**FICHE de PRÉSENTATION d’activités**

|  |  |  |
| --- | --- | --- |
| ***Niveau*** | ***seconde*** | |
| ***Séquence*** | ***Décrire un mouvement*** | |
| ***Titre de l’activité*** | ***Tracé de positions et vecteurs vitesses d’un système en utilisant le langage Python VERSION 2***  ***(chute libre sans vitesse initiale***  ***et approche de par )*** | |
| ***Type d'activité*** | ***Expérimentale et programmation en ½ groupe*** | |
| ***Références au programme*** | Notions et contenus  *Position. Trajectoire d’un point*  *Vecteur déplacement d’un point.*  *Vecteur vitesse d’un point..* | Capacités exigibles  *Caractériser différentes trajectoires.*  ***Capacité numérique :*** *représenter les positions successives d’un système modélisé par un point lors d’une évolution unidimensionnelle ou bidimensionnelle à l’aide d’un langage de programmation.*  *Réaliser et/ou exploiter une vidéo ou une chronophotographie d’un système en mouvement et représenter des vecteurs vitesse. Décrire la variation du vecteur vitesse.*  ***Capacité numérique :*** *représenter des vecteurs vitesse d’un système modélisé par un point lors d’un mouvement à l’aide d’un langage de programmation.* |
| ***Compétences mobilisées*** | ❑ Analyser/raisonner ❑ Réaliser | |
| ***Mise en œuvre*** | Pré-requis:  Quelques notions de python (exécution d’un programme sous Edupython) | |
| Durée : 1h30 | |
| Contraintes matérielles :  - salle informatique avec EduPython  - vidéo lancer parabolique et ou chute libre sans Vitesse initiale  - Logiciel de pointage : Latipro ou avimeca ou avistep | |
| *Liens photos* | *aucun* | |
| *Auteur* | **olivier.chaumette@ac-lyon.fr** | |
| **pour le GRD groupe lycée de l’académie de LYON** | |

**Fiche élève : activité**

Cette activité a pour but de pointer les positions prises par un système, d’importer les données dans un programme python et de faire tracer trajectoire et vecteurs vitesse à l’aide du programme.

**1. Positions successives du système**

***1.1. Pointage des différentes positions du système***

***(Texte volontairement non finalisé)*** faire pointer les positions d’un objet sur une vidéo de chute libre sans vitesse initiale (mouvement rectiligne) et exporter au format TXT par Latispro, Avimeca ou Regressi…

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

**1.** Copier les fichiers «**import\_donnees\_meca.py**» et **«2nde\_Position\_et\_vecteur\_V2.py»** dans votre dossier personnel.

**2.** Lancer l’éditeur Python (Edupython.exe) et ouvrir le programme **« 2nde\_Position\_et\_vecteur\_V2.py »** que vous venez de copier.

***1.3. Tracé des positions successives du système : sa trajectoire***

*Pour l’instant, le programme n’affiche rien. Vous devrez le modifier.* ***Attention à bien respecter la casse****.*

**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.

***1.4. Interprétation***

***(Texte volontairement non finalisé)*** En s’appuyant sur la trajectoire, qualifier le mouvement

**2. Tracé des vecteurs vitesses**

***2.1. Travail préparatoire pour définir les coordonnées du vecteur vitesse***

Le vecteur vitesse au point 5 ne peut pas être dessiné de manière exacte.

Mais **il est possible de l’approcher en traçant le vecteur vitesse moyenne entre les points 5 et 6** :

où M5 et M6 sont les noms des points n° 5 et 6

et t5 et t6 sont les dates de passage du système aux points n°4 et n°6



L’abscisse Vx du vecteur vitesse au point n°5 (par exemple) s’exprime :



où x5 et x6 sont les abscisses du système au points M5 et M6

et t5 et t6 les dates de passage du système aux points M5 et M4

De la même manière, écrire la formule donnant l’expression de l’ordonnée Vy du vecteur vitesse au point n°5 (M5).

***2.2. Tracé du vecteur vitesse au point 5 dans le programme Python***

En Python, l’abscisse du système au point x6 se tape : **x[6]** La date t6 au point 6 se tape : **t[6]**

**1. TRAVAIL 3 :** **Calcul des coordonnées du vecteur vitesse au point 5.** Lire les consignes du programme situées sous TRAVAIL 3 :

- Taper le code permettant de créer la variable Vx (coordonnées du vecteur vitesse au point M5).

- Créer également Vy (au point M5 toujours)

**2. TRAVAIL 4 :** **Tracé du vecteur vitesse au point 5.**

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

**draw\_Vector(*numéro du point* , *Abscisse du vecteur* , *Ordonnée du vecteur* , “k”)**

*où “k” représente la couleur du vecteur (k : noir, voir les couleurs dans l’annexe en fin de TP)*

Dans le programme, en dessous des consignes correspondantes à TRAVAIL 4, taper le code permettant de tracer le vecteur  au point 5 en bleu.

***2.3. Tracé du vecteur vitesse en un point n quelconque***

**1.** Définir, d’une manière générale, l’abscisse Vx du vecteur vitesse en un point **n** quelconque.

**2.** Définir, d’une manière générale, l’ordonnée Vy du vecteur vitesse en un point **n** quelconque.

**TRAVAIL 5 :** Suivre les consignes en rose dans le programme au niveau de « TRAVAIL 5» :

- Créer une variable **n** qui contiendra le numéro du point auquel on veut tracer le vecteur (n valant 3 au début)

- Taper le code permettant de créer la variable Vx (coordonnées du vecteur vitesse au point n).

- Créer également Vy (au point n toujours)

- Taper le code permettant de tracer le vecteur  au point n en vert.

***2.4. Utilisation du programme Python***

***Le fait de tracer***  ***en un point quelconque à l’aide du programme est intéressant : il suffit de changer uniquement la valeur de la variable n (donc du point) pour que le vecteur se trace automatiquement en ce point.***

***Utiliser le programme afin de répondre aux questions suivantes :***

**1.** La valeur du vecteur vitesse au point 1 est-elle inférieure ou supérieure à celle au point 5 ?

**2.** Même question au point 10.

**3.** Comment évolue Vy? Pour répondre à cette question, il est possible de faire afficher par Python la valeur de Vy.

**4.** Comment évolue Vx ? Pour répondre à cette question, il est possible de faire afficher par Python la valeur de Vy.

**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 |

**Fiche professeur**

Une fiche expliquant comment exporter des données depuis AVIMECA ou AVISTEP est fournie à a fin de ce document. Cela permettra au professeur d’adapter le paragraphe 1.1 de l ‘énoncé au logiciel qu’il utilise.

Dans cette version 2 de l’activité, l’auteur a réduit l’étude du mouvement au seul mouvement rectiligne comme le demande le programme (même si, dans les capacités numériques, il est possible de faire tracer la trajectoire d’autres types de mouvements).

Pour l’approche numérique de la vitesse, même si la dérivée numérique centrée donne de meilleurs résultats, l’auteur a préféré revenir à la dérivée numérique à droite pour être en lien avec l’approche préconisée par le BO : la vitesse en un point est approchée par la vitesse moyenne entre ce point et le suivant.

Il est toujours possible de dire aux élèves qu’il existe une autre méthode dite « centrée » mais, n’ayant pas à notre disposition la notion de tangente en 2nde, il sera difficile de montrer aux élèves que la méthode centrée est plus pertinente pour des points discrets car elle permet au vecteur vitesse tracé de se rapprocher de la tangente. Ceci sera montré en première spécialité.

***ATTENTION :*** le fichier **import\_donnees\_meca.py** doit être présent dans le même dossier que le fichier python que complètent les élèves (car c’est lui qui permet de transformer le fichier exporté par le logiciel en tableaux numpy dans Python)

***Voici ci-dessous le code corrigé :***

# ##########################################################################

# Programme Python permettant de tracer les positions d'un point M

# à partir d'abscisses et d'ordonnées issues d'un pointage (ou d'équations).

# et de tracer le vecteur vitesse en un point

# à destination des élèves qui peuvent modifier le code sous les parties roses

# L'utilisation des listes NUMPY permet de simplifier le code

# ##########################################################################

# O. CHAUMETTE - Lycée JP SARTRE - 69500 BRON - olivier.chaumette@ac-lyon.fr

# ##########################################################################

# ##########################

# IMPORTATION DES MODULES #

# #########################

# importation de NUMPY (pour gestion tableaux et calculs) sous l'alias "np"

**import** numpy **as** np

# importation de PYPLOT (du module MATPLOTLIB, pour le tracé de courbes) sous l'alias "plt"

**import** matplotlib**.**pyplot **as** plt

# modules permettant d'importer des données

# le fichier "import\_donnees.py" est indispensable

**from** import\_donnees\_meca **import** traiteDonnees

# Initialisation des tableaux numpy contenant x,y,t

x**=**np**.**array**([])**

y**=**np**.**array**([])**

t**=**np**.**array**([])**

# #################################

# Fonction qui simplifie le tracé du vecteur vitesse

# afin que l'élève n'ait pas à définir l'origine

# du vecteur en termes de coordonnées

# (pour ne pas mélanger des positions et des coordonées de vecteurs)

# #################################

**def** draw\_Vector**(**numero\_du\_point**,** abscisse\_vecteur**,** ordonnee\_vecteur**,** couleur**):**

**global** x**,**y

plt**.**quiver**(**x**[**numero\_du\_point**],** y**[**numero\_du\_point**],**abscisse\_vecteur**,**ordonnee\_vecteur**,**color**=**couleur**,**scale**=**30**)**

""" ---------------------------------

CHOIX du PROFESSEUR

------------------------------------- """

# Logiciel utilisé à choisir parmi :

# 'pymecavideo' (exporter un fichier TXT)

# 'latispro' (fichier TXT créé par LATISPRO, avec décimale VIRGULE et séparation POINT VIRGULE)

# 'avimeca' (crééer un fichier TXT en export)

# 'avistep' (pointage exporté depuis AVISTEP:

# 'equation' (équation paramétrées saisies dans le fichier "import\_donnees")

# 'entrees' (données de pointage saisies "à la main" dans le fichier "import\_donnees.py)

Logiciel\_Utilise **=** 'latispro'

""" ---------------------------------

FIN du CHOIX du PROFESSEUR

------------------------------------- """

t**,** x**,** y **=** traiteDonnees**(**Logiciel\_Utilise**)**

""" ###########################################

Ici commence le travail à réaliser par les élèves

################################################# """

# #####################################

# Tracé de la trajectoire

# #####################################

""" TRAVAIL 1:

Taper ci-dessous le code Python permettant de tracer l'ordonnée du point en fonction de son abscisse. """

plt**.**plot**(**x**,**y**,**"+r"**)**

""" TRAVAIL 2:

taper, à la place des points entre guillemets, les légendes pour les axes,

le titre du graphique"""

plt**.**title**(**"Trajectoire du point point M"**)** # titre du graphique

plt**.**xlabel**(**"Abscisse X du système (en m)"**)** # nom de la grandeur sur l'axe X

plt**.**ylabel**(**"Ordonnée Y du système (en m)"**)** # nom de la grandeur sur l'axe Y

""" TRAVAIL 3:

Taper ci-dessous le code python permettant:

- de mettre dans une variable Vx l'abscisse du vecteur vitesse au point 5

- de mettre dans une variable Vy l'ordonnée du vecteur vitesse au point 5 """

Vx **=** **(**x**[**6**]-**x**[**5**])/(**t**[**6**]-**t**[**5**])**

Vy **=** **(**y**[**6**]-**y**[**5**])/(**t**[**6**]-**t**[**5**])**

""" TRAVAIL 4:

Taper ci-dessous le code python permettant

de tracer le vecteur vitesse au point 5 en bleu"""

draw\_Vector**(**5**,**Vx**,**Vy**,**"b"**)**

""" TRAVAIL 5:

Généralisation

Taper ci-dessous le code python permettant:

- créer une variable nommée n correspondant à un numéro de point quelconque (3 par exemple)

- de mettre dans une variable Vx l'abscisse du vecteur vitesse au point n

- de mettre dans une variable Vy l'ordonnée du vecteur vitesse au point n

- de tracer le vecteur vitesse au point n en vert"""

n**=**10

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

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

draw\_Vector**(**n**,**Vx**,**Vy**,**"g"**)**

""" fin du travail """

# On affiche une grille

plt**.**grid**()**

# le repère doit être orthonormé:

plt**.**axis**(**'equal'**)**

# on affiche la fenêtre

#plt.get\_current\_fig\_manager().window.state('zoomed') # <-- Affiche la fenêtre maximisée sous windows uniquement

plt**.**show**()**

**MODE D’EMPLOI des programmes Python de mécanique (formation 2019)**

*L’objectif de cette fiche est d’expliquer comment utiliser les programmes Python de mécanique proposés lors de la formation Académique de 2019.*

**1. Pourquoi cette notice ?**

Les programmes py de mécanique proposés nécessitent l’importation de données de pointage (abscisse, ordonnée et date). **Une ligne de ces programmes dit être modifiée par le professeur** en tenant compte du logiciel de pointage utilisé par les élèves.

**Attention, pour tous les programmes Python de mécanique, le fichier “import\_donnees\_meca.py” doit être présent dans le même dossier que le programme (car c’est lui qui permet d’importer les données)**

**2. Choix, par le professeur, de la manière d’importer des données dans le programme**

**La ligne à modifier se situe, dans le programme, au niveau du commentaire “CHOIX DU PROFESSEUR” :**

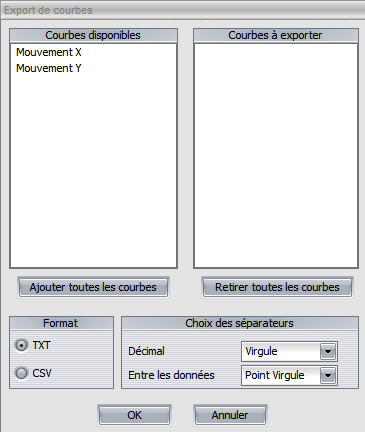
Dans la version par défaut, la ligne est la suivante : **Logiciel\_Utilise = 'latispro'**

S’il le souhaite, le professeur peut remplacer **‘latispro’** par un des choix proposés : **‘pymecavideo’** ou **‘avimeca’** ou **‘avistep’** ou **‘equations’** ou **‘entree’** (ATTENTION : bien respecter la casse et ne pas mettre d’accents)

**‘latispro’** ou **‘pymecavideo’** ou **‘avimeca’** ou **‘avistep’**: nom du logiciel utilisé.

**‘equations’** : le professeur saisit les équations horaires dans le fichier “import\_donnees\_meca.py” (voir explication § 7)

**‘entree’** : le professeur saisit les données « à la main » dans le fichier “import\_donnees\_meca.py” (voir explication § 8)



**3. Export des données depuis LATISPRO**

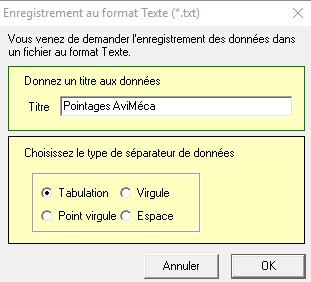
**1.** Une fois le pointage réalisé (Mouvement X, Mouvement Y et Temps seulement), choisir Menu **Fichier** >> **Exportation**.

**2. “Ajouter toutes les courbes”**. Vérifier que TXT est coché (dans “Format”) et que le **Choix des séparateurs** est **Virgule** (pour “Décimal”) et **Point Virgule** (pour “Entre les données”).

**4. Export des données depuis AVISTEP**

**1.** Une fois le pointage réalisé, choisir Menu **Résultats** >> **Tableau des valeurs**.

**2.** Dans la fenêtre Tableau des valeurs, choisir Menu **Fichier** >> **Enregistrer**.



**5. Export des données depuis AVIMECA**

**1.** Une fois le pointage réalisé, choisir Menu **Fichier** >> **Mesures >> Enregistrer dans un Fichier >> Format Texte (\*.txt)**.

**2.** Dans la fenêtre, vérifier que « **Tabulation** » est bien coché. Puis **OK**.

**6. Export des données depuis PYMECAVIDEO**

PyMecaVideo est un logiciel gratuit, libre et multiplateforme disponible à l’adresse : <http://outilsphysiques.tuxfamily.org/wiki/index.php?title=Pymecavideo>

Une fois le pointage réalisé, choisir Menu **Fichier >> Enregistrer les données…..**

**7. Choix EQUATIONS**

**1.** Charger le programme “**import\_donnees\_meca.py**” dans l’éditeur Python (EduPython ou Spyder par exemple)

**2.** Modifier la durée et les équations horaires dans le code suivant :

**elif origine=='equation':**

**""" CAS 2:**

**X ET Y ISSUES D'EQUATIONS HORAIRES DONNEES PAR LE PROFESSEUR**

**(méthode rapide)"""**

**tmax = 2 Saisir ici la durée maximale du temps en seconde**

**Nbre\_Mesures = 20 Saisir ici le nombre de points voulus. L’intervalle de temps sera tmax/Nombre\_Mesures**

**t = np.linspace(0, tmax,Nbre\_Mesures)**

**x=2\*t Saisir ici l’équation horaire x(t)**

**y=-4.9\*t\*t+2\*t+1 Saisir ici l’équation horaire y(t)**

**3.** Enregistrer le programme “**import\_donnees\_meca.py**” (sans changer son nom)

**8. Choix ENTREE**

**1.** Charger le programme “**import\_donnees\_meca.py**” dans l’éditeur Python (EduPython ou Spyder par exemple)

**2.** Modifier le contenu des tableaux x, y et t dans le code suivant :

**if origine=='entree':**

**""" CAS 1:**

**X ET Y ISSUES DU POINTAGE et entrées à la main"""**

**x=np.array([0,0,0,0,0,0,0,0,0]) Saisir ici les valeurs de x (séparées par une virgule)**

**y=np.array([9.86E-3,-5.92e-2,-1.48E-1,-2.76e-1,-4.24E-1,-6.02E-1,-8.48E-1,-1.07,-1.39])**

**Saisir ici les valeurs de y (séparées par une virgule)**

**t=np.array([0,0.033,0.067,0.1,0.133,0.167,0.2,0.233,0.267]) Saisir les valeurs de t**

**3.** Enregistrer le programme “**import\_donnees\_meca.py**” (sans changer son nom)

**9. En cas d’erreurs d’importation**

Il peut arriver que l’exportation par les logiciels enregistre des données absurdes ou fausses. Ces données font que Python renvoie une erreur.

Dans ce cas, le professeur peut ouvrir le fichier exporté avec le bloc-note Windows et supprimer « à la main »les lignes à l’origine de l’erreur.

***Exemple d’un fichier Latispro incorrect :***

Depuis le bloc-note, supprimer cette ligne « à la main » et enregistrer le fichier

**Temps;Mouvement X;Temps;Mouvement Y**

**0;0,00902378999179662;0;1,18229227704207**

**NAN ;NAN ;NAN,NAN**

**0,04;0,0986757295206844;0,04;1,32995429508965**

**0,08;0,20414859955467;0,08;1,47761631313723**