BLAISE PASCAL
PT 2020-2021

TP 5 – Informatique

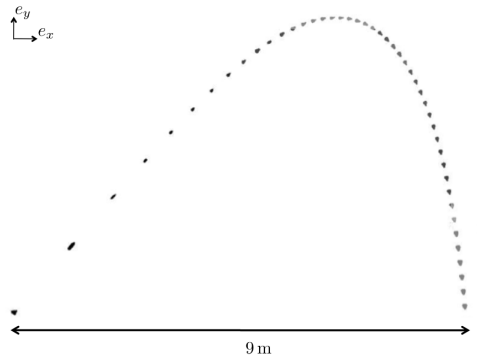
Modélisation de la trajectoire d'un volant de badminton

L'objectif de ce TP mi-informatique mi-physique est d'analyser la force de frottement subie par un volant de badminton à partir de l'étude de sa trajectoire basée sur une chronophotographie. Nous résoudrons numériquement l'équation du mouvement du volant pour deux modèles de frottements, linéaire et quadratique, en utilisant la méthode d'Euler puis une fonction issue d'une bibliothèque. Les résultats obtenus seront ensuite comparés à la chronophotographie.

Bibliothèques utilisées : importer les bibliothèques `numpy` et `matplotlib.pyplot`.

Données : masse du volant étudié $m = 5{,}3\,\text{g}$ et accélération de la pesanteur $g = 9{,}81\,\text{m}\cdot\text{s}^{-2}$; à définir comme deux variables globales.

I - Trajectoire expérimentale



Notre étude se base sur un enregistrement vidéo de la trajectoire du volant, extrait d'une thèse[1], où des images du volant sont prises à la caméra rapide avec une période d'échantillonnage $T_\text{e} = 45\,\text{ms}$. Toutes ces images ont été superposées pour donner la chronophotographie ci-contre.

Les positions successives du volant ont été repérées à l'aide du logiciel ImageJ, puis exportées au format texte dans le fichier `pointage.txt` à récupérer sur l'ENT. Enregistrer ce fichier *dans le même dossier* que votre programme Python.

1 - Recopier les lignes suivantes dans votre fichier Python :

```
1 pointage = np.loadtxt("pointage.txt",skiprows=1)
2 Xexp = pointage[:,5]
3 Yexp = pointage[:,6]
4
5 x = Xexp - Xexp[0]
6 y = -(Yexp - Yexp[0])
```

Ouvrir le fichier `pointage.txt`, par exemple avec le bloc-notes, et afficher dans la console les deux listes `Xexp` et `Yexp`. En déduire ce que font les trois premières lignes proposées ci-dessus. Les lignes 5 et 6 permettent de corriger deux « défauts » de l'export réalisé par ImageJ : l'origine du repère est replacé à la position initiale du volant (au lieu d'un coin d'image), et l'axe des ordonnées réorienté vers le haut (alors qu'ImageJ l'oriente vers le bas).

2 - Afficher la trajectoire expérimentale, c'est-à-dire `y` en fonction de `x`. On pourra utiliser l'option d'affichage `marker='o'` pour représenter des points.

II - Modélisation

II.1 - Équation du mouvement

Rappelons que l'objectif du TP est de comparer les trajectoires prévues par deux modèles de frottement à la trajectoire expérimentale pour déterminer lequel est le plus proche de la réalité. Ces deux modèles sont ceux d'une force de frottement linéaire, c'est-à-dire de norme proportionnelle à la vitesse du volant, ou quadratique, c'est-à-dire de norme proportionnelle au carré de sa vitesse :

$$\vec{F}_\text{lin} = -a\,\vec{v} \qquad \text{ou} \qquad \vec{F}_\text{quadr} = -bv\,\vec{v}$$

1. Thèse de Baptiste Darbois-Texier, réalisée au laboratoire PMMH de l'ESPCI.

Étienne Thibierge, 4 décembre 2020, www.etienne-thibierge.fr

où on note comme d'habitude $v = ||\vec{v}||$. Les deux coefficients de frottement a et b sont phénoménologiques : ils ne sont pas connus a priori, et dépendent entre autres de la forme du volant et de la façon dont il se déforme au cours de son mouvement.

Par application du théorème de la résultante cinétique au volant dans le référentiel terrestre considéré galiléen, on obtient l'équation du mouvement sous la forme

$$\frac{\mathrm{d}\vec{v}}{\mathrm{d}t} + \frac{a}{m}\vec{v} = \vec{g} \qquad \text{ou} \qquad \frac{\mathrm{d}\vec{v}}{\mathrm{d}t} + \frac{b}{m}v\,\vec{v} = \vec{g}\,.$$

La première est la classique équation linéaire du premier ordre, en revanche la deuxième est non-linéaire, ce qui la rend moins simple à résoudre analytiquement.

II.2 - Estimation des coefficients de frottement

3 - Sur chacune des deux équations, montrer qu'après une phase transitoire la vitesse du volant tend vers une vitesse limite $\vec{V}_{\mathrm{lim}}$ verticale. En déduire que la mesure de cette vitesse limite permet de déterminer a et b.

4 - À partir des deux tableaux `numpy` de coordonnées `x` et `y`, estimer numériquement les valeurs de a et b. Les conserver sous forme de deux variables globales `a` et `b`, au même titre que g, m ou encore T_{e}. À partir de l'observation de la chronophotographie, indiquer ce qui limite la précision de l'estimation.

II.3 - Estimation de la vitesse initiale

5 - Pour pouvoir résoudre les équations du mouvement, il est nécessaire de connaître les composantes de la vitesse initiale $\vec{v}_0$ du volant. En adoptant la même démarche que précédemment, déterminer numériquement `V0x` et `V0y`

III - Résolution par la méthode d'Euler

6 - Implémenter deux fonctions `acc_lin` et `acc_quadr` prenant comme arguments deux tableaux `numpy Vx` et `Vy` contenant les deux composantes du vecteur vitesse du volant et renvoyant deux tableaux `numpy Ax` et `Ay` avec les composantes de son vecteur accélération, respectivement pour le modèle linéaire et quadratique.

7 - Coder une fonction `avec_euler` capable de résoudre l'équation du mouvement, c'est-à-dire de renvoyer trois tableaux de même longueur `t` (temps), `Vx` et `Vy` (composantes de la vitesse au cours du temps). Cette fonction prend comme seul argument une fonction `acceleration`, qui prend elle-même les mêmes arguments que `acc_lin` et `acc_quadr`. On rappelle que toutes les autres variables, notamment les conditions initiales, sont définies comme des variables globales puisqu'il s'agit d'analyser la chronophotographie. Choisir un pas de temps $\Delta t = T_{\mathrm{e}}/50$.

8 - En s'inspirant de la méthode d'Euler, écrire une fonction `position` prenant pour arguments deux listes `Vx` et `Vy` contenant les composantes du vecteur vitesse et renvoyant deux listes `X` et `Y` contenant les composantes du vecteur position du volant. On rappelle qu'à l'instant initial le volant se trouve au point de coordonnées $(0, 0)$.

9 - En combinant les différentes fonctions, calculer puis représenter les trajectoires prévues par le modèle linéaire et par le modèle quadratique. Conclure sur le meilleur modèle.

10 - L'auteur de la thèse indique une vitesse initiale $V_0 = 40\,\mathrm{m}\cdot\mathrm{s}^{-1}$ formant un angle $\theta_0 = 52°$ avec l'horizontale, et une vitesse limite $V_{\mathrm{lim}} = 6{,}7\,\mathrm{m}\cdot\mathrm{s}^{-1}$. Remplacer vos valeurs estimées par celles de l'auteur (**Attention !** commentez vos expressions, mais ne les effacez pas, pour pouvoir revenir en arrière.). Commenter l'influence de ce changement.

IV - Résolution avec une bibliothèque

Pour analyser les performances de la méthode d'Euler, nous allons comparer les résultats de la section précédente à ceux obtenus en utilisant une fonction déjà implémentée dans une bibliothèque de calcul scientifique : la fonction `odeint` du module `scipy.integrate`, que l'on importera par `from scipy.integrate import odeint`.

11 - La fonction `odeint` prend trois arguments obligatoires, nommés `func`, `y0` et `t` dans la documentation, ainsi qu'un certain nombre d'arguments optionnels que l'on n'utilisera pas. Ouvrir la documentation (`help(odeint)` dans la console) et comprendre ce qu'ils représentent et les informations physiques qu'ils contiennent.

12 - Coder les fonctions utiles à la résolution de l'équation par `odeint`. On gagnera à travailler avec un tableau `cinematique` contenant quatre éléments, eux-mêmes des tableaux, et représentant respectivement `X`, `Y`, `Vx` et `Vy`.

13 - Résoudre l'équation du mouvement avec `odeint` et comparer les résultats à ceux donnés par la méthode d'Euler. On s'intéressera en particulier à l'influence du pas de temps et des conditions initiales.