

Homogénéité et incertitudes

1. Calculs

$$1. a^2 - b^2 = (a + b)(a - b)$$

$$2. (a + b)^2 = a^2 + b^2 + 2ab \text{ et non } a^2 + b^2$$

$$3. 4 \text{ divisé par } 1/2 \text{ vaut } 8 \text{ et non } 2$$

$$4. 10^6 \text{ divisé par } 2 \text{ vaut } 5 \cdot 10^5$$

$$5. 10^{-10} \text{ divisé par } 10^{-5} \text{ donne } 10^{-5} \text{ et non } 10^{-15}$$

$$6. \sqrt{16ab} = 4\sqrt{ab} \text{ et non } 4ab$$

$$7. \frac{b}{a} + \frac{c}{d} = \frac{bd + ac}{ad} \text{ et non } \frac{b+c}{a+d} \text{ (contre-exemple)}$$

possible si $a = b = c = d = 1$)

$$8. \frac{1}{a+b} \text{ ne vaut pas } \frac{1}{a} + \frac{1}{b} = \frac{b+a}{ab}$$

$$9. \frac{b-a}{ab} = \frac{1}{a} - \frac{1}{b}$$

$$10. \ln(ab) = \ln(a) + \ln(b) \text{ et non } \ln(a) \times \ln(b)$$

$$11. \exp(a-b) = \frac{e^a}{e^b}$$

$$12. \cos(a+b) = \cos a \cos b - \sin a \sin b \text{ et non } \cos a + \cos b$$

2. Quiz

Faux : Par exemple si on évalue l'évolution d'un temps caractéristique τ au cours du temps t , la dérivée de τ par rapport à t sera sans dimension. Autre exemple, l'évolution d'une altitude z en fonction de l'abscisse x : dz/dx est sans dimension.

Vrai : Il manque le temps T et l'intensité I et on a la vitesse V et la charge Q à la place. Or, un temps est constructible par L/V et le courant vaut une charge par unité de temps donc $I = Q/T = QV/L$.

Vrai : Un angle est sans dimension de même que le résultat d'une fonction mathématique.

Faux : Il ne faut pas utiliser les arrondis intermédiaire et toujours garder la valeur exacte en mémoire dans la calculatrice pour ne pas que les erreurs ne se propagent.

Faux : Si l'incertitude est uniquement due aux incertitudes des appareils, ce serait vrai, malheureusement, il se rajoute souvent des incertitudes de lecture ou liées au protocole opératoire, ce qui fait que ce n'est pas vrai dans le cas général.

3. Conversions en unités SI

$$1. 50 \text{ km/h} = 14 \text{ m/s}$$

$$2. 0,2 \text{ g/cm}^3 = 200 \text{ kg/m}^3$$

$$3. 30 \text{ mA.cm}^2 = 3 \cdot 10^{-6} \text{ A.m}^2$$

$$4. 25 \text{ mN.cm} = 0,25 \cdot 10^{-3} \text{ N.m}$$

$$5. 350 \text{ mL} = 3,5 \cdot 10^{-4} \text{ m}^3$$

$$6. 15^\circ = 0,26 \text{ rad}$$

$$7. 2'15'' = 6,5 \cdot 10^{-4} \text{ rad}$$

4. Unités diverses

Masse volumique en kg/m^3

Volume massique en m^3/kg

Transfert thermique en joule, soit $\text{kg.m}^2.\text{s}^{-2}$

Moment d'une force en N.m soit $\text{kg.m}^2.\text{s}^{-2}$

Conductance en Ω^{-1} soit A/V . Or une puissance (en watt, ou joule par seconde ou $\text{kg.m}^2.\text{s}^{-3}$) s'écrit aussi $u \times i$, donc les volts sont en fait des $\text{kg.m}^2.\text{s}^{-3}.\text{A}^{-1}$. Donc finalement la conductance s'exprime en $\text{A}^2.\text{kg}^{-1}.\text{m}^{-2}.\text{s}^3$.

Travail en joule, soit $\text{kg.m}^2.\text{s}^{-2}$

Coefficient d'autoinduction tel que $u = L \frac{di}{dt}$, soit L en $\text{V.A}^{-1}.\text{s}$ ou encore $\text{kg.m}^2.\text{s}^{-2}.\text{A}^{-2}$.

Différence de potentiel en volt V soit des $\text{kg.m}^2.\text{s}^{-3}.\text{A}^{-1}$.

Intensité en ampère A

5. Une tasse de café? En prenant $m_e = 50$ g, $m_t = 90$ g, $\Delta T = 60^\circ\text{C}$, il vient

$$Q = m_e c \Delta T = 1,3 \cdot 10^4 \text{ J} \quad h = \frac{Q}{(m_t + m_e)g} = 9,1 \cdot 10^3 \text{ m} \quad \text{et} \quad v = \sqrt{\frac{2Q}{m_t + m_e}} = 4,2 \cdot 10^2 \text{ m.s}^{-1}$$

6. Mesure et précision du résultat None available

7. Une année, vision du physicien None available

8. L'instrument de mesure None available

9. Batman* None available.

10. Problèmes de poids* None available

11. Maquette mécanique des fluides* None available

12. Anecdote historique sur la puissance de l'analyse dimensionnelle* None available.

13. Analyse dimensionnelle pour trouver une loi* None available.

14. Chiffres significatifs et encadrement* None available

15. Précision de mesure* None available

16. Je pense donc je me trompe!* Calculons l'incertitude-type sur G vue comme une fonction de trois variables M_1 , M_2 et M_3 . On a

$$\begin{aligned} u(G) &= \sqrt{\left(\frac{\partial G}{\partial M_1}\right)^2 (u(M_1))^2 + \left(\frac{\partial G}{\partial M_2}\right)^2 (u(M_2))^2 + \left(\frac{\partial G}{\partial M_3}\right)^2 (u(M_3))^2} \\ &= \sqrt{\frac{(u(M_1))^2}{(M_3 - M_2)^2} + \left(\frac{M_1}{(M_3 - M_2)^2}\right)^2 (u(M_2))^2 + \left(\frac{M_1}{(M_3 - M_2)^2}\right)^2 (u(M_3))^2} \end{aligned}$$

Avec les valeurs numériques fournies, on voit que les termes en $M_1 / (M_3 - M_2)^2 = 0,20$ sont plus grands que le terme en $1 / |M_3 - M_2| = 0,033$. La précision de la mesure sur M_1 est donc moins critique que les deux autres puisque ceux-ci interviennent dans une différence et donc que l'incertitude relative sur cette différence peut être bien plus élevée que sur chacune des mesures M_2 et M_3 prises séparément.

17. Incertitude en optique**

1. Les graduations sont données tous les 10° sur le dessin, on peut donc estimer une incertitude-type sur les mesures de $u(i) = u(r) \approx 5^\circ$ environ.

2. L'indice n est une fonction des angles i et r en suivant la loi de Descartes : $n = f(i, r) = \frac{\sin i}{\sin r}$. On doit donc calculer l'incertitude-type d'une grandeur composée à l'aide des dérivées partielles.

$$u(n) = \sqrt{\left(\frac{\partial f}{\partial i}\right)^2 (u(i))^2 + \left(\frac{\partial f}{\partial r}\right)^2 (u(r))^2}$$

Les dérivées partielles se calculent comme suit. Celle par rapport à i se fait à r constant, donc $\sin r$ est une constante qui peut sortir de la dérivée.

$$\frac{\partial f}{\partial i} = \frac{\partial}{\partial i} \left(\frac{\sin i}{\sin r} \right) = \frac{1}{\sin r} \times \frac{\partial}{\partial i} (\sin i) = \frac{1}{\sin r} \times \cos i$$

De même, celle par rapport à r se fait à i constant donc $\sin i$ sort

$$\frac{\partial f}{\partial r} = \frac{\partial}{\partial r} \left(\frac{\sin i}{\sin r} \right) = \sin i \times \frac{\partial}{\partial r} \left(\frac{1}{\sin r} \right) = \sin i \times \left(-\frac{\cos r}{\sin^2 r} \right)$$

Ainsi,

$$u(n) = \sqrt{\left(\frac{\cos i}{\sin r} \right)^2 (u(i))^2 + \left(-\frac{\sin i}{\sin^2 r} \cos r \right)^2 (u(r))^2}$$

Pour l'application numérique, il faut bien prendre garde à convertir $u(i)$ et $u(r)$ en radians (qui est l'unité réelle des fonctions mathématiques sinusoïdale, le degré n'étant qu'une facilité introduite pour manipuler des nombres à peu près entiers)

$$u(n) = \sqrt{\left(\frac{0,766}{0,423} \right)^2 \times (0,0873)^2 + \left(\frac{0,643 \times 0,906}{(0,423)^2} \right)^2 \times (0,0873)^2} = 0,33$$

à comparer avec la valeur de $n = 1,52\dots$ On aurait donc $n = 1,5$ avec $u(n) = 0,33$. Si on considère que $u(i) = u(r) = 1^\circ$, on aurait alors $n = 1,52$ avec $u(n) = 0,065$ ce qui est déjà bien meilleur comme estimation.

18. Étude d'une pile** La correction ci-après est donnée en Python et sera illustrée durant l'année, mais d'autres moyen (regressi, calculatrice ou graphes faits à la main) sont tout à fait possible.

On commence par recopier les données de l'énoncé (stockées dans U et I) et calculer les incertitudes (stockées dans dU et dI) dont la valeur en chaque point est fonction de la valeur lue sur le multimètre. Les valeurs de I sont rentrées sur deux lignes uniquement pour des raisons de visibilité et celles de dI du fait qu'il y a deux calibres selon que le courant est inférieur ou supérieur à 200 μA .

```

1  # Import des outils utiles
2  import numpy as np
3  import scipy as sp
4  import scipy.optimize
5  import matplotlib.pyplot as plt
6
7  # Définition des données et incertitudes
8  U = [4.731,4.731,4.73,4.728,4.724,4.722,4.721,4.719,4.716]
9  I = [9.283e-05,1.1545e-04,1.5265e-04,2.352e-04]
10 I+= [4.686e-04,5.200e-04,6.661e-04,7.750e-04,9.264e-04]
11 dU = [0.05e-2*u + 3e-3 for u in U]
12 dI = [0.2e-2 *i + 3e-2*1e-6 for i in I if i < 200e-6]
13 dI+= [0.2e-2 *i + 3e-4*1e-3 for i in I if i > 200e-6]
14
15 # Pour éviter tout souci, il faut que les tableau soit des np.array
16 U = np.array(U)
17 I = np.array(I)
18 dU = np.array(dU)
19 dI = np.array(dI)

```

À présent que les données sont accessibles, il s'agit de faire un ajustement de u en fonction de i selon la loi $u = E - r i$ signalée dans l'énoncé.

```

1 # La fonction d'ajustement
2 f = lambda i,E,r: E - r*i
3
4 # L'ajustement proprement dit sur p=[E,r]
5 p,pvar = sp.optimize.curve_fit(f,I,U)          # L'ajustement
6 E,r = p                                       # Les valeurs
7 dE,dr = [np.sqrt(pvar[i,i]) for i in range(2)] # Les erreurs
8 fit = f(I,E,r) # Et la courbe correspondante pour affichage plus tard.

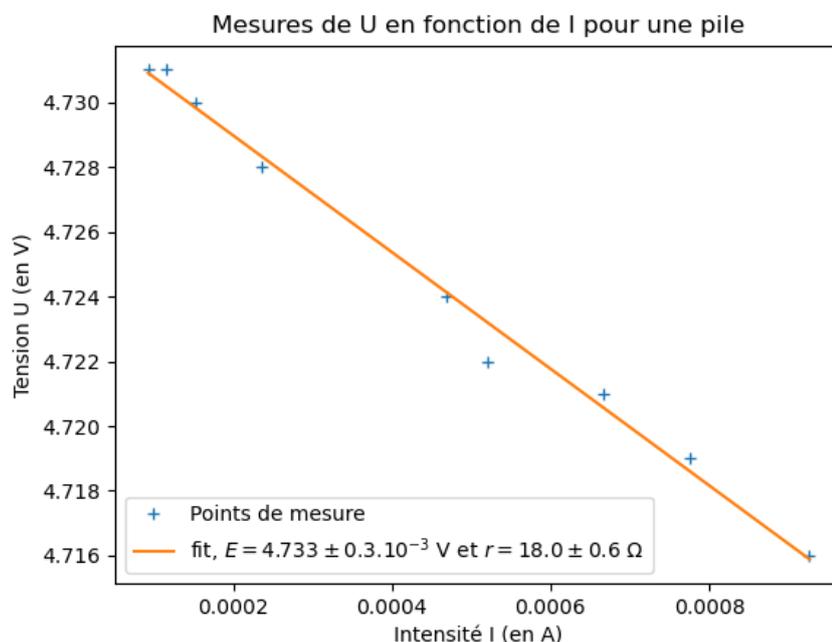
```

Il ne reste plus qu'à demander à Python de faire le dessin pour obtenir le graphe suivant

```

1 def arrondi(nb):
2     '''Pour avoir un beau rendu en LaTeX'''
3     puissance = int(np.log10(nb))
4     return '%.1f.10~{%d}' % (nb/10**puissance,puissance)
5
6 def legende_titre_et_axes():
7     '''Pour éviter de le retaper deux fois'''
8     plt.legend(loc='lower left')
9     plt.title('Mesures de U en fonction de I pour une pile')
10    plt.xlabel('Intensité I (en A)')
11    plt.ylabel('Tension U (en V)')
12
13    # Représentation graphique sans barres d'erreur
14    plt.plot(I,U,'+',label='Points de mesure') # Les points de mesure
15    lab = 'fit, $E=%.3f\pm%s$ V et $r=%.1f\pm$ %.1f$ $\Omega$' % (E,arrondi(dE),r,dr)
16    plt.plot(I,fit,label=lab) # L'ajustement
17    legende_titre_et_axes() # On affiche légende, titre et axes
18    plt.savefig('pile1.png') # On sauvegarde
19    plt.clf() # et on nettoie.

```



L'ajustement semble plutôt bon et l'estimation de l'incertitude est donnée par la méthode d'ajustement (ce sont les valeurs données par Regressi par exemple). Néanmoins, quand on rajoute les barres d'erreurs, on se rend compte de

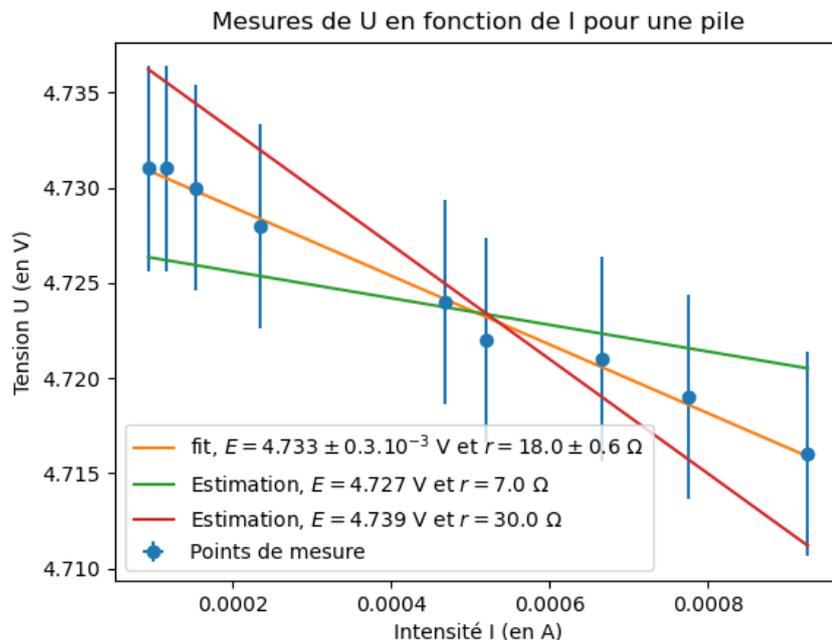
deux choses :

- celles sur i sont totalement négligeables;
- en revanche celles sur u sont très importantes et si on bouge « à la main » les valeurs de E et r pour faire passer des droites extrêmes dans toutes les barres d'erreur présentées, on se rend compte qu'au lieu des 3% d'erreur sur r annoncées par l'ajustement automatique (plutôt correct à l'œil), l'incertitude sur la valeur de r pourrait être de l'ordre de 60%. En d'autres termes, on peut penser observer un comportement qui n'existe pas en réalité!

```

1  # Estimation "à l'oeil"
2  Ebasse,rbasse = 4.727,7      # Estimation basse pour r
3  Ehaute,rhaute = 4.739,30    # Estimation haute pour r
4  basse = f(I,Ebasse,rbasse)  # Les valeurs pour l'affichage
5  haute = f(I,Ehaute,rhaute) # pour les deux estimations.
6
7  def make_lab(E,r):
8      '''Représentation unifiée des label'''
9      return 'Estimation, $E=%.3f$ V et $r=%.1f$ $\Omega$' % (E,r)
10
11 # Représentation graphique avec barres d'erreur
12 plt.errorbar(I,U,xerr=dI,yerr=dU,fmt='o',label='Points de mesure')
13 plt.plot(I,fit,label=lab)    # L'ajustement
14 plt.plot(I,basse,label=make_lab(Ebasse,rbasse)) # Estimation1
15 plt.plot(I,haute,label=make_lab(Ehaute,rhaute)) # Estimation2
16 legende_titre_et_axes()     # On affiche légende, titre et axes
17 plt.savefig('pile2.png')    # On sauvegarde
18 plt.clf()                   # et on nettoie.

```



Bien entendu, cela demanderait que toutes les erreurs aient à chaque fois été faites dans le même sens à droite et à gauche de l'ajustement (ce qui ne serait vraiment « pas de bol »), mais les données ne permettent pas de rejeter ce comportement comme complètement aberrant. La prise en compte des barres d'erreur modifie *de fait* la confiance que l'on peut porter à la détermination de cette résistance interne.

Finalement on peut en déduire les valeurs suivantes pour E et r :

$$E = 4,733 \pm 0,006 \text{ V} \quad \text{et} \quad r = 18 \pm 12 \Omega$$

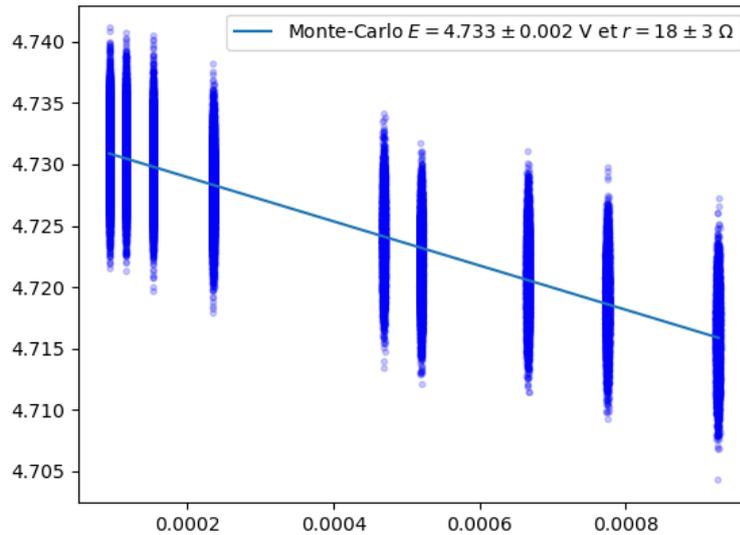
L'incertitude absolue sur E passe de 0,3 mV à 6 mV par rapport à l'évaluation statistique (multiplication par 20, mais l'incertitude relative reste inférieure à 0,1%) et surtout celle sur r passe de 0,6 Ω à 12 Ω (à nouveau une multiplication par 20 mais une incertitude relative supérieure à 50%!).

À noter qu'une estimation à base d'un tirage Monte-Carlo est finalement plus raisonnable puisque l'incertitude sur E est de 2 mV et celle sur r est de 3 Ω , ce qui montre que les droites limites ont vraiment tendance à surestimer les incertitudes, histoire d'être sûr que l'intervalle de confiance soit assez grand.

```

1  # Faisons la méthode de Monte-Carlo pour estimer l'erreur
2
3  nbMC = 10000 # Nombre de tirages
4  E_MC,r_MC = [],[] # Pour stocker les valeurs trouvées pour E et r
5  X,Y = [],[]
6  for i in range(nbMC): # Boucle sur tous les tirages
7      if i%100 == 0: print(i) # Un peu de feedback pour visualiser l'avancée des calculs
8      # On fait un tirage par une loi normale centrée sur la valeur de U avec un écart-type
9      # de dU/2 avant autant de point (len(U)) qu'on a de valeurs
10     U_MC = np.random.normal(U,dU/2,len(U))
11     # Pareil pour les valeurs de I
12     I_MC = np.random.normal(I,dI/2,len(I))
13     # Reste à faire l'ajustement proprement dit
14     p,pvar = sp.optimize.curve_fit(f,I_MC,U_MC) ## L'ajustement
15     E,r = p ## Les valeurs
16     # Stockage des valeur de E et r et des points intermédiaires
17     E_MC.append(p[0])
18     r_MC.append(p[1])
19     X.append(I_MC)
20     Y.append(U_MC)
21
22 # On rajoute les points tirés au graphe en mode presque transparent
23 plt.plot(X,Y, '.b',alpha=0.2)
24
25 # Maintenant qu'on a toutes les valeurs possibles pour E et r, on peut faire les statistiques
26 E_moyen = np.mean(E_MC)
27 r_moyen = np.mean(r_MC)
28 dE_moyen= np.std(E_MC)
29 dr_moyen= np.std(r_MC)
30
31 rE = round(E_moyen,3)
32 rr = int(round(r_moyen))
33 rdE= round(dE_moyen,3)
34 rdr= int(round(dr_moyen))
35
36 plt.plot(I_MC,f(I_MC,E_moyen,r_moyen),
37 label='Monte-Carlo $E={}\pm{}\$ V et $r={}\pm{}\sim\Omega\$'.format(rE,rdE,rr,rdr))
38 plt.legend()
39 plt.savefig('pile3.png')
40 plt.clf()

```



19. Chiffres significatifs et mesures suivies d'un calcul* None available.

20. Homogénéité**

1. On cherche tout d'abord à déterminer la dimension de G et celle de \hbar :

$$\text{Comme } \|\vec{F}_{\text{grav}}\| = \mathcal{G} \frac{m_1 m}{r^2}, \quad \text{M.L.T}^{-2} = [F_{\text{grav}}] = \frac{[\mathcal{G}].\text{M}^2}{\text{L}^2}$$

d'où

$$[\mathcal{G}] = \text{M}^{-1}.\text{L}^3.\text{T}^{-2}$$

En ce qui concerne \hbar

$$[\hbar] = \frac{[E]}{[\omega]} = \frac{\text{M.L}^2.\text{T}^{-2}}{\text{T}^{-1}}$$

d'où

$$[\hbar] = \text{M.L}^2.\text{T}^{-1}$$

2. Une fois ces dimensions trouvées, on peut déterminer

(a) **La masse de Planck** m_p en notant $m_p = \mathcal{G}^\delta . \hbar^\eta . c^\mu$.

En notant que chaque terme de l'égalité doit avoir même dimension, il vient :

$$M = (\text{M}^{-\delta} . \text{L}^{3\delta} . \text{T}^{-2\delta}) . (\text{M}^\eta . \text{L}^{2\eta} . \text{T}^{-\eta}) . (\text{L}^\mu . \text{T}^{-\mu}) = \text{M}^{(-\delta+\eta)} . \text{L}^{(3\delta+2\eta+\mu)} . \text{T}^{(-2\delta-\eta-\mu)}$$

$$\text{On obtient le système } \begin{cases} -\delta + \eta = 1 \\ 3\delta + 2\eta + \mu = 0 \\ -2\delta - \eta - \mu = 0 \end{cases}$$

Soit finalement : $\delta = -\frac{1}{2}$, $\eta = \frac{1}{2}$ et $\mu = \frac{1}{2}$. On peut donc écrire :

$$m_p = \sqrt{\frac{\hbar c}{\mathcal{G}}}$$

(b) **Le temps de Planck** t_p peut effectivement être déterminé en notant $t_p = \mathcal{G}^\alpha . \hbar^\beta . c^\gamma$. On propose ici une autre méthode possible :

- En regardant les dimensions de \mathcal{G} et \hbar , on voit que le produit $\mathcal{G}\hbar$ a la dimension L^5T^{-3} (élimination de la dimension M).
- Pour n'avoir plus que la dimension T, on voit que l'on doit diviser $\mathcal{G}\hbar$ par c^5 . Il reste alors T^2 , soit en récapitulant : $\left[\frac{\mathcal{G}\hbar}{c^5} \right] = \text{T}^2$.

— On en déduit que le temps de Planck t_P s'écrit :

$$t_P = \sqrt{\frac{\mathcal{G} \hbar}{c^5}}$$

(c) **La longueur de Planck** ℓ_P peut aussi être déterminée en notant $\ell_P = \mathcal{G}^\Delta \cdot \hbar^\nu \cdot c^\xi$. Ceci étant, on peut aussi écrire $\ell_P = c t_P$, soit

$$\ell_P = \sqrt{\frac{\hbar \mathcal{G}}{c^3}}$$

3. *Application numérique :*

$$\begin{aligned} m_P &= 2,18 \cdot 10^{-8} \text{ kg} \\ t_P &= 5,38 \cdot 10^{-44} \text{ s} \\ \ell_P &= 1,61 \cdot 10^{-35} \text{ m} \end{aligned}$$

Temps de Planck et longueur de Planck constituent respectivement les plus petites mesures temporelle et spatiale ayant une signification physique dans le cadre de nos théories présentes. Quant à la masse de Planck, d'ordre de grandeur plus familier (celle d'un grain de sable), elle peut s'interpréter comme la masse maximale que pourrait avoir une particule fondamentale.

21. Exploitation de mesures expérimentales : chute libre d'une bille**

1. La mesure de hauteur se fait au centimètre près. Si l'observateur est attentif, on peut dire qu'on est sûr que la valeur est connue au demi-centimètre près pour une distribution rectangulaire, donc pour avoir l'incertitude-type, on peut encore diviser ce résultat par $\sqrt{3}$. Ainsi,

$$u(h) = \frac{0,5}{\sqrt{3}} = 0,29 \text{ cm}$$

L'incertitude-type sur la valeur moyenne des temps de chute se trouve en calculant σ_{n-1} et en le divisant par la racine du nombre n de mesures. Informatiquement, avec Python, cela donne

```

1 import numpy as np
2
3 h = 1.80
4 u_h = 5e-3 / 3**0.5 # 5mm divisé par racine(3)
5 t = np.array([0.61, 0.62, 0.60, 0.63, 0.62, 0.61, 0.58, 0.59, 0.6, 0.64, 0.62, 0.61, 0.63])
6
7 # Calcul de la valeur moyenne
8 moyenne_t = np.mean(t)
9
10 # ddof = 1 permet de diviser par n-1 plutôt que n dans l'expression de l'écart-type
11 sigma_nmoins1 = np.std(t, ddof = 1)
12
13 # On divise par racine(n) pour l'incertitude-type sur moyenne(t)
14 u_t = sigma_nmoins1 / (len(t)) ** 0.5

```

Après calculs, on obtient alors

$$\bar{t} = 0,61 \quad \text{et} \quad u(\bar{t}) = \frac{\sigma_{n-1}}{\sqrt{n}} = 0,0047 \text{ s}$$

2. Pour calculer les quantités adéquates pour g , on a déjà besoin de savoir le lien entre hauteur h , temps de chute t et accélération de la pesanteur g . Le cours de mécanique nous montrera que $g = 2h/t^2$. Ici on s'intéresse à ce qui se passe d'un point de vue statistique selon que l'on calcule $g(h, \bar{t})$ ou la moyenne des $g(h, t)$. On va aussi faire

un tirage Monte-Carlo pour montrer que la formule de propagation des incertitudes est cohérente avec cela. En l'occurrence, la formule de propagation des incertitudes est une application du cas $x^p y^q$ quand $p = 1$ et $q = -2$, soit

$$\frac{u(g)}{g} = \sqrt{\left(\frac{u(h)}{h}\right)^2 + 4\left(\frac{u(\bar{t})}{\bar{t}}\right)^2}$$

Les calculs Python donnent alors

```

1 def g(h, t):
2     """ Donne l'accélération g de la pesanteur en fonction de la hauteur h de chute
3     et du temps t de ladite chute. """
4     return 2*h / t**2
5
6 # Premier cas: on calcule directement g avec h et moyenne(t) et on estime
7 # l'incertitude-type par formule de propagation des incertitudes
8 g_de_moyenne_de_t = g(h, moyenne_t)
9 u_g1 = g_de_moyenne_de_t * np.sqrt( (u_h/h)**2 + 4*(u_t/moyenne_t)**2)
10
11 # Deuxième cas, on calcule g pour chacun des t et on estime son incertitude
12 # par sigma/sqrt(n)
13 moyenne_des_g = np.mean(g(h, t))
14 u_g2 = np.std(g(h, t), ddof = 1) / np.sqrt(len(t))
15
16 # Troisième cas, on fait nos propres tirages en faisant varier h et t autour
17 # des valeurs moyennes et on calcule moyenne et écart-type
18 import numpy.random as rd
19 N = 10**5 # Nombre de tirages
20 h_MC = rd.normal(h, u_h, N)
21 t_MC = rd.normal(moyenne_t, u_t, N)
22 g_MC = g(h_MC, t_MC)
23 moyenne_g_MC = np.mean(g_MC)
24 u_g3 = np.std(g_MC)

```

Les résultats sont les suivants :

$$g_1 = g(h, \bar{t}) = 9,60 \text{ m.s}^{-2} \quad \text{et} \quad u(g_1) = 0,15 \text{ m.s}^{-2} \quad (1)$$

$$g_2 = \overline{g(h, t)} = 9,62 \text{ m.s}^{-2} \quad \text{et} \quad u(g_2) = 0,15 \text{ m.s}^{-2} \quad (2)$$

$$g_3 = \overline{g(h_{MC}, t_{MC})} = 9,60 \text{ m.s}^{-2} \quad \text{et} \quad u(g_3) = 0,15 \text{ m.s}^{-2} \quad (3)$$

Les trois valeurs sont parfaitement cohérentes les unes avec les autres aux incertitudes près (on est clairement à moins de deux incertitudes-type de distance les unes des autres) avec des valeurs similaires pour les dites incertitudes-type, mais est-ce cohérent avec la valeur connue de $g_{\text{réf}} = 9,81 \text{ m.s}^{-2}$? Pour le vérifier, on va calculer le Z-score dans les deux cas (si $g_{\text{réf}}$ est certain donc $u(g_{\text{réf}}) = 0$ ou alors avec une incertitude-type $u(g_{\text{réf}}) = 0,01 \text{ m.s}^{-2}$). Ainsi,

```

1 def Z_score(v1, v2, u1, u2):
2     return abs(v2 - v1) / np.sqrt(u1**2 + u2**2)
3
4 g_ref = 9.81
5
6 Z1 = Z_score(g_ref, g_de_moyenne_de_t, 0, u_g1)
7 Z2 = Z_score(g_ref, g_de_moyenne_de_t, 0.01, u_g1)

```

Les Z-scores obtenus valent respectivement

$$Z_1 = 1,41 \quad \text{et} \quad Z_2 = 1,40$$

La différence n'est pas notable puisque l'incertitude-type sur $g_{\text{réf}}$ est 15 fois moindre que celle sur les valeurs mesurées donc sans effet sur le résultat. En revanche, comme le Z-score est inférieur à 2 dans chaque cas, on en déduit que la valeur mesurée, bien qu'un peu faible, visiblement, est tout à fait compatible avec la valeur de référence habituelle. La Terre ne s'est pas mise en vacances pendant l'expérience.