 итераций

Найдем размерность сорняка.



1) N = 5

2) Построим систему координат. Максимально удаленные друг от друга точки находятся на расстоянии трех шагов друг от друга (очевидно по рисунку).

Следовательно, r = 3

$$d = -\frac{\ln N}{\ln(1/r)} = -\frac{\ln 5}{\ln(1/3)} = 1.46497$$

axiom = F

newF = -F+F+[+F-F-]-[-F+F+F]

θ = π/8

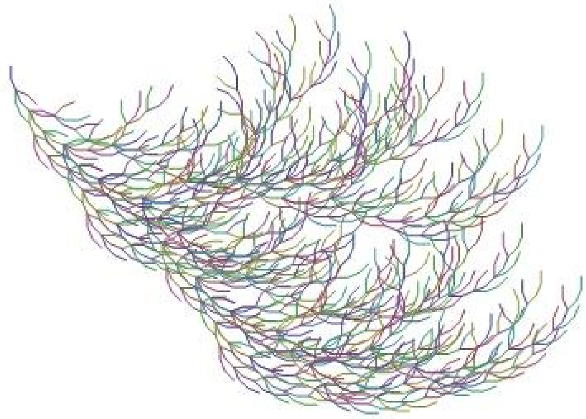
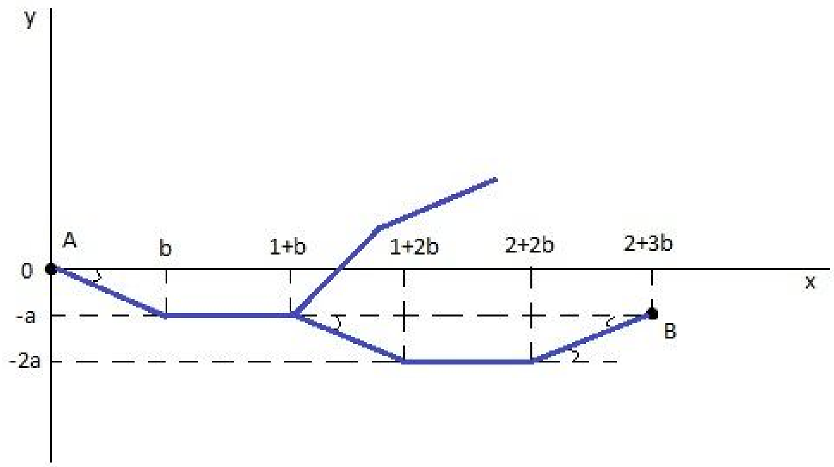


Рис.10. Куст после 4 итераций

Найдем размерность куста.



1) N = 7

2) т.А и т.В – максимально удаленные друг от друга точки.

Построим систему координат с началом в т.А.

$$a = \sin\theta = \sin\frac{\pi}{8} = 0.382683$$

$$b = \cos\theta = \cos\frac{\pi}{8} = 0.873305$$

$$2 + 3b = 2 + 3 * 0.873305 = 4.619915$$

=> B(4,619915; -0,382683)

Расстояние между т.А и т.В ищем по формуле:

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} = \sqrt{4.619915^2 + 0.382683^2} = 4.63573736$$

Следовательно, r = 4,63573736

$$d = -\frac{\ln N}{\ln(1/r)} = -\frac{\ln 7}{\ln(1/4.63573736)} = 1.26869$$

axiom = F[+F+F][-F-F][++F][--F]F

newF = FF[++F][+F][F][-F][--F]

$\alpha = \pi/2$

$\theta = \pi/16$

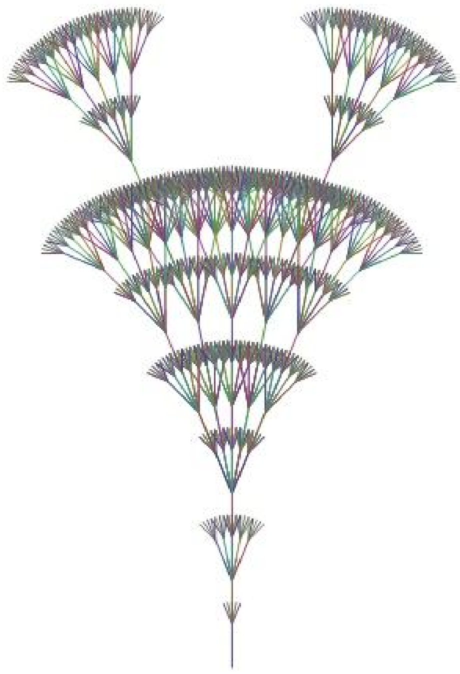
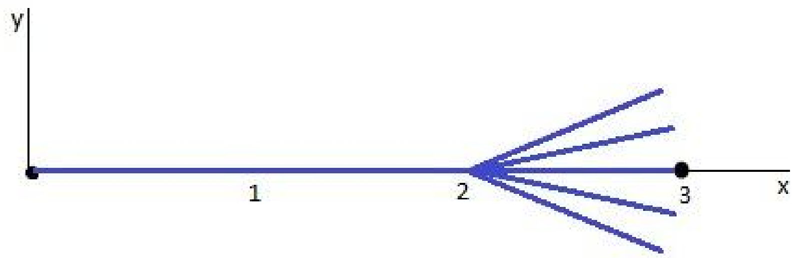


Рис.11. Цветок после 3 итераций

Найдем размерность цветка.



1) N = 7

2) Построим систему координат. Максимально удаленные друг от друга точки находятся на расстоянии трех шагов друг от друга (очевидно по рисунку).

Следовательно, r = 3

$$d = -\frac{\ln N}{\ln(1/r)} = -\frac{\ln 7}{\ln(1/3)} = 1.77124$$

axiom = X

newF = FF

newX = F-[[X]+X]+F[+FX]-X

θ = π/8

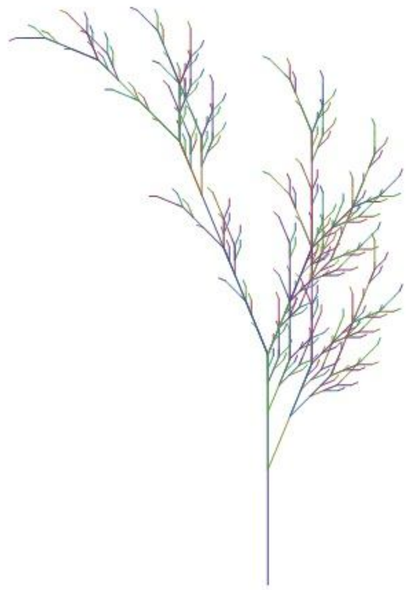
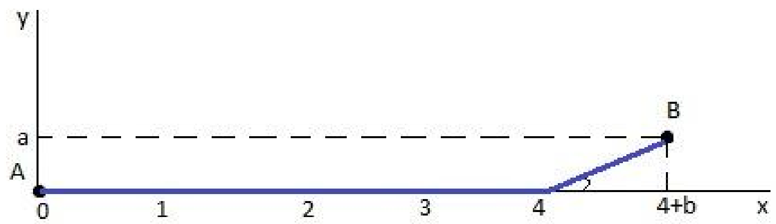


Рис. 12. Дерево после 5 итераций

Найдем размерность дерева.



1) N = 5

2) т.А и т.В – максимально удаленные друг от друга точки.

Построим систему координат с началом в т.А.

$$a = \sin\theta = \sin\frac{\pi}{8} = 0.382683$$

$b = \cos\theta = \cos\frac{\pi}{8} = 0.873305$

=> B(4,873305; 0,382683)

Расстояние между т.А и т.В будем искать по формуле:

$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} = \sqrt{4.873305^2 + 0.382683^2} = 4.88830726$

Следовательно, r = 4,88830726

$$d = -\frac{\ln N}{\ln(1/r)} = -\frac{\ln 5}{\ln(1/4.88830726)} = 1.01424$$

Стохастические L-системы

Стохастические (или недетерминированные) L-системы содержат элемент случайности в своем развитии. Они допускают по несколько правил для каждого символа – предшественника. Одно из нескольких правил для данного символа строки состояния каждый раз выбирается случайно, с некоторой заданной вероятностью. Разумеется, каждому из правил для данного предшественника приписывается вероятность выбора правила – число от нуля до единицы (сумма таких чисел должна равняться единице). В результате итерационного процесса получаются объекты довольно похожие на природные фракталы – несимметричные деревья, изрезанные лагунами береговые линии островов и многое другое. Двумерные стохастические фракталы используются преимущественно при моделировании рельефа местности и поверхности моря.

Наиболее яркими примерами стохастических фракталов являются:

- Траектория броуновского движения на плоскости и в пространстве. А также граница броуновского движения на плоскости
- Эволюции Шрама-Лёвнера – конформно – инвариантные фрактальные кривые, возникающие в критических двумерных моделях статистической механики
- Разнообразные виды рандомизированных фракталов. Это фракталы, образованные с помощью рекурсивной процедуры, в которую на каждом шаге вводится случайный параметр. Очень часто эту разновидность фракталов используют в компьютерной графике для изображения плазмы

Не так давно в изобразительном искусстве сформировалось новое направление – фрактальная монотипия или стохатипия, целью которых является получение изображения случайного фрактала.

Псевдокод для итерирования порождающих правил, когда используются правила F=newF, b=newb, X=newX, Y=newY и вероятность p:

Назначение: реализует правила F=newF, b=newb, X=newX, Y=newY.

Вход:

axiom (слово инициализация)

newF (порождающее правило)

newb (порождающее правило)

newX (порождающее правило)

newY (порождающее правило)

level (число итераций)

p (вероятность от 0 до 1)

Выход:

word (слово – результат)

Инициализация:

W=axiom

T=“” (пустое множество)

Шаги:

```
For [j=1, j≤StringLength[W], l=StringTake[W,{j}];
a=RandomReal;
If[l=="+", T=T<>"+"];
If[l=="-", T=T<>"-"];
If[l=="b", T=T<>newb];
If[l==”X”, T=T<>newX];
If[l==Y, T=T<>newY];
If[l==”[“, T=T<>”[“];
If[l==”]”, T=T<>”]”];
If[a<=p, T=T<>newF1];
If[a>p, T=T<>newF2];
j++];
```

W=T; i++]

W;

На рис.13 и рис.14 изображены стохастические фракталы с одинаковыми аксиомами и порождающими правилами после 4 итераций.

axiom = F

newF1 = F+[F-F]

newF2 = F-[F+F]

$\theta = \pi/8$

p = 0.7

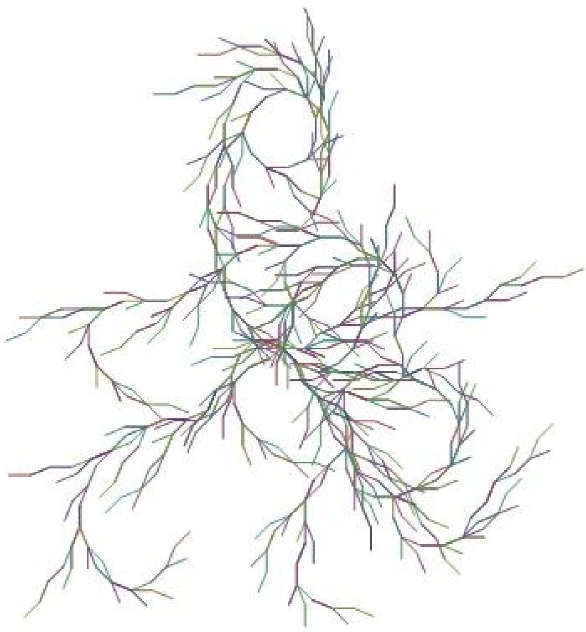


Рис.13

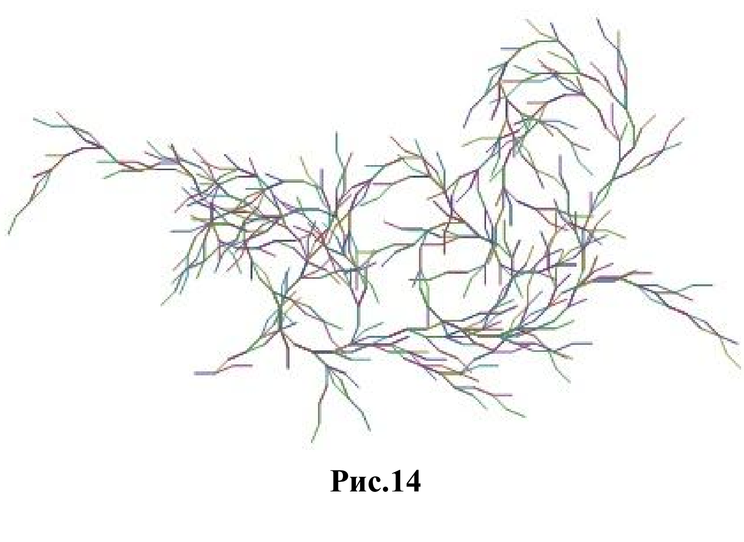
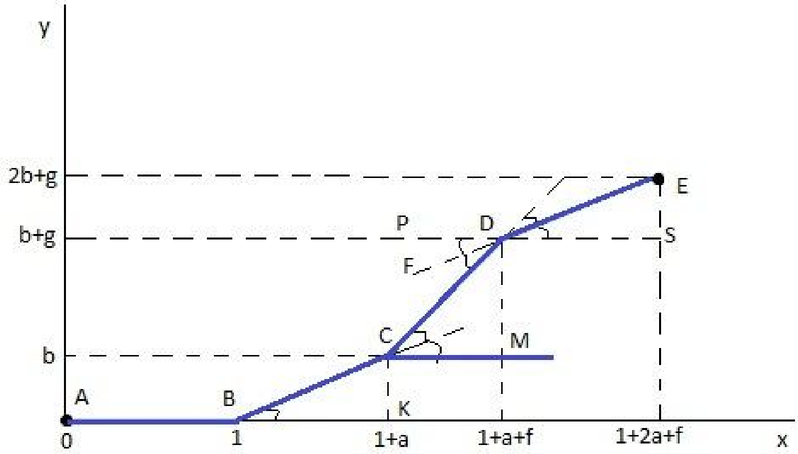


Рис.14

Найдем размерность.



1) N = 6

2) т.А и т.Е – максимально удаленные друг от друга точки.

Построим систему координат с началом в т.А.

A(0; 0), B(1; 0)

ΔBCK: $a = \cos\theta = \cos\frac{\pi}{8} = 0.873305$

$b = \sin\theta = \sin\frac{\pi}{8} = 0.382683$

=> C(1+a; b)

ΔCDM: $f = \cos 2\theta = \cos\frac{\pi}{4} = 0.707107$

$g = \sin 2\theta = \sin\frac{\pi}{4} = 707107$

=> D(b+g; 1+a+f)

ΔEDS: $ES = \sin\theta = b$

$DS = \cos\theta = a$

=> E(1+2a+f; 2b+g) => E(3,55487; 1,47247)

Находим расстояние между т.А и т.Е по формуле:

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} = \sqrt{3{,}55487^2 + 1{,}47247^2} = 3{,}84776$$

Следовательно, r = 3,84776

$$d = -\frac{\ln N}{\ln(1/r)} = -\frac{\ln 6}{\ln(1/3{,}84776)} = 1.3297$$

На рис.15 и рис.16 изображены стохастические фракталы с одинаковыми аксиомами и порождающими правилами после 5 итераций.

axiom = FX+[F]-[FF+XF]

newF1 = F-X

newF2 = F+X

newX = X

p = 0.3

$\theta = \pi/3$

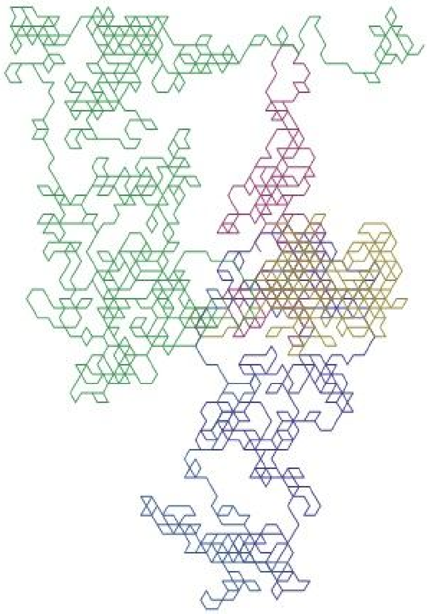


Рис15

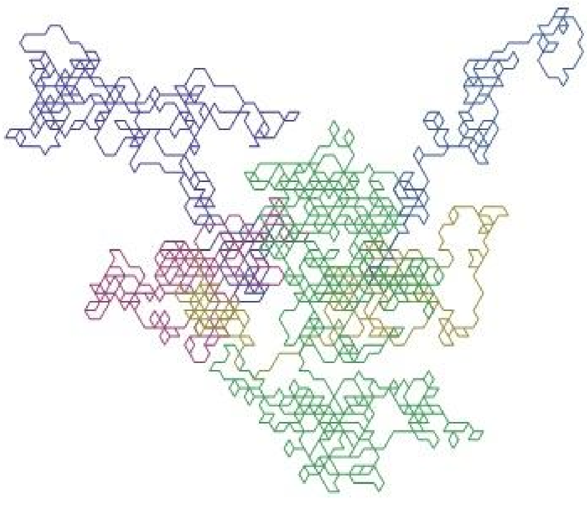
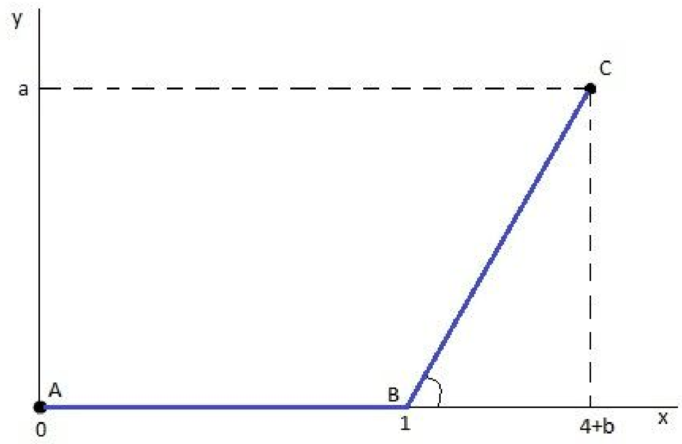


Рис. 16

Найдем размерность.



1) N = 2

2) т.А и т.С – максимально удаленные друг от друга точки.

Построим систему координат с началом в т.А.

$$a = \sin\theta = \sin\frac{\pi}{3} = 0.86602541$$

$$b = \cos\theta = \cos\frac{\pi}{3} = 0.5$$

=> C(4,5; 0,86602541)

Расстояние между т.А и т.С ищем по формуле:

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} = \sqrt{4.5^2 + 0.86602541^2} = 4.58258$$

Следовательно, r = 4,58258

$$d = -\frac{\ln N}{\ln(1/r)} = -\frac{\ln 3}{\ln(1/4.58258)} = 0.45534$$

Заключение

В последнее время применение L–систем получило особую актуальность для построения фрактальных объектов. Теория фракталов нашла широкое применение в физике, астрономии, биологии, а также экономических науках, в частности для анализа финансовых рынков. С точки зрения этих приложений использование фракталов рассматривается как наиболее перспективное современное направление математики.

В своей дипломной работе я исследовала самоподобные свойства фракталов, вычислила размерность подобия, изучила алгоритм построения фракталов на основе L-систем, реализовала алгоритм в программе Wolfram Mathematica.

Список использованных источников

1. Р.М.Кроновер. Фракталы и хаос в динамических системах. Основы теории. – М.: Постмаркет, 2000. – 352 с.

2. Трошин П.И. Моделирование фракталов в среде Maxima. Часть I. Фракталы на плоскости и в пространстве. – Казань: Казанский федеральный университет, 2014. – 92 с.

3. Scheinerman, Edward R. Invitation to Dynamical Systems / Edward R. Scheinerman // Department of Mathematical Sciences The Johns Hopkins University. – 1996.

4. Alligood, Kathleen T. Chaos - an introduction to dynamical systems / Kathleen Alligood, Tim Sauer, James A. Yorke // Springer-Verlag New York, Inc. – 1996.

5. Федер Е. Фракталы. Пер. с англ. -М.: Мир, 1991. - 254с.

6. Bian Runqiang Derivation of L-system Models from Measurements / Bian Runqiang, Phoebe Chen, Kevin Burrage, Jim Hanan, Peter Room, John Belward — 2002.

7. Bruno F. Lourenco L-Systems, scores, and evolutionary techniques / Bruno F. Lourenco, Jose C. L. Ralha, Marcio C. P. Brandao // Proceedings of the SMC 2009 — 6th Sound and Music Computing Conference, 23-25 July 2009 — Porto, Portugal, 2009.

8. Christian Jacob Genetic L-System Programming / Christian Jacob — Erlangen, Germany. – 1994.

9. H. Abelsonand Turtle geometry / H. Abelsonand, A.A. diSessa — Cambridge: M.I.T. Press. - 1982.

10. Jean-Eudes Marvie The FL-system a functional L-system for procedural geometric modeling / Jean-Eudes Marvie, Julien Perret, Kadi Bouatouch // Rennes, France. - 2005.

11. Zamir Arterial Branching within the Confines of Fractal L-System Formalism / M. Zamir // New York: The Rockefeller University Press. - 2001.

12. Przemyslaw Prusinkiewicz An L-System-Based Plant Modeling Language / Przemyslaw Prusinkiewicz, Jim Hanan and Radomir Mech // Berlin. - 2000.

Приложения

Текст программы для детерминированных L-систем

```
axiom = InputString["Введите аксиому"];
newF = InputString["Введите порождающее правило newF"];
newX = InputString["Введите порождающее правило newX"];
newY = InputString["Введите порождающее правило newY"];
newb = InputString["Введите порождающее правило newb"];
level = Input["Введите число итераций"];
W = axiom;
For[i = 1, i ≤ level, T = "";
 For[j = 1, j ≤ StringLength[W], l = StringTake[W, {j}];
  If[l == "+", T = T <> "+"];
  If[l == "-", T = T <> "-"];
  If[l == "F", T = T <> newF];
  If[l == "[", T = T <> "["];
  If[l == "]", T = T <> "]"];
  If[l == "b", T = T <> newb];
  If[l == "X", T = T <> newX];
  If[l == "Y", T = T <> newY];
  j++];
 W = T; i++]
W;
T = {0, 0};
stack = {};
alpha = Input["Введите начальное направление движения"];
theta = Input["Введите величину приращению по углу тета"];
For[j = 1; i = 1; path[i] = {T}, j ≤ StringLength[W], l = StringTake[W, {j}];
 If[l == "+", alpha = alpha + theta];
 If[l == "-", alpha = alpha - theta];
 If[l == "F", T = T + {Cos[alpha], Sin[alpha]};
  path[i] = path[i] ~Join~ {T}];
 If[l == "b", i++; T = T + {Cos[alpha], Sin[alpha]}; path[i] = {T}];
 If[l == "[", i++; stack = stack ~Join~ {Append[T, alpha]};
  path[i] = {T}];
 If[l == "]", i++; T = stack[[-1, 1 ;; 2]]; alpha = stack[[-1, 3]];
  stack = stack[[1 ;; -2]]; path[i] = {T}]; j ++]
Print[ListLinePlot[Table[path[k], {k, i}], Axes → False, PlotRange → All, AspectRatio → Automatic]]
```

Текст программы для стохастических L-систем

```
axiom = InputString["Введите аксиому"];
newF1 = InputString["Введите порождающее правило newF1"];
newF2 = InputString["Введите порождающее правило newF2"];
newX = InputString["Введите порождающее правило newX"];
newY = InputString["Введите порождающее правило newY"];
newb = InputString["Введите порождающее правило newb"];
level = Input["Введите число итераций"];
p = Input["Введите вероятность от 0 до 1"];
W = axiom;
For[i = 1, i ≤ level, T = "";
 For[j = 1, j ≤ StringLength[W], l = StringTake[W, {j}];
  a = RandomReal[];
  If[l == "+", T = T <> "+"];
  If[l == "-", T = T <> "-"];
  If[l == "[", T = T <> "["];
  If[l == "]", T = T <> "]"];
  If[l == "b", T = T <> newb];
  If[l == "X", T = T <> newX];
  If[l == "Y", T = T <> newY];
  If[a <= p, T = T <> newF1];
  If[a > p, T = T <> newF2];
  j++];
 W = T; i++]
W;
T = {0, 0};
stack = {};
alpha = Input["Введите начальное направление движения"];
theta = Input["Введите величину приращению по углу тета"];
For[j = 1; i = 1; path[i] = {T}, j ≤ StringLength[W], l = StringTake[W, {j}];
  If[l == "+", alpha = alpha + theta];
  If[l == "-", alpha = alpha - theta];
  If[l == "F", T = T + {Cos[alpha], Sin[alpha]};
   path[i] = path[i] ~Join~ {T}];
  If[l == "b", i++; T = T + {Cos[alpha], Sin[alpha]}; path[i] = {T}];
  If[l == "[", i++; stack = stack ~Join~ {Append[T, alpha]};
   path[i] = {T}];
  If[l == "]", i++; T = stack[[-1, 1 ;; 2]]; alpha = stack[[-1, 3]];
   stack = stack[[1 ;; -2]]; path[i] = {T}]; j++];
Print[ListLinePlot[Table[path[k], {k, i}], Axes → False, PlotRange → All, AspectRatio → Automatic]]
```