



## I. Introduction

Niels Fabian Helge Von Koch, (Suédois 1870-1924) est un mathématicien qui a donné son nom à l'une des premières fractales : le flocon de Koch ou flocon de neige.

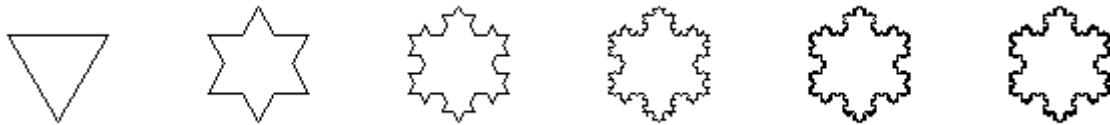
Il a décrit le flocon auquel on a donné son nom en 1904 dans un article intitulé Sur une courbe continue sans tangente, obtenue par une construction géométrique élémentaire.

Le mot « fractal » vient du latin « fractus » qui signifie « brisé ». En effet, une figure fractale est une figure géométrique dont la structure est invariante par changement d'échelle, en zoomant sur une partie de la figure, on peut retrouver toute la figure, on dit qu'elle est auto similaire.

On attribue la découverte des fractales à un polytechnicien français, Benoît Mandelbrot (1924-2010). Ses premières recherches datent de 1964 où il emploie le terme de self-similar lors d'une étude réalisée chez IBM. Mais c'est en 1975 qu'il expose ses travaux et donne le nom de « fractale » dans son ouvrage « Les objets fractals ».

## II. Définition

Le flocon de Von Koch est défini à partir d'un triangle équilatéral de périmètre 1 (étape  $i=0$ ) auquel on construit extérieurement au triangle de manière régulière trois triangles équilatéraux de côté le tiers du triangle précédent (étape  $i=1$ ) on répète la construction aux étapes suivantes.



## III. Travail à faire

1. Pour tout entier  $i$  on note :

- $n_i$  le nombre de côtés de la figure à l'étape  $i$
- $c_i$  la longueur d'un côté de la figure à l'étape  $i$
- $a_i$  l'aire de la figure à l'étape  $i$
- $p_i$  le périmètre de la figure à l'étape  $i$ .

Compléter l'algorithme et le programme sur Python qui donnent les quatre variables de la figure de Von Koch à l'étape  $I$ .

$n \leftarrow 3$

$c \leftarrow \frac{1}{3}$

$a \leftarrow \dots$

$p \leftarrow \dots$

Pour  $i$  variant de 1 à  $I$  faire

$c \leftarrow \dots$

$a \leftarrow \dots$

$n \leftarrow \dots$

$p \leftarrow \dots$

Fin pour

---

```

1 #flocon de Von koch
2 from math import*
3
4 def VonKoch(I):
5     n=3
6     c=1/3
7     A=...
8     p=...
9     for i in range(1,I+1):
10        c=...
11        A=...
12        n=...
13        p=...
14     return (n, c, A, p)

```

---

vonkoch.py

2. Justifier les expressions suivantes :

- $n_i = 3 \times 4^i$

- $A_i = \frac{2\sqrt{3}}{45} - \frac{3\sqrt{3}}{180} \times \left(\frac{4}{9}\right)^i$  (voir annotation 1)

- $c_i = \frac{1}{3^{i+1}}$

- $p_i = \left(\frac{4}{3}\right)^i$

*note 1* : calculer de deux manières  $(a_i - a_{i-1}) + (a_{i-1} - a_{i-2}) + \dots + (a_1 - a_0)$ .

3. Donner les limites des suites  $(a_i)$  et  $(p_i)$

4. (pour aller plus loin : programmer la figure)

---

```

1 from turtle import *
2
3 def koch(longueur, n):
4
5     if n == 0:
6         forward(longueur)
7     else:
8         koch(longueur/3, n-1)
9         left(60)
10        koch(longueur/3, n-1)
11        right(120)
12        koch(longueur/3, n-1)
13        left(60)
14        koch(longueur/3, n-1)
15
16 def flocon(taille, etape):
17     koch(taille, etape)
18     right(120)
19     koch(taille, etape)
20     right(120)
21     koch(taille, etape)
22
23
24 for i in range(0,6):
25     up()
26     setheading(0)
27     goto(-280+i*100, 100)
28     down()
29     speed(0)
30     flocon(50, i)
31 hideturtle()
32 done()

```

---

vonkochfigure.py

---

## Correction du programme

---

```
1
2 from math import*
3
4 def VonKoch(I):
5     n=3
6     c=1/3
7     A=c**2*sqrt(3)/4
8     p=n*c
9     for i in range(1,I+1):
10        c=c/3
11        A=A+n*c**2*sqrt(3)/4
12        n=n*4
13        p=n*c
14    return (n,c,A,p)
```

---

vonkochcorrection.py

