

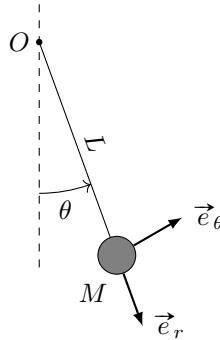
TP20 : Pendule simple

1 Objectif du TP

L'objectif de ce TP est de faire l'acquisition de l'accélération d'un pendule simple à l'aide de l'accéléromètre d'un téléphone portable et d'analyser les données obtenues.

Ne pas oublier qu'une mesure physique doit toujours être associée à une incertitude expérimentale. Penser à lire la notice des appareils pour connaître l'incertitude liée aux valeurs qu'ils fournissent.

2 Aspects théoriques



L'application des lois du mouvement (lois de Newton) au pendule simple permet d'obtenir l'équation différentielle du mouvement suivante :

$$\ddot{\theta} + \frac{g}{l} \sin \theta = 0$$

qui nécessite une résolution numérique. Cependant, lorsque $\theta \ll 1$ on peut faire l'approximation $\sin \theta \simeq \theta$ et on obtient l'équation d'un oscillateur harmonique

$$\ddot{\theta} + \omega_0^2 \theta = 0$$

Avec $\omega_0^2 = \frac{g}{l}$. La période $T_0 = \frac{2\pi}{\omega_0} = 2\pi\sqrt{\frac{l}{g}}$ des oscillations ne dépend alors pas de leur amplitude. Pour des oscillations d'amplitude supérieures (mais ne dépassant pas 60° , on peut montrer que la période est donnée par la formule de Borda :

$$T \simeq T_0 \left(1 + \frac{\theta_0^2}{16} \right) \quad (1)$$

où θ_0 est l'amplitude angulaire des oscillations.

Lorsqu'on ajoute une force de frottement fluide proportionnelle à la vitesse du pendule, on obtient l'équation d'un oscillateur harmonique amorti :

$$\ddot{\theta} + \frac{\omega_0}{Q} \dot{\theta} + \omega_0^2 \sin \theta = 0$$

qui, lorsque Q est élevé (c'est en général le cas) possède une solution pseudo-périodique :

$$\theta(t) = \theta_0 e^{-t/\tau} \sin(\omega t + \varphi)$$

où $\tau = 2Q/\omega_0$ est le temps caractéristique d'amortissement du pendule et φ la phase à l'origine.

Dans ce TP, nous allons mesurer l'accélération à l'aide d'un smartphone placé à la place de la masse. On s'arrangera pour qu'un des axes de l'accéléromètre du smartphone pointe dans la direction \vec{e}_r et l'autre dans la direction \vec{e}_θ .

Le mouvement du pendule étant circulaire, l'accélération suivant \vec{e}_r est :

$$a_r = -L\dot{\theta}^2 \quad (2)$$

On peut alors montrer que $a_r(t)$ est de la forme



$$a_r(t) = L\dot{\theta}^2 = A e^{-\frac{2t}{\tau}} \sin(\omega_0 t + \varphi')^2 \quad (3)$$

On peut aussi montrer que sur une période d'oscillation, l'accélération normale est maximale lorsque le point M se trouve à la verticale du point O et vaut

$$a_{rm} = 2g(1 - \cos(\theta_0)) \quad (4)$$


3 Manipulations

Pour faire les mesures d'accélération, vous utiliserez votre smartphone avec l'application **phyphox** que vous pouvez installer à partir de f-droid, Google Play ou App Store.

Dans l'application **phyphox**, choisir le capteur "Accélération (sans g)". Vous pouvez alors lancer l'acquisition avec l'icône . À la fin de l'expérience, arrêtez l'acquisition en appuyant sur .

On cherche ici à mesurer l'accélération d'un pendule simple en fonction du temps. Il faudra donc fabriquer un pendule simple dont la masse oscillante est un smartphone contenant un accéléromètre (tous les smartphones contiennent des accéléromètres!). Les axes de l'accéléromètre du téléphone sont alignés avec les axes du téléphone. On s'arrangera pour que l'un de ces axes soit dans la direction du fil du pendule.

Écarter le pendule de sa position d'équilibre avec un angle assez grand, et faire l'acquisition des oscillations jusqu'à ce qu'elles soient très faibles.

Une fois que l'acquisition que vous vouliez faire est terminée, il faut exporter les données vers un fichier pour pouvoir les traiter avec python. Pour cela, appuyer sur , puis "Exporter les mesures" et choisir "CSV (Tabulator, decimal point)". Enregistrer le fichier sur le téléphone et le transférer sur l'ordinateur, vous devriez avoir un fichier compressé au format zip duquel il faudra extraire le fichier csv (probablement appelé `Raw Data.csv`).

Une fois que vous avez le fichier `.csv` enregistré sur l'ordinateur, vous pouvez l'importer dans un tableau **numpy** pour commencer le traitement des données.

4 Traitement des données

Supposons que le fichier contenant les données d'accélération que vous avez enregistrées s'appelle `data.csv` et que votre script python se trouve dans le même dossier que ce fichier. Pour importer les données dans un tableau **numpy**, vous pouvez faire :

```
import numpy as np
data = np.loadtxt('data.csv', skiprows=1)
```

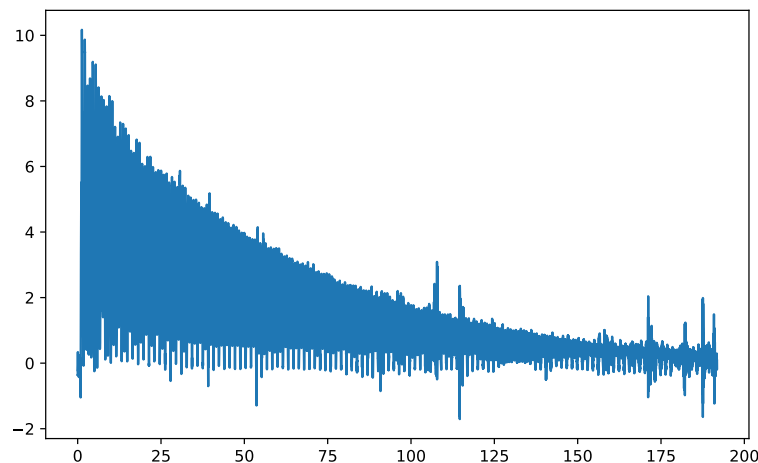
Il faudra ensuite déterminer à quel axe de l'accéléromètre correspond l'accélération normale du pendule. Pour cela on pourra tracer les accélérations des trois axes en fonction du temps et cherche celle qui oscille le plus. On peut faire :

```
import matplotlib.pyplot as plt
plt.plot(data[:,0], data[:,1]) # Axe x
plt.plot(data[:,0], data[:,2]) # Axe y
plt.plot(data[:,0], data[:,3]) # Axe z
```

Une fois l'axe de l'accélération normale déterminé, on pourra garder les données correspondantes dans les tableaux `t` et `ar` avec

```
t, ar = data[:,0], data[:,2]
```

Vous devriez obtenir quelque-chose qui ressemble à ça :

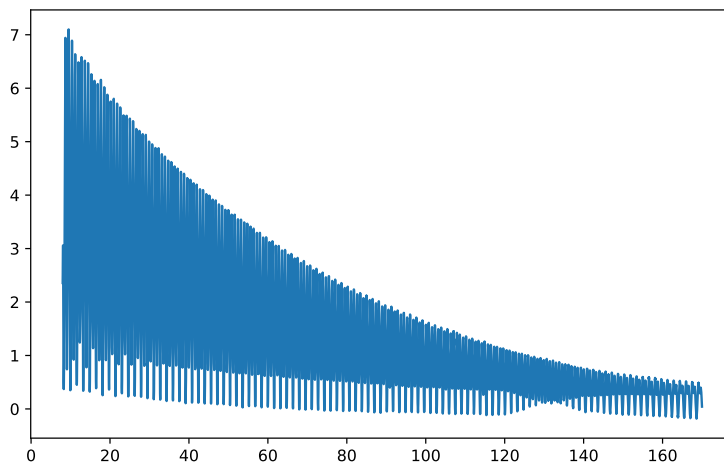


Filtre ensuite ces deux tableaux pour ne conserver que les données correspondant aux oscillations du pendule en faisant :

```
tmin = 2.2 # début des oscillations libres en s
tmax = 74 # fin des oscillations libres en s
tf = t[(t>tmin) & (t<tmax)]
arf = ar[(t>tmin) & (t<tmax)]

# On applique une moyenne glissante sur les données pour les lisser (filtre passe-bas)
n = 20 # nombre de points de la moyenne glissante
arf = np.convolve(arf,np.ones(n)/n,'same')
```

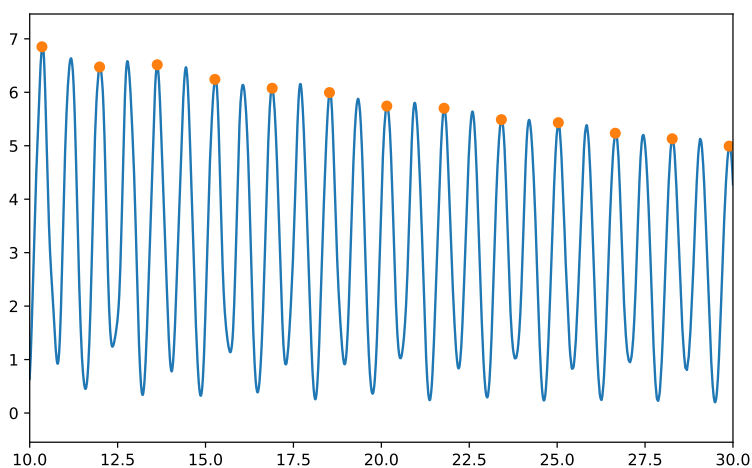
Pour obtenir ceci :



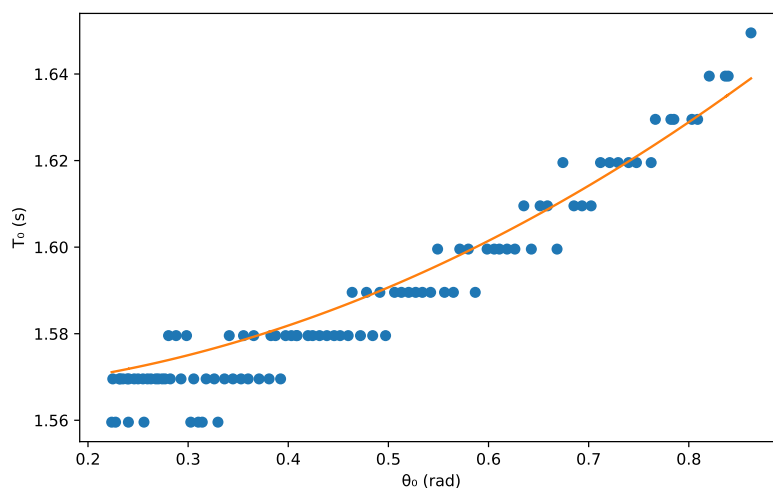
On peut maintenant écrire un programme qui détecte les maxima d'accélération et qui détermine l'évolution de la période au cours du temps

```
tdiff = tf[:-1]
diff = arf[1:]-arf[:-1] # liste des différences entre deux points successifs
diff = np.convolve(diff, np.ones(n)/n, 'same') # moyenne glissant pour lisser la courbe
i=0
lt = [] # liste des temps
la = [] # liste des accélérations
while i<len(diff)-1:
    if diff[i+1]<0 and diff[i]>0: # On cherche un changement de signe
        lt.append(tf[i])
        la.append(arf[i])
        i+=50 # On passe un certain nombre de points pour s'écarter du max
    i+=1
lt = np.array(lt[1:][::2]) # On oublie le premier et on garde un maximum sur 2
la = np.array(la[1:][::2])
```

On devrait pouvoir détecter les maxima et les accélérations correspondantes



On utilise enfin la relation entre l'accélération normale maximale et l'angle θ_0 pour tracer la période en fonction de θ_0 . (En orange, la période calculée à partir de la formule de Borda (1)).

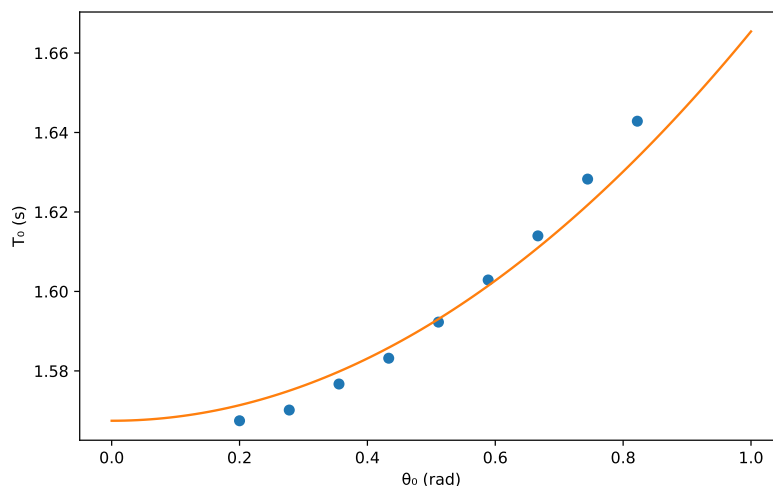


On peut faire une moyenne des points qui ont des valeurs de θ_0 et T_0 similaires,

```
theta0m = np.linspace(0.2, 0.9, 10)
numero = np.digitize(theta0, theta0m)
Tm = [T[numero==i].mean() for i in range(1,len(theta0m))]

plt.plot(theta0m[:-1], Tm, "o")
plt.plot(theta0, Tm[0]*(1+theta0**2/16))
```

On obtient finalement le graphique suivant :



On observe sur ce graphique que la loi de Borda ne semble pas décrire parfaitement les données. L'écart est probablement dû, au moins en partie, au fait que le filtrage que nous avons effectué sur l'accélération ait réduit son amplitude, ce qui nous a fait sous-estimer θ_0 et donc les points expérimentaux de ce graphique sont décalés vers la gauche. On pourrait essayer de corriger cet effet en comparant les données brutes aux données filtrées.

Avec ces données, on peut aussi, par exemple :

- Vérifier que l'amplitude des oscillations décroît exponentiellement avec le temps.
- Déterminer le facteur de qualité de l'oscillateur et sa pulsation propre. Comparer la pulsation propre de l'oscillateur à l'expression théorique. En fonction de l'orientation du téléphone durant l'expérience, on pourra éventuellement en déduire une estimation de l'endroit où se trouve l'accéléromètre dans le téléphone (deux mesures sont nécessaires pour localiser la position de l'accéléromètre).