

TP 13 : tracés de fonctions et fractales

Dans ce TP, nous utiliserons la fonction `plot` pour tracer des fonctions simples et pour construire des courbes fractales. La fonction `plot` est définie dans le module `pyplot` de la bibliothèque `matplotlib`. Les modules `math` et `numpy` seront aussi nécessaires pour avoir accès aux fonctions et grandeurs mathématiques de bases (π , \sin , \cos ...). Ainsi au début de votre programme vous devez importer ces bibliothèques en tapant les commandes suivantes :

```
#importation de toutes les fonctions de la bibliothèque math
from math import *

#importation de toutes les fonctions de la bibliothèque pyplot
from matplotlib.pyplot import *

#importation de toutes les fonctions de la bibliothèque numpy
from numpy import *
```

Partie 1 : utilisation basique de la fonction `plot`

La fonction `plot` s'écrit ainsi: `plot(x, y, motif)` où `x` et `y` sont des listes, appelés vecteurs, de même longueur. La fonction `plot` trace sur l'écran les points de coordonnées $(x[i], y[i])$ où i est l'indice de l'élément `x` et `y`. L'argument `motif` est optionnel et permet de définir la couleur et le style des points.

Exercice 1 :

Taper les lignes suivantes et tester les :

```
plot([1,2,3],[2,4,6],'.')
plot([1,2,3],[2,4,6],'r-')
plot([1,2,3],[2,4,6],'g*')
plot([1,2,3],[2,4,6],'-')
plot([1,2,3],[4,5,2])
```

Exercice 2:

La fonction `linspace(a,b,n)` permet de créer une liste de `n` flottants régulièrement espacés entre `a` et `b`. Taper le code suivant et tester le :

```
x=linspace(0,10,10)
y=3*x
plot(x,y)
```

Exercice 3 :

Ecrire un programme qui permet de tracer la fonction `sin` entre 0 et 2π .

Exercice 4 :

Modifier votre programme précédent pour tracer à l'écran la fonction `sin` en faisant varier le nombre d'éléments des vecteurs `x` et `y` de 10 à 100. Conclure.

Partie 2 : création d'une tortue graphique

Une tortue graphique est un petit programme qui permet de dessiner à l'écran. Le principe de base est le déplacement dans l'écran d'un point appelé « tortue ». La tortue, comme l'animal, peut uniquement avancer et tourner. Le dessin est obtenu en traçant la trajectoire suivie par la tortue.

Le déplacement de la tortue sera obtenu par la lecture d'une chaîne de caractère contenant les caractères A, + et - :

- si la tortue lit "+" alors elle tourne d'un angle α fixé dans le sens trigonométrique. Dans ce cas, la tortue ne change pas de coordonnées. Seule son orientation change, elle passe de θ à $\theta + \alpha$;
- si la tortue lit "-" alors elle tourne d'un angle α fixé dans le sens horaire. Dans ce cas, la tortue ne change pas de coordonnées. Seule son orientation change, elle passe de θ à $\theta - \alpha$;
- si la tortue lit "A" alors elle avance d'un incrément. Dans ce cas, les coordonnées de la tortue changent. Elle passe de (x, y) à $(x + \cos\theta; y + \sin\theta)$ où θ est l'orientation de la tortue.

Exercice 5 :

Supposons que la position de la tortue soit définie par la liste de 3 paramètres : $P = [x, y, \theta]$.

Nous souhaitons créer une fonction qui renvoie la nouvelle position de la tortue après la lecture du caractère c qui peut être A, + ou -.

Ecrire et commenter la fonction `deplacement(P, c, alpha)` qui renvoie la nouvelle position de la tortue : $P = [x', y', \theta']$.

Tester votre fonction :

```
>>deplacement([0,0,0],'+',pi/6)
P=[0,0,0.523]
>>deplacement([1,2,pi/4],'A',0)
P=[1.707,2.707,0.785]
```

Exercice 6 :

Pour définir la trajectoire de la tortue, nous souhaitons obtenir 2 listes LX et LY contenant les coordonnées successives de la tortue. Ces 2 listes doivent être obtenues à partir :

- de la chaîne de caractère `str` (contenant une suite de A, + et -) que doit lire la tortue ;
- d'un angle `alpha` qui définira le changement d'orientation de la tortue.

Ecrire et commenter la fonction `coordonnees(str, alpha)` qui lit un à un les caractères de la chaîne `str` et qui renvoie une liste L contenant les 2 listes de coordonnées LX et LY. Le point de départ est défini par la liste $[0, 0, 0]$.

Tester votre fonction :

```
>>coordonnees('AA+AA-AA',pi/2)
[[0, 1.0, 2.0, 2.0, 2.0, 3.0, 4.0],
 [0, 0.0, 0.0, 1.0, 2.0, 2.0, 2.0]]
```

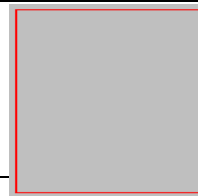
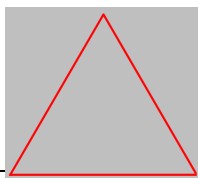
Exercice 7:

Maintenant, il faut tracer la trajectoire suivie par la tortue en utilisant la commande `plot`. Nous souhaitons écrire une fonction `tracer` qui à partir d'une chaîne de caractère `str` et d'un angle `alpha` trace la trajectoire décrite par les coordonnées successives LX et LY. Ecrire et commenter la fonction `tracer(str, alpha)`.

Tester votre fonction :

```
>>tracer('+A- - A- -A',pi/3)

>>tracer('AA+AA+AA+AA',pi/2)
```

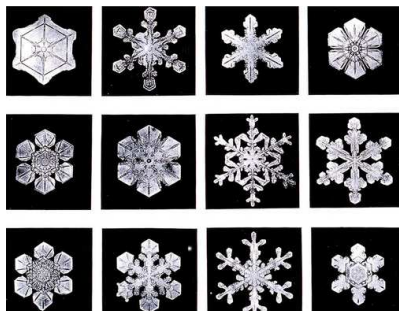


Exercice 8 :

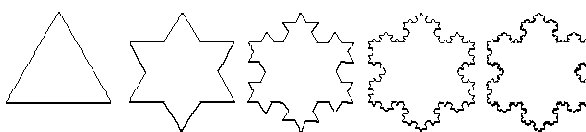
Ecrire une fonction qui trace un polygone à n cotés avec votre tortue. Vous utiliserez les fonctions précédentes.

Partie 3 : construction de fractales**Exercice 9 : le flocon de Von Koch**

Une courbe fractale est obtenue en effectuant une succession de transformations géométriques à une courbe simple. L'une des premières courbes fractales définie par le mathématicien Von Koch est le « Flocon de Von Koch ». En observant la nature, Von Koch (ci-contre) a remarqué que de nombreuses structures se développaient en répétant un motif. Le flocon de neige en est un exemple simple.



Flocons de neige



Flocons de Von Koch



Helmut Von Koch

La construction du flocon de Von Koch s'obtient en partant d'un triangle équilatéral.

Ensuite, chaque côté est remplacé par le motif



. On recommence l'opération sur la forme obtenue.

- Tracer un triangle équilatéral avec votre tortue. Quelle est la chaîne que doit lire la tortue pour ce tracé ? Quel est l'angle associé ?
- Tracer le motif ci-dessus avec votre tortue. Quelle est la chaîne que doit lire la tortue pour ce tracé ? Quel est l'angle associé ?
- Pour commencer le flocon de Von Koch, il suffit de remplacer dans la chaîne de caractère du triangle chaque A par le motif de la question b. : A+A- - A+A. Tracer la deuxième figure du flocon de Von Koch.

Pour continuer, il faut créer une fonction qui remplace chaque A d'une chaîne de caractère par le motif A+A- -A+A.

- Créer une fonction `remplacer(str, c, motif)` qui remplace le caractère `c` par le motif défini.

Tester votre fonction :

```
>>remplacer('+A- - A- -A', 'A', 'A+A- - A+A')
+A+A- -A+A- -A+A- -A+A - -A+A- -A+A
```

- Créer une fonction `vonkoch(n)` qui trace le flocon de Von Koch après n transformations (ne pas dépasser $n=5$).