# TP à la maison : Pendule avec votre smartphone

### But du TP :

Réaliser un pendule maison avec le smartphone et exploiter les données grâce à un programme Python.

### Matériel nécessaire à la construction du pendule et à la prise de mesure

- Votre smartphone
- Un ordinateur connecté au WiFi du smartphone.
- Une feuille de papier, du papier bulle (facultatif).
- Une agrapheuse, (du scotch), des ciseaux, une épingle ou un objet pointu
- de la ficelle
- une tringle à rideaux (installée et sans rideaux), ou un bout de table, ou un lustre, ou tout autre système permettant d'accrocher votre pendule !

## I Rappels théoriques

L'équation non-linéaire du pendule avec frottement fluide linéaire est de la forme suivante :

$$\ddot{\theta} + \frac{\omega_0}{Q}\dot{\theta} + \omega_0^2 \sin(\theta) = 0 \tag{1}$$

où $\theta$ est l'angle que fait le système avec la verticale, $\omega_0$ est la pulsation propre et $Q$ le facteur de qualité. Cette équation se linéarise en une équation d'oscillateur amorti aux petits angles ($\sin(\theta) \simeq \theta$) :

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

Le pendule considéré est très faiblement amorti, on peut donc considérer que la pseudo-période $T_{pp}$ des oscillations est égale à la période propre $T_0 = 2\pi\sqrt{\frac{\ell}{g}}$ pour un pendule simple de longueur $\ell$. $T_0$ ne dépend pas de l'amplitude des oscillations dans l'approximation des petits angles : il y a isochronisme.

Pour prendre en compte les non-linéarités, on peut faire un développement limité du sinus : $\sin(\theta) \simeq \theta - \frac{\theta^3}{6}$. Ce développement permet de trouver une période $T$ du pendule qui dépend de l'amplitude du mouvement. C'est la formule de Borda : $T = T_0\left(1 + \frac{\theta_m^2}{16}\right)$ où $\theta_m$ est l'amplitude. Dans ce TP, on va notamment montrer que même aux petits angles, le comportement non-linéaire du pendule apparaît.

## II Construction du pendule à deux fils.

Commençons par construire le dispositif expérimental.

Il faut d'abord créer l'enveloppe pour mettre le smartphone. Celle-ci doit être ajuster pour que le mouvement du smartphone soit confondu avec celui de l'enveloppe.

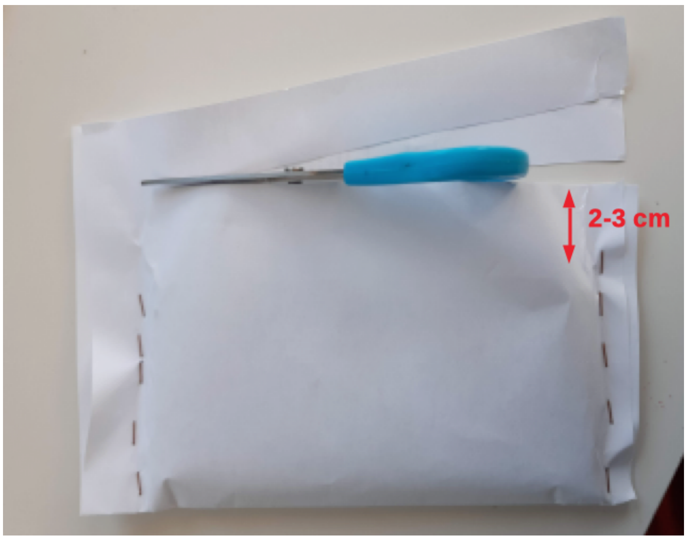
**Protection du téléphone (facultatif)**

Pour le protéger pendant cette expérience, vous pouvez faire une première enveloppe en papier bulle. Pour cela, découper puis agrapher un morceau de papier bulle aux dimensions du portable (figure haut de page suivante à gauche). Le retourner ensuite de manière à mettre les agraphes à l'intérieur afin de faciliter l'insertion dans l'enveloppe que l'on crée par la suite (figure haut de page suivante à droite).


## Création de l'enveloppe

On utilise une feuille de papier A4 (du brouillon) pour créer l'enveloppe qui accueillera le portable (figure ci-dessous à gauche). Pour ne pas que le portable puisse bouger, on agraffe les côtés au plus près de celui-ci (figure ci-dessous à droite).


On découpe ensuite de manière à ce que la feuille soit ajustée (laisser 2-3 cm de rab pour percer les trous). Percer deux trous aux extrémités (en étant le plus symétrique possible par rapport au portable. Vous pouvez commencer avec une aiguille ou une épingle à nourrice puis agrandir le trou avec les ciseaux.



**Finalisation du pendule**

Faites passer les deux bouts de ficelle dans les trous. Faire un nœud coulant qui permettra d'ajuster les longueurs de chaque fil. Accrocher le tout à une tringle ou un séchoir à linge ou tout autre système permettant un mouvement pendulaire.
Aider vous d'un niveau à bulle (ou d'un store comme sur la photo) pour vérifier que votre pendule est droit. Sinon ajuster l'un ou l'autre fil. Vous obtenez ainsi votre "pendule 2 fils fait maison"!!

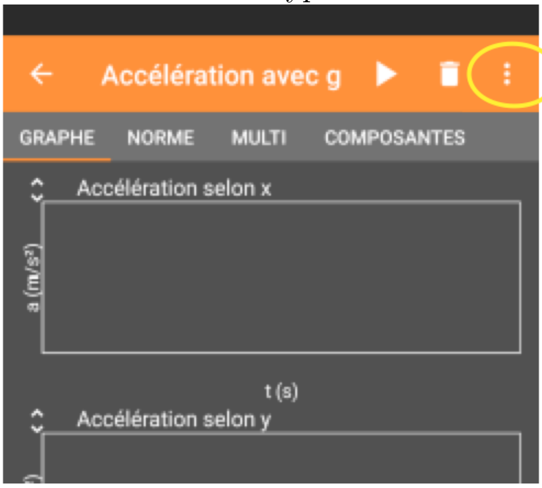
# III Utilisation de Phyphox

## III.1 Installation et Présentation

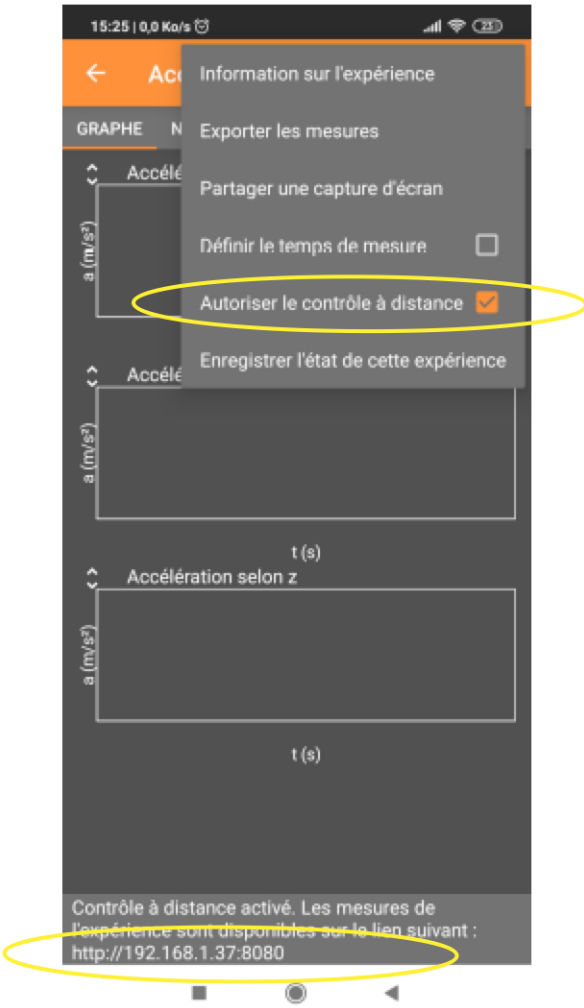
L'application que nous allons utiliser est l'application Phyphox, disponible gratuitement sur Playstore, ou Applestore.
Une fois téléchargée, vous pouvez ouvrir l'application. L'application vous présente tous les capteurs disponibles dans votre téléphone. Si certains capteurs sont grisés, cela signifie qu'ils ne sont pas présents dans votre téléphone. C'est souvent le capteur de pression qui peut manquer.
L'application permet également un contrôle à distance des acquisitions.

### III.2 Faire une acquisition à distance

Voici les différentes étapes pour prendre une acquisition à distance sur Phyphox :




- ➢ Configurer votre téléphone portable en "point d'accès wifi".
- ➢ Connecter votre ordinateur sur le portable. (Si vous ne l'avez jamais fait, il vous faut récupérer le mot de passe de votre point d'accès dans les paramètres de connexion de votre smartphone).
- ➢ Lancer l'application Phyphox et choisir le programme « Acceleration avec g ».
- ➢ Dans la bande orange, ouvrir le menu en haut à droite (3 points verticaux). (cf photo ci-contre).



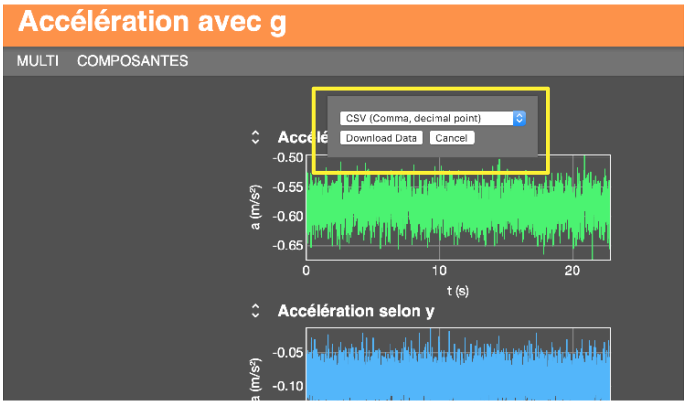

- ➢ Cocher "allow remote access" ou "autoriser le contrôle à distance" : une adresse s'affiche en bas de l'écran : http ://192.168.1.37 :8080 dans le cas ci-contre.
- ➢ Rentrer cette adresse dans le navigateur de votre ordinateur .
- ➢ Vous avez maintenant le même écran sur votre ordinateur que sur votre smartphone. Et vous pouvez le piloter à distance.
  Remarque importante : l'application Phyphox doit rester ouverte sur le smartphone pour que le contrôle à distance soit toujours possible. Si vous la fermez, le contrôle à distance est impossible. Par contre il suffit de revenir sur Phyphox et sur le capteur choisi pour reprendre le contrôle à distance.

### III.3 Acquisition et récupération des données

Une fois le contrôle à distance activé,

- ➢ Positionner le téléphone dans l'enveloppe.
- ➢ Décaler le pendule de sa position d'équilibre et le lâcher sans vitesse initiale. Un petit décalage (moins de 5 cm) est suffisant et permettra une étude aux petits angles à comparer avec la théorie. Veillez à ce que le pendule ait un mouvement dans le plan vertical et évitez les mouvements parasites dans le plan horizontal. Pour cela, il faut lancer le pendule dans le plan formé par les fils, la tringle et l'enveloppe au repos.

- ➢ Enfin, une fois votre pendule déjà en mouvement, lancez l'acquisition à distance grâce à votre ordinateur puis enregistrez les données dans votre ordinateur.
  Vous verrez que le temps d'amortissement est assez long et que l'on a beaucoup de pseudo-périodes. Vous devez prendre un temps d'acquisition assez long pour voir une décroissance exponentielle de l'amplitude de l'accélération.
- ➢ Il faut maintenant enregistrer les mesures. Toujours dans le menu en haut à droite (3 petits points verticaux), cliquer sur "Exporter les mesures" et choisir le format "CSV (Comma, decimal point)". Pour finir, cliquer sur "Download data".



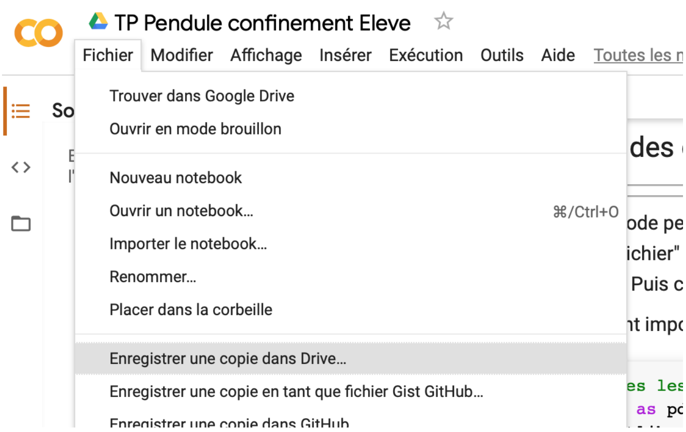

- ➢ Le fichier de données nommé "Raw Data.csv" se trouve dans votre dossier de téléchargement. Vous pouvez le renommer, pour vous souvenir de ce qu'il contient.

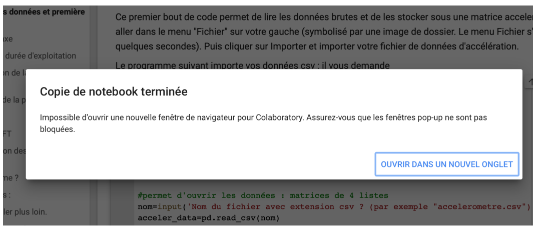
N'hésitez pas à faire et enregistrer plusieurs essais.

# IV Exploitation des mesures

## IV.1 Utilisation de Google Collaboratory

Pour exploiter les mesures, le programme Python est sur Google Collaboratory. Pour y accéder, vous devez aller sur le lien suivant (également envoyé par mail) :
`https://colab.research.google.com/drive/1f6o9u5kdTuO9yQYKdvqa9cXBkewEJkTZ?usp=sharing`
Afin que vous puissiez tous exploiter vos données dans de bonnes conditions, je vous demande de **faire une copie** de ce document dans votre drive.
Pour cela, aller dans "Fichier", puis "Enregistrer une copie dans Drive" et vous l'ouvrez dans un nouvel onglet.



puis


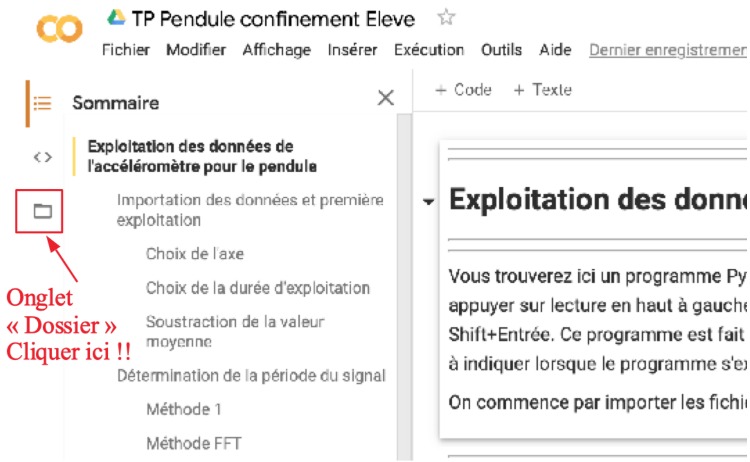

Ainsi vous travaillez sur la copie et non sur le fichier source. Pour les plus férus de Python, vous pourrez également modifier le code (cf fin du TP et uniquement dans votre fichier).

Une fois la copie du document collaboratif faite, vous pouvez commencer l'exploitation. Il faut suivre les étapes proposées et lancer les différents bouts de code à la suite. Le code est commenté pour vous aider à le comprendre.

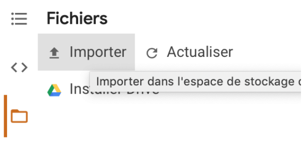
Une fois le fichier enregistrer dans le Drive, lorsque vous souhaitez le réouvrir en cliquant dessus, une page s'ouvre et il faut cliquer sur "Open with Google Collaboratory" en haut de la page.

## IV.2 Importation des données

Pour exploiter les données, il faut d'abord les importer. Pour cela, dans le menu de gauche, il vous faut aller dans l'onglet "Dossier".
Le menu du dossier peut mettre quelques secondes à s'ouvrir. Soyez patients.



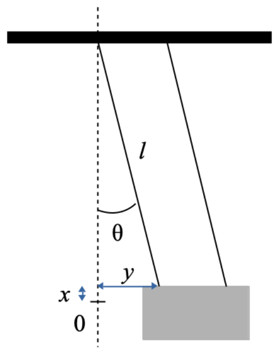

Cliquez ensuite sur "Importer" (ou sur l'icône avec une flèche vers le haut) et importez vos données.
Une fois que cette étape est faite, vous pouvez suivre les instructions du programme sur Python.




Les parties qui suivent proposent une explication plus complète de ce que fait le programme Python.

## IV.3 Premières exploitations

Dans la première partie du programme, on commence par trier les accélérations sur les 3 axes et ne garder que celle qui nous intéresse. Les capteurs accéléromètres ont les axes comme représentées sur la figure ci-dessous à gauche,


Pour de petits angles, et donc de petites amplitudes du mouvement, on peut considérer le mouvement comme horizontal. En effet d'après le schéma ci-dessus à droite, le mouvement sur $x$ est faible par rapport au mouvement sur l'axe $y$. Ainsi, le mouvement pendulaire, quasiment horizontal, est mesuré avec l'accélération sur l'axe $y$. Et c'est cette courbe que l'on va exploiter. On fait ainsi l'approximation de l'accélération sur $y$ :

$$\ddot{y} \simeq \ell\ddot{\theta} \simeq \theta_0 \ell \omega_0^2 \exp(-t/\tau)(\cos(\omega t + \Psi) \tag{3}$$

.
Ensuite, on vous demande de déterminer le temps initial et le temps final sur votre acquisition. Cela permet de s'affranchir d'éventuels débuts d'acquisition posant problème ou d'enlever la fin si vous avez une valeur nulle pendant un long moment. Concernant l'acquisition, pour pouvoir interpréter

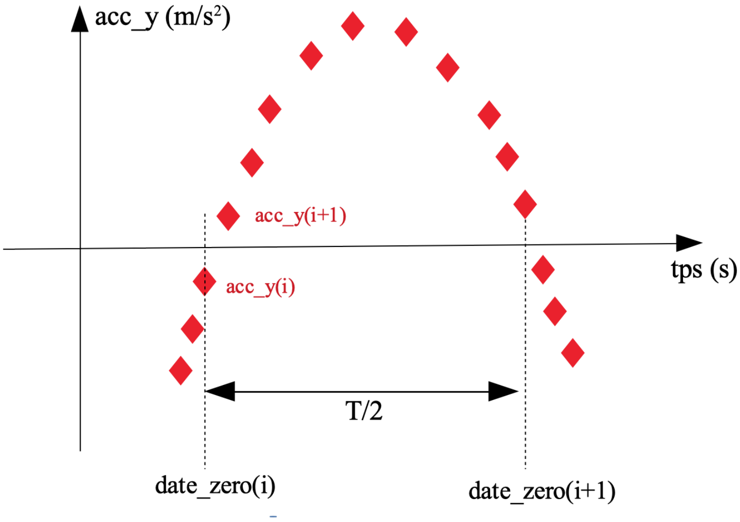
correctement la suite, il faut cependant voir une décroissance significative de l'amplitude.
La fin de cette première partie permet de fixer la valeur moyenne à une valeur quasi-nulle. Nous en aurons besoin dans le calcul de la période.

## IV.4 Calcul de la période

### IV.4.1 Méthode zéro

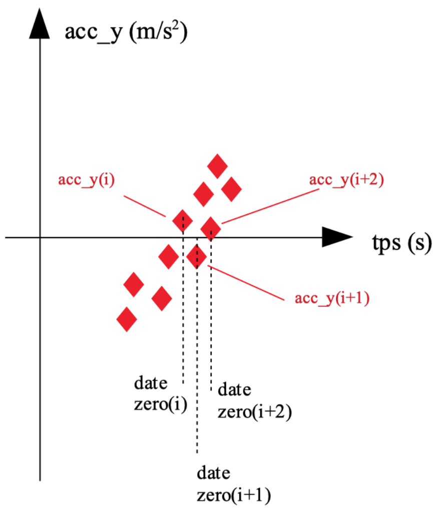
Comme pour Python, le numéro des méthodes commence à zéro et non à 1 !

Pour mesurer la période et surtout son évolution au fur et à mesure que l'amplitude diminue, nous sélectionnons le passage par zéro de l'accélération. En effet, l'accélération ayant maintenant une valeur moyenne nulle, les instants de passage par zéro sont séparés d'une demi-période, comme le montre le schéma ci-contre.
Dans Python, cela consiste à créer deux tableaux : un premier tableau donne le signe de l'accélération. L'autre donne également le signe de l'accélération mais est décalé d'une ligne par rapport au premier. On fait la somme de ces deux tableaux, ce qui revient à sommer les signes de deux accélérations successive. Si elles sont de même signe, on trouve $+2$ ou $-2$. Mais lorsque l'accélération change de signe entre un point et le suivant, le tableau somme renvoie zéro.
On récupère les instants de passage par zéro dans le tableau "date_zero". Il ne reste plus qu'à faire la différence entre deux points successifs de ce tableau pour en déduire la demi-période "demi_periode".




En faisant le calcul, on peut trouver une demi-période, qui prend des valeurs nulles par instant. (cf résultat du premier code de la partie 2.

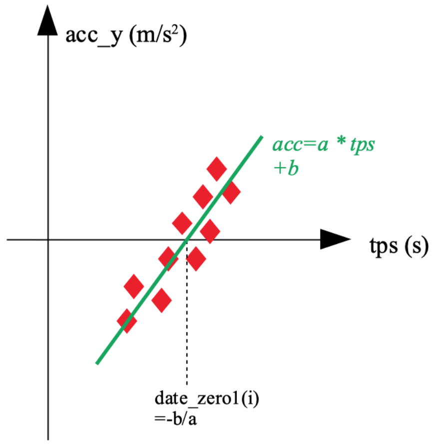
Cela s'explique par le bruit que peut avoir le signal. On peut ainsi avoir deux instants de passage par zéro très proches comme le montre le schéma ci-contre. La demi-période calculée est ainsi très proche de zéro et ne donne pas la valeur attendue.




Pour s'en affranchir, une méthode radicale, consiste à trier les valeurs de demi-période et à ne garder que les valeurs plus grande qu'une valeur limite. Pour cette valeur limite, on peut prendre un quart de période (soit une demi demi-période !).
Une amélioration de la méthode est un ajustement des points autour des instants de passage par zéro comme expliqué dans la suite.

### IV.4.2 Méthode 1 : régression linéaire des passages par zéro.

Une façon d'améliorer la méthode de mesure est donc de faire un ajustement afine $acc\_y = a \times t + b$ des points proches de la date zéro (10 avant et 10 après) et de déterminer l'instant où la régression linéaire croise l'axe des abscisses : $t_1 = -b/a$.
Sur Python, il faut donc faire les régressions linéaires de chaque passage par zéro, à l'aide d'une boucle for.
Cette méthode permet de diminuer le bruit pour la mesure de la période.



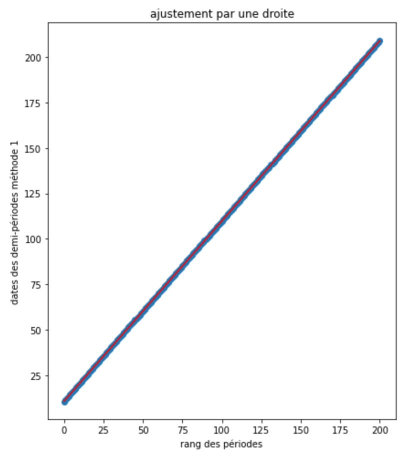

### IV.4.3 Méthode FFT

Cette méthode permet de déterminer la période moyenne, mais ne rend pas compte de la dépendance ou non de la période en fonction de l'amplitude. Ainsi elle permet de comparer les périodes moyennes déterminées avec les deux méthodes précédentes.

### IV.4.4 Isochronisme et résidus

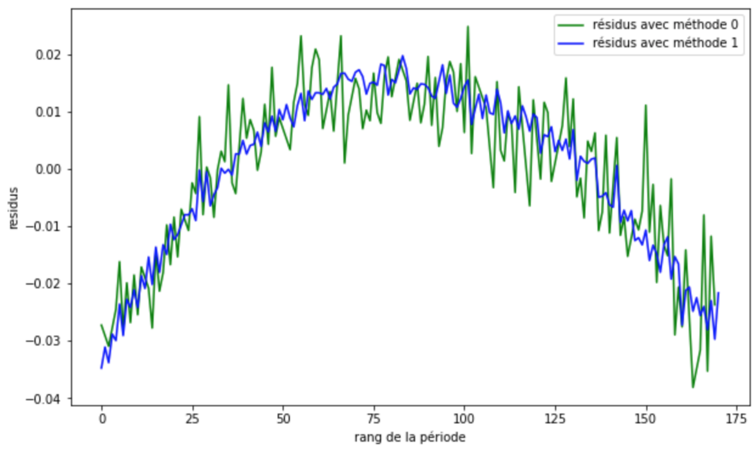
Aux petits angles, on peut considérer qu'il y a isochronisme des oscillations. Pour le vérifier sur Python, la demi-période doit rester constante et l'instant de passage par zéros en fonction du rang doit être une droite affine si on les trace en fonction de leur rang. En effet, les instants de passage par zéro sont espacés d'un intervalle constant si la demi-période est constante.

Pour s'en assurer, on peut faire une régression linéaire des dates de passage par zéro en fonction de leur rang. En traçant les points expérimentaux et la régression linéaire, celle-ci semble s'ajuster parfaitement aux points expérimentaux (cf courbe ci-contre issue de mes résultats).




Une façon de s'assurer que la modélisation s'ajuste bien sur les points expérimentaux est de tracer le graphe des résidus : on trace la différence entre les points expérimentaux et la régression linéaire. La régression linéaire est adaptée et correcte si le graphe des résidus ne présentent pas de structure particulière et si l'amplitude maximale des résidus est faible devant les amplitudes des points : on doit donc avoir des résidus qui sont du bruit blanc autour de zéro.

Cependant, si on observe la structure que l'on obtient pour les deux méthodes, on voit nettement une structure parabolique apparaître dans le graphe des résidus (ci-contre mes résultats). Cela s'explique par le fait que pour un pendule, la période dépend de l'amplitude, et qu'il y a donc non-isochronisme des oscillations. Cette structure n'apparaît que si vous avez un nombre de période suffisant et donc une décroissance significative de l'amplitude.



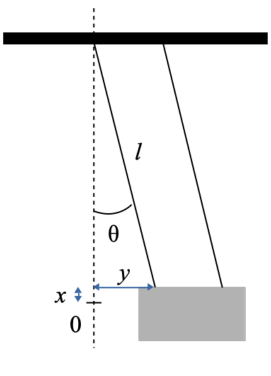
On voit que le résidu est en ordre de grandeur au maximum à 3 % de la période, ce qui reste faible sans être complètement négligeable.
De plus, la structure des résidus montre que la modélisation affine des dates de passage par zéros au fur et à mesure n'est pas satisfaisante : ainsi les dates de passages par zéro n'augmentent pas de manière régulière et donc la période du pendule n'est pas constante . L'approximation des petits angles est ainsi mise en défaut puisqu'on observe, grâce aux résidus, une dépendance de la période à l'amplitude. Les dates de passage par zéro, d'après la théorie proposée, doivent être de plus en plus proche au fur et à mesure que l'amplitude des oscillations diminue. La modélisation affine réalisée tend à minimiser l'écart avec les points expérimentaux : elle est donc centrée en zéro mais l'écart entre les points expérimentaux et la modélisation est un peu trop faible e pour les premiers et derniers rangs et un peu trop grande pour des rangs du milieu.

En conclusion, même avec une expérience faite maison, on peut mettre en avant la non-linéarité intrinsèque de l'équation du pendule même en se plaçant dans des conditions de bonne approximation !

## IV.5 Portrait de phase et étude énergétique

Pour reprendre les notions vues en cours, on peut demander à Python de tracer le portrait de phase du pendule. Pour cela, nous avons besoin d'intégrer l'accélération une première fois pour en déduire la vitesse et une deuxième fois pour en déduire la position. Cependant, les données sont en général assez bruitées. Pour la suite, nous allons donc travailler avec des modèles et nous intégrerons les modèles. Les modèles choisis sont des modèles de sinus amorti exponentiellement. La première modélisation se fait sur l'accélération. Il peut arriver qu'elle donne des résultats abberants (modèle qui ne s'ajuste pas sur les points expérimentaux) : dans ce cas, reprenez les valeurs "guess_" du début qui sont les paramètres initiaux : vous pouvez changer la valeur de la phase à l'origine avec celle trouvée après une première modélisation par exemple. Ce choix de paramètre initiaux permet au programme de moins se tromper. Précisons cependant que vu, les résultats précédents, ces modèles ne sont pas les plus adaptées puisqu'on a vu que les non-linéarités devaient être pris en compte. Une autre possibilité serait de filtrer préalablement le signal pour éliminer le bruit avant d'intégrer.
Cependant, ce premier modèle permet d'avoir un tracé d'un portrait de phase et de vérifier la décroissance de l'énergie mécanique.
La méthode d'intégration utilisée est la méthode des trapèzes (cf cours d'info) et la fonction Python qui permet d'intégrer l'accélération pour en déduire la vitesse en fonction du temps est "integrate.cumtrapz" de la bibliothèque "scipy". On en déduit après intégration et modélisation successive, la vitesse $v\_y$ et la position $y$ du pendule.
On retrouve ainsi le portrait de phase en spirale, caractéristique d'un oscillateur amorti.

Pour déterminer les énergies cinétiques et potentielles, il faut mesurer la masse $m$ du pendule (masse du téléphone + masse de l'enveloppe avec ou sans papier bulle) et la longueur $\ell$ du fil. La longueur du fil est récupérée ici à partir de la période calculée et de la valeur de $g$ ($g = 9,80428$ m.s$^{-2}$.
L'énergie cinétique est donnée par : $E_c = \dfrac{1}{2}mv_y^2$.
Pour l'énergie potentielle, pour un pendule simple, $E_p = mg\ell(1-\cos(\theta))$. Or ici, on a $\theta \simeq \dfrac{y}{\ell}$ dans l'approximation des petits angles (cf schéma ci-contre). De plus, dans cette approximation, le cosinus se développe en $\cos(\theta) = 1 - \dfrac{\theta^2}{2}$ (dl à l'ordre 2). Ainsi, l'énergie potentielle de pesanteur s'écrit :




$$E_p = mg\ell(1-\cos(\theta)) = mg\ell\frac{\theta^2}{2} = \frac{mgy^2}{2\ell} \quad (4)$$

L'énergie mécanique s'en déduit ensuite. Vous devez observer une énergie cinétique et potentielle en opposition de phase et une énergie mécanique qui décroit au cours du temps.

# V Travail à réaliser

## V.1 Travail expérimental

- Construire le pendule. (Faire une photo de votre montage maison !!)
- Récupérer les données et faire une première exploitation avec le programme Python.
- N'hésitez pas à refaire l'expérience sur une durée différente ou avec d'autres longueurs de fil !

## V.2 Travail théorique

- Retrouver l'équation du pendule simple avec frottements linéaire (1) et les résultats de la partie théorique (période,...)
- Dans l'approximation des petits angles, la solution $\theta(t)$ du pendule s'écrit : $\theta(t) = \theta_0 \exp(-t/\tau)\cos(\omega t+\phi)$. Expliciter $\tau$ et $\omega$ en fonction de $Q$ et de $T_0$.
- Justifier, d'après votre courbe, que le facteur de qualité $Q$ est très grand (de l'ordre de la centaine) et qu'on peut ainsi considérer que $\omega \simeq \omega_0$ et donc que la pseudo-période et la période propre sont égales. Que peut-on dire de $1/\tau$ par rapport à $\omega_0$ ?
- A partir des résultats précédents, déterminer des expressions approchées de $\dot{\theta}$ et $\ddot{\theta}$ puis retrouver le résultat (3) proposé à la partie IV.3.
- Retrouver l'expression de l'énergies potentielle (4).

## V.3 Exploitation et critique des résultats

A partir de l'exploitation des résultats réalisée :

- Préciser la période déterminée à l'aide des différentes méthodes. Comparer la période propre à celle attendue à partir de la mesure du fil. Commenter.
- Fournir le graphe des résidus. Expliquer en quoi le graphe des résidus permet de montrer le caractère non-linéaire du mouvement.
- Tracer et fournir le portrait de phase. Vérifier qu'il correspond à celui d'un oscillateur amorti.
- Pourquoi, d'après le programme, le logarithme de l'énergie mécanique est forcément une droite affine décroissante ? Pensez-vous que c'est réellement le cas ?
- Proposer des critiques constructives sur cette expérience : les approximations faites sont-elles toutes justifiées ? Quels faits sont passés sous silence et pourraient être pris en compte ? Comment feriez-vous pour les prendre en compte dans l'exploitation ou dans le montage ?

## V.4 Travail sur le programme Python (facultatif)

Pour ceux qui ont envie, voici quelques pistes d'amélioration ou d'ajout du programme Python :

- Pour éliminer le bruit, on peut faire un prétraitement avec un filtrage passe-bas de l'accélération.
- Améliorer les différentes étapes (créer des fonctions, etc...). Notamment, l'étape de la méthode 1.
- Pour la modélisation de l'accélération, prendre en compte le fait que la fréquence dépend de l'amplitude.
- Déterminer l'amplitude maximale à chaque période, prendre son logarithme et vérifier que la décroissance est exponentielle : le log de l'amplitude doit donc décroître linéairement.
- Tracer le portrait de phase en fonction de grandeurs adimensionnées : $q = \dfrac{y}{y_0}$ et $\dot{q} = \dfrac{\dot{y}}{\dot{y}_0}$ où $y_0$ et $\dot{y}_0$ sont respectivement les amplitudes de la position $y(t)$ et de la vitesse $\dot{y}(t)$ (reliées à l'angle $\theta(t)$ et à la vitesse angulaire $\dot{\theta}(t)$).
- Et toute autre amélioration qui vous viendrait à l'esprit !

Merci de me faire parvenir vos améliorations et suggestions.