

PROGRAMMER EN PYTHON

FICHE N°8 : TRAJECTOIRE PARABOLIQUE ET ÉNERGIE

L'étude du mouvement d'une balle de golf à partir d'une chronophotographie permet de construire les différentes formes d'énergies associées au système, puis de tester l'hypothèse de la conservation de l'énergie mécanique.

Capacité numérique mise en œuvre

Représenter, à partir de données expérimentales variées, l'évolution des grandeurs énergétiques d'un système en mouvement dans un champ uniforme à l'aide d'un langage de programmation ou d'un tableur.

Présentation

On utilise des valeurs expérimentales (issues d'un pointage sur une vidéo) pour créer et déterminer des grandeurs énergétiques (énergie cinétique, potentielle et mécanique) au cours du mouvement.

Les mesures pointées sont :

```
8 #%%
9 import numpy as np
10 import matplotlib.pyplot as plt
11 # valeurs experimentales
12
13 x=np.array([0.41,0.91,1.39,1.86,2.33,2.81,3.33,3.79,4.31,4.83,5.36,5.86,6.34])
14 y=np.array([0.52,1.01,1.42,1.77,2.02,2.16,2.19,2.13,1.96,1.71,1.38,0.97,0.50])
15 t=0.1*np.arange(len(x))
```

On pourra utiliser la fiche 7 pour illustrer le fait que les variations du vecteur vitesse sont pratiquement verticales et illustrer la seconde loi de Newton.

Construction de la liste des vecteurs vitesse

Le script utilise deux listes : l'une contenant les valeurs successives de la vitesse horizontale, l'autre, celles de la vitesse verticale. Ces valeurs sont calculées une à une au moyen d'une boucle **for** puis ajoutées aux listes grâce à la fonction **append**. Le script débute l'initialisation de la liste : la ligne `vx=[]` qui crée une liste vide. Puis pour chaque élément d'une liste redimensionnée (il est impossible de calculer $x[i+2]$), la vitesse horizontale v_x est calculée, puis

ajoutée à la liste `vx1`, grâce à `vx1.append(vx)`. On transforme ensuite ces listes en `np.array` pour pouvoir effectuer des opérations sur elles – par exemple, pour calculer leur carré, les sommer et prendre la racine de la somme, ce qui permettra de calculer la norme de la vitesse, qui intervient dans l'énergie cinétique.

```
vx1=[]
m=np.arange(len(t)-2)
for i in m :
    vx=(x[i+2]-x[i])/(t[i+2]-t[i])
    vx1.append(vx)
vx1l=np.array(vx1)
vy1=[]
m=np.arange(len(t)-2)
for i in m :
    vy=(y[i+2]-y[i])/(t[i+2]-t[i])
    vy1.append(vy)
vy1l=np.array(vy1)
```

Construction de la liste des vecteurs variation de vitesse par unité de temps

Avec exactement la même démarche, il est possible de faire calculer les variations des vitesses horizontale et verticale par unité de temps. Il faut simplement, là encore, prendre garde à redimensionner les listes.

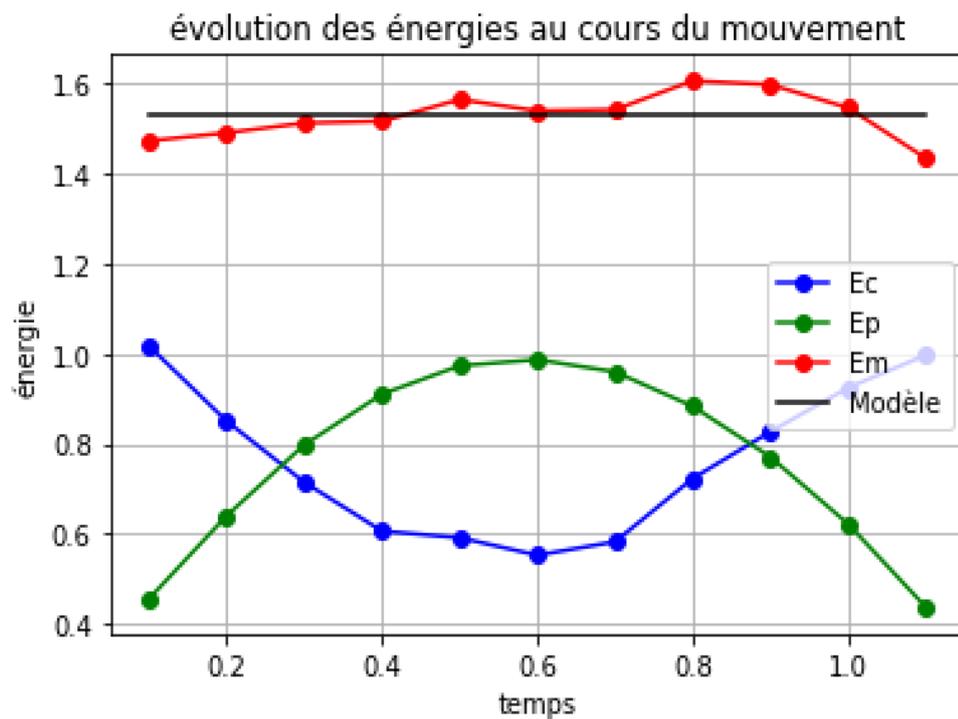
Représentation des énergies pertinentes

Il est alors possible de tracer les énergies cinétique, potentielle et mécanique au cours du mouvement, notamment pour vérifier s'il y a conservation d'énergie mécanique. Il faut prendre garde à la taille des listes manipulées, car la construction des listes de vitesse fait que celle-ci comporte forcément deux valeurs de moins que la liste des positions. Ainsi, il faut redimensionner les listes de temps et de position pour pouvoir tracer les différentes grandeurs et effectuer des opérations mathématiques conjointes sur celles-ci.

```
32 m=0.046
33 g=9.8
34 v=(vx1l**2+vy1l**2)**(1/2)
35 ec=0.5*m*v**2
36 y2=y[1:-1]
37 ep=m*g*y2
38 em=ec+ep
39 t2=t[1:-1]
40
41 plt.plot(t2,ec,'bo-',label='Ec')
42 plt.plot(t2,ep,'go-',label='Ep')
43 plt.plot(t2,em,'ro-',label='Em')
44 plt.xlabel("temps")
45 plt.ylabel("énergie")
46 plt.title("évolution des énergies au cours du mouvement")
47 plt.legend()
48 plt.grid()
49 plt.show()
```

Retrouvez éducol sur





Conclusion

On observe que l'énergie mécanique n'évolue pas significativement au cours du temps car on note une absence de tendance marquée, notamment à la baisse.

Retrouvez éducol sur

