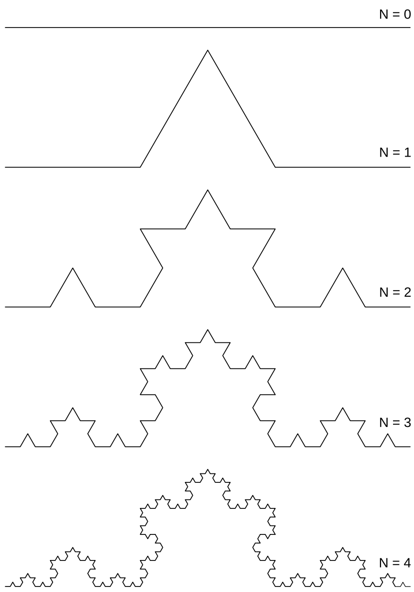
AUTOUR DE LA COURBE DE KOCH

1. Énoncé



La courbe de Koch (prononcé « *Korr* ») est une fractale célèbre, inventée par le mathématicien suédois Helge von Koch en 1904. Trois de ces courbes autosimilaires forment le flocon de Koch.

À partir d'un segment de longueur L, on crée 4 segments de longueur $\frac{L}{3}$, disposés comme sur l'illustration. À chaque étape, la longueur d'un segment est divisée par 3, le nombre de segment est multiplié par 4. Ainsi, la longueur totale de la courbe suit une progression géométrique de raison $\frac{4}{3}$.

Dans ce document, nous allons tracer la courbe de Koch grâce au module `turtle` de Python et laisser en exercices plusieurs variantes.

2. Code

Koch.py

```python
from turtle import *
from re import sub

N = 4

setup(width=800, height=600)
hideturtle()
penup()
setpos(-729/2, -50)
pendown()

vk = 'A'

for _ in range(N):
    vk = sub('A', 'AGADAGA', vk)

L = 729/3**N                # Longueur d'un segment
vk = sub('A', 'fd(L);', vk)     # Avancer
vk = sub('G', 'lt(60);', vk)    # Gauche
vk = sub('D', 'rt(120);', vk)   # Droite

tracer(0)
exec(vk)
update()
getscreen().getcanvas().postscript(file="Koch.eps")
done()
```

3. Explications

Pour commencer, nous importons le module `turtle` et la fonction `sub` (*substitute*) du module `re` (*regular expressions*). Nous définissons une variable `N` qui correspond à l'étape que l'on souhaite représenter. Quelques lignes permettent de configurer la fenêtre de dessin de `turtle`.

Le principe du programme est de démarrer par l'instruction fondamentale : « avancer » (étape 0), puis de faire une boucle remplaçant `N` fois chaque instruction « avancer » par le mouvement décrit dans le processus (ligne 15).

Nous utilisons pour cela 3 lettres : `A` pour avancer, `G` pour tourner à gauche d'un angle de 60° et `D` pour tourner à droite d'un angle de 120°.

Ces lettres ne sont pas liées au module `turtle`, nous aurions pu en choisir d'autres. Une fois le mouvement complet de toute la courbe décrit par ces trois lettres, elles sont remplacées par les vraies commandes de `turtle`.

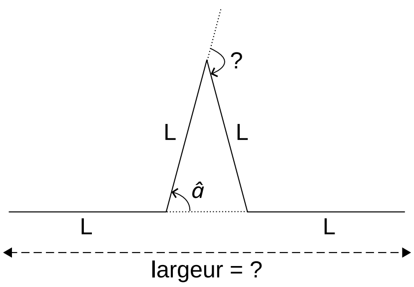
La longueur d'un segment est donnée par `L = 729/3**N`. Le nombre de 729 a été choisi car c'est une puissance de 3 ($729 = 3^6$) et c'est un nombre cohérent de pixels pour une largeur d'image. Comme la longueur d'un segment est égale à $\dfrac{1}{3^N}$, la variable `L` est un entier jusqu'à `N = 6` (et choisir un `N>5` est trop élevé ici).

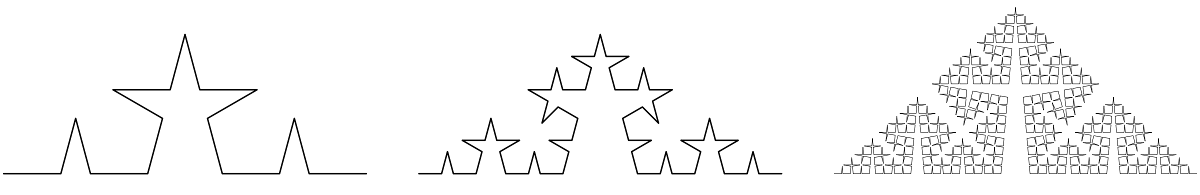
`tracer(0)` permet un tracé instantané. La courbe est sauvegardée au format `.eps` qui peut être converti, avec un logiciel externe (comme Inkscape) en `.svg` ou `.pdf` pour du vectoriel, ou en `.png` ou `.jpg` pour du matriciel.

4. Courbe de Cesàro

La courbe de Cesàro est une généralisation de la courbe de Koch. Au lieu de tourner d'un angle de 60°, on tourne d'un angle $\widehat{a}$ de mesure strictement comprise entre 60° et 90°.

Après avoir préparé les questions ci-dessous, s'inspirer du code précédent pour afficher une courbe de Cesàro à l'étape `N`, pour une rotation à gauche d'angle `a`.





$\hat{a}$ = 75°
N = 2

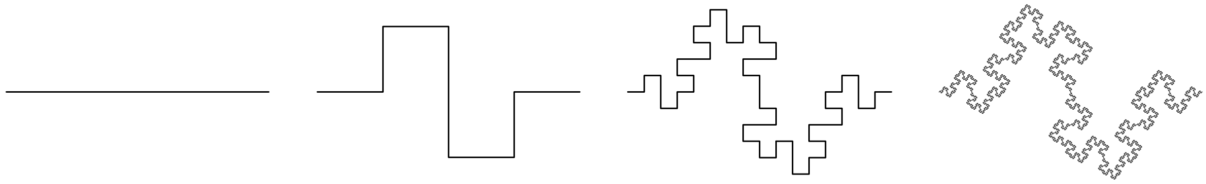
$\hat{a}$ = 75°
N = 3

$\hat{a}$ = 85°
N = 5

Questions préliminaires :

- Si l'angle de rotation à gauche est égal à $\widehat{a}$, de combien sera celui de la rotation à droite ?
- À l'étape 1, que vaut la largeur de la courbe, en fonction de L et de $\widehat{a}$?
- À l'étape N, combien doit valoir la longueur L_N d'un seul segment, en fonction de $\widehat{a}$ et de N, pour avoir une largeur de courbe fixée à 750 pixels ?

5. Courbe quadratique de Koch (type 2)



On réalise maintenant une variante du processus : chaque segment de longueur L est partagé en 8 segments de longueur $\frac{L}{4}$ (il est plus simple de considérer que le segment central de longueur $\frac{L}{2}$ correspond en fait à deux segments). Tracer la courbe en choisissant une valeur pertinente pour `L`.

6. Flocon de Koch

Quelle ligne changer dans le code initial pour obtenir le flocon de Koch ?

SOLUTIONS

Cesaro.py

```
a = 85  # mesure en degrés
# une largeur = (2+2*cos(a*pi/180))*L avec cos en radians. D'où :
L = 750 / (2+2*cos(a*pi/180))**N   # donnera une largeur de 750 pixels
C = sub('A', 'fd(L);', C)
C = sub('G', 'lt(a);', C)
C = sub('D', 'rt(2*a);', C)  # rotation à droite de 2*a
```

Koch-quadratique.py

```
setup(width=1200, height=1200)
(...)
vkq = 'A'

N = 4

for _ in range(N):
   vkq = sub('A', 'AGADADAAGAGADA', vkq)

L = 1024/4**N

vkq = sub('A', 'fd(L);', vkq)
vkq = sub('G', 'lt(90);', vkq)
vkq = sub('D', 'rt(90);', vkq)
```

Koch-flocon.py

```
vk = 'ADADAD' # au lieu de vk = 'A' (et changer la fenêtre)
```