

# TP n°12 : Trajectoire d'un volant de badminton

## Comment modéliser numériquement la trajectoire d'un point ?

L'objectif de ce TP est d'analyser la force de frottements subie par un volant de badminton à partir de l'étude de sa trajectoire. Nous établirons l'équation du mouvement du volant pour un modèle de frottements linéaire, et la résoudrons numériquement par la méthode d'Euler. Les résultats obtenus seront ensuite confrontés à la chronophotographie du vol d'un volant.

### Méthodes mises en oeuvre

- ▷ Résoudre numériquement une équation différentielle avec la méthode d'Euler ;
- ▷ Réaliser une chronophotographie ;
- ▷ Confronter les résultats expérimentaux aux expressions théoriques.

## Étude théorique du mouvement

On suppose que le volant de badminton de masse  $m = 5,3 \text{ g}$  est soumis à une force de frottements fluide de la forme :

$$\vec{F} = -\alpha \vec{v} \quad (1)$$

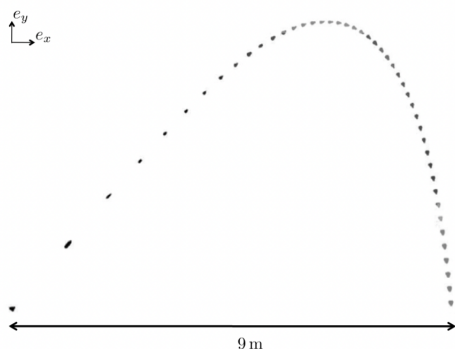
où  $\alpha$  est un coefficient de frottement positif (en  $\text{kg}\cdot\text{s}^{-1}$ ).

🔗 Appliquer le principe fondamental de la dynamique au volant. En déduire une équation différentielle sur la vitesse  $\vec{v}$  du volant. Projeter ensuite cette équation différentielle selon les axes  $Ox$  et  $Oy$ .

🔗 Montrer qu'après une phase transitoire la vitesse du volant tend vers une vitesse limite  $V_{\text{lim}}$  verticale. En déduire que la mesure de cette vitesse limite permet de déterminer  $\alpha$ .

🔗 En déduire les deux relations de récurrence, sur  $v_x$  et  $v_y$ , afin de résoudre numériquement l'équation différentielle. Vous pourrez vous aider de la **fiche méthode Euler**, distribuée en début d'année, disponible sur Cahier de prépa.


## I - Chronophotographie





Récupérer le fichier image *chronophotobadminton.png* (dans le dossier PTSI 2022/2023), extrait d'une thèse, et l'ouvrir avec le logiciel *ImageJ*. Les photos du volant sont prises à intervalle régulier  $T_e = 45 \text{ ms}$  avec une caméra rapide, puis toutes superposées sur la même image.


L'objectif de cette première partie est de récupérer dans un fichier texte les coordonnées successives du volant pour pouvoir ensuite les importer sous Python et y visualiser la trajectoire.

## • Pointage sous ImageJ


 **Imposer une échelle sur l'image.** À l'aide de l'outil "ligne droite" tracer une ligne sur l'image entre deux points séparés d'une distance connue. Pour passer des pixels à une longueur physique, aller dans le menu `Analyze` → `Set Scale` et indiquer la longueur connue dans la case `Known distance`, sans oublier de préciser l'unité de la longueur.

 **Pointer les positions successives de l'avant du volant.** Utiliser pour cela l'outil `Point`, et cliquer sur les positions successives du volant tout en maintenant la touche Maj enfoncée. Une fois toutes les positions sélectionnées, aller dans les menus `Analyze` → `Set Measurements` et décocher les 3 cases cochées par défaut.

 Cliquez sur l'onglet `Measure` dans le menu `Analyze` : les résultats s'affichent dans un tableau, qui doit contenir 46 lignes si vous n'avez oublié aucun point. La première colonne de ce tableau contient le numéro du point, la deuxième correspond aux coordonnées  $x$  et la troisième aux coordonnées  $y$  du point que l'on cherche à récupérer.


 Enregistrer ce tableau sous forme d'un fichier `pointage.txt`, placé dans le même dossier que celui où vous allez créer votre programme Python. Si le logiciel vous propose un autre format (par exemple `.xls`), remplacez-le par `txt`.


## • Import sous Python

 Ouvrir le fichier Python "ScriptTP12" dédié à ce TP, **situé dans le même dossier que celui où est enregistré votre fichier `pointage.txt`**

Les lignes suivantes permettent d'ouvrir le fichier :

```
1 | pointage = np.loadtxt("pointage.txt",skiprows=1)
2 | Xexp = pointage[:,1]
3 | Yexp = pointage[:,2]
```

 Ouvrir le fichier `pointage.txt`, par exemple avec le bloc-notes, et afficher dans la console (ou l'explorateur de variables) Python les deux listes `Xexp` et `Yexp`.

 En déduire ce que font les trois lignes proposées ci-dessus.

## • Affichage de la trajectoire

Les lignes de code ci-dessous permettent d'afficher la trajectoire issue du pointage sous ImageJ :

```
1 | plt.figure()
2 | plt.title('Trajectoire du volant de badminton')
3 | plt.grid()
4 | plt.plot(x, y, '+',label='Trajectoire expérimentale')
5 | plt.xlabel('$x$ (en m)')
6 | plt.ylabel('$y$ (en m)')
```


 Afficher la trajectoire issue de votre pointage ImageJ, et vérifier la cohérence avec la trajectoire attendue.

## II - Modélisation

### ● Estimation du coefficient de frottement

✎ Justifier que la ligne de code ci-dessous permet d'estimer la vitesse limite expérimentale  $V_{lim}$ .

```
1 | Vlim = np.sqrt( (x[-2] - x[-1])**2 + (y[-2] - y[-1])**2 ) / Te
```

 Compléter la ligne de code ci-dessous pour calculer numériquement la valeur du coefficient  $\alpha$ .

```
1 | alpha =
```

### ● Estimation de la vitesse initiale


Pour pouvoir résoudre les équations du mouvement, il est nécessaire de connaître les composantes de la vitesse initiale  $\vec{v}_0$  du mouvement.

 Compléter les deux lignes de code ci-dessous pour calculer les vitesses initiales  $v_{0x}$  et  $v_{0y}$  :


```
1 | v0x =
2 | v0y =
```

### ● Implémentation du schéma numérique d'Euler

Compléter la fonction ci-dessous résolvant l'équation du mouvement grâce au schéma numérique proposé.


 Cette fonction prendra en argument le pas de temps  $dt$  et le nombre de points  $N$ , et renverra trois listes de même longueur  $t$  (temps),  $vx$  et  $vy$  (composantes de la vitesse à chaque instant).

```
1 | def vitesse_euler(N,dt) :
2 |
3 |
4 |
5 |
6 |
7 |
```

 Représenter l'allure de la vitesse obtenue en fonction du temps. On prendra  $dt = Te/50$  et  $N = 50 * \text{len}(x)$ .

✎ Est-ce cohérent avec la trajectoire du volant ? Justifier qualitativement. En déduire une lecture graphique de la vitesse limite, et comparer avec la valeur obtenue dans le début de partie.

En reprenant le schéma d'Euler, compléter la fonction ci-dessous prenant pour arguments le pas de temps  $dt$  et deux listes  $V_x$  et  $V_y$  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 par convention au point de coordonnées  $(0,0)$ .

```
1 | def position (dt,vx,vy) :
2 |
3 |
4 |
5 |
6 |
7 |
```

### III - Résultats numériques



Représenter la trajectoire obtenue précédemment. Comparer à la trajectoire obtenue par la chronophotographie.

L'auteur de la thèse indique une vitesse initiale  $v_0 = 40 \text{ m}\cdot\text{s}^{-1}$  formant un angle  $\theta_0 = 52^\circ$  avec l'horizontale, et une vitesse limite  $V_{\text{lim}} = 6,7 \text{ m}\cdot\text{s}^{-1}$ .



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 - Modèle de frottements quadratiques

Si l'on adopte un modèle de frottements quadratiques, la force s'écrit cette fois :

$$\vec{F} = -kv\vec{v} \quad (2)$$

où  $v$  désigne la norme du vecteur vitesse :  $v = \sqrt{v_x^2 + v_y^2}$ .

✎ En déduire, dans le cadre des frottements quadratiques, les deux relations de récurrence, sur  $v_x$  et  $v_y$ , afin de résoudre numériquement l'équation différentielle.



Modifier la fonction EULER implémentée précédemment pour modéliser la trajectoire à l'aide de frottements quadratiques.

✎ Comparer au cas des frottements linéaires. Quel modèle de frottement faut-il donc adopter ?