

# TP n°11 : Viscosimétrie

## Comment mesurer la viscosité d'un fluide ?

L'objectif de ce TP est de déterminer la valeur de la viscosité du glycérol à l'aide du mouvement de chute d'une bille en son sein.

↪ Cliquez ou flashez le QR code ci-contre pour visualiser les effets de la viscosité sur un mouvement de chute !



### Matériel à disposition

Bille en fer, éprouvette graduée, glycérol, pied à coulisse, caméra vidéo (votre téléphone).

### Méthodes mises en oeuvre

- ▷ Réaliser et exploiter l'enregistrement vidéo d'un mouvement.
- ▷ s'approprier un script Python afin de modéliser un mouvement.
- ▷ Confronter les résultats expérimentaux aux expressions théoriques.

## Étude théorique du mouvement

On considère la chute d'une bille de rayon  $R$ , de masse volumique  $\rho_b$  dans un fluide très visqueux, de masse volumique  $\rho_g$ , appelé glycérol. Lors de sa chute, la bille est soumise à son poids  $\vec{P}$ , sa poussée d'Archimède  $\vec{\Pi}_A$  ainsi qu'à la force de frottement fluide  $\vec{f}$  liée à la viscosité :

$$\vec{f} = -6\pi\eta R \vec{v}$$

où  $\vec{v}$  désigne la vitesse de la bille, et  $\eta$  la viscosité du fluide, en Poiseuille (1 Pl = 1 Pa.s).

✎ Exprimer le poids ainsi que la poussée d'Archimède exercés sur la bille, en fonction notamment de  $\rho_b$ ,  $\rho_g$  et du rayon  $R$ .

✎ Appliquer le principe fondamental de la dynamique à la bille lors de sa chute. Montrer que cette équation différentielle en projection selon l'axe  $(Oz)$  s'écrit

$$\frac{dv}{dt} + \frac{1}{\tau}v = \frac{v_{\text{lim}}}{\tau} \quad (1)$$

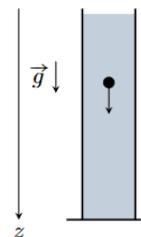
où  $\tau = \frac{2R^2\rho_b}{9\eta}$  et  $v_{\text{lim}} = \frac{2R^2(\rho_b - \rho_g)g}{9\eta}$ .

✎ Résoudre cette équation différentielle en prenant la vitesse initiale nulle.

✎ Intégrer cette équation pour obtenir l'expression de l'équation horaire  $z(t)$ .



À partir de l'équation horaire  $z(t)$  et du script Python donné en page 4, proposer un protocole expérimental permettant de déterminer la viscosité du fluide  $\eta$ .



## I - Enregistrement vidéo et exploitation

### ● Mesures sur la bille



Mesurer le diamètre de la bille à l'aide du  **pied à coulisse**  (cf document 1) et mesurer sa masse.  
En déduire sa masse volumique  $\rho_b$ .

### ● Enregistrement vidéo et exploitation sous Tracker



Enregistrer le mouvement de la chute libre d'une bille à l'aide de votre téléphone portable (la position initiale de la bille et l'échelle doivent être visibles).

🚫🚫 **Attention** : la bille doit être lâchée sans vitesse initiale dans le glycérol, il ne doit pas y avoir d'espace entre le glycérol et la bille à l'instant initial!



Analyser la vidéo avec le logiciel *Tracker*. Un document de prise en main du logiciel est disponible sur cahier de prépa, dans le dossier *TP*. Ajouter un système d'axes, un bâton de calibration **en veillant à utiliser des virgules et non des points pour les décimales**) et suivre la bille comme une "Masse ponctuelle" (onglet "Trajectoire" puis "Nouveau") de masse  $m$  à partir de l'instant où la bille plonge dans le glycérol.



Dans la fenêtre "Données" cliquer sur "Tableau de données" et sélectionner seulement  $y$ . Vérifier que le tableau de données contient seulement les valeurs de  $t$  et  $y$  et cliquer sur  $t(s)$  pour ordonner les résultats par ordre croissant des instants  $t$ . Dans l'onglet "Fichier" cliquer sur "Exporter" puis "Fichier de données", sélectionner toutes les cellules, pleine précision et onglet, puis cliquer sur "Enregistrer sous", aller dans le dossier "TP Mécanique" et nommer ce fichier selon le numéro de votre groupe (par exemple : "billegroupeXX").

### ● Exploitation sous Python



Dans la partie modifiable du script Python, changer la valeur de la variable "nomdefichier" afin qu'elle corresponde au nom du fichier de votre groupe "billegroupeXX".



Dans la partie modifiable du script Python, compléter la fonction ajustement afin qu'elle corresponde à la fonction  $y(t)$  trouvée plus tôt (les variables "t", "vlim" et "tau" correspondent respectivement aux grandeur  $t$ ,  $v_{\text{lim}}$  et  $\tau$ ; on utilisera la fonction exponentielle de  $x$  "np.exp(x)").



Exécuter le script pour obtenir les valeurs de  $v_{\text{lim}}$  et  $\tau$  ainsi que leur incertitude associée.

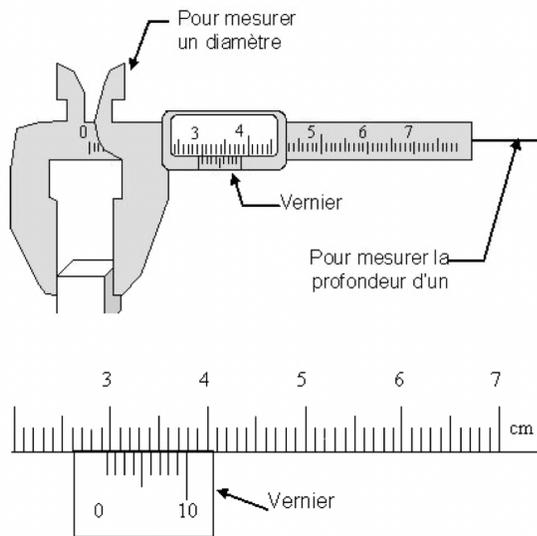
## II - Détermination de la viscosité

📎 Déterminer la valeur de  $\eta$  ainsi que son incertitude associée à partir de la valeur de  $v_{\text{lim}}$  et son incertitude associée. Discuter de la compatibilité des résultats. La valeur attendue est d'environ 1 Pl.

📎 Parmi les corrections discutées dans le document 2, lesquelles doivent être prises en compte pour exploiter l'expérience réalisée et lesquelles s'avèrent négligeables?

### III - Documents

#### Document 1 : Mesurer une longueur au pied à coulisse



Un pied à coulisse est un appareil permettant de mesurer une longueur (épaisseur, diamètre ou profondeur) avec une précision de 0,1 mm. Une telle précision est atteinte grâce à un vernier, dispositif inventé par le mathématicien Pierre Vernier au XVI<sup>e</sup> siècle.

L'échelle principale sur le pied à coulisse est graduée en millimètres tandis qu'un centimètre sur le vernier est divisé en dix intervalles de 0,9 mm. Pour lire l'ouverture des mâchoires du pied à coulisse, on lit la position indiquée sur la graduation par la première ligne du vernier. La décimale suivante est donnée par la division du vernier qui coïncide avec une des graduations du pied à coulisse.

Sur l'exemple ci-contre, le zéro du vernier (premier trait vertical) correspond à la graduation 2,9 de l'échelle principale. La décimale suivante est la ligne du vernier qui coïncide avec une ligne de la graduation : ici c'est la huitième. La longueur mesurée est alors de 2,98 cm.

*Adapté du site du Cégep de Sainte-Foy (Québec)*

#### Document 2 : force de Stokes

L'expression usuelle de la force de frottement fluide exercée par le glycérol sur la bille est appelée *force de Stokes*, et s'exprime :

$$\vec{f} = -3\pi\eta d \vec{v}$$

où  $d$  désigne le diamètre de la bille. Or cette formule n'est en toute rigueur valable que dans un fluide "infini". Dès que le fluide est limité par les parois latérales ou la base d'un récipient, celles-ci modifient l'écoulement par rapport à la situation où la bille serait seule. De façon plus mathématique, cela se traduit par des conditions aux limites supplémentaires. Ces effets entraînent les corrections suivantes, qui peuvent être calculées indépendamment les unes des autres par des développements perturbatifs :

▷ effet des parois latérales : une bille de diamètre  $d$  évoluant dans un cylindre de diamètre  $D$  subit la force (toutes les autres corrections étant négligées) :

$$\vec{f} = -3\pi\eta d \left(1 + 2,104 \frac{d}{D}\right) \vec{v}$$

▷ effet du fond : une bille de diamètre  $d$  évoluant à une altitude  $z$  du fond du récipient subit la force (toutes les corrections étant négligées) :

$$\vec{f} = -3\pi\eta d \left(1 + 2 \frac{d}{z}\right) \vec{v}$$

## ● Script Python

```

1
2 # --- Bibliothèques ---
3 from IPython.display import display, Latex
4 import matplotlib.pyplot as plt
5 import numpy as np
6 from scipy.optimize import curve_fit
7
8 # --- A modifier ---
9 m = # masse de la bille
10 g = # accélération de pesanteur
11 d = # diamètre de la bille
12 diam = # diamètre de l'éprouvette
13 plt.close('all')
14 def fonction_d_ajustement(t,vlim,tau,v0): #Fonction d'ajustement théorique, cf. I du TP
15
16 # --- Ouverture du fichier Tracker (non modifiable) ---
17 pointage = np.loadtxt("",skiprows=2,delimiter='\t')
18 t = pointage[:,0]
19 y = pointage[:,1]
20 t = t - t[0]
21 fig, ax1 = plt.subplots(1, 1)
22 ax1.set_title(r'Évolution temporelle de la position de la bille $y(t)$')
23 ax1.set_xlabel('$t$ (s)')
24 ax1.set_ylabel(r'$y$ (m)')
25
26 # --- Ajustement des données expérimentales ---
27 popt, pcov = curve_fit(fonction_d_ajustement, t, y)#p0=[y[-1]/t[-1],t[-1]*0.1,-0.02])
28
29 # --- Valeurs des coefficients de l'ajustement et leur incertitude associée ---
30 v_lim_ajuste = popt[0]
31 tau_ajuste = popt[1]
32 Delta_v_lim_ajuste = np.sqrt(np.diag(pcov))[0]
33 Delta_tau_ajuste = np.sqrt(np.diag(pcov))[1]
34 vth = v_lim_ajuste*(1 - np.exp(-t/tau_ajuste))
35
36 # --- Partie graphique ---
37 t_ajustement = np.linspace(t[0],t[-1],1000) #Abscisses de la fonction d'ajustement
38 y_ajustement = fonction_d_ajustement(t_ajustement,*popt)
39
40 ax1.plot(t,y,'xb',label=r'$y(t)$ mesuré')
41 #ax1.plot(t,vth,'xb',label=r'$\ell v(t)\ell$')
42 ax1.plot(t_ajustement,y_ajustement,'r',label=r'$y(t)$ ajusté')
43 ax1.legend()
44 ax1.grid(True, which='major', color='gray', alpha=0.6, linestyle='dashdot', lw=1.5)
45 ax1.minorticks_on()
46 ax1.grid(True, which='minor', color='beige', alpha=0.8, ls='-', lw=1)
47 eta = m*g/(3*np.pi*d*(1 + 2.104*(d/diam))*v_lim_ajuste)

```