



Préparation à l'oral

Math II - Informatique
Semaine du 2 Juin

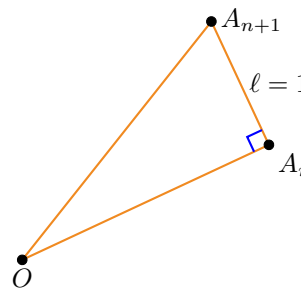
Sujet OB-INFO-4

Dans le plan affine, muni d'un repère orthonormé direct d'origine O , on considère la suite de points $(A_n)_{n \in \mathbb{N}}$ tels que :

- ✗ A_0 a pour coordonnées $(1, 0)$.
- ✗ pour tout entier naturel non nul n , le triangle OA_nA_{n+1} est rectangle en A_n , la distance A_nA_{n+1} vaut 1 et l'angle $(\overrightarrow{OA_n}, \overrightarrow{OA_{n+1}})$ est direct.

Si x_n et y_n sont les coordonnées de A_n , on a :

$$\begin{cases} x_{n+1} = x_n - \frac{y_n}{\sqrt{x_n^2 + y_n^2}} \\ y_{n+1} = y_n + \frac{x_n}{\sqrt{x_n^2 + y_n^2}} \end{cases}$$



1. Écrire une fonction `A` d'argument N renvoyant la liste des coordonnées des points A_n pour $0 \leq n \leq N$.
2. Écrire une fonction `afficher` d'argument N affichant la figure représentant les $(N + 1)$ premiers points A_n . La tester pour $N = 60$.

Les points tournent autour de l'origine O du repère. On note $T(k)$ la valeur de n telle que le point A_n commence le k -ième tour.

3. Écrire une fonction `tours` d'argument m qui renvoie la liste des $T(k)$ pour k variant de 1 à m .
Afficher `tours(5)`.
4. Faire tracer chacun des cinq premiers tours avec une couleur différente.
5. Déterminer les abscisses des dix premiers points d'intersection de la ligne reliant les A_n avec la partie positive de l'axe des abscisses.
Vérifier qu'à chaque tour, on s'est éloigné du centre d'une distance environ égale à π .

Sujet OB-INFO-4 : Math II 2018, Solution

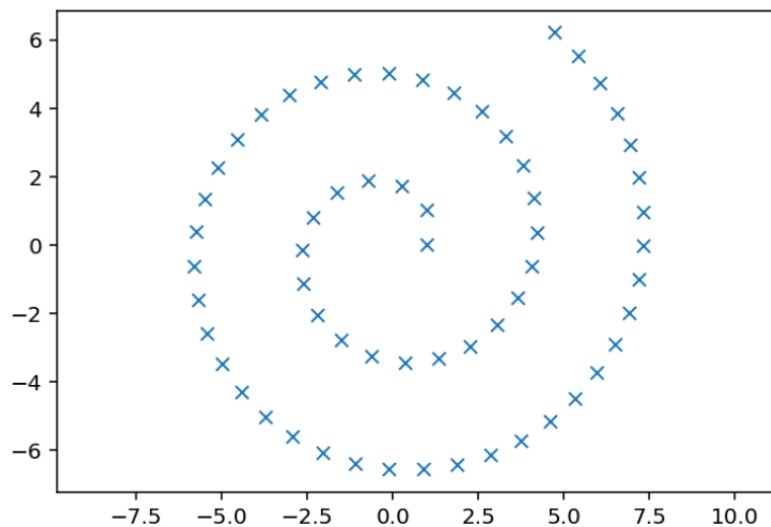
On commence par importer les packages dont on aura besoin

```
import matplotlib.pyplot as plt
import numpy as np
```

```
1. def A(N):
    L = []
    M = [1,0]
    for i in range(N+1):
        L.append(M)
        x = M[0] - (M[1] / (np.sqrt(M[0]**2 + M[1]**2)))
        y = M[1] + (M[0] / (np.sqrt(M[0]**2 + M[1]**2)))
        M = [x, y]
    return L
```

```
2. def afficher(N):
    Res = A(N) # liste des An
    X = [p[0] for p in Res] # abscisses
    Y = [p[1] for p in Res] # ordonnées
    plt.plot(X, Y, 'x')
    plt.axis('equal')
    plt.show()
```

Pour $N = 60$, on a la figure ci-dessous



3. D'abord une fonction auxiliaire qui calcule le point A_{n+1} à partir de A_n

```
def aux(M):
    x = M[0] - (M[1] / (np.sqrt(M[0]**2 + M[1]**2)))
    y = M[1] + (M[0] / (np.sqrt(M[0]**2 + M[1]**2)))
    return [x, y]

def tours(m):
    T = []
    n = 0
    M = [1,0]
    for i in range(m):
        T.append(n)
        while M[1] >= 0: # itérer tant qu'on est dans le demi-plan supérieur,
            n += 1
```

```

    M = aux(M)
    while M[1] < 0: # itérer tant qu'on est dans le demi-plan inférieur
        n += 1
        M = aux(M)
    return T

```

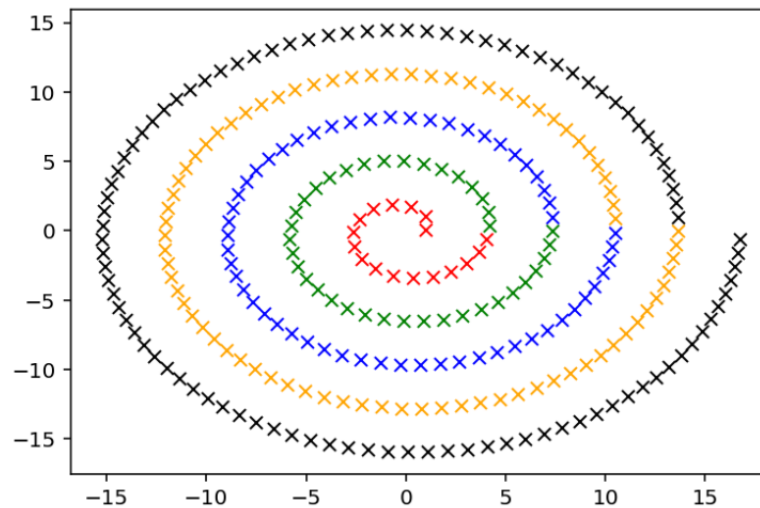
```

4. L = tours(6) # la liste des indices des points
    # qui commencent les 6 premiers tours
N = L[-1] # indice du point commençant le 6eme tour
points = A(N) # liste des An des cinq premiers tours (et début du 6e)

# partition des points en fonction du nombre de tours effectués
partition = [[points[i] for i in range(L[k],L[k+1])] for k in range(5)]

couleurs = ['red','green','blue','orange','black']
i = 0
for tour in partition:
    abscisses = [p[0] for p in tour]
    ordonnees = [p[1] for p in tour]
    plt.plot(abscisses, ordonnees, 'x', color = couleurs[i])
    i += 1
plt.show()

```



```

5. L = tours(11)
N = L[-1]
points = A(N) # liste de tous les points ayant effectué N tours
for i in range(len(L)-1):
    courant = points[L[i]][0] # abscisse du point commençant le tour i
    suivant = points[L[i+1]][0] # abscisse du point commençant le tour i+1
    distance = suivant - courant
    print(distance)
# effectivement ces distances ont proches de pi

```