Thème II. Mouvements et interactions (Mécanique)

TP n°20 Étude d'un ressort par pointage vidéo

Vendredi 5 ou 12 avril 2024

Capacité exigibles du programme :

✓ **Réaliser et exploiter quantitativement un enregistrement vidéo d'un mouvement : évolution temporelle des vecteurs vitesse et accélération.**

✓ Enregistrer un phénomène à l'aide d'une caméra numérique et repérer la trajectoire à l'aide d'un logiciel dédié, en déduire la vitesse et l'accélération.

Objectif

L'objectif de ce TP est d'étudier le système masse-ressort par pointage vidéo pour en mener une analyse énergétique.

I Vidéo et pointage

Expérience

☞ Avec votre téléphone, filmer l'oscillation de la masse (sur laquelle on aura mis un repère) accrochée à l'extrémité du ressort. On veillera à choisir un nombre d'images par seconde suffisant, à avoir la totalité du ressort sur l'image et à être bien en face du système.

☞ Ouvrir l'application `Fizziq`, et sélectionner « Analyse cinématique », puis « Cinématique par vidéo » , puis « Mes vidéos ». Sélectionner la vidéo que vous venez de faire.

☞ Définir l'origine du repère, et l'échelle (pour la conversion pixel → mètre). Pour que l'origine soit indépendante du choix d'échelle, il faut cliquer sur le petit plus rouge en haut à droite (qui devient alors un moins). Cf Figure 1.

☞ Effectuer le pointé sur au moins 4 ou 5 périodes d'oscillations.

☞ Une fois le pointé effectué, cliquer sur « Résultat ». Sélectionner : T(s), x(m) et y(m). Puis cliquer sur « Cahier ».

☞ Partager ce cahier en sélectionnant « Créer un fichier CSV ou Python », puis « Tableau Python ». Envoyer ce fichier par mail. Cf Figure 2.

☞ Enregistrer le fichier Python précédent dans votre dossier du serveur du lycée.

FIGURE 1 – Choix de l'origine et définition de l'échelle

FIGURE 2 – Exporter en python

II Paramètres de l'oscillateur

Exploitation

☞ Ouvrir `Spyder`.

☞ Importer les bibliothèques `numpy` (avec l'alias `np`) et `matplotlib.pyplot` (avec l'alias `plt`).

☞ Ouvrir le fichier issu du pointage avec `Spyder`.

☞ Créer deux tableau numpy `t` et `y` des temps et des ordonnées verticales.

☞ Représenter le graphe de la position en fonction du temps. *(On s'aidera du document « Boîte à outils python »)*

Q1. Noter vos observations : recopier l'allure de $y(t)$, commenter son allure.

☞ Exploiter le graphe pour déterminer la constante de raideur k.

Q2. Noter les mesures effectuées pour déterminer k, et noter sa valeur.

Q3. Comment déterminer la position d'équilibre $y_{éq}$ en utilisant le tableau de `y` ?

☞ Calculer, avec python, $y_{éq}$, et l'affecter à une variation `y_eq`.

III Vitesse

Exploitation : vitesse

La vitesse à l'instant t_i est approximée par : $v(t_i) = \dfrac{y(t_{i+1}) - y(t_i)}{\Delta t}$, avec $\Delta t = t_{i+1} - t_i$ le pas de temps.

☞ 💻 Calculer le tableau des vitesses de la masse. On pourra s'aider du code ci-dessous :

```
v=np.zeros(len(y)-1) # initialisation du tableau des vitesses, qui a un élément de moins que y, car la dernière vitesse ne peut pas être calculée selon la définition précédente
for i in range(0,len(y)-1):
    v[i]=          # (à compléter) : v[i] est la vitesse à l'instant ti
```

☞ 💻 Représenter la vitesse en fonction du temps.
On fera attention au fait que la vitesse n'est pas définie au dernier instant. Par conséquent le tracé doit se faire en utilisant, en abscisse, le tableau `t` privé de son dernier élément : `t[:-1]` ou `t[:len(t)-1]`.

Q4. Recopier l'allure de $v(t)$. Commenter.

IV Aspect énergétique

Exploitation : Énergies

☞ 💻 Sans boucle `for`, en utilisant la facilité permise par les tableaux `numpy`, calculer le tableau `Ec` des énergies cinétiques (une ligne suffit).

L'énergie potentielle totale du système s'exprime selon :

$$\begin{aligned}\mathscr{E}_p &= \mathscr{E}_{p,él} + \mathscr{E}_{pp} \\ &= \frac{1}{2}k(\ell - \ell_0)^2 - mgy + K\end{aligned}$$

En choisissant l'origine de l'énergie potentielle à la position d'équilibre, $y_{éq} = \ell_0 + \dfrac{mg}{k}$, déterminée précédemment, on peut montrer que l'énergie potentielle totale s'écrit

$$\mathscr{E}_p = \frac{1}{2}k(y - y_{éq})^2$$

☞ 💻 Comme pour l'énergie cinétique, calculer le tableau `Ep` des énergies potentielles (une ligne suffit).

☞ 💻 Obtenir le tableau `Em` des énergies mécaniques (attention, l'énergie cinétique n'est pas définie pour le dernier instant puisque la vitesse ne l'est pas).

☞ 💻 Tracer sur le même graphe, les courbes de l'énergie cinétique, de l'énergie potentielle et de l'énergie mécanique (attention, l'énergie cinétique et l'énergie mécanique ne sont pas définies pour le dernier instant). Les courbes devront être légendées.

Q5. Recopier l'allure des courbes.

Q6. Commenter les courbes obtenues.

V Amélioration

Le calcul numérique de la dérivée de données expérimentales donne souvent des résultats « moches » . En effet, une variation importante de la donnée expérimentale liée à une erreur de pointé (par exemple), donne une valeur de la dérivée bien plus importante que celle qu'elle devrait avoir.

Pour remédier à cela, on peut commencer par « lisser » les données expérimentales par calcul de moyenne glissante. On choisit de calculer la moyenne sur trois points :

$$y_m(t_i) = \frac{y(t_{i-1}) + y(t_i) + y(t_{i+1})}{3}$$

```
# Tableau ym des valeurs de y moyennées
ym=np.zeros(len(y))
for i in range(1,len(y)-1): # le 1er instant et le dernier ne peuvent pas être calculées de la même manière
    ym[i]=              # à compléter

ym[0]=(y[0]+y[1])/2 # instant t=0
ym[-1]=(y[-1]+y[-2])/2 # dernier instant
```

Exploitation

☞ Reprendre toutes les exploitations précédentes en utilisant la liste `ym` à la place de `y`.

Q7. Commenter les allures, et notamment l'effet du lissage.