Министерство науки и высшего образования РФ
ФГБОУ ВО «Удмуртский государственный университет»
Институт математики, информационных технологий и физики

Лабораторная работа

"Вычисление фрактальной размерности"

Выполнил: студент 4 курса
группы ОАБ-02.03.01-42
Лаврентьев Михаил Андреевич

Ижевск
2022

Задание

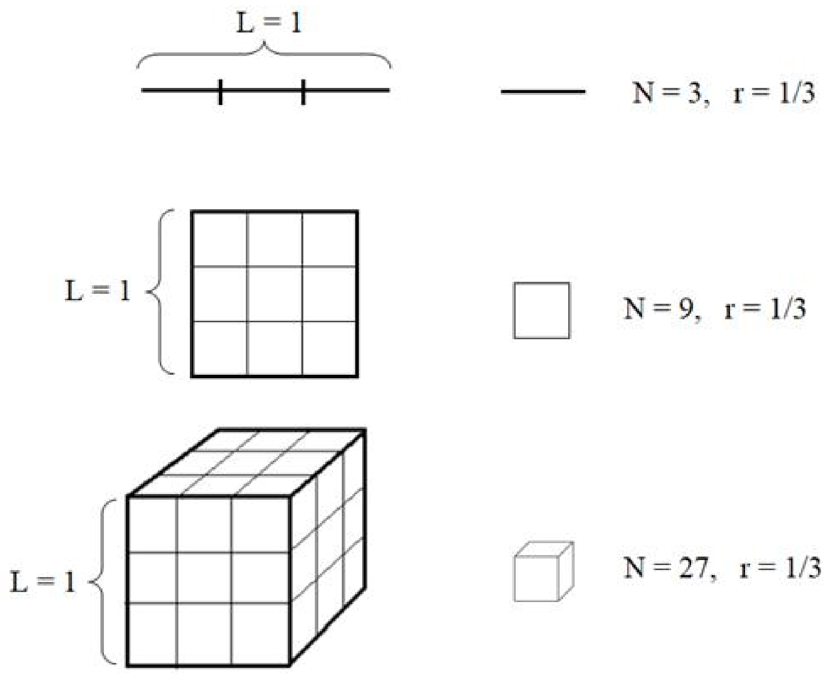
Написать программу для вычисления фрактальной размерности для
а) конструктивных фракталов

Конструктивные фракталы

Конструктивные или геометрические фракталы являются самыми наглядными, в них самоподобие видно сразу. Для построения геометрических фракталов характерно задание "основы" и "фрагмента", повторяющегося при каждом уменьшении масштаба. Методика: сначала изображается основа. Затем некоторые части основы заменяются на фрагмент. На каждом следующем этапе части уже построенной фигуры, аналогичные замененным частям основы, вновь заменяются на фрагмент, взятый в подходящем масштабе. Всякий раз масштаб уменьшается. Для получения самого фрактала нужно бесконечное число этапов. Меняя основу и фрагмент, можно получать разные геометрические фракталы. Рассмотрим построение классических фракталов.

Чтобы получить формулу расчета размерности фигуры, можно рассуждать следующим образом. Если взять линейный отрезок и разделить его на N = 3 равные части, то длина каждого фрагмента будет в три раза меньше исходной длины. Пусть начальная длина, условно равна 1, тогда длины фрагментов будут равны r = 1/3. Очевидно, что общая длина отрезка, равна:

$$N \cdot r = 1$$



Проделаем ту же операцию с квадратом. Каждую из его сторон длиной в одну единицу также разделим на три равные части. То есть, линейные размеры маленьких квадратов будут равны r = 1/3. И таких квадратов всего N = 9. В этом случае площадь большого квадрата будет равна:

$$N \cdot r^2 = 1$$

Как вы уже догадались, если взять куб и каждую из его сторон также разбить на три равных отрезка, то получим N = 27 кубиков со сторонами r = 1/3. Тогда объем куба можно выразить формулой:

$$N \cdot r^3 = 1$$

То есть, смотрите, размерность фигуры проявляется как степень коэффициента подобия r. И в общем случае можно записать:

$$N \cdot r^d = 1$$

Чтобы из этой формулы выразить степень d, прологарифмируем левую и правую части этого уравнения, получим:

$$\begin{aligned}
&\log\left(N \cdot r^d\right) = \log 1 \\
&\log N + d \cdot \log r = 0 \\
&d \cdot \log r = -\log N \\
&-d \cdot \log\left(1/r\right) = -\log N \\
&d = \frac{\log N}{\log 1/r}
\end{aligned}$$

Логарифм можно взять по любому основанию, обычно это 2, 10 или e – натуральный логарифм. Величину d называют **фрактальной размерностью** или **размерностью подобия**.

Множество Кантора

В 1883 году немецкий математик Георг Кантор описал множество, которое теперь называют его именем. Оно конструируется следующим образом. Берется единичный отрезок. Этот отрезок делится на три части, средняя удаляется. Получается два отрезка длиной ⅓ каждый. Далее с каждым из отрезков проделывается та же процедура (удаление серединной части); и так до бесконечности. Полученное множество состоит из точек отрезков не удаленных алгоритмом и является самоподобным (его фрагмент выглядит как все множество в целом.

Построение множества Кантора приведем с помощью Maple

```
> restart; with(plots) :
```

Множество Кантора

Процедура ***basic*** получает на входе координаты концов исходного отрезка и возвращает концевые точки двух производных отрезков

```
> n := 3;
  basic := proc(p1, p2)
    local dx, p3, p4, p5, p6 :
    dx := (p2[1] - p1[1]) / n :
    p3 := [p1[1] + dx, p1[2]] :
    p4 := [p2[1] - dx, p2[2]] :
    p1, p3, p4, p2;
  end proc:
```

$$n := 3 \tag{1.1}$$

```
> pt := basic([0, 0], [1, 0]);
```

$$pt := [0, 0], \left[\frac{1}{3}, 0\right], \left[\frac{2}{3}, 0\right], [1, 0] \tag{1.2}$$

Процедура ***m_k*** берет список точек, которые представляют собой концы отрезков исходного множества, и возвращает список, в котором каждый отрезок заменен двумя, посредством процедуры ***basic***

```
> m_k := proc( )
    local i, lin, k_0, j :
    k_0 := NULL :
  for i by 2 to nargs - 1 do
      k_0 := k_0, basic(args[i], args[i + 1]) :
  end do:
  k_0 :
  end proc:
```

Например, обратимся к ***m_k*** со списком ***pt***

```
> m_k(pt);
```

$$[0,0],\left[\frac{1}{9},0\right],\left[\frac{2}{9},0\right],\left[\frac{1}{3},0\right],\left[\frac{2}{3},0\right],\left[\frac{7}{9},0\right],\left[\frac{8}{9},0\right],[1,0] \qquad (1.3)$$

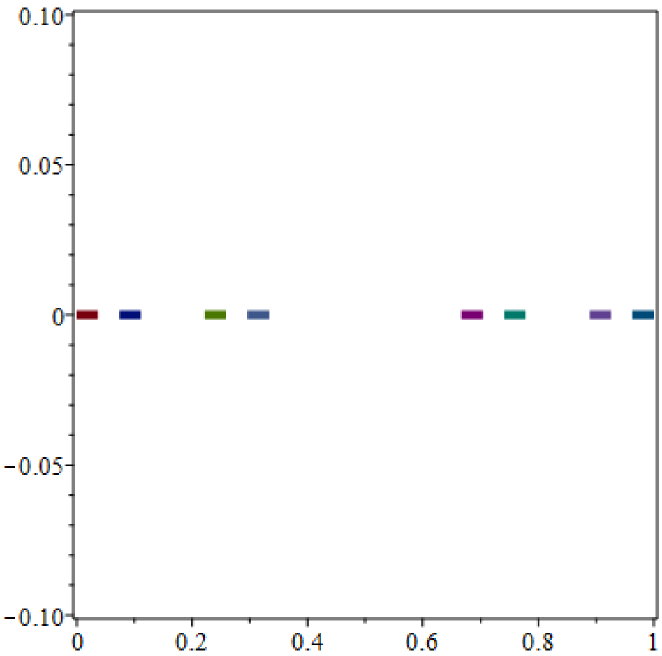
Теперь можно определить ***n-й*** шаг построения множества Кантора

```
> kantor := proc(n :: integer)
    local i, j, lin, kan, kant :
    lin := [0, 0], [1, 0] :
    for i to n do lin := m_k(lin) end do:
   j := 1 :
   for i by 2 to nops([lin]) do
      kan[j] := [lin[i], lin[i + 1]] :
      j := j + 1 :
   end do:
   kant := convert(kan, list);
   plot(kant, axes = box, thickness = 5, view = [0 ..1, -0.1 ..0.1]) :
   end proc:
```

Обратившись к процедуре ***kantor***, сделаем три шага построения множества

```
>
  K := 3;
  kantor(K);
```

$$K := 3$$



Расчитаем фрактальную размерность

```
> 
```

$$N := 2^K;$$

$$r := \left(\frac{1}{n}\right)^K;$$

$$d := \frac{\ln(N)}{\ln\left(\frac{1}{r}\right)};$$

$evalf(d);$

$$N := 8$$

$$r := \frac{1}{27}$$

$$d := \frac{\ln(2)}{\ln(3)}$$

$$0.6309297534 \tag{1.4}$$

Кривая Коха

Норвежский математик Гельге фон Кох в 1904 году описал одну кривую линию, необычную тем, что она не имеет касательной ни в одной своей точке. Эта кривая определяется как предел последовательности кривых, порождаемых процедурами, аналогичными описанными выше (для множества Кантора). Точно так же рассматривается отрезок единичной длины, из которого удаляется средняя треть. Отличие в том, что разорванный отрезок дополняется двумя отрезками длиной ⅓ , превращаясь в ломаную из четырех звеньев. Далее та же самая операция применяется к каждому из отрезков ломаной. Следующий набор процедур позволяет отображать замкнутую кривую (снежинку) Коха.

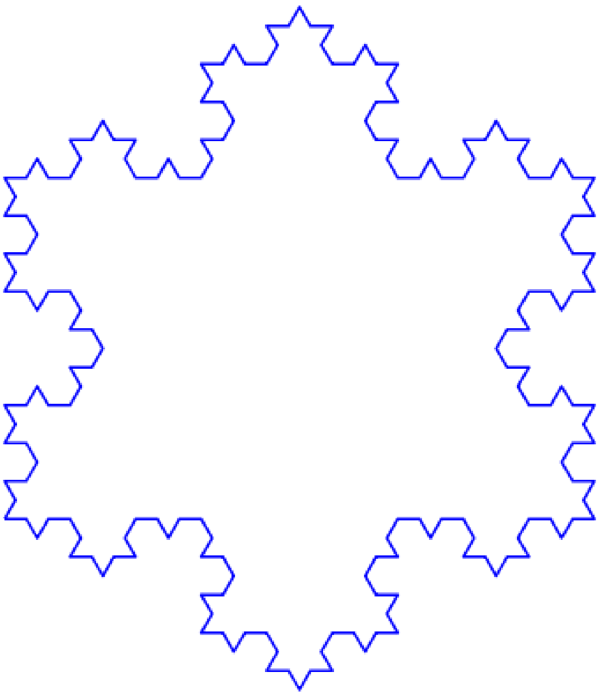
Построим снежинку Коха с помощью Maple

```
> restart; with(plots) :
```

Снежинка Коха

```
> basic := proc(p1, p2)
    local dx, dy, p3, p4, p5 :
    dx := (p2[1] - p1[1])/3 :   dy := (p2[2] - p1[2])/3 :
    p3 := [p1[1] + dx, p1[2] + dy] :
    p4 := [p1[1] + 2·dx, p1[2] + 2·dy] :
    p5 := [p1[1] + 1.5·dx - dy·sqrt(3)/2,  p1[2] + 1.5·dy + dx·sqrt(3)/2] :
   p3, p5, p4, p2;
  end proc:
> flake := proc(fl)
    local i, curve;
    curve := fl[1] :
    for i to nops(fl) - 1 do
      curve := curve, basic(fl[i], fl[i + 1]) :
    end do:
  end proc:
> snowflake := proc(n)
    local i, curve :
    curve := [[-0.5, 0], [0, sqrt(3)/2], [0.5, 0], [-0.5, 0]] :
    for i from 2 to n do   curve := [flake(curve)] od;
  plot(curve, color = blue, scaling = constrained, axes = none) :
  end proc:
```

```
> snowflake(4);
```



Расчитаем фрактальную размерность

```
> K := 4 ; #порядок кривой
r := (1/3)^K ; #длина отрезка
N := 4^K; #кол-во полученных отрезков
d := ln(N)/ln(1/r);
evalf(d);
```

$$K := 4$$

$$r := \frac{1}{81}$$

$$N := 256$$

$$d := \frac{2\ln(2)}{\ln(3)}$$

$$1.261859507 \qquad \textbf{(1.1)}$$

Кривая Пеано

Особенность следующей непрерывной кривой, которую описал в 1891 году Давид Гильберт, состоит в том, что заполняет собою некоторое заданное пространство, таким образом, в пределе можно сказать, что ее длина составляет определенное количество квадратных метров.

```
> restart; with(plots) :
```

Кривая Пеано

```
> basic := proc(p1, p2, p3)
    local p4, p5, p6, p7, p8, p9;
  p4 := 0.5·(p1 + p2);
  p9 := 0.5·(p2 + p3);

  p5 := p4 + (p2-p9);
  p6 := p2 + (p2 - p9);
  p7 := p2 + (p4 - p1);
  p8 := p9 + (p4 - p1);
  p4, p5, p6, p2, p7, p8, p9, p3;
  end proc:
> peano := proc(fl)
    local i, cur;
  cur := [fl[1]];
  for i by 2 to nops(fl) - 2 do
     cur := [op(cur), basic(fl[i], fl[i + 1], fl[i + 2])]
  end do;
  end proc:
```

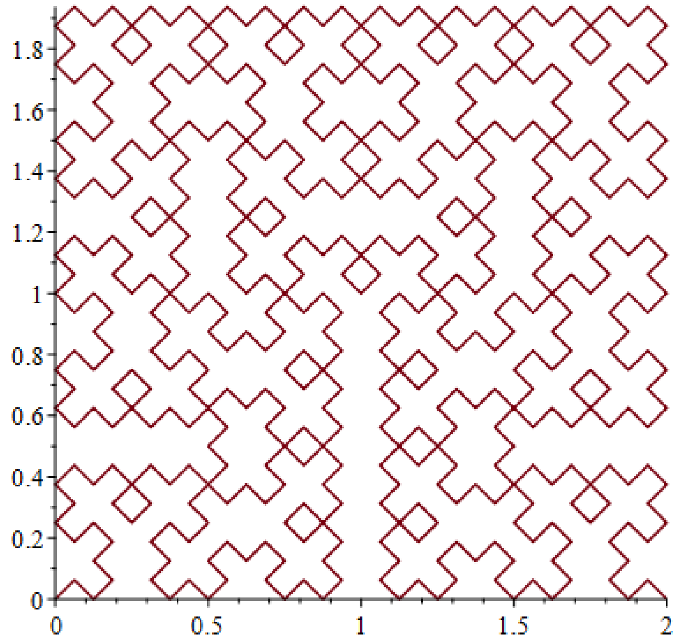
Начальная ломаная:

```
> fl := [[0, 0], [1, 1], [2, 0]] :
```

Выполним четыре шага построения кривой Пеано

```
> for i to 4 do fl := peano(fl) end do:
```

> $plot(fl, style = line, scaling = constrained);$



Расчитаем фрактальную размерность

> $K := 4$; #порядок кривой

$r := \left(\frac{1}{2}\right)^K$; #длина отрезка

$N := 4^K$; #кол-во полученных отрезков

$d := \frac{\ln(N)}{\ln\left(\frac{1}{r}\right)}$;

$evalf(d)$;

$$K := 4$$

$$r := \frac{1}{16}$$

$$N := 256$$

$$d := 2$$

$$2. \quad (1.1)$$