M1. Illustration de quelques lois de la dynamique newtonienne

Préparation à l'Agrégation de Physique

Introduction

L'objectif de ce notebook est d'illustrer deux piliers de la mécanique classique à travers une exploitation de données numériques :

1. **La conservation du moment cinétique** via l'étude d'une force centrale (Loi des aires).
2. **La dynamique d'un oscillateur amorti** soumis à une force de frottement fluide visqueux.

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from scipy.optimize import curve_fit
from scipy.signal import find_peaks
def f_lin(x, a, b):
    return a*x+b
from PIL import Image
```

1. Illustration de la Loi des Aires

Rappels théorique

Pour un point matériel M de masse m soumis à une force centrale $\vec{F}$ de centre O, le moment cinétique $\vec{L}_O = \vec{OM} \wedge m\vec{v}$ est une constante du mouvement.

Cette conservation entraîne deux conséquences géométriques majeures :

- **Planéité du mouvement :** La trajectoire est contenue dans un plan perpendiculaire au vecteur $\vec{L}_O$.
- **Loi des aires :** L'aire balayée par le rayon vecteur $\vec{OM}$ par unité de temps, appelée vitesse aérolaire, est constante :

$$\frac{dA}{dt} = \frac{L_O}{2m} = \text{cte}$$

Dans l'expérience du mobile autoporteur, nous vérifions cette constante en calculant l'aire élémentaire entre deux positions successives M_i et M_{i+1} :

$$\Delta A_i \approx \frac{1}{2}|x_i y_{i+1} - x_{i+1} y_i|$$

Pour faire cette expérience, nous filmons le mobile avec une caméra qui se situe au niveau de l'accroche du ressort (point A sur le schéma ci-dessous).

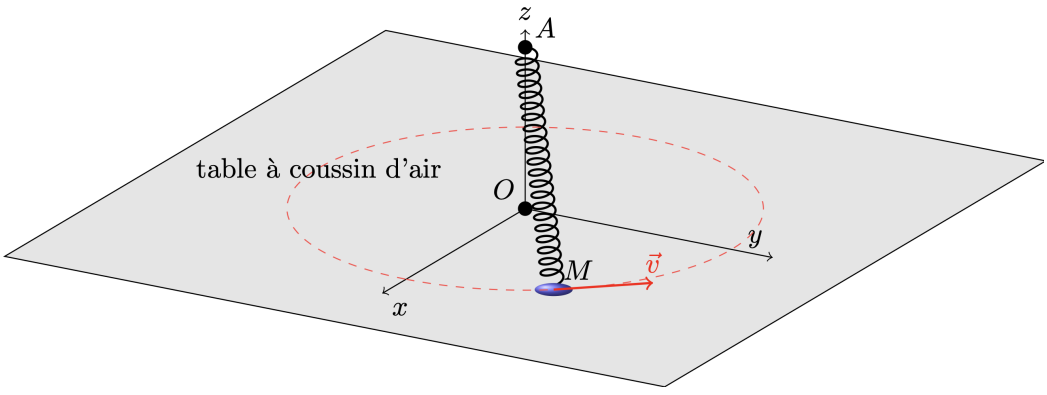


Figure 1: Schéma de l'expérience

Position et vitesse

A l'aide du logiciel Tracker, nous relevons la position et la vitesse du mobile, et traçons leurs évolutions en fonction du temps.

```
datA=pd.read_csv("Loi_Des_Aires.csv", decimal=",", delimiter=";")
tAire = datA["t1"].values
xAire = datA["x1"].values
yAire = datA["y1"].values
L = datA["L"].values
L = np.delete(L, [68])

T = 0.033
vit = pd.read_csv("Vitesse_Tracker.csv", decimal=".", delimiter=";")
vt = vit["v"].values
tAire = np.delete(tAire, [67,68])

plt.figure(0, figsize=(14,6))

plt.subplot(1,2,1)
plt.plot(xAire, yAire, 'r+', label="mobile")
plt.plot(0,0, 'ko')
plt.text(0.02,0,"O")
plt.xlabel("y(m)", color="green", size=15)
plt.ylabel("x(m)", color="green", size=15)
plt.title("Trajectoire du Mobile", color="green")
plt.legend()
plt.grid()

plt.subplot(1,2,2)
plt.plot(tAire, vit, 'r+', label="vitesse du mobile")
plt.ylabel("v(m.s-1)", color="green", size=15)
plt.xlabel("t(s)", color="green", size=15)
plt.title("Vitesse du Mobile", color="green")
plt.legend()
plt.grid()
```

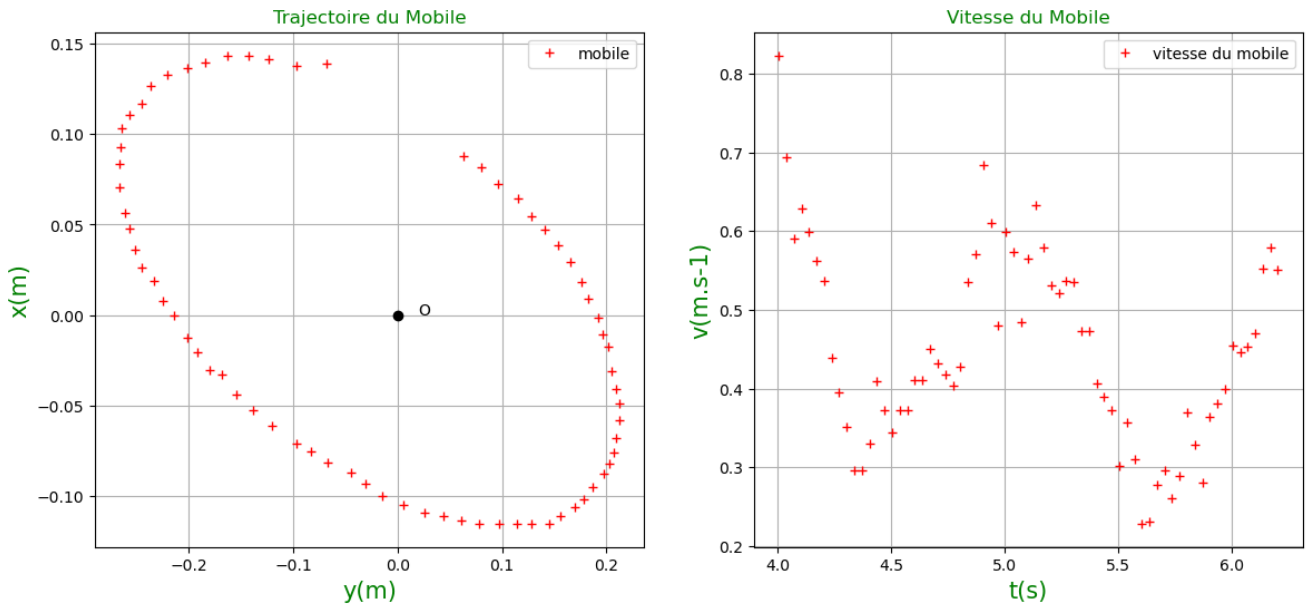
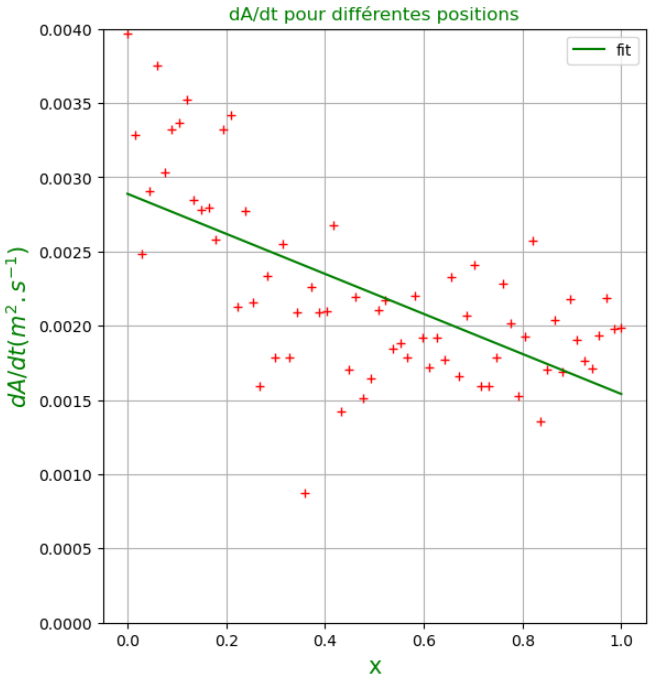


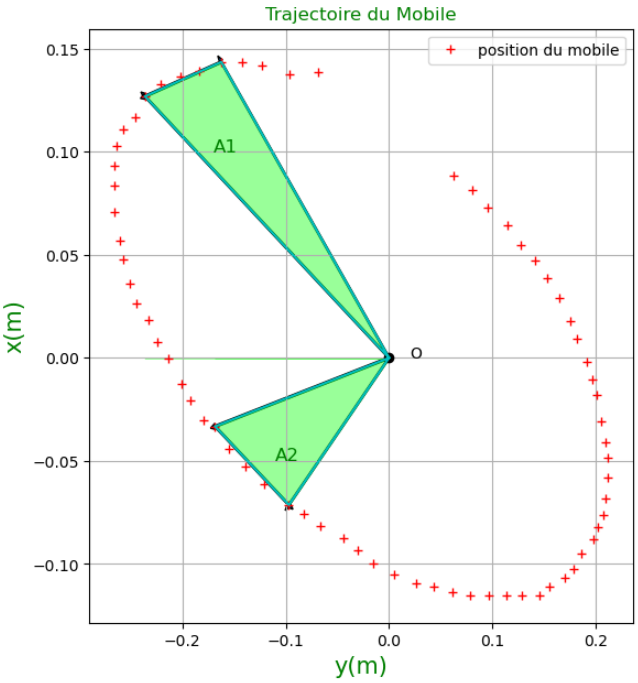
Figure 2: Evolution de la position et de la vitesse au cours du temps

Etude des aires balayées

```
x = np.linspace(0,1,len(L))
Lf = L/(2*0.5)

plt.figure(0, figsize=(14,7))

plt.subplot(1,2,1)
plt.plot(x, Lf, 'r+')
plt.ylim(0,0.004)
pop2, pcov2 = curve_fit(f_lin, x, Lf)
aop2, bop2 = pop2
yth2 = []
yth2 = aop2*x+bop2
plt.plot(x, yth2, 'g-', label="fit")
plt.xlabel("x", color="green", size=15)
plt.ylabel("$dA/dt (m^2.s^{-1})$", color="green", size=15)
plt.title("dA/dt pour différentes positions", color="green")
plt.legend()
plt.grid()

delta_aop2 = np.sqrt(pcov2.diagonal(0))
delta_aop2 = delta_aop2[0]
err2 = -delta_aop2/aop2
print("Incertitude sur le fit :", np.round(err2, 4)*100,"%")

plt.subplot(1,2,2)
plt.plot(xAire, yAire, 'r+', label="position du mobile")
plt.arrow(0,0, xAire[4],yAire[4])
plt.arrow(0,0, xAire[8],yAire[8])
plt.arrow(xAire[4],yAire[4],xAire[8]-xAire[4], yAire[8]-yAire[4])
plt.arrow(0,0, xAire[25],yAire[25])
plt.arrow(0,0, xAire[29],yAire[29])
plt.arrow(xAire[25],yAire[25],xAire[29]-xAire[25], yAire[29]-yAire[25])
plt.plot(0,0, 'ko')
plt.text(0.02,0,"O")
plt.text(-0.17, 0.1, "A1", color="green", size=12)
plt.text(-0.11, -0.05, "A2", color="green", size=12)
plt.xlabel("y(m)", color="green", size=15)
plt.ylabel("x(m)", color="green", size=15)
plt.title("Trajectoire du Mobile", color="green")
plt.legend()
plt.grid()

area_x = [0,xAire[4],xAire[8],0]
area_y = [0,yAire[4],yAire[8],0]
plt.plot(area_x,area_y,"-c")
plt.fill_between(area_x, area_y,alpha=0.4, color="lime")
area_x = [0,xAire[25],xAire[29],0]
area_y = [0,yAire[25],yAire[29],0]
plt.plot(area_x,area_y,"-c")
plt.fill_between(area_x, area_y,alpha=0.4, color="lime")

print("y=",np.round(aop2,4),"*x")
```

```
incertitude sur le fit : 1.4 %
V = 71.09 * d  -12.894
```





Analyse des résultats expérimentaux

Étude de la conservation du moment cinétique – mobile autoporteur + ressort

---

Observation

- Les points expérimentaux (croix rouges) présentent une **dispersion notable**.
- Une **tendance linéaire décroissante** est mise en évidence par la droite de régression.
- La grandeur mesurée (\frac{dA}{dt}) diminue lorsque (x) augmente.

Interprétation physique

- En théorie, dans un système **isolé soumis à une force centrale** (comme un ressort idéal), on doit avoir :

$$\frac{d\vec{L}}{dt} = \vec{M}_{\text{ext}} = \vec{0} \quad \Rightarrow \quad \frac{dA}{dt} = \text{constante}$$

- Or ici, $ \frac{dA}{dt} $ **n'est pas constant** :
  - La pente négative indique une **diminution progressive du moment cinétique**.

Causes possibles d'écart à la théorie

- **Frottements résiduels** (air ou table)
- **Ressort non idéal** (force pas parfaitement centrale)
- **Erreurs expérimentales** :
  - bruit de mesure (visible dans la dispersion)
  - incertitudes sur la position
- **Mauvais centrage du point O**

Conclusion

Le système n'est **pas parfaitement conservatif**, mais la variation reste **faible**, ce qui suggère une **approximation raisonnable de la conservation du moment cinétique**.

---

2. Étude du Ressort Fluide : De l'amortissement au coefficient C_x

2.1. Modélisation de l'oscillateur harmonique

On étudie le mouvement d'une masse m reliée à un ressort de raideur k, plongée dans un fluide de masse volumique ρ. Le système est soumis à une force de rappel élastique et une force de frottement fluide $\vec{f}$.

Selon la vitesse et la géométrie de l'objet (nombre de Reynolds Re), deux modèles de frottement sont envisageables :

1. **Modèle linéaire (Stokes) :** $\vec{f} = -h\vec{v}$, l'amplitude décroît exponentiellement.
2. **Modèle quadratique (Newton) :** $\vec{f} = -\alpha|v|\vec{v}$, l'amplitude décroît de manière hyperbolique.

2.2. Justification du modèle d'amortissement quadratique

Dans ce régime, la perte d'énergie mécanique sur une période conduit à une équation pour l'enveloppe des maxima de la forme :

$$\frac{dA}{dt} = -\gamma A^2 \implies \frac{1}{A(t)} = \frac{1}{A_0} + a_{op} \cdot t$$

La linéarité de $1/A$ en fonction du temps valide donc l'utilisation du coefficient de frottement quadratique $\alpha = \frac{1}{2}\rho S C_x$.

2.3. Détermination du coefficient de traînée C_x

Pour faire le lien entre la pente expérimentale a_{op} (issue du fit de $1/A$) et le coefficient adimensionnel C_x, on utilise une méthode de moyennage de la puissance dissipée sur une pseudo-période T. On établit la relation suivante :

$$a_{op} = \frac{4\rho S C_x \omega_0}{3\pi m}$$

En isolant le coefficient de traînée, on obtient la formule d'exploitation utilisée dans le code :

$$C_x = \frac{3\pi m a_{op}}{2\rho S \omega_0}$$

Où :

- m est la masse du système.
- S est la section droite de l'objet ($S = \pi R^2$).
- ρ est la masse volumique du fluide.
- $\omega_0 = \sqrt{k/m}$ est la pulsation propre de l'oscillateur.

2.4. Protocole d'exploitation numérique

1. **Détection de pics :** Repérer les sommets (t_i, x_i) et corriger l'offset pour obtenir les amplitudes A_i.
2. **Régression linéaire :** Effectuer le fit de $1/A_i$ en fonction du temps pour extraire la pente a_{op}.
3. **Incertitudes :** L'incertitude sur a_{op} est calculée à partir de la matrice de covariance du fit (`pcov`), permettant de quantifier la précision sur le C_x final.

Étalonnage du capteur

La principale difficulté de cette expérience réside dans la capacité à mesurer précisément l'évolution de la position de la masse au cours du temps lorsqu'elle est en mouvement dans le fluide.

Une première approche consisterait à réaliser une acquisition vidéo du mouvement, puis à effectuer un pointage afin d'extraire la position en fonction du temps. Toutefois, cette méthode ayant déjà été exploitée dans l'expérience précédente, il est pertinent ici de mettre en œuvre une technique de mesure différente.

Principe de la méthode

On impose une différence de potentiel entre deux électrodes placées aux extrémités du tube rempli d'eau. Il s'établit alors un **gradient de potentiel électrique** entre le bas et le haut de la colonne de fluide.

Dans ces conditions, la tension électrique devient une fonction de la position verticale dans le tube. Il est alors possible de réaliser un **étalonnage** en mesurant la tension pour différentes positions connues.

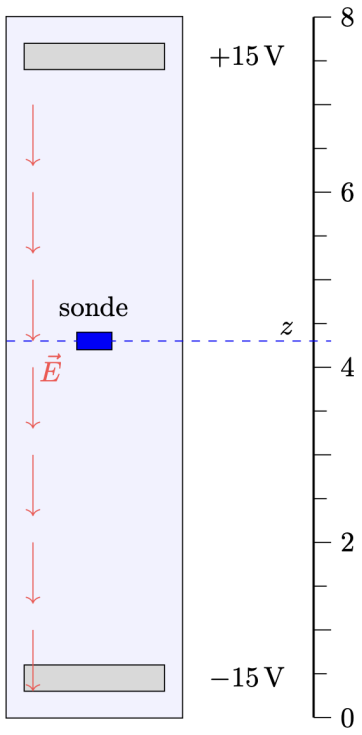
Objectif

Cet étalonnage permet d'établir une **relation affine** entre la tension mesurée U et la position z :

$$U(z) = az + b$$

Cette relation sera ensuite utilisée pour convertir directement les mesures de tension en position lors de l'étude dynamique du système.

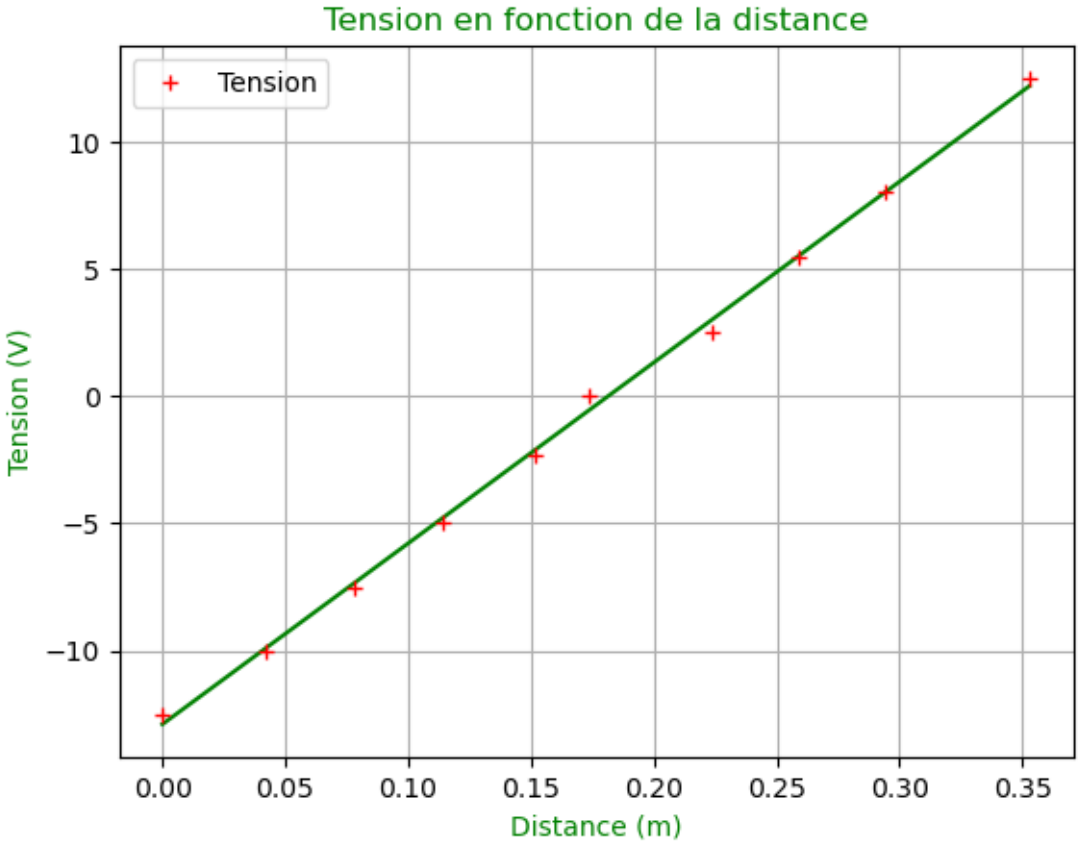
Tube rempli d'eau soumis à une ddp



Mesure de z à la règle et de $U \Rightarrow$ étalonnage $U(z)$

```
lin = pd.read_csv("Capteur_Lineaire.csv", delimiter=";", decimal=",")
Volt = lin["V"].values
Distance = lin["d"].values

popl, pcovl = curve_fit(f_lin, Distance, Volt)
aopl, bopl = popl
ythl=[]
ythl = aopl*Distance+bopl
plt.plot(Distance,ythl,'g-')

delta_aopl = np.sqrt(pcovl.diagonal(0))
delta_aopl = delta_aopl[0]
err = delta_aopl/aopl
print("incertitude sur le fit :",np.round(err*100,2),"%")

plt.plot(Distance,Volt,'r+', label="Tension")
plt.xlabel("Distance (m)",color="green")
plt.ylabel("Tension (V)",color="green")
plt.title("Tension en fonction de la distance",color="green")
plt.legend()
plt.grid()
print("V =", np.round(aopl,3),"* d ", np.round(bopl,3))
```

```
incertitude sur le fit : 1.4 %
V = 71.09 * d  -12.894
```



Étalonnage du ressort

Afin d'exploiter quantitativement le système masse–ressort, il est nécessaire de déterminer la **constante de raideur** du ressort.

La méthode retenue repose sur une étude en régime statique, dans laquelle la masse est immobile et soumise à l'action de son poids et de la force de rappel du ressort.

Principe de la méthode

On suspend successivement différentes masses au ressort et on mesure, pour chacune d'elles, l'**allongement** du ressort par rapport à sa position à vide.
À l'équilibre, les forces s'exerçant sur la masse se compensent :

- le poids, dirigé vers le bas,
- la force de rappel du ressort, dirigée vers le haut.

On obtient alors la relation :

$$k\,\Delta z = mg$$

où :

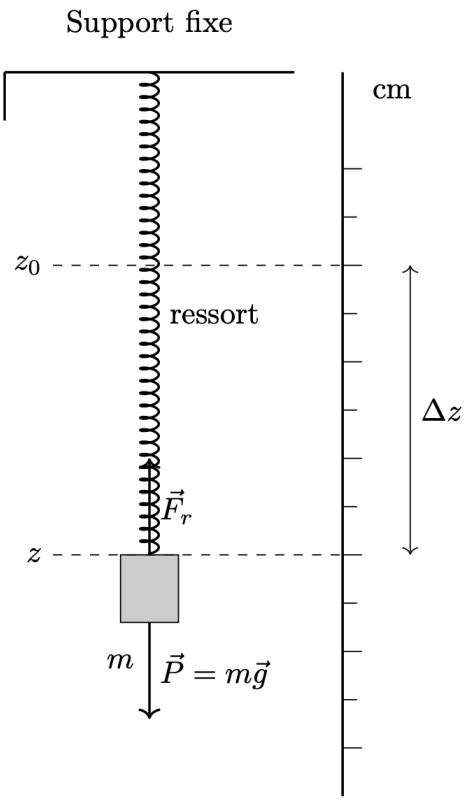
- k est la constante de raideur du ressort,
- Δz l'allongement,
- m la masse suspendue,
- g l'intensité de la pesanteur.

En traçant la courbe $\Delta z = f(m)$, on met en évidence une **relation linéaire** :

$$\Delta z = \frac{g}{k}\,m$$

La pente de la droite permet alors de déterminer la constante de raideur :

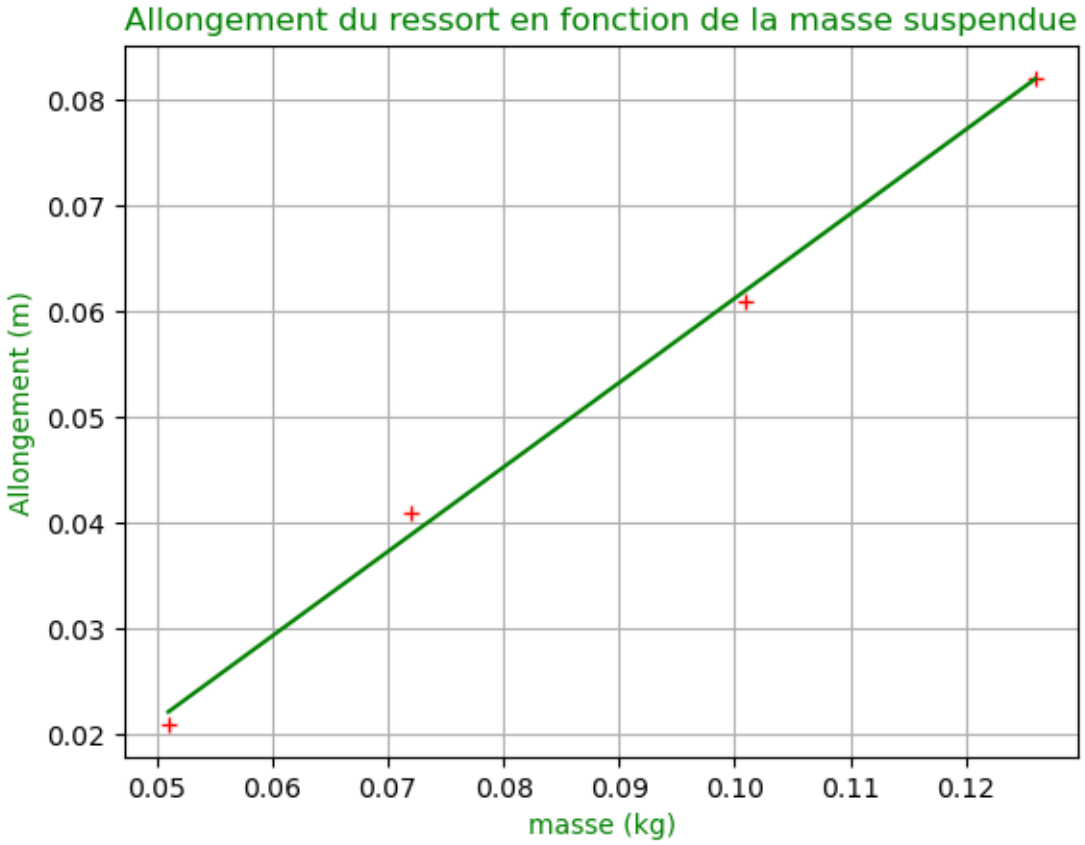
$$k = \frac{g}{\text{pente}}$$



Mesure de l'allongement Δz en fonction de la masse m

```
Lo = 0.102            ### longueur du ressort à l'équilibre (masse pendue mais immobile)
g = 9.81
mressort = np.array([0.051, 0.072, 0.101, 0.126])             ### masse ressort
AllongeRessort = np.array([0.021, 0.041, 0.061, 0.082])       ### longueur ressort
plt.plot(mressort, AllongeRessort, 'r+')

pop, pcov = curve_fit(f_lin, mressort, AllongeRessort)
aop, bop = pop

delta_aop = np.sqrt(pcov.diagonal(0))
delta_aop = delta_aop[0]
err1 = delta_aop/aop
print("incertitude sur le fit :",np.round(err1*100,2),"%")

yth = aop*mressort+bop
plt.plot(mressort, yth, "g-")
k = g/aop
print("k=",np.round(k,3),"N/m")                                ### k vaut g/pente
plt.xlabel("masse (kg)",color="green")
plt.ylabel("Allongement (m)", color="green")
plt.title("Allongement du ressort en fonction de la masse suspendue", color="green")
plt.grid()
```

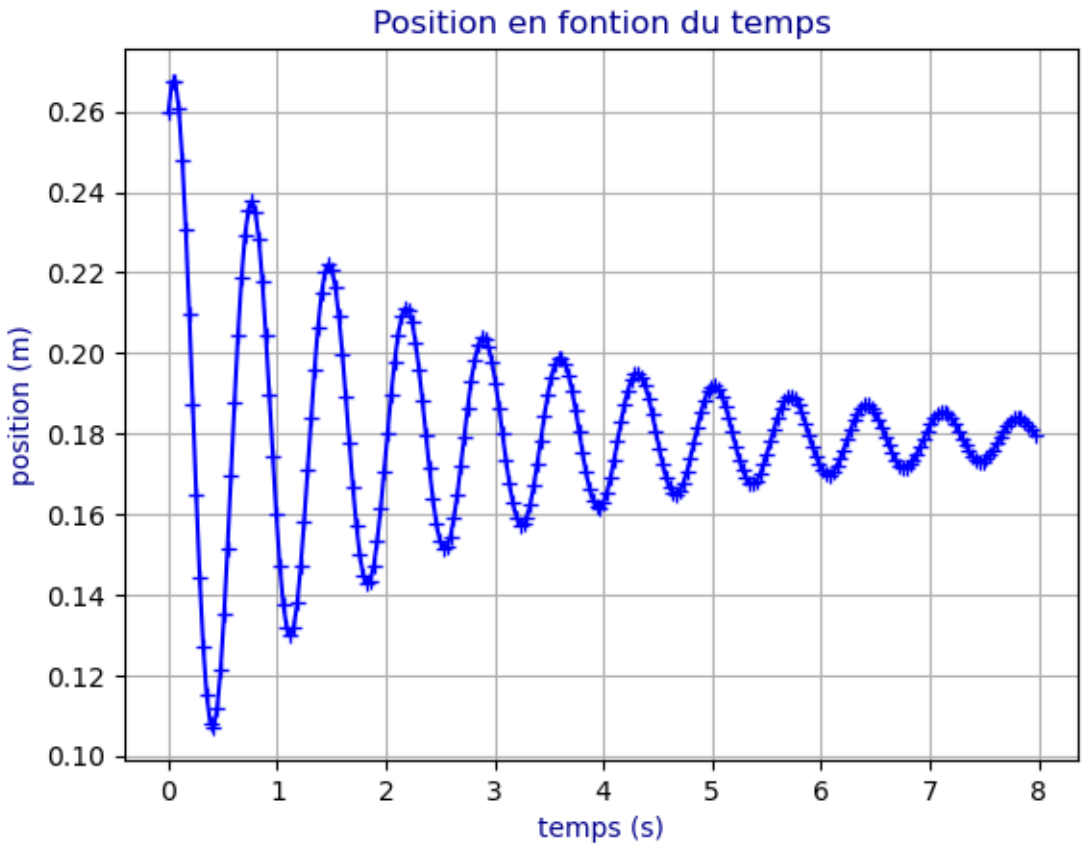
```
incertitude sur le fit : 4.07 %
k= 12.315 N/m
```



Évolution temporelle

Une fois la constante de raideur k déterminée par l'étude statique, nous passons à l'étude du comportement dynamique du système. Le graphique ci-dessous présente l'évolution de la position $z(t)$ de la masse après avoir été écartée de sa position d'équilibre. Cette série temporelle permet d'observer la nature des oscillations (fréquence et amortissement) et constitue la base de l'extraction des paramètres de frottement.

```
osci = pd.read_csv("Oscillateur_Amorti.csv", delimiter=";", decimal=",")
V = osci["V"].values
t = osci["t"].values
dt = t[2]-t[1]
pos =  (V-bopl)/aopl                          ### Conversion grâce à l'étallonage
v_osci = np.gradient(V)/dt
m = 0.170
plt.plot(t, pos, 'b+-')
plt.ylabel("position (m)",color="darkblue")
plt.xlabel("temps (s)", color="darkblue")
plt.title("Position en fontion du temps", color="darkblue")
plt.grid()
```



Nous observons bien un régime pseudopériodique

Analyse Énergétique

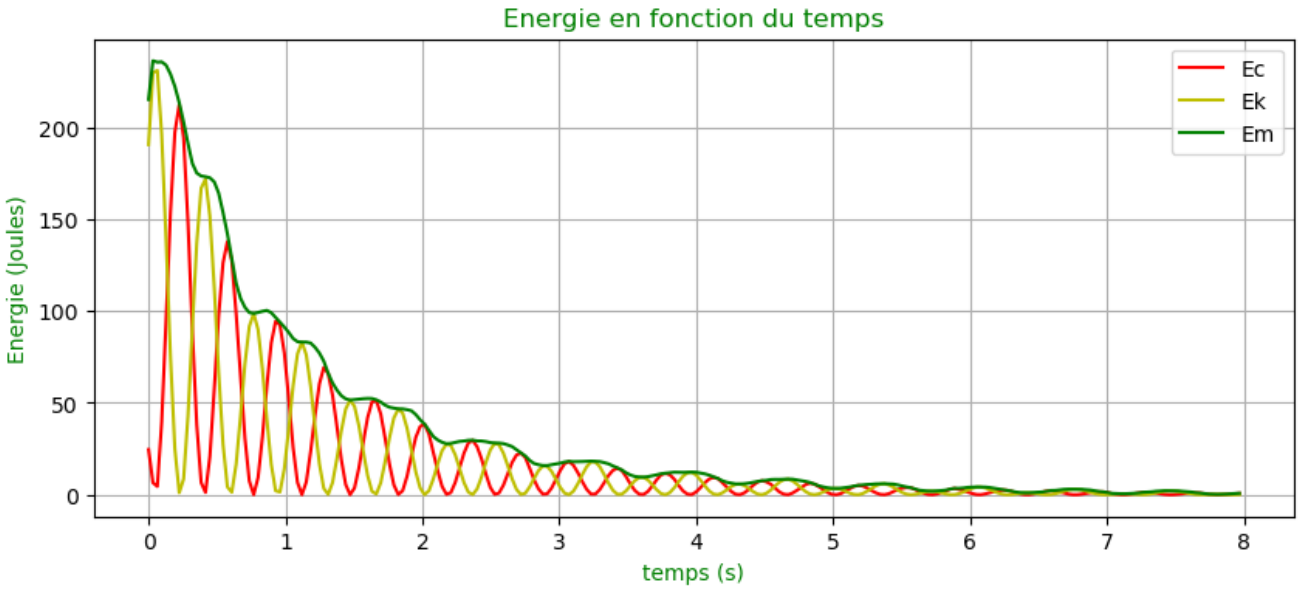
L'étude du bilan énergétique permet de caractériser finement les transferts et les pertes au sein de l'oscillateur. Le graphique ci-dessous met en évidence plusieurs phénomènes fondamentaux :

- **Quadrature de phase et transfert d'énergie :** On observe un "chassé-croisé" caractéristique entre l'énergie cinétique E_c et l'énergie potentielle élastique E_{pe}. À l'instant initial ($t = 0$), l'énergie est exclusivement sous forme élastique. Au cours de la période, ces deux formes d'énergie s'échangent mutuellement, l'une atteignant son maximum quand l'autre s'annule.
- **Dissipation de l'énergie mécanique :** L'énergie mécanique totale $E_m = E_c + E_{pe}$ n'est pas conservée. On observe une décroissance monotone de E_m au cours du temps, ce qui traduit la présence de forces de frottement.
- **Travail des forces non-conservatives :** Cette perte d'énergie correspond au travail dissipatif de la force de traînée. La morphologie de cette décroissance (ici liée à un modèle en $1/A$) confirme la nature du régime de frottement identifié précédemment.

Bien que cela nous éloigne de l'objectif initial, cela nous permet d'avoir une discution intéressante sur les échanges d'énergie en jeux dans cette manipulation

---

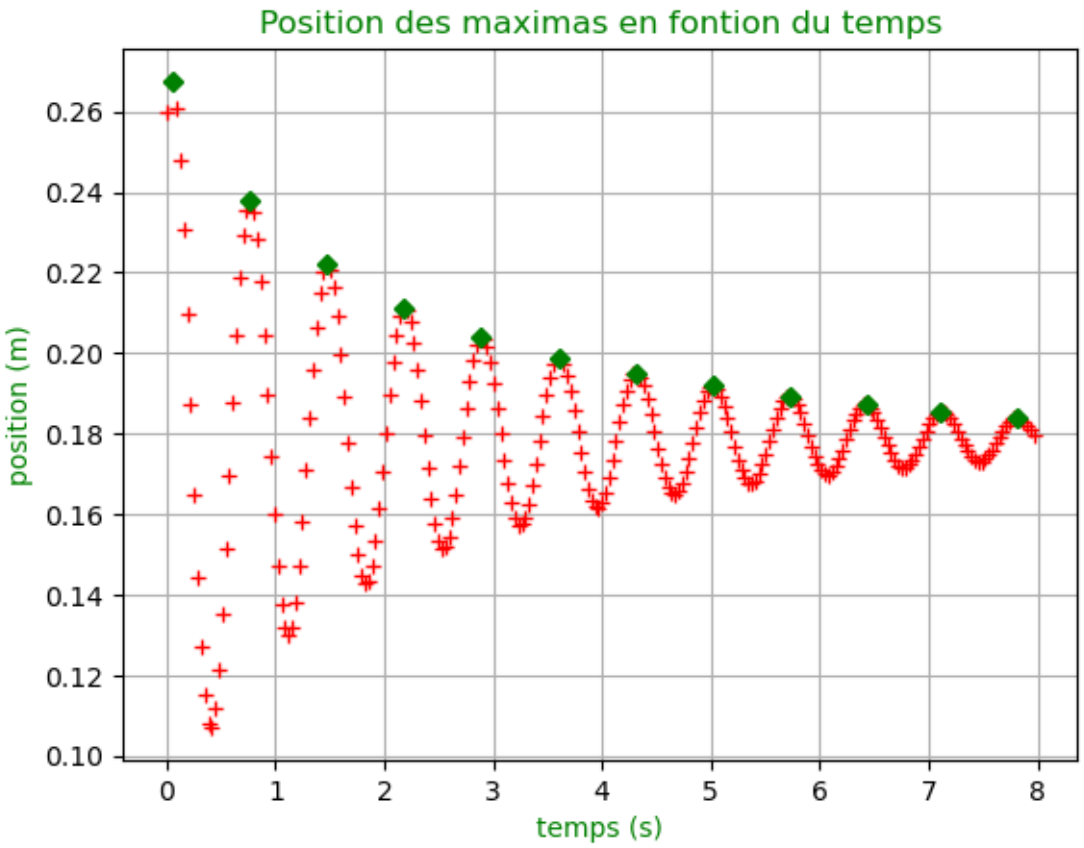
```
plt.figure(0, figsize=(10,4))

### Calcul des grandeurs énergétiques

Ec = 0.5*m*v_osci**2
Ek = 0.5*k*V**2
Em = Ec+Ek


plt.plot(t, Ec, 'r-', label="Ec")
plt.plot(t, Ek, 'y-', label="Ek")
plt.plot(t, Em, 'g-', label="Em")
plt.xlabel("temps (s)",color="green")
plt.ylabel("Energie (Joules)",color="green")
plt.title("Energie en fonction du temps", color="green")
plt.legend()
plt.grid()
```



Revenons en au traitement numérique pour trouver le coefficient aérodynamique Cx :

1. **Détection de pics :** Repérer les sommets (t_i, x_i) et corriger l'offset pour obtenir les amplitudes A_i.

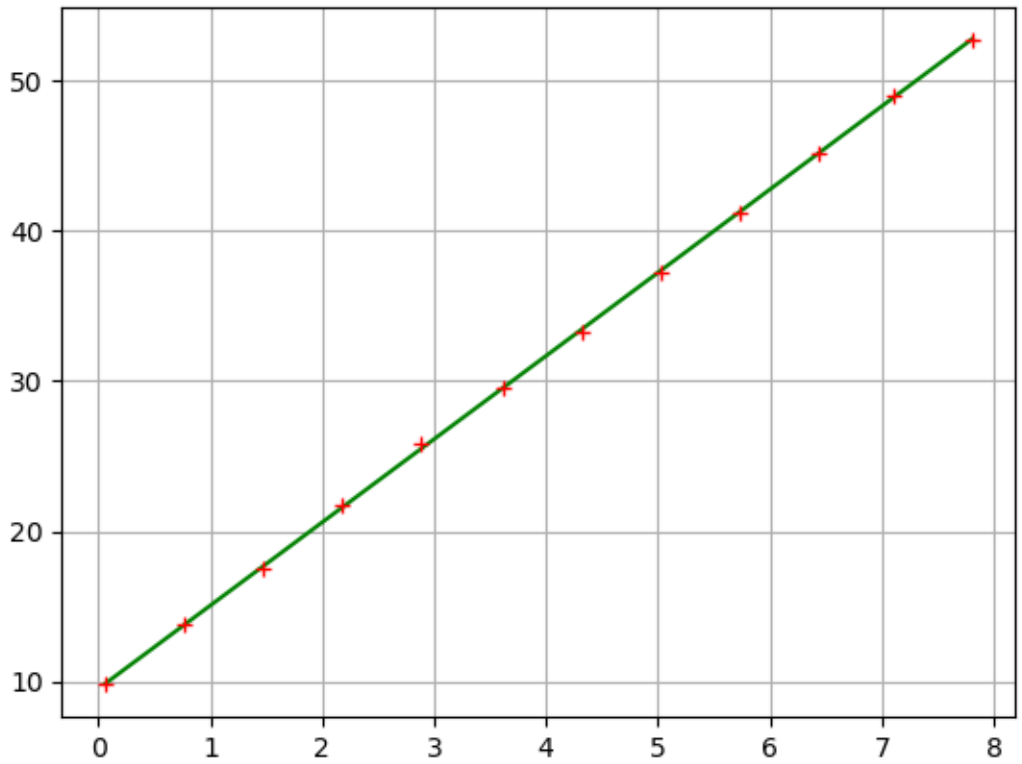
```
index_peak = find_peaks(pos, distance=8)    ### Fonction qui "trouve" les valeurs max
A = pos[index_peak[0]]-0.165                ### Ajuste l'offset
tA2 = t[index_peak[0]]

plt.plot(t, pos, 'r+')
plt.plot(tA2, A+0.165, 'gD', ms=5)
plt.ylabel("position (m)",color="green")
plt.xlabel("temps (s)", color="green")
plt.title("Position des maximas en fontion du temps", color="green")
plt.grid()
```



1. **Régression linéaire :** Effectuer le fit de $1/A_i$ en fonction du temps pour extraire la pente a_{op} et calculer la valeur du coefficient de frottement dynamique.

```
pop, pcov = curve_fit(f_lin, tA2, 1/A)
aop, bop = pop
yth = []
yth = aop*tA2+bop
plt.plot(tA2, yth, 'g-', label="fit")

delta_aop = np.sqrt(pcov.diagonal(0))
delta_aop = delta_aop[0]
err = delta_aop/aop
print("incertitude sur le fit :",np.round(err*100,2),"%")

plt.plot(tA2, 1/A, 'r+')
plt.grid()

R = 0.025/2
S = np.pi*R**2
Rho = 1000

Cx2 = (3*np.pi*m*np.round(aop,3))/(Rho*S*k)
print("Le coefficient de frottement aérodynamique est : Cx =",np.round(Cx2,3))
```

```
incertitude sur le fit : 0.29 %
Le coefficient de frottement aérodynamique est : Cx = 1.471
```



Conclusion – Étude du ressort fluide

L'étude de l'oscillateur amorti a permis de mettre en évidence un **régime pseudopériodique**, caractéristique d'un système soumis à des forces de frottement. L'analyse des maxima de l'oscillation montre que l'amplitude ne décroît pas exponentiellement mais suit une loi en $1/A$, ce qui valide l'hypothèse d'un **frottement quadratique**.

La linéarité de la représentation de $1/A$ en fonction du temps constitue un argument expérimental fort en faveur de ce modèle. Elle permet également d'accéder quantitativement au coefficient de traînée C_x, dont la valeur obtenue est cohérente avec l'ordre de grandeur attendu pour un objet se déplaçant dans un fluide.

Conclusion générale – Ensemble des deux expériences

Les deux expériences réalisées offrent une illustration complémentaire de concepts fondamentaux de la mécanique newtonienne.

- La première expérience met en évidence la **quasi-conservation du moment cinétique** dans un système soumis à une force centrale. Les écarts observés rappellent les limites expérimentales et l'influence des forces dissipatives réelles.
- La seconde expérience montre explicitement le rôle des **forces non conservatives**, en mettant en évidence la dissipation d'énergie et en caractérisant quantitativement un frottement fluide.

Dans leur ensemble, ces manipulations soulignent un point essentiel :

les lois de conservation sont des idéalisations puissantes, mais leur validité expérimentale dépend des conditions réelles du système étudié.

Elles illustrent également la démarche du physicien :

- modéliser un système,
- confronter théorie et expérience,
- quantifier les écarts,
- et en déduire des propriétés physiques mesurables.

Ces expériences constituent ainsi un excellent support pour traiter le cas de la **dynamique newtonienne**.

H. VERRAC, agrégé de physique