**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***

***(Texte volontairement non finalisé)*** faire pointer les positions d’un objet sur une vidéo et exporter au format TXT par Latispro, Avimeca ou Regressi… **Les coordonnées des positions M0, M1 etc. sont créées.**

***1.2. Préparation et ouverture du fichier Python***

**1.** Copier les 2 fichiers «**import\_donnees\_meca2.py**» et **«1ere\_Spe\_Variation\_vect\_vitesse1.py»** dans votre dossier personnel.

**2.** Lancer l’éditeur Python (Edupython.exe) et ouvrir le programme **« 1ere\_Spe\_Variation\_vect\_vitesse1.py »** que vous venez de copier.

***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 Mn s’exprime de la manière suivante :

où Mn+1 et Mn-1 sont les noms des positions n+1 et n-1

et tn+1 et tn-1 sont les dates de passage à ces positions

L’abscisse Vx du vecteur vitesse au point n s’exprime donc :

où xn+1 et xn-1 sont les abscisses du système au points Mn+1 et Mn-1



et tn+1 et tn-1 les dates de passage du système aux points Mn+1 et Mn-1

Il en est de même pour Vy.

***En Python :***

l’abscisse du système au point xn se tape : **x[n]** La date tn 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 Vx et de Vy au point Mn

Vx = (x[n+1]-x[n-1])/(t[n+1]-t[n-1])

Vy = (y[n+1]-y[n-1])/(t[n+1]-t[n-1])

**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 Vx la coordonnée du vecteur Vx à la position Mn

- Faire de même pour Vy

**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 = Nbre\_mesures.

Pourquoi la boucle proposée par le professeur commence à 1 et se termine à Nbre\_mesures-1 ?

Car calcul de Vx utilise le point « d’avant » et celui « d’après » donc pas possible en n=0 et au dernier point.

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

***Vecteur « variation de vitesse » ***

Le vecteur variation de vitesse à la position Mn est :

Il se construit graphiquement de la manière suivante :



Ses coordonnées s’expriment donc, à la position Mn : **ΔVxn = Vxn+1-Vxn-1** et **ΔVyn = Vyn+1-Vyn-1**

**TRAVAIL 4 :** Lire les consignes du programme situées sous TRAVAIL 4 :

- créer une boucle (compteur de boucle nommé "n") allant de 1 à 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 (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**

***(Texte volontairement non finalisé et laissé à l’initiative du professeur)***

**1.** Que remarque-t-on pour le vecteur variation de vitesse ? aux erreurs de pointage près, même sens, même direction.

**2.** Pourquoi n’y a-t-il pas de vecteur ******au niveau des 2 premiers points et des 2 derniers ? car pour calculer les coordonnées de *, il faut utiliser les 2 points « avant » et les 2 points « après »*

**3.** Faire le bilan des forces s’exerçant sur le système. Bien préciser sens et direction de la résultats de ces forces.

Poids uniquement donc vertical vers le bas… comme *** !***

**4.** En comparant sens et direction de *et de la résultante des forces, que peut-on en conclure.*

Aux erreurs de pointage près, *et somme forces colinéaires et de même sens.*

**5. Discussion autour de la validité de la conclusion**

***(Texte volontairement non finalisé et laissé à l’initiative du professeur)***

En réalité, on voit bien que nos vecteurs ne sont pas tous parallèles et colinéaires à la résultante des forces.

**1.** Proposer une explication.

Pointage pas top donc calcul des variations pas exact donc logique à priori que les vecteurs ne soient pas parallèles exactement.

**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

**Cette fois, OK donc cela venait bien du pointage.**

**REMARQUE POUR LE PROFESSEUR :**

L’auteur de la fiche propose aussi un code appelé « 1ere\_Spe\_Variation\_vect\_vitesse2.py » qui reprend le code précédent et qui trace également Vx et Vy en fonction de t. Cela peut permettre de montrer aux élèves qu’il y a des petites variations de Vx dûes au pointage et donc mieux expliquer le fait que les DetaV ne sont pas parfaitement parallèles.

Avec les équations horaires, on leur montre que Vx est bien constant.

**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 | Croix + | Triangle | Rouge | Bleu | vert | noir | magenta |