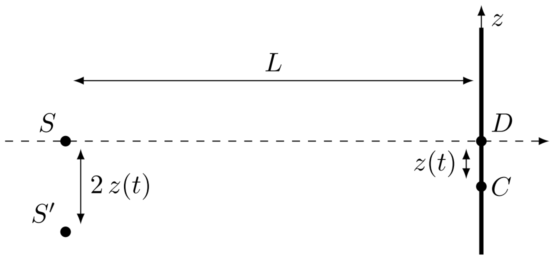
OP3-TD

Correction

OP3 – 06 Mesure de l'accélération de la pesanteur

1) Le miroir crée une source virtuelle S' image de S. Le dispositif est donc équivalent à celui ci-dessous :



La distance entre les deux sources est $2\,z(t)$, celle entre le point d'observation D et le centre de la figure C est $z(t)$, et la distance entre les sources et l'écran est L. Par analogie avec les trous d'Young, on donne alors directement la différence de marche

$$\boxed{\delta(t) = \frac{2\,z^2(t)}{L}} \tag{1}$$

en considérant $n = 1$ pour l'air.

2) Le miroir tombe, donc $z(t)$ augmente au fil du temps. En conséquence, la différence de marche $\delta(t)$ évolue également et au niveau du détecteur D fixe, on verra défiler des franges brillantes et sombres alternativement.

3) Décomposons l'étude en trois morceaux indépendants. **a) Déjà, étudions la chute du miroir**. Le miroir de masse m est soumis uniquement à son poids, c'est-à-dire qu'on néglige les forces de frottements. Dans le référentiel du laboratoire supposé galiléen, le théorème du centre de masse conduit à

$$m\,\vec{a} = m\,\vec{g}$$

Avec l'axe z vers le haut, la projection sur l'axe z et les deux intégrations successives conduisent à

$$z(t) = z(0) + v(0)\,t - \frac{1}{2}\,g\,t^2$$

Le miroir étant laché à $t = 0$ en $z = 0$ sans vitesse initiale, on a

$$z(0) = 0 \qquad \text{et} \qquad v(0) = 0 \qquad \text{soit} \qquad \boxed{z(t) = -\frac{1}{2}\,g\,t^2} \tag{2}$$

b) Ensuite, étudions les interférences en D. On obtient un maximum d'intensité, c'est-à-dire une interférence constructive, lorsque l'ordre d'interférence p est entier, soit

$$p = \frac{\delta}{\lambda} = m \qquad \text{avec} \qquad m \in \mathbb{Z}$$

En utilisant l'expression de la différence de marche (1) de concert avec celle (2) de $z(t)$, on aboutit à

$$\frac{g^2\,t_m^4}{2\,L\,\lambda} = m \qquad \text{soit} \qquad \boxed{t_m^4 = \frac{2\,L\,\lambda}{g^2}\,m} \tag{3}$$

Les temps t_m sont les temps pour lesquels on observe un maximum d'éclairement en D.

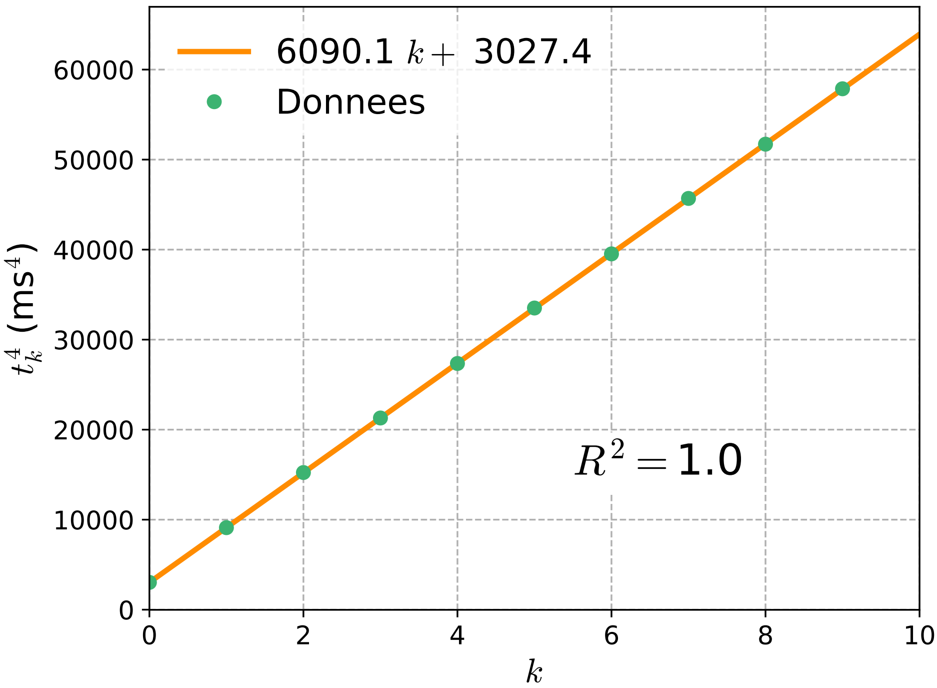
c) Enfin, discutons l'exploitation des données expérimentales. Il y a deux méthodes possibles.

- **1 ▶** par régression linéaire (avantage : exploite tous les points donc plus précise, inconvénient : il nous faut un logiciel pour ça) ;
- **2 ▶** en exploitant seulement deux mesures (moins précise mais plus rapide).

1 ► Pour exploiter toutes les valeurs de l'énoncé, on peut mener une régression linéaire de t_k^4 en fonction de k. Ce tracé conduit effectivemment à une droite d'après l'équation (3). Attention cependant, rien ne nous dit que le temps $k = 0$ est le premier maximum observé, et ce n'est en fait certainement pas le cas vu les temps correspondants. Il faut ici comprendre que les indices k de l'énoncé ne sont pas les indices m de notre étude analytique. Il y a un décalage entier $\alpha \in \mathbb{Z}$ (inconnu mais dont la connaissance est inutile) entre les deux

$$k = m + \alpha$$

Du fait de ce décalage, il faut bien penser à modéliser les données par une loi **affine** et non pas par une loi linéaire... Cette étude par python conduit à la courbe suivante



La pente $p = 6090,1$ ms^4 obtenue par la régression affine (le coefficient de régression est excellent) donne

$$\frac{2\,L\,\lambda}{g^2} = p \qquad \text{soit} \qquad \boxed{g = \sqrt{\frac{2\,L\,\lambda}{p}} = 9,8\ \mathrm{m\cdot s^{-2}}}$$

qui est en accord avec la valeur connue de l'accélération de la pesanteur.

2 ► Une autre manière de procéder, moins précise mais qui est **plus facile à mettre en œuvre le jour d'un oral**, est d'exploiter seulement deux points. En considérant les deux points extrémaux, le $k = 0$ et le $k = 9$, la différence d'indice est 9 *(que l'on parle en terme de k ou de m, c'est tout l'intérêt d'utiliser une différence et non pas une seule mesure !)* donc

$$t_9^4 - t_0^4 = \frac{18\,L\,\lambda}{g^2}$$

Alors directement

$$\boxed{g = \sqrt{\frac{18\,L\,\lambda}{t_9^4 - t_0^4}} = 9,8\ \mathrm{m\cdot s^{-2}}}$$

qui est tout à fait en accord avec la valeur connue de l'accélération de la pesanteur.

Remarque. Notez que tout le problème ici est que le temps $k = 0$ du tableau de l'énoncé n'est pas le premier maximum rencontré lorsqu'on laisse tomber le miroir (qui correspond à $m = 0$). Si c'était le cas, une seule des mesures suffirait. Pour s'affranchir ici du décalage d'indice, on utilise une différence de valeurs.

Remarque. On donne ci-dessous le code python qui a permis d'obtenir la courbe présentée

```
import numpy as np
from scipy import stats as scps
from matplotlib import pyplot as plt
plt.rc('mathtext', fontset="cm")

k = np.array(range(0,10))
Tk = np.array([7.42, 9.77, 11.11, 12.08, 12.86, 13.53, 14.10, 14.62, 15.08, 15.51])

y = Tk**4

slope, intercept, r_value, p_value, std_err = scps.linregress(k, y)
t = np.linspace(0.,10.,100)

plt.xlim([0.0,10.0])
plt.grid(linestyle='--')
plt.xlabel(r"$k$", fontsize=16)
plt.ylabel(r"$t_k^4$ (ms$^4$)", fontsize=16)

plt.plot(t, slope*t + intercept, color='darkorange', linewidth=2.5,
              label= str(round(slope,1)) + r" $k +$ " + str(round(intercept,1)))
plt.plot(k, y, color='mediumseagreen', linestyle='None', marker='o', markersize=6.,
              label=r"Donnees")

plt.text(5.5,15000.,r"$R^2 =$" + str(round(r_value**2,3)), fontsize=20,
              backgroundcolor='white')
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
legend = plt.legend(loc = 2, numpoints = 1, handlelength=2., frameon=1, fontsize=16)
frame = legend.get_frame()
frame.set_facecolor('white')
frame.set_edgecolor('white')
plt.tight_layout()
plt.savefig("reglin.pdf")
plt.show()
```