

## Exercice de mécanique

Vous pouvez créer un fichier `S1-exo-meca.py`; Copiez-coller à l'intérieur le code présent à cette adresse.

### 1 Un aperçu de la trajectoire

1. En vous servant de ce qui a déjà été fait, représentez la trajectoire du point au cours de son mouvement.

### 2 Étude pour un point particulier

2. En utilisant les éléments des listes `X`, `Y` et `T`, calculer les coordonnées du vecteur vitesse du 3<sup>ème</sup> point de la trajectoire (`vx` et `vy`), puis affichez-les.

RAPPEL :

- Python numérote les listes à partir du numéro "0" ;
- Afin de se mettre en concordance avec la définition de dérivée en maths, on privilégiera en PC comme formule de calcul de vitesse :

$$\vec{v}(n) = \frac{\vec{r}(n+1) - \vec{r}(n)}{t(n+1) - t(n)}$$

3. Déduisez-en alors la norme du vecteur vitesse pour le troisième point de la trajectoire (`v`). Calculez-la puis faites afficher cette norme.

AIDES :

- Mettre une expression  $x$  au carré sous Python s'écrit : `x**2`.
  - On peut utiliser la fonction racine carrée de la bibliothèque `math` : `sqrt()`
  - On peut aussi mettre l'expression à la puissance un demi : `(xxxx)**(1/2)`
4. On souhaite maintenant tracer le vecteur vitesse en ce point. Il existe pour cela en Python la fonction `arrow()` et la fonction `quiver()` du paquet `matplotlib.pyplot`.

Nous nous intéressons uniquement ici à la fonction `arrow`.

POINT COURS :

Une fois importé le paquet `matplotlib.pyplot` sous l'alias `plt`, la fonction `arrow` est appelée par :

```
plt.arrow(x,y, dx, dy, length_includes_head = "true", head_width = 0.02, color = "...")
```

avec :

- `x` : la coordonnée selon  $Ox$  du point de base du vecteur
- `y` : la coordonnée selon  $Oy$  du point de base du vecteur
- `dx` : la longueur selon  $Ox$  du vecteur
- `dy` : la longueur selon  $Oy$  du vecteur
- `color` (optionnel) : à choisir parmi `red`, `blue`, `green`, `cyan`, `black`, `yellow`, ...
- le paramètre `length_includes_head = "true"` (optionnel) permet d'avoir la longueur de la pointe de la flèche dans la longueur totale de la flèche
- le paramètre `head_width` (optionnel) permet de jouer sur l'épaisseur de la pointe de la flèche

À l'usage, les vecteurs sont souvent trop grands ou trop petits ; il faudra donc appliquer un facteur multiplicatif aux longueurs `dx` et `dy` selon  $Ox$  et  $Oy$ .

En vous servant de cela, tracez le vecteur vitesse au troisième point de la trajectoire.

### 3 Étude pour tous les points

5. En utilisant une boucle, tracez tous les vecteurs possibles pour les points où cela est possible.

AIDES :

— On pourra penser à l'utilisation d'une boucle `for` associée à la fonction `range` :

`for i in range(0,10) :` `i` va parcourir toutes les valeurs de 0 à 9

— le nombre d'éléments d'une liste `L` peut être donné par `len(L)`

Entrez le code ci-dessous pour comprendre les différents éléments à utiliser dans cette question :



CODE PYTHON™

```
((
)) >>> L = [12, 13, 14]
((
)) >>> for i in range(0, len(L)) :
((
))     print("élément ", i, "   valeur ", L[i])
```

6. Comment comprenez-vous la ligne du programme :

```
plt.text(0,0, "vecteurs vitesse", color = "magenta")
```

À quoi sert-elle ? Comment le vérifier ? Vous pouvez rédiger ces réponses en commentaires, dans le code.

7. Rajoutez un nom aux axes et un titre au graphique.

↗ aide à la résolution : S1-exo-meca-aide.py

► lien vers la correction : S1-exo-meca-correction.py◀