


Étude du vecteur variation de vitesse à l'aide d'un programme Python

Nous allons étudier, à l'aide d'un programme python, le vecteur « variation de vitesse » lors d'un lancer parabolique sans frottement.

1. Positions successives du système (déjà traité en classe de seconde)

1.1. Pointage des différentes positions du système

Ouvrir latis pro (tous les programmes/physique-chimie/Latispro) puis icône video (le cinquième de la liste ). Ouvrir la vidéo *parabole.avi* que vous avez téléchargé, faire l'étalonnage et pointer les positions de la balle sur la vidéo et exporter au format TXT **Les coordonnées des positions M_0 , M_1 etc. sont créées.**

1.2. Préparation et ouverture du fichier Python

Lancer l'éditeur Python (Edupython.exe) et ouvrir le programme « **1ere_Spe_Variation_vect_vitesse1.py** » que vous venez de télécharger.

1.3. Tracé des positions successives du système : sa trajectoire

1. TRAVAIL 1 : Tracé de la trajectoire.

Cas général en Python (à adapter à ce que l'on veut faire) :

Pour tracer la courbe représentant une grandeur A en fonction d'une grandeur B, il faut taper : `plt.plot(B,A,"kx")` "kx" signifie que les points affichés seront noirs ("k") et représentés par des croix ("x")

Dans le programme, en dessous de "TRAVAIL 1", taper (en adaptant le cas général) le code permettant de tracer **l'évolution de l'ordonnée y du système en fonction de son abscisse x** avec des points de couleur rouge représentés par une croix sans être reliés (voir annexe à la fin du TP pour le nom des couleurs).

2. TRAVAIL 2 : Préparation du graphique. Suivre les consignes en rose dans le programme au niveau de «TRAVAIL 2»:

Donner un nom pertinent au graphique, écrire les noms des grandeurs représentées sur les axes **et leur unité.**

3. Exécuter le programme (petit triangle vert en haut de l'écran). Une fenêtre vous permet d'aller chercher le fichier que vous avez exporté depuis LATISPRO. La courbe doit s'afficher dans une nouvelle fenêtre. Fermer cette fenêtre.

2. Coordonnées des vecteurs vitesse

RAPPEL de Seconde

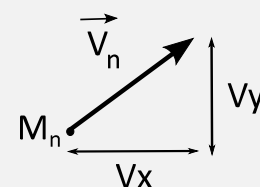
Le vecteur vitesse à la position M_n s'exprime de la manière suivante :

$$\vec{V}_n = \frac{\overrightarrow{M_{n+1}M_{n-1}}}{t_{n+1}-t_{n-1}} \quad \text{où } M_{n+1} \text{ et } M_{n-1} \text{ sont les noms des positions } n+1 \text{ et } n-1 \text{ et } t_{n+1} \text{ et } t_{n-1} \text{ sont les dates de passage à ces positions}$$

L'abscisse V_x du vecteur vitesse au point n s'exprime donc :

$$V_x = \frac{(x_{n+1}-x_{n-1})}{(t_{n+1}-t_{n-1})} \quad \text{où } x_{n+1} \text{ et } x_{n-1} \text{ sont les abscisses du système au points } M_{n+1} \text{ et } M_{n-1} \text{ et } t_{n+1} \text{ et } t_{n-1} \text{ les dates de passage du système aux points } M_{n+1} \text{ et } M_{n-1}$$

Il en est de même pour V_y .



En Python :

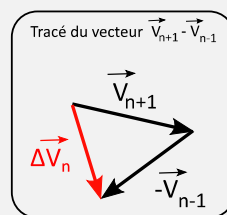
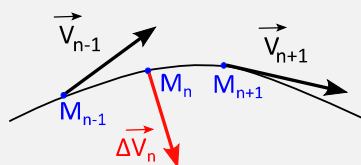
l'abscisse du système au point x_n se tape : `x[n]` La date t_n au point n se tape : `t[n]`

Pour ajouter un élément « A » à un tableau « T », l'instruction est la suivante : `T = np.append(T,A)`

1. Écrire ci-dessous, en langage python, l'expression de V_x et de V_y au point M_n
2. **TRAVAIL 3** : Dans le code python, supprimer : `print("")` (qui permettait au programme incomplet de fonctionner quand même)
Lire les consignes du programme situées sous TRAVAIL 3 :
 - Dans la boucle, taper le code permettant d'ajouter au tableau V_x la coordonnée du vecteur V_x à la position M_n
 - Faire de même pour V_y
3. Vérifier la cohérence de votre code en traçant les vecteurs vitesse en tapant, toujours dans la boucle : `draw_Vector2(n, Vx[n-1], Vy[n-1], "b", 20)`
4. La première position correspond à $n=0$ et la dernière à $n = \text{Nbre_mesures}$.
Pourquoi la boucle proposée par le professeur commence à 1 et se termine à $\text{Nbre_mesures}-1$?

3. Coordonnées et tracé des vecteurs « variation de vitesse »

Vecteur « variation de vitesse » $\Delta \vec{V}$
Le vecteur variation de vitesse à la position M_n est : $\Delta \vec{V}_n = \vec{V}_{n+1} - \vec{V}_{n-1}$
Il se construit graphiquement de la manière suivante :



Ses coordonnées s'expriment donc, à la position M_n : $\Delta V_{x_n} = V_{x_{n+1}} - V_{x_{n-1}}$ et $\Delta V_{y_n} = V_{y_{n+1}} - V_{y_{n-1}}$

- TRAVAIL 4** : Lire les consignes du programme situées sous TRAVAIL 4 :
- créer une boucle (compteur de boucle nommé "n") allant de 1 à $\text{Nbre_vitesses}-1$
 - mettre dans une variable DeltaVx l'abscisse du vecteur variation de vitesse au point n
 - mettre dans une variable DeltaVy l'ordonnée du vecteur variation de vitesse au point n
 - tracer le vecteur variation vitesse au point "n+1" en vert avec une échelle de 5 (n+1 car le numéro de point et l'index du vecteur dans son tableau sont décalés de 1)

En Python, pour tracer un vecteur en un point, il faut utiliser la fonction suivante :

`draw_Vector2 (numéro du point , Abscisse du vecteur , Ordonnée du vecteur , "k" , échelle)`

où "k" représente la couleur du vecteur (k : noir, voir les couleurs dans l'annexe en fin de TP)
« échelle » est un nombre entier permettant de tracer le vecteur à une bonne échelle par rapport au reste du graphique

4. Interprétation

1. Que remarque-t-on pour le vecteur variation de vitesse ?
2. Pourquoi n'y a-t-il pas de vecteur $\Delta \vec{V}$ au niveau des 2 premiers points et des 2 derniers ?
3. Faire le bilan des forces s'exerçant sur le système. Bien préciser sens et direction de la résultante de ces forces.
4. En comparant sens et direction de $\Delta \vec{V}$ et de la résultante des forces, que peut-on en conclure.

5. Discussion autour de la validité de la conclusion

En réalité, on voit bien que nos vecteurs $\Delta\vec{V}$ ne sont pas tous parallèles et colinéaires à la résultante des forces.

1. Proposer une explication.

2. Vérifier votre réponse en utilisant, à la place de votre pointage, les équations horaires modélisant la trajectoire (la détermination des équations horaires sera vue en terminale spécialité)

Pour cela : dans la partie « choix du professeur » du code python, taper :

```
Logiciel_Utilise = 'equation'
```

Exécuter le programme et conclure

ANNEXE - Quelques options de mise en forme du tracé avec le module MATPLOTLIB utilisé par Python

Tracé		Type de points tracés					Couleurs				
-	--	o	.	x	+	v	r	b	g	k	m
Points reliés	Points reliés en pointillé	Gros «ronds»	Petit point	Croix x	Croix +	Triangle	Rouge	Bleu	vert	noir	magenta