

TP d'informatique n°14

Graphiques avec Python

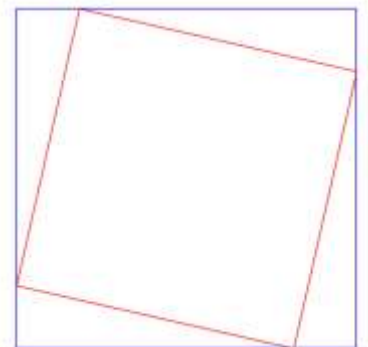
L'objectif du TP est d'apprendre à construire des graphes avec Python

1) Fonctionnalités de base de matplotlib

1) Saisir les commandes suivantes :

```
import matplotlib.pyplot as plt
X , Y = [1, 3, 9, 6] , [2, 8, 14, -1]
plt.plot(X, Y, 'r*-')
X2, Y2 = [2, 5, 7, -6] , [3, 8, -4, 2]
plt.plot(X2, Y2, 'g+:')
plt.show()
```

- a) Exécutez les commandes précédentes. Que constatez-vous ?
- b) Comment peut-on corriger les commandes précédentes pour tracer un carré (fermé)?
- c) On suppose qu'un point peut être représenté par une liste de 2 nombres : la liste de ses coordonnées. Ecrire une fonction, appelée `polygon`, prenant comme argument une liste de points (donc une liste de listes...) et qui renvoie le tracé du polygone (fermé) reliant les points de la liste donnée. On utilisera la fonction `plt.plot` pour définir le tracé et la fonction `plt.draw()` pour "forcer le tracé"
- d) Etant donné un carré ABCD et un réel p compris entre 0 et 1, on lui fait subir une transformation géométrique consistant à :
 - i. Considérer le point A_1 barycentre de (A,p) , $(B,1-p)$
 - ii. Considérer de même B_1 , C_1 et D_1 les barycentres respectifs de $\{(B,p), (C,1-p)\}$, $\{(C,p), (D,1-p)\}$ et $\{(D,p), (A,1-p)\}$
 - iii. Construire le carré $A_1B_1C_1D_1$



Par exemple, avec $p = 0.9$, cette transformation change le carré bleu en le carré rouge...

Ecrire une fonction `transfocarre` qui prend comme argument une liste L de points (formant un carré) et un flottant p , et qui donne la liste des sommets du carré obtenu par la transformation.

- e) A l'aide d'une boucle `for`, effectuer le tracé, sur une même figure, de plusieurs itérations de la transformation précédente en partant d'un carré donné. Vous pourrez faire varier le

paramètre p et le nombre de carrés tracés. Vous pouvez même agir sur les couleurs si vous le souhaitez...

f) Amélioration : Question à ne traiter que si vous êtes "bien en avance"


Matplotlib permet de créer des graphiques dynamiques à l'aide du mode interactif.

Dans un nouveau fichier source, reprenez les définitions des fonctions `polygon` et `transfocarre`, "ouvrez" une figure interactive (par l'instruction `plt.figure(5)` par exemple), indiquez l'ouverture d'une session interactive (par l'instruction `plt.ion()`), reprenez une des séquences de création de plusieurs carrés dans une même figure, terminez chacune des itérations de la boucle par `plt.draw()`, enfin, après la boucle `for`, terminez par les instructions `plt.ioff()` puis `plt.show()`


2) Séparation d'une figure entre plusieurs sous figures

- 1) On considère la transformation de Von-koch, qui consiste à, étant donné un segment $[AB]$, on découpe ce segment en 3 en créant les points C et D , on considère le sommet E du triangle équilatéral direct CED . Enfin on relie les points $ACEDB$. Par exemple si A est à gauche et B à droite dans la première figure, on trouve la ligne brisée de la seconde figure. On peut montrer que, si les coordonnées de A sont (x,y) et celles de B sont (x', y') , celles de E sont : $(x+x')/2 - (y - y')*(\sqrt{3})/6$ et $(y+y')/2 + (x - x')*(\sqrt{3})/6$. On supposera qu'un point est représenté par la liste de ses deux coordonnées. Ecrire une fonction `trans_seg` qui prend comme argument 2 points A et B , et qui retourne la liste des 5 points $[A,C,E,D,B]$

$n=0$



$n=1$


- 2) Ecrire une fonction `von_koch` qui prend comme argument une liste de plusieurs points `Liste1` et qui retourne la liste des points obtenue à partir de `Liste1` en insérant entre 2 points consécutifs de `Liste1` les trois points obtenus par la transformation de la question précédente.
- 3) En reprenant en grande partie les instructions de la fonction `polygon` de l'exercice précédent, écrire une fonction `trace_ligne` qui prend comme argument une liste de points et retourne le tracé de la ligne brisée reliant ces points.
- 4) En partant de la liste `Liste = [[0,0], [1.0], [0.5, sqrt(3)/2], [0,0]]`, puis en effectuant plusieurs réaffectations successives du type `Liste = von_koch(Liste)`, tracer plusieurs itérations du flocon de Von Koch.
- 5) Une fois l'instruction `plt.figure(1)` validée, l'instruction `plt.subplot(npq)` découpe la fenêtre de graphique en n lignes et p colonnes, et concerne la q -ième partie de ce tableau. Ecrire des instructions permettant de tracer sur une même fenêtre mais sur des graphes différents les 6 premières étapes du flocon de Von Koch.