

I. Le triangle de Sierpinsky

Modules graphiques

Plusieurs modules graphiques existent sur Python :

- Tkinter est le module graphique d'origine de Python. Il permet la gestion de fenêtres avec graphiques, menus, boutons, ...
- Pygame (non installé avec Pyzo) est orienté vers la création de jeux.
- Matplotlib est le module graphique adapté aux mathématiques.

Matplotlib contient le sous-module pyplot (qui contient notamment la fonction plot). Le module pylab, qui s'installe avec matplotlib, combine les fonctionnalités de pyplot et de numpy.

Dans ce TP, nous utilisons pylab (`import pylab as pl`).

Dessin d'un triangle

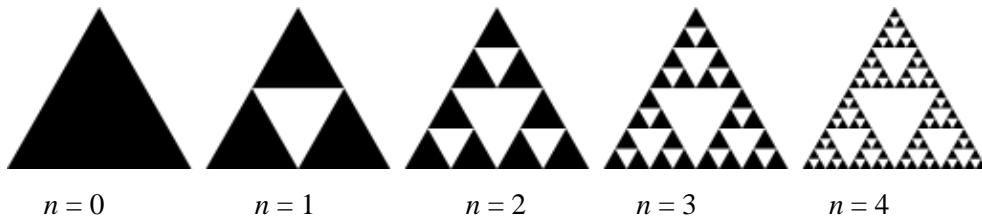
`plot([x1, x2, ..., xn],[y1, y2, ..., yn])` crée automatiquement la fenêtre graphique avec les axes et relie les points de coordonnées (x_i, y_i) . Tester :

```
pl.plot([0,1,0.5,0],[0,0,3**0.5/2,0])
```

`luc=pl.Polygon([[x1,y1],[x2,y2],...,[xn,yn]])` crée un polygone de sommets (x_i, y_i) nommé luc que l'on affiche avec la commande `pl.gca().add_patch(luc)`. Tester :

```
triangle=pl.Polygon([[0,0],[1,0],[0.5,3**0.5/2]],color='black')
pl.gca().add_patch(triangle)
```

Voici des triangles de Sierpinsky d'ordre n :



1. Dessiner un triangle de Sierpinsky d'ordre 0.
2. Rajouter un triangle blanc pour dessiner un triangle de Sierpinsky d'ordre 1.

Un triangle de Sierpinsky est formé de plusieurs triangles noirs.

On décide d'identifier chaque triangle noir par un triplet (x, y, c) où (x, y) désignent les coordonnées de son coin inférieur gauche et c la longueur de son côté.

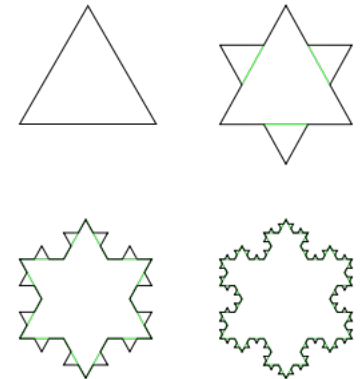
Dans les questions 3 et 4, on tracera le triangle noir avant l'appel de la fonction.

3. Écrire la fonction d'entête `sierpinski(x, y, c)` « découpant » dans le triangle noir (x, y, c) un triangle blanc reliant les milieux des côtés. Utiliser cette fonction pour construire le triangle de Sierpinsky d'ordre 2.
4. Transformer la fonction `sierpinski` en fonction récursive traçant le triangle de Sierpinsky d'ordre n .

5. Écrire une fonction `sierpinski` traçant le triangle de Sierpinsky d'ordre n non plus en découpant des triangles blancs dans un triangle noir mais en traçant directement les triangles noirs.

II. Le flocon de von Koch

On construit le flocon de von Koch F_n par récurrence. F_1 est un triangle équilatéral de côté 1. Pour obtenir F_{n+1} à partir de F_n , on remplace, dans chaque côté de F_n , le tiers central par deux segments de même longueur que celui-ci de manière à ce que F_{n+1} soit un polygone convexe.



Ecrire la fonction `vonKoch` appelée par le programme suivant, qui trace le flocon de von Koch d'ordre n .

```
import pylab as pl

def vonKoch(n):
    .....

n=int(input("Entrer l'ordre : "))
pl.axes()
pl.xlim([-0.2,1.2])
pl.ylim([-0.4,1])
vonKoch(n)
pl.show()
```