Space pesée

ou

comment peser sans gravité

Olympiades de Physique 2023

Alexandre ADOSHVILI
Mathis GAUDILLAT
Simon GOULERET

Lycée Lalande
Bourg en Bresse
encadrés par JB BUTET

SOMMAIRE

INTRODUCTION

Pour des expériences de physique ou de chimie dans l'ISS, comme pour surveiller l'état de forme des astronautes, il est important de pouvoir peser dans l'espace, c'est-à-dire de déterminer la masse du système considéré.

Si sur Terre cela ne pose aucun problème particulier, il devient déjà plus difficile de peser correctement sur la Lune. Seule une balance de Roberval pourra nous donner une masse correcte en comparant à des masses étalons, toutes les autres balances “normales” ne nous donnant pas accès à la masse mais au poids, c'est-à-dire à la force liée à la gravité.

Comment arriver à peser dans l'espace quand on est en apesanteur, par exemple en orbite autour de la Terre alors que presque toutes les balances sont faites pour mesurer une force -le poids-?

Nous allons vous présenter ici deux manières de mesurer une masse sans gravité que nous avons mises en place, dont l'une d'elles qui s'est avérée être très concluante.

La théorie générale de la pesée dans l'espace :

Pour peser une masse sur Terre, rien de plus facile : prendre un objet déformable (un ressort, un quartz par exemple), faire en sorte que le poids de l'objet à peser déforme l'objet, mesurer l'ampleur de la déformation et ensuite en déduire la masse de l'objet par étalonnage ou via une relation.

Dans l'espace, en apesanteur, il est tout de suite plus difficile de mesurer une masse. Prenons l'exemple de l'ISS : la station spatiale internationale est en orbite autour de la terre, ce qui revient schématiquement à dire qu'elle est en chute libre perpétuelle. Prenons l'exemple d'un astronaute voulant mesurer la masse d'un objet quelconque. Il subirait la même accélération que la station et que l'objet qu'il tente de mesurer. De fait, dans le référentiel de l'astronaute, l'objet ne serait soumis à aucune force, car il serait immobile.

Alors comment mesurer une masse dans l'espace s'il est impossible de le mesurer grâce au poids ?

Pour notre astronaute, une approche possible serait de faire bouger l'objet pour qualifier la résistance que celui-ci applique : créer une gravité artificielle.

On ne mesure plus alors la masse “pesante” (liée à la gravité) mais la masse inertielle (la quantité de matière qui s'oppose à la variation de mouvement)

Cet exposé portera sur différentes possibilités envisageables pour mesurer la masse d'un système en utilisant son inertie, c'est-à-dire mesurer sa masse inertielle et non pas sa masse gravitationnelle.

I) Première expérience : la balance rotative

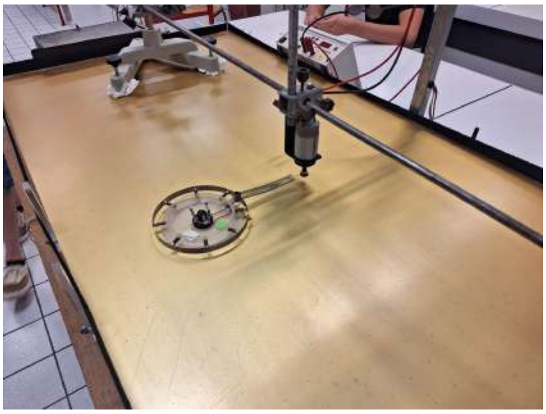
Comme le poids est une force verticale, pour s'en affranchir, nos expériences devaient se dérouler sur un plan horizontal.

Nous avons alors pensé à faire tourner le système et mesurer la "force centrifuge". Voici le dispositif que nous avons construit :

Il est composé d'une table à coussin d'air, d'un mobile relié à un moteur électrique par un ressort, le tout solidement agencé à l'aide de potences et diverses pinces. Le moteur électrique est alimenté par un générateur.

Il y a beaucoup d'oscillations parasites, il a fallu solidariser par des tiges la plupart des parties pour les éviter.

Grâce à ce mécanisme, la seule force agissant sur le mobile est donc la force de rappel du ressort, notée $\vec{T}$. La trajectoire du système étant plane, on peut utiliser un système de coordonnées à deux dimensions pour établir la position du mobile.



L'idée était de faire tourner différentes masses et de voir si la mesure de vitesse angulaire et l'allongement du ressort variaient et pouvait nous donner des informations sur la masse attachée.

a) Théorie de la balance rotative

Le référentiel du laboratoire étant considéré galiléen, on peut appliquer la seconde loi de Newton. Le poids et la réaction du support se compensant, il ne reste que la force liée au ressort on obtient :

$m \times \vec{a} = \vec{F}_{ressort} = \vec{T}$

Le mouvement étant circulaire, le plus simple est alors de se placer dans un repère polaire et non pas cartésien et de projeter.

la seconde loi de Newton devient alors :

$m \times \vec{a} = m \times \left(\left(\frac{d^2r}{dt^2} - r\omega^2\right)\vec{u}_r + \left(r\frac{d^2\theta}{dt^2}{}^2 + 2\frac{dr}{dt}\omega\right)\vec{u}_\theta\right) = -k(l - l_0)\,\vec{u}_r$

Le mouvement étant circulaire et uniforme (hypothèse de travail), les dérivées du rayon et d'ω sont nulles.

En simplifiant les termes nuls, on obtient : $m\vec{a} = -mr\omega^2\,\vec{u}_r$ et $\vec{F} = -k(l - l_0)\,\vec{u}_r$

En projetant selon le vecteur $\vec{u}_r$:

on obtient $mr\omega^2 = k(l - l_0) \Leftrightarrow m = \frac{k(l-l_0)}{r\omega^2}$

On obtient alors la masse avec le calcul suivant :

$$\boxed{m = \frac{k\times(l-l_0)}{r\times\omega^2}}$$

Pour mesurer la masse, Il nous faut donc trouver k, R, l et la vitesse angulaire ω

- trouver k

Cette mesure étant juste liée au ressort, elle n'aura à être faite qu'une fois. Nous avons attaché une masse au ressort et mesuré l'allongement. On sait que le poids d'un objet accroché au ressort s'exprime $P = k(l - l_0) \Leftrightarrow k = \frac{P}{l-l_0}$

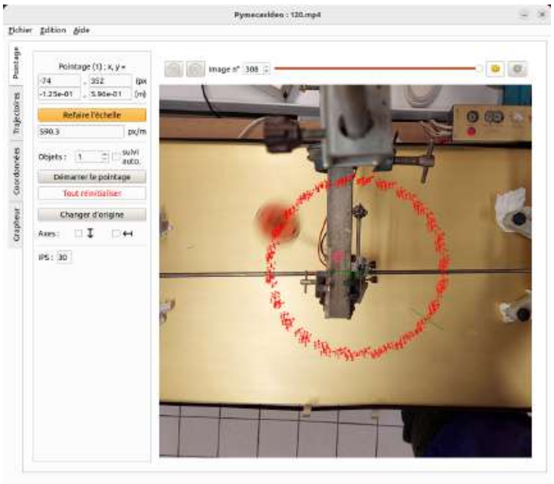
On trouve alors que :

$k = \frac{10}{19{,}5.10^{-2}-11{,}7.10^{-2}}$

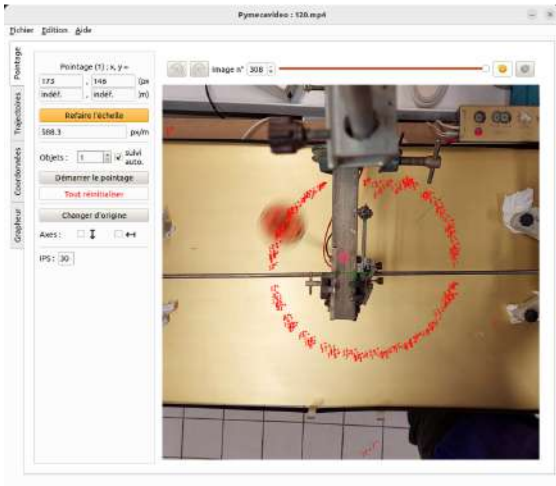
- mesurer ω et R.

Pour cela, nous avons installé un téléphone portable au-dessus du dispositif, et pris des vidéos. Nous nous sommes assurés que la position du téléphone restait la même, afin de réduire les incertitudes de mesures.

Ensuite, nous avons commencé à pointer les points individuellement à l'aide du logiciel pymecavideo. Cette opération fût très longue, car il y a plus de 300 points par vidéo.



Pointage manuel : plus grande précision globale



Pointage automatique : certains points sont placés en dehors de la trajectoire

Nous avons décidé de refaire des mesures en laissant le logiciel pointer automatiquement, mais cela faisait un peu de données erronées (à cause des barres situées dans le champ de la caméra) qu'il nous a fallu trier après coup. Finalement, nous avons gardé nos points faits à la main.

Le pointage nous a permis d'obtenir un fichier de données CSV pour chaque vidéo, dans lequel sont inscrits les coordonnées et le temps associé à chaque point. Notre première idée a été d'écrire un programme en Python afin de rendre les données exploitables, soit convertir les coordonnées cartésiennes des points en coordonnées polaires, afin de déterminer la vitesse angulaire moyenne du mobile dans chacune des vidéos.

Afin d'exploiter les données issues du pointage, nous avons fait usage des librairies `NumPy` (pour manipuler les données plus rapidement), `math` (pour les formules de trigonométries, les racines carrés, etc.), `os` (pour accéder à la liste des fichiers dans un dossier) et `matplotlib.pyplot` (pour réaliser des graphes).

Nous avons fait face à une première difficulté, trouver le centre de chacune des trajectoires circulaires. En effet, sans cette information, il est impossible d'exploiter les points.

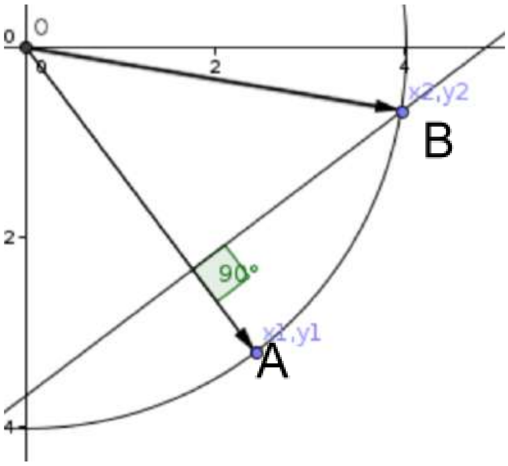
Nous avons donc implémenté dans le programme une méthode itérative pour trouver le centre du cercle, en déterminant pour chaque point le milieu du segment formé par le point et celui le plus éloigné (segment qui est en théorie un diamètre du cercle), puis de calculer la moyenne des points ainsi obtenus.

Ensuite, nous avons implémenté le calcul du rayon dans le programme, pour passer à la réelle utilisation de nos données : le calcul de la vitesse angulaire. Nous avons finalement laissé de côté la transformation des coordonnées en coordonnées polaires, afin d'utiliser la trigonométrie.

Pour trouver la vitesse angulaire et faire une mesure qui soit reproductible tout au long de la vidéo, nous avons dû avoir quelques idées et quelques fausses routes…

b) Mesurer la vitesse angulaire

- Notre première idée : **utiliser le produit scalaire.**



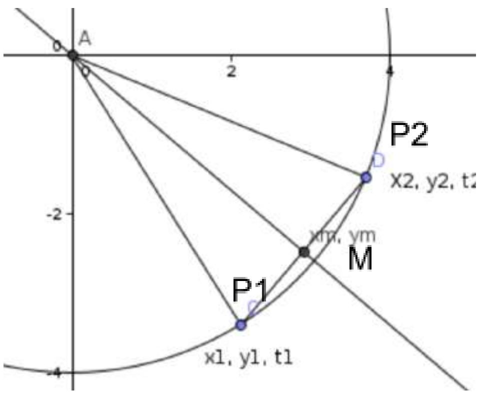
On trouve que le produit scalaire des deux vecteurs $\vec{OA} \cdot \vec{OB}$ est :

$\vec{OA}.\vec{OB} = R \times R \times cos(OA\ OB) \ = R \times R \times cos(\theta)$
$= (x_1 - x_0)(x_2 - x_0) + (y_1 - y_0)(y_2 - y_0)$

d'où on tire le code suivant :
$\theta = acos\left((x_1 - x_0) * (x_2 - x_0) + (y_1 - y_0) * (y_2 - y_0)) / R^2\right)$
il suffit alors de diviser par la durée t_1-t_0 pour obtenir la vitesse angulaire, ce qui donne en Python :

```python
# Méthode produit scalaire
def calcul_vitesse_angulaire(coordonnees, centre,rayon, methode=0, debug=0):
    echantillonnage = 5
    échantillons_points = coordonnees[::echantillonnage]
    vitesses_angulaires = np.array([])
    x0,y0 = centre
    for i in range(len(échantillons_points)-1):
        try:
            x1,y1,t1 = échantillons_points[i]
            x2,y2,t2 = échantillons_points[i+1]
            omega = acos(((x1-x0)*(x2-x0)+(y1-y0)*(y2-y0))/rayon**2)/(t1-t2)
            vitesses_angulaires = np.append(vitesses_angulaires, -omega)
        except  ValueError :
            pass
    vitesse_angulaire = np.mean(vitesses_angulaires)
    return vitesse_angulaire
```

- Notre deuxième idée: **faire encore un peu de géométrie.**



Soit l'image ci-contre une représentation théorique et simplifiée, avec une échelle arbitraire, d'une partie de notre expérience (avec A le centre du cercle formé par la trajectoire du mobile). On considère deux positions (appelées P_1 et P_2, et un point M milieu du segment formé par P_1 et P_2) du mobile. Afin de trouver l'angle θ entre ces deux points, on multiplie par 2 la moitié de l'angle entre P_1 et P_2, connu grâce à la trigonométrie. En effet, on sait que les triangles AMP_1 et AMP_2 sont rectangles en M, et on connaît le rayon (segment AM) et la distance P_1P_2, donc on peut utiliser la relation $sin(\theta) = \frac{opposé}{hypoténuse}$, soit dans notre cas $\theta = 2 \times asin\left(\frac{P_2M}{P_2A}\right)$. Ensuite, on divise θ par le temps Δt entre les mesures des deux points pour obtenir la vitesse angulaire ω.

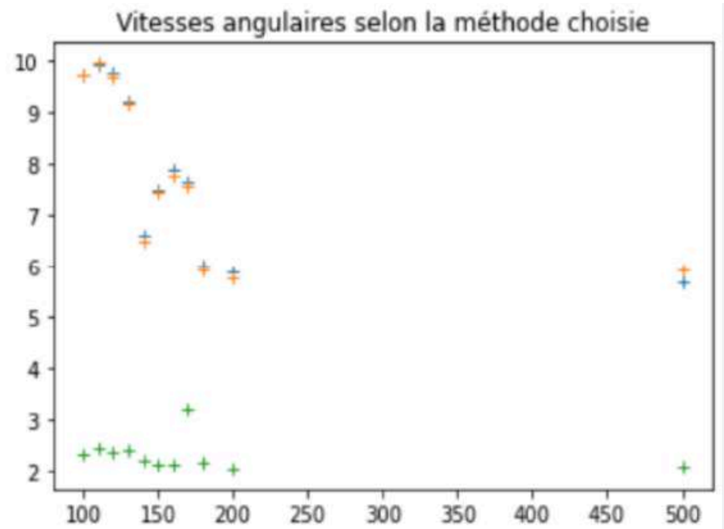
Dans le programme Python, nous utilisons les coordonnées des points pour calculer ω. Voici la formule utilisant les coordonnées des points :

$$\omega = \frac{2 \times asin\frac{\sqrt{(x_0-x_m)^2+(y_0-y_m)^2}}{r}}{t_2-t_1}$$

Le code Python correspondant est celui-ci :

```
# Méthode géométrique
def calcul_vitesse_angulaire(coordonnees, centre,rayon, methode=0, debug=0):
    echantillonnage = 5
        échantillons_points = coordonnees[::echantillonnage]
        vitesses_angulaires = np.array([])
        for i in range(len(échantillons_points)-1):
            try:
                x1,y1,t1 = échantillons_points[i]
                x2,y2,t2 = échantillons_points[i+1]
                AB = sqrt((x1-x2)**2+(y1-y2)**2)
                omega = 2*asin(AB/(2*rayon))/(t2-t1)
                vitesses_angulaires = np.append(vitesses_angulaires, omega)
            except  ValueError :
                pass
        vitesse_angulaire = np.mean(vitesses_angulaires
    return vitesse_angulaire
```

Ces deux méthodes se ressemblent beaucoup et donnent les mêmes résultats. Ils sont peu dépendants de l'échantillonnage (que l'on prenne tous les points ou 1 point sur 5 alors qu'un tour fait 20 points environ), les résultats sont donc très reproductibles.
Si l'on trace les deux résultats sur un même graphique les points se juxtaposent quasiment. (points bleus et oranges).



Les vitesses angulaires calculées par les deux méthodes précédentes sont les deux plus hautes (bleues et orange). (échantillonnage choisi : tous les 2 points)

Les valeurs trouvées sont cohérentes avec ce qui est mesuré directement sur la vidéo avec un chronomètre.

- Troisième idée : **calculer le temps mis par la masse pour faire un tour.**

Le pointage automatique nous fait du déchet en bas à gauche car il y a un tube qui cache le mobile. Nous avons donc décidé de filtrer les points, pour garder uniquement ceux qui sont situés en haut du cercle. (NB : sur les captures d'écran incluses, la vidéo a été retournée de 180° sur l'axe horizontal, donc les points gardés sont sur les images ceux situés en bas.)

Les données nécessitent nécessitent du filtrage. Nous avons implémenté une fonction dans notre programme Python qui sert à effectuer un nettoyage de la liste de points pour ne garder que les points "pas trop éloignés du rayon moyen".

La fonction "nettoyer()" fait un premier filtrage large de 5cm qui permet d'enlever les points trop éloignés et induisant donc une marge d'erreur, puis recalcule le centre, le rayon, et refait un filtrage plus fin à moins de 1 cm.

```python
# Nettoyer une liste de coordonnées
def nettoyer(coordonnees, centre, rayon) :
    # Premier nettoyage grossier
    coordonnees_filtree1= []
    for point in coordonnees :
        distance = sqrt((point[1]-centre[1])**2+(point[0]-centre[0])**2)
        if abs(rayon-distance)<0.05 :
            coordonnees_filtree1.append( point)
    centre = trouver_centre(coordonnees_filtree1)
    rayon = trouver_rayon(centre, coordonnees_filtree1)

    #2e nettoyage
    coordonnees_filtree2= []
    for point in coordonnees_filtree1 :
        distance = sqrt((point[1]-centre[1])**2+(point[0]-centre[0])**2)
        if abs(rayon-distance)<0.01 :
            coordonnees_filtree2.append( point)
    coordonnees_filtree =coordonnees_filtree2

    return coordonnees_filtree
```

On obtient alors un nouveau centre et un nouveau rayon, en appelant une nouvelle fois les fonctions suivantes :

```python
# Obtenir le rayon d'un cercle à partir de points
def trouver_rayon(centre, cercle):
    rayons = np.array([])
    for point in cercle:
        rayons = np.append(rayons, sqrt((point[1]-centre[1])**2+(point[0]-centre[0])**2))
    return np.mean(rayons)
```

```python
# Obtenir le centre d'un cercle à partir de points
def trouver_centre(points):
    distances = []
    centres = []
    for point in points:
        distance_max = [(float,float),(float,float),0]
        for point2 in points:
            d = [(point[0],point[1]),(point2[0],point2[1]),sqrt((point[1]-point2[1])**2+(point[0]-point2[0])**2)]
            if d[2] > distance_max[2]:
                distance_max = d
        distances.append(distance_max)
    for element in distances:
        centres.append(((element[0][0]+element[1][0])/2,(element[0][1]+element[1][1])/2))
    centres = np.array(centres)
    return np.mean(centres, axis=0)
```

Ensuite, nous sélectionnons les points qui sont proches du haut du cercle, donc qui remplissent ces deux conditions : abscisse proche de 0 et ordonnée proche du rayon.

```python
# Méthode des points hauts
def calcul_vitesse_angulaire(coordonnees, centre,rayon, methode=0, debug=0):
    echantillonnage = 5
    liste_points_hauts = []
    for x,y,t in coordonnees :
        tolerance = 0.04
        if -tolerance<x-centre[0]<tolerance and rayon-tolerance <y-centre[1]<rayon+tolerance :
            liste_points_hauts.append([x,y,t])
    liste_points_moyens = []
    t_moyen = None
    x_moyen = None
    y_moyen = None
    for i in range(len(liste_points_hauts)) :
        x,y,t = liste_points_hauts[i]
        if t_moyen is None :
            t_moyen = [t]
            x_moyen = [x]
            y_moyen = [y]
        else :
            if abs(t-t_moyen[0]) < 0.05: #points proches en temps à moyenner
                t_moyen.append(t)
                x_moyen.append(x)
                y_moyen.append(y)
            else : #points éloignés en temps
                #fais la moyenne et l'ajoute comme un point
                t_moyen = np.mean(np.array(t_moyen))
                x_moyen = np.mean(np.array(x_moyen))
                y_moyen = np.mean(np.array(y_moyen))
                liste_points_moyens.append([x_moyen, y_moyen, t_moyen])
                t_moyen = [t]
                x_moyen = [x]
                y_moyen = [y]
    T = 0
    for i in range(len( liste_points_moyens)-1) :
        T += liste_points_moyens[i+1][2]-liste_points_moyens[i][2]
    T = T/len(liste_points_moyens)
    vitesse_angulaire = 2*pi*rayon  / T

    return vitesse_angulaire
```

De ces points “hauts” nous regardons si ils sont du même tour ou s' il s’agit d’un tour après. Si c’est du même tour, on l’ajoute, sinon on fait la moyenne des points précédents et on recommence.

Ceci permet d’obtenir une liste qui comporte la moyenne des positions et du temps des points hauts.

Ensuite, il suffit de faire la moyenne des temps obtenus, ce qui donne une moyenne des durées pour faire un tour.

Vidéo d'exécution du script :
[lien vers l'exécution du script]

Cette méthode n’était pas fidèle, la vitesse angulaire mesurée ne variant pas ou peu (courbe verte) alors que si on mesure directement au chronomètre, on voit bien une modification de la vitesse angulaire.
Nous avons gardé les mesures des deux premières méthodes (quasi équivalentes) qui étaient corroborées par une mesure au chronomètre.

Évaluation des incertitudes sur la vitesse angulaire.

Nous utilisons la méthode des incertitudes de type A, ayant un grand nombre de points. On trouve alors les résultats suivants :

m	vitesse angulaire		
100	7.6	±	0.2
110	9.8	±	0.2
120	6.0	±	0.2
130	5.9	±	0.2
140	6.6	±	0.2
150	7.5	±	0.1
160	9.2	±	0.2
170	7.9	±	0.2
180	9.7	±	0.2
200	5.7	±	0.1
500	9.9	±	0.2

NB : les calculs ont été produits par le code suivant :

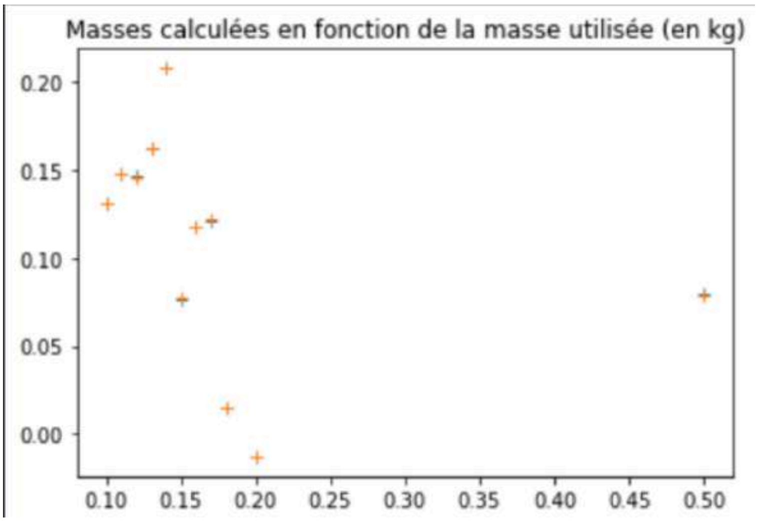
```
print(f"{vitesse_angulaire:.1f} {ceil(2*np.std(vitesses_angulaires)/sqrt(len(vitesses_angulaires))*10)/10:.1f} avec {len(vitesses_angulaires)}" )
```

On note une incertitude relative d’environ 20%... la méthode n’est pas très précise.

c) Les résultats obtenus

Nous avons donc fait des premières mesures pour la série suivante de masses :
100, 110, 120, 130, 140, 150, 160, 170, 180, 200 et 500 grammes.

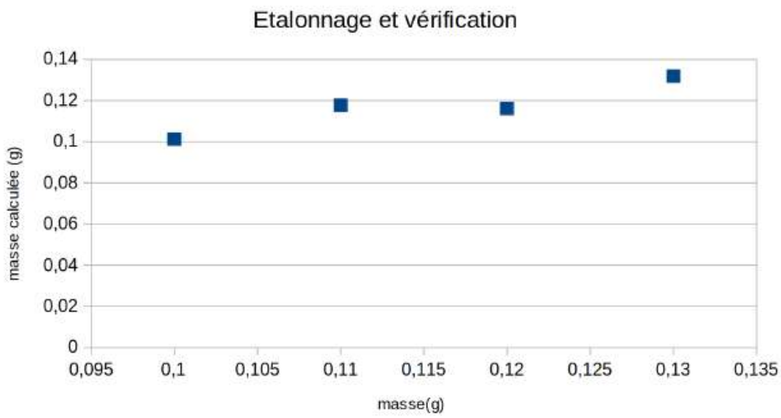
Nous avons tracé la masse obtenue en fonction de la masse à mesurer pour voir la relation entre elles.



En dehors du fait qu'il n'y a aucune corrélation, nous avions assez vite remarqué qu'au-delà d'une certaine masse, les frottements avec la table à coussin d'air n'étaient plus négligeables. Cela se voit assez bien à partir de 150g.

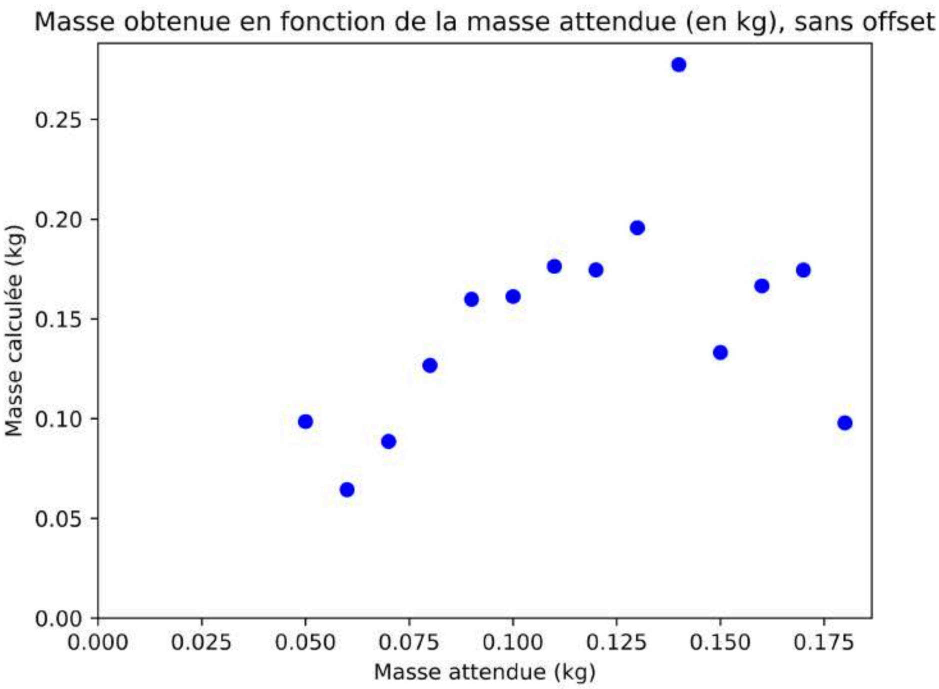
Pourtant, en y regardant de plus près, il semble y avoir une corrélation pour les masses faibles.

Ceci étant compris, nous avons décidé de nous focaliser sur l'intervalle 100-130g, considérant que le système est alors sans frottements.



On remarque alors ici une corrélation possible, car on a une correspondance quasi 1:1 entre les masses attendues et les masses calculées sur cet intervalle. Nous avons ajouté 39 grammes à chaque masse calculée afin d'avoir cette quasi-correspondance. Nous pensons que cette valeur correspond au poids du mobile. Il est important de noter qu'afin d'obtenir une correspondance quasi-exacte des masses calculées et attendues, nous avons rajouté une constante de 39 grammes aux masses calculées. Nous pensons que cette dernière correspond à la masse du mobile.

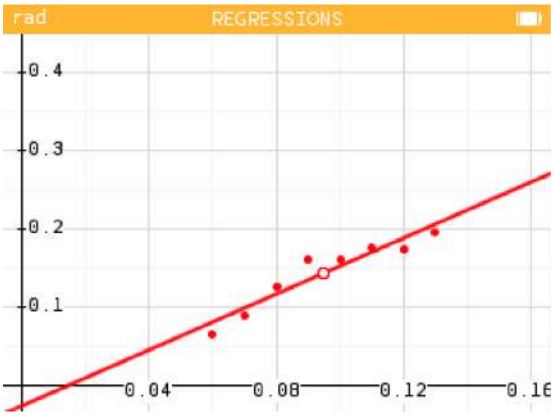
Ayant été sélectionnés pour la finale des Olympiades de Physique, nous avons eu le temps de refaire des mesures, cette fois-ci avec des masses étalon comprises entre 50 et 100 grammes. Il est utile de noter que le dispositif sur la table à coussin d'air a dû être remonté, car il avait été défait afin de permettre aux professeurs d'utiliser la table en cours.



Analyse : On observe 3 zones distinctes :

- le point “50g”, dont la valeur est très élevée par rapport à la masse attendue, nous avons supposé qu'il en était ainsi car nous avions dépassé la limite minimale de mesure de la balance,
- l'intervalle entre 60 et 130g paraît cohérent, en effet on observe une tendance croissante quand la masse attendue augmente,
- les points au-delà de 130g ne présentent aucune corrélation comme déjà observé précédemment.

Nous avons alors décidé d'effectuer une régression linéaire des données entre 60 et 130 grammes afin de déterminer la pente de la courbe et le coefficient de détermination.



Le calcul renvoie une pente de 0.849659, avec un coefficient R^2 de 0.897 environ.

Interprétation : la pente semble assez proche de 1, la pente théorique de la droite obtenue en traçant les masses calculées en fonction des masses attendues, mais le modèle n'est pas assez précis afin d'être utilisé pour faire des mesures.

d) Conclusions :

Avec nos limitations matérielles (moteur à courant continu qui ne tourne pas à vitesse constante quelque soit la masse, un seul ressort qui s'adaptait correctement aux contraintes de notre dispositif), nous avons découvert que notre idée amenait tout de même des résultats, certes entachés de beaucoup d'erreurs (20%) pour une plage de mesure restreinte.

Nous nous sommes rendus compte aussi qu'un phénomène d'oscillations venait perturber nos mesures faisant sortir du modèle "circulaire uniforme" et pouvant expliquer ces incertitudes. Nous nous sommes même demandés si nous pouvions utiliser ces oscillations pour mesurer la masse, mais cela nous avons manqué de temps.

Dommage car peser avec une perceuse, notre ressort et une caméra dans l'espace, nous aurait plu. Compliqué mais peut-être pas impossible !

Actuellement, c'est possible avec notre moteur et notre ressort, mais seulement des masses comprises entre 60 et 130g… et encore, avec une très grande incertitude.

Après avoir creusé cette idée, nous avons décidé de créer une autre balance inertielle, à partir d'une oscillation cette fois-ci.

II) La balance “oscillante” :

L’idée est de pouvoir accéder à une fréquence dépendante de la masse d’un objet mis au bout d’une tige flexible sur une seule dimension.

Au début de nos essais avec une lame de scie à métaux, nous avons très vite remarqué que plus la masse était élevée, plus la lame oscillait lentement.

Il nous fallait aussi une grandeur à mesurer et un capteur qui puisse transformer cette grandeur en tension.

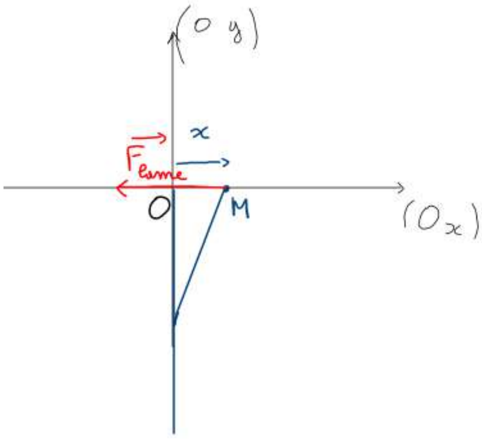
Avec un aimant placé au bout à proximité d’une bobine reliée à un fréquencemètre, nous avons pu prendre des mesures en exploitant le phénomène d’induction.

Nous avons d’abord tracé $m = f(f)$ sans voir de courbes faciles à analyser. Nous avons donc décidé d’étudier le phénomène :

a) La théorie

Nous allons faire quelques hypothèses pour simplifier l’analyse du système:

- nous négligeons tous frottements.
- nous négligeons le poids de la lame de scie
- nous négligeons la déformation de la lame dans toute les directions qui ne sont pas dans le sens de déformation (on ne garde que les déformations en x)
- on considère que la force de rappel de la lame est proportionnelle à la déformation en x



Dans le référentiel de laboratoire, considéré galiléen, on peut appliquer la seconde loi de Newton sur le point M de masse m.

$m\vec{a} = \vec{F}_{lame}$ que nous projetons sur l'axe des abscisses, soit :

$$m\ddot{x} = -kx$$

On obtient une équation différentielle du second degré. Comme nous avons une oscillation, la solution doit être sinusoïdale, donc on peut essayer une solution de la forme :
$x = X_m \times sin(\omega t + \varphi)$
ce qui donne :
$m \times (-\omega^2) \times X_m \times sin(\omega t + \varphi) = -k \times X_m \times sin(\omega t + \varphi)$
Ceci devant être valable tout le temps, quelque soit t, on peut diviser par $X_m \times sin(\omega t + \varphi)$
On obtient alors : $m \times \omega^2 = k$ c'est à dire $\omega^2 = \frac{k}{m}$.
Conclusion : la solution peut-être sinusoïdale si $\omega^2 = \frac{k}{m}$.
On sait que $\omega = 2\pi f$ donc finalement, on obtient :

$$\boxed{m = \frac{k}{4\pi^2} \cdot \frac{1}{f^2}}$$

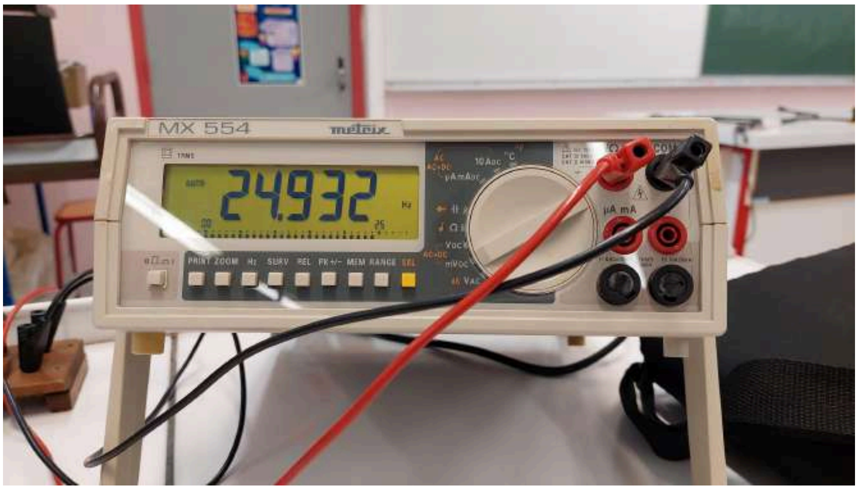
Pour obtenir la mesure de la masse, il suffit d'avoir accès à la fréquence d'oscillation du point M.

b) Mesure de la fréquence

Nous avons utilisé un fréquencemètre et vérifié à l'aide de Latis Pro et d'une transformée de Fourier du signal que les résultats des deux étaient fiables.

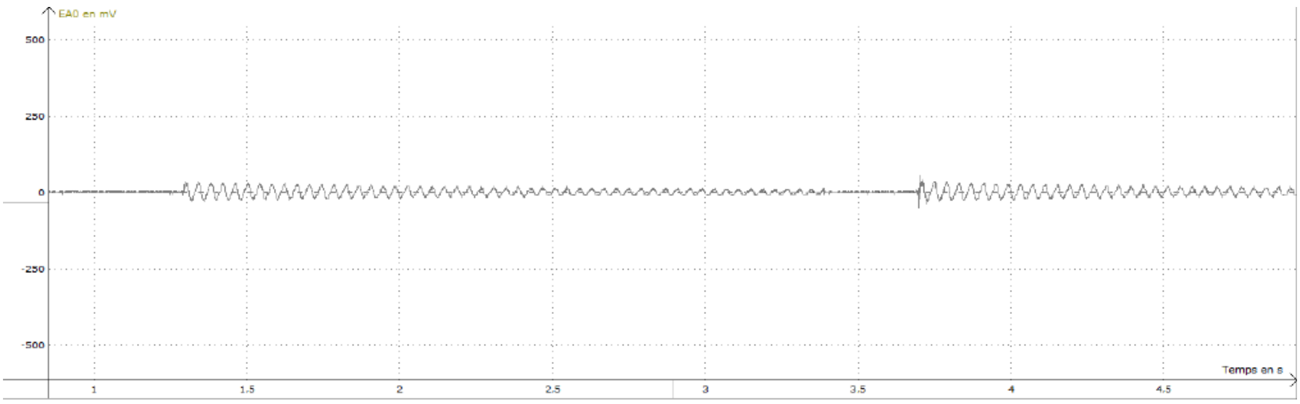
La méthode

Pour mesurer avec un facilité la fréquence d'oscillation, nous avons d'abord utilisé un fréquencemètre.



Mesure d'une fréquence avec le fréquencemètre (dans le cadre d'une expérience)

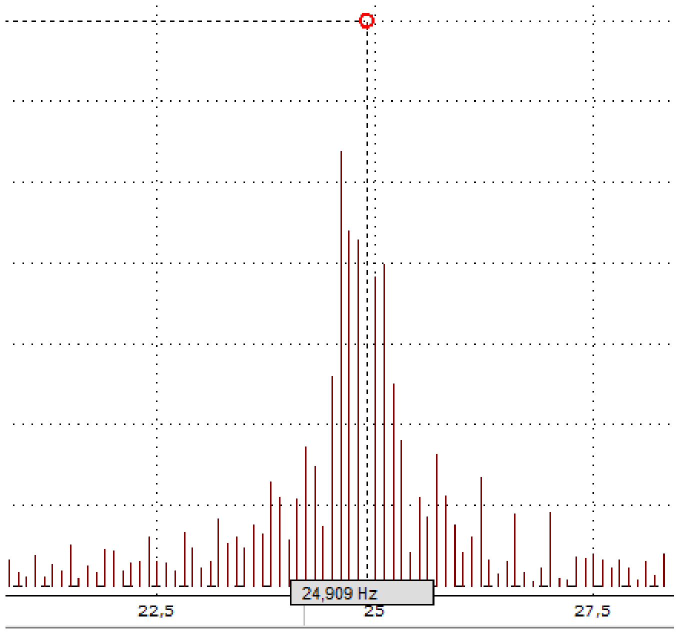
Il s'est avéré que les valeurs indiquées par le fréquencemètre étaient changeantes dans le temps, nous avons donc décidé d'enregistrer les mesures avec le dispositif d'acquisition $SYSAM - SP5^{®}$ pour analyser ensuite les résultats avec une transformée de Fourier avec le logiciel $Latis\ Pro^{®}$.



Signal enregistré (dans le cadre de la même expérience que la photo précédente)

On peut remarquer en faisant la transformée de Fourier que la fréquence ayant la plus grande amplitude est très proche de celle obtenue avec le fréquencemètre.

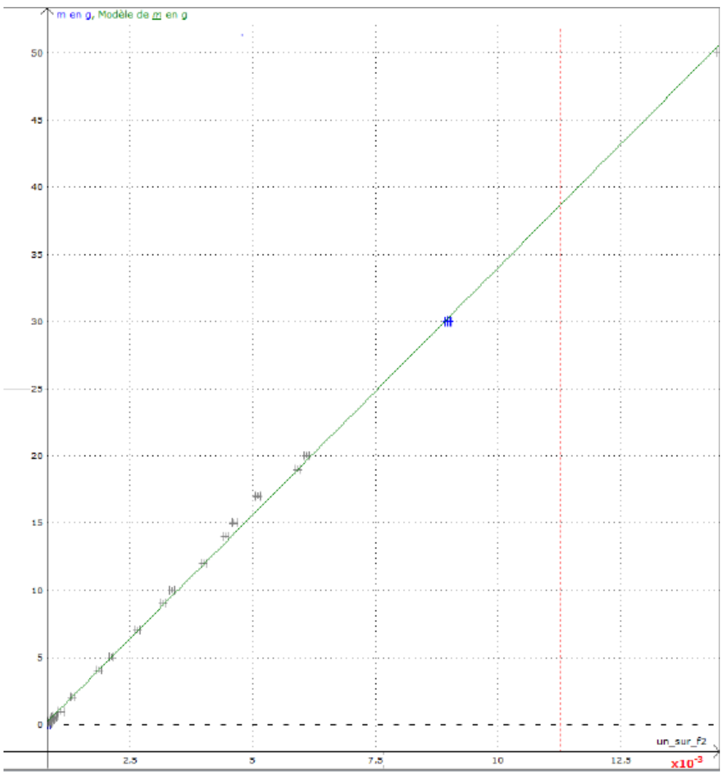
Nous pouvons en conclure que les valeurs données par le fréquencemètre sont fiables à 10^{-1} Hz près.



Transformée de Fourier du signal enregistré afin de vérifier la fiabilité du fréquencemètre

c) Les résultats

Nous avons donc tracé m en fonction de $\frac{1}{f^2}$ pour valider nos hypothèses :



La répartition selon une droite fonctionne extrêmement bien.

Analyse : cette méthode est très reproductible. Les points formant la droite sont très alignés. Il y a peu de dispersion. Le coefficient de corrélation est très proche de 1 : $R^2 = 0,999$.

Nous avons évalué l'incertitude relative sur le fréquence à environ 1 % et l'incertitude sur nos masses à 0,5%. (nous avons ajouté les masses certaines fois pour créer d'autres masses, par exemple : 8g = 5g + 2g + 1g)

Cependant, on obtient une fonction affine et non linéaire donnant la masse (en g) en fonction de $\frac{1}{f^2}$:

$$m = 3688 \times \frac{1}{f^2} - 2,83$$

Nous nous attendions à une droite passant par 0. Cependant, il est assez facile de trouver une signification physique à cette ordonnée à l'origine : il s'agit de la masse à vide de la lame de scie, qui prend en compte l'aimant, la patafix et aussi la masse de la partie de la lame qui oscille.

Nous avons donc mesuré à la balance de précision divers objets puis nous avons mesuré ces masses à l'aide de notre technique, à partir de la mesure de la fréquence.

	masse attendue (g)	fréquence (Hz)	masse (g)	Δm (g)	erreur relative (%)
pièce 5 centimes	3,895	23,7	3,733	0,2	4,2
grosse vis noire	23,087	12,45	20,96	2,1	9,2
gros boulon noir	13,892	14,52	14,66	-0,8	5,5
bout de bois	2,192	27,89	1,909	0,3	12,9
grande vis galva. bout noir	8,902	16,55	10,63	-1,7	19,4

Il n'y a pas de règles particulières pour l'erreur relative qui peut être grande ou pas. Cependant, les objets denses et qui sont bien en place au bout de la lame ont les meilleurs résultats. Les "vis" qui sont longues, posent quant à elles souci.

En tout cas, cette méthode est particulièrement intéressante et facile à mettre en œuvre. Par contre, il faudrait un "jeu" de lames pour pouvoir peser une large gamme d'objets.

CONCLUSION

Dans le but de déterminer une masse sans gravité, nous avons mis en place plusieurs protocoles, le premier faisant tourner un objet et mesurant la force centrifuge et le deuxième faisant osciller un mobile sur une lame.

Nous avons essayé pour la première méthode d'automatiser au possible l'étude des vidéos. Peser dans l'espace avec une perceuse et un ressort nous paraissait drôle. Cependant, après nous être cassé les dents plusieurs fois sur les mesures, ne trouvant aucun résultat probant, nous avons décidé de passer à la balance oscillante. Seul un examen un peu plus détaillé nous a fait prendre conscience que nous avions peut-être des résultats plus significatifs pour de faibles masses, ce qui s'est avéré réel.

Les résultats obtenus avec la balance oscillante sont de loin plus précis et plus fidèles, voire bluffants.

Au départ, pour notre projet, nous avions décidé de faire nos mesures en utilisant un matériel ayant des propriétés piézoélectriques comme les quartz. Nous avions mesuré la tension aux bornes de capteurs piézoélectriques en céramique en fonction de différentes fréquences pour déterminer lesquelles correspondaient à la résonance. Grâce aux résultats obtenus il serait possible de déterminer la masse d'objet très léger comme avec les microbalances à quartz.
Car même sur Terre, pour les objets très légers, on doit passer par une balance inertielle !

REMERCIEMENTS

Nous remercions chaleureusement M. BUTET pour son accompagnement et ses conseils tout au long de nos recherches, le personnel du laboratoire de physique du lycée Lalande pour leur soutien logistique, M. RABAN, professeur de physique en CPGE, pour la brillante idée d'utiliser l'induction avec un aimant et une bobine afin de mesurer les oscillations du réglet, M. GIRARD, également professeur de physique en CPGE, pour son aide au calcul des incertitudes de type A, et finalement l'administration du lycée Lalande pour son organisation impeccable.