

Vecteur-vitesse d'un avion à l'atterrissage avec Python


Dans la colonne de gauche du tableau ci-dessous, on donne le programme (à compléter) qui permet de tracer les vecteurs vitesses à l'atterrissage d'un avion à l'aéroport de Nice. Une version de ce programme partiel est téléchargeable à l'adresse suivante :

En vous aidant des instructions et commentaires donnés dans la colonne de droite du tableau, répondre aux questions en dessous du tableau

Programme	Compréhension du programme (Commentaires – Instructions)
<pre>import matplotlib.pyplot as plt xc=[-22.69,-22.11,-22.11,-21.01,-20.81,-21.01,-20.57,-20.01,-20.29,-21.23,.....] yc=[-20.03,-19.66,-19.05,-19,-18.58,-18,-17.69,-17.52,-16.74,-16.41,-15.96,.....] t1=[-0.215,-0.207,-0.199,-0.19,-0.182,-0.174,-0.165,-0.157,-0.149,-0.14,-0.135,.....] x1=[6.9,8.24,9.71,11.14,12.47,13.69,14.87,16.07,17.2,18.39,18.96,19.73,19.46,.....] y1=[44.54,39.51,33.99,28.63,23.62,18.97,14.39,9.72,5.34,0.77,-1.45,-4.61,-7.23,.....] t2=[-0.274,-0.266,-0.257,-0.249,-0.241,-0.235,-0.223,-0.215,-0.207,-0.198,-0.19,.....] x2=[2.97,5.07,7.34,9.41,11.4,12.65,11.99,11.22,10.46,9.64,8.86,8.21,6.92,4.27,.....] y2=[44.77,39.63,34.07,28.93,24.03,20.51,13.49,8.53,3.67,-1.53,-6.39,-9.94,-12.13,....] plt.figure('Atterrissage', figsize=(11,11), dpi=80) plt.title('Atterrissage à Nice') plt.xlabel('x (km)') plt.ylabel('y (km)') plt.xlim(-25,45) plt.ylim(-30,40) plt.plot(xc,yc,'k',ms=5,lw=2) plt.plot(x1,y1,'r',ms=2,lw=0.5) n=len(t1) vx1=[] vy1=[] e=0.001 for i in range(0,n-1):</pre>	<p>Commande appelant une librairie de fonction mathématiques</p> <p>Listes correspondant aux abscisses et ordonnées des points de la côte entre Mandelieu et San Remo, autour de l'aéroport de Nice</p> <p>Listes correspondant aux positions de l'avion 1 (Strasbourg-Nice) à l'approche de Nice en fonction du temps (condition météorologique : léger vent d'est et bonne visibilité)</p> <p>Listes correspondant aux positions de l'avion 2 (Strasbourg-Nice) à l'approche de Nice en fonction du temps (conditions météorologiques : vent d'est et visibilité réduite)</p> <p>Définit la taille du graphique Définit le titre du graphique Définit les titres des axes</p> <p>Définit les valeurs minimales et maximales de l'axe des abscisses Définit les valeurs minimales et maximales de l'axe des ordonnées</p> <p>Affiche les coordonnées de la côte (ligne de couleur « k », de largeur 2) Affiche les coordonnées de l'avion 1 (ligne de couleur « r », de largeur 0.5)</p> <p>Définit le nombre de positions (longueur de la liste t1) Créer des liste vides dans lesquelles figureront les abscisses et ordonnées des vitesses instantanées de l'avion 1 Définit l'échelle avec laquelle on va représenter les vecteurs</p>

<pre> vx1.append(????) vy1.append(????) plt.arrow(x1[i], y1[i], e*vx1[i], e*vy1[i],length_includes_head="true",width=0.1, fc='r',ec='r') plt.show() </pre>	<p>Boucle itérative qui permet de remplir les listes des coordonnées des vitesses instantanées à partir des listes de coordonnées x1 et y1 de l'avion aux dates t1</p> <p>Affiche les vecteurs (arrow) représentant les vecteurs vitesses de coordonnées vx1 et vy1, à partir de chaque position x1 et y1, à l'échelle « e », à la largeur 0.1, avec la couleur « r »</p> <p>Affiche le graphe</p>
---	--

Travail à réaliser :

1. Ouverture du programme :
 - Télécharger le programme « AtterrissageNice » à l'adresse suivante :
 - Ouvrir l'interface Spyder
 - A partir de l'interface, ouvrir le programme « AtterrissageNice »
 - Tester le programme en utilisant l'icone 
 - Attention, ce programme n'est pas complet... Il n'affiche que la trajectoire d'un seul avion
2. On veut maintenant faire calculer les coordonnées du vecteur vitesse de l'avion qui devraient s'afficher.
 - Remplacer les points d'interrogation du programme par les formules adéquates
 - Tester le programme
3. Ajouter les lignes de programme qui permettent d'afficher la trajectoire et les vecteurs vitesse de l'avion 2
4. Télécharger le programme « AtterrissageNice9 » dans lequel on veut on suivre l'atterrissage de 9 avions, avec une couleur différent pour chaque avion.
 - En testant le programme, on peut voir que celui n'affiche que la trajectoire et les vecteurs-vitesses d'un seul avion
 - Transformer le programme en créant une fonction qu'on appellera « landing »
 - Rappel : une fonction est une instruction qui permet de réaliser plusieurs fois la même opération au sein d'un programme.
 - Il faut identifier :
 - les tâches à répéter
 - le nombre de variables à définir qu'elles nécessitent (l'abscisse et l'ordonnée de la position de l'avion sont 2 variables par exemple)

<pre> Def landing (... , ..., ...,): </pre>	<p>On définit le nom de la fonction au-dessus des lignes du programme qui la définisse en précisant entre parenthèse les variables</p> <p>Les lignes de programme définissant la fonction s'écrivent en retrait (on parle d'incrémentation)</p>
<pre> landing (t1, x1,,) </pre>	<p>Dans le programme principal (plus d'incrémentation), on appelle la fonction en faisant correspondre dans l'ordre les variables qui concernent l'avion 1</p>

