TP2 : courbes de Bézier, transformations, ...

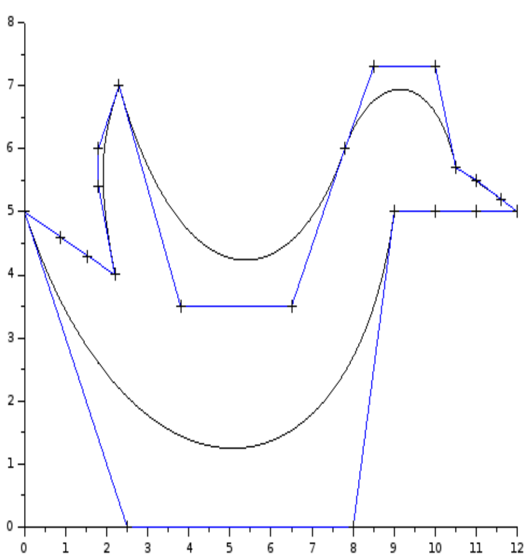
Vous avez programmé en Python, lors du dernier TP, un traceur pour les courbes de Bézier définies par la donnée de 4 points (2 points d'ancrage A_0 et A_3 et 2 "poignées" A_1 et A_2). Cette fonction prend en argument la matrice à 2 lignes et 4 colonnes $M = (A_0A_1A_2A_3)$ où la $i^{ème}$ colonne contient les coordonnées du point A_i (pour $i \in \{0, 1, 2, 3\}$).

Voici pour mémoire cette fonction traceur :

```python
import numpy as np
import matplotlib.pyplot as plt
def traceur(P):
    B=np.array([[1.,-3,3,-1],[0,3,-6,3],[0,0,3,-3],[0,0,0,1]])
    T=np.linspace(0,1,1001)
    Vect=np.array([np.ones(1001),T,T**2,T**3])
    res=np.dot(P,np.dot(B,Vect))
    plt.plot(res[0,:],res[1,:])
    return(res)
```

Pour tracer des arcs plus complexes la donnée de 4 points de contrôle est insuffisante. D'un autre côté, plus le nombre de points augmente plus les équations définissant la courbe de Bézier sont compliquées et de degré élevé en t. Pour palier ces difficultés, il est pratique de mettre bout à bout plusieurs courbes de Bézier définies uniquement par 4 points. On se donnera tout d'abord 4 points $A_0A_1A_2A_3$ pour la première courbe puis ensuite 4 autres points pour la seconde courbe $B_0B_1B_2B_4$...etc ...

Par exemple, pour obtenir le dessin de l'oiseau ci-dessous, on se donne 7 courbes de Béziers, donc 28 points de contrôle au total. Sur la figure ci-dessous, ces 28 points correspondent aux croix et délimitent le polygone entourant l'oiseau.



1. Pour simplifier les notations, on mettra dans la même matrice M tous les points de contrôle des courbes de Bézier successives ($M = (A_0A_1A_2A_3B_0B_1B_2B_3 \dots)$). Une telle matrice sera donc suffisante pour décrire entièrement un arc complexe.

   (a) A partir de la fonction **traceur** définie à la séance précédente, écrivez une fonction **tr(M)** permettant de tracer un contour défini par une matrice M de la forme décrite ci-dessus.

   (b) On se donne la matrice M suivante qui correspond à l'oiseau ci-dessus :

```
         column  1 to 26

 0.   2.5   8.   9.   9.   10.   11.   12.   12.   11.6   11.   10.5   10.5   10.   8.5   7.8   7.8   6.5   3.8   2.3   2.3   1.8   1.8   2.2   2.2   1.54
 5.   0.    0.   5.   5.   5.    5.    5.    5.    5.2    5.5   5.7    5.7    7.3   7.3   6.    6.    3.5   3.5   7.    7.    6.    5.4   4.    4.    4.3

         column 27 to 28

 0.88   0.
 4.6    5.
```

Tracez l'oiseau correspondant en utilisant la fonction **tr(M)**.

2. Dans cette partie nous allons effectuer des transformations affines sur l'oiseau défini par la matrice M précédente. Nous vous rappelons qu'un point du plan de coordonnées $\begin{pmatrix} x \\ y \end{pmatrix}$ dans un repère orthonormal $\left(0, \overrightarrow{i}, \overrightarrow{j}\right)$ a pour image par une application affine f le point de coordonnées $\begin{pmatrix} x' \\ y' \end{pmatrix}$ que l'on obtient par le produit matriciel suivant :

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} a & c & h \\ b & d & k \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

(c'est l'écriture en coordonnées homogènes de la transformation affine f)
On peut obtenir avec un seul produit matriciel toutes les coordonnées des images des points contenus dans la matrice M de la question 1 de la façon suivante : On remplace la matrice M par la matrice M_H obtenu à partir de M en rajoutant une ligne de 1 ($MH = np.concatenate((M, np.ones(1, 28)), axis = 0)$) et on effectue le produit matriciel suivant :

$$N_H = \begin{pmatrix} a & c & h \\ b & d & k \\ 0 & 0 & 1 \end{pmatrix} M_H$$

N_H contient les coordonnées des images et une ligne de 1.

(a) Que représentent les coefficients de la matrice $\begin{pmatrix} a & c & h \\ b & d & k \\ 0 & 0 & 1 \end{pmatrix}$ par rapport à la transformation affine f ?

(b) On considère la transformation affine f_1 définie par la matrice $A_1 \begin{pmatrix} 1 & 1 & 12 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$. on note $N_1 = A_1 M_H$. Tracez l'image de l'oiseau par cette application affine. Vous donnerez les commandes python successives.

(c) On considère la transformation affine f_2 définie par la matrice $A_2 = \begin{pmatrix} 2 & 0 & -12 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix}$. on note $N_2 = A_2 M_H$.

   i. Représentez l'image de l'oiseau par cette application affine sur le même graphique.
   ii. Déterminez par le calcul les points invariants (s'ils existent) de cette transformation (c'est à dire les points P du plan tels que $P = f(P)$).
   iii. Comment décrire précisément cette transformation ? (à l'aide des transformations usuelles décrites dans le cours)

(d) Soit $a = \frac{\sqrt{2}}{2}$ .On considère la transformation affine f_3 définie par la matrice $A_3 = \begin{pmatrix} a & -a & 0 \\ a & a & 10 \\ 0 & 0 & 1 \end{pmatrix}$. on note $N_3 = A_3 M_H$.

   i. Représentez l'image de l'oiseau par cette application affine sur le même graphique. .
   ii. Déterminez les points invariants (s'ils existent) de cette transformation
   iii. Comment décrire précisément cette transformation ?

(e) On veut effectuer une symétrie centrale de centre $C\begin{pmatrix} 5 \\ -5 \end{pmatrix}$. Donnez la matrice A_4 permettant d'effectuer cette transformation affine.