
CHAPITRE D3 – INCERTITUDES

La mesure est une activité fondamentale en sciences expérimentales. Connaître la valeur de la grandeur mesurée est tout aussi important que d'être capable d'estimer la précision avec laquelle on la connaît.

I) Évaluation une incertitude-type

1) Approche statistique – évaluation de type A

Imaginons une expérience physique dont le but est de mesurer un paramètre noté x .

Soit une série de N mesures, dont les résultats sont notés $\{x_i\}$. On définit la **valeur moyenne**, notée $\bar{x}$, et l'**écart-type**, noté σ_x , de la distribution :

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i \quad \text{et} \quad \sigma_x = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2}$$

La valeur moyenne indique le « centre » de la distribution et l'écart-type indique son « l'étalement » : plus σ_x est grand, plus la distribution est étalée.

En physique, l'écart-type s'appelle **incertitude-type** et est notée $u(x) = \sigma_x$. Elle quantifie l'incertitude de l'expérimentateur sur sa connaissance de chacune des mesures x_i .

Propriété :

L'incertitude-type de la valeur moyenne de la distribution est donnée par :

$$\overline{u}(x) = \frac{u(x)}{\sqrt{N}}$$

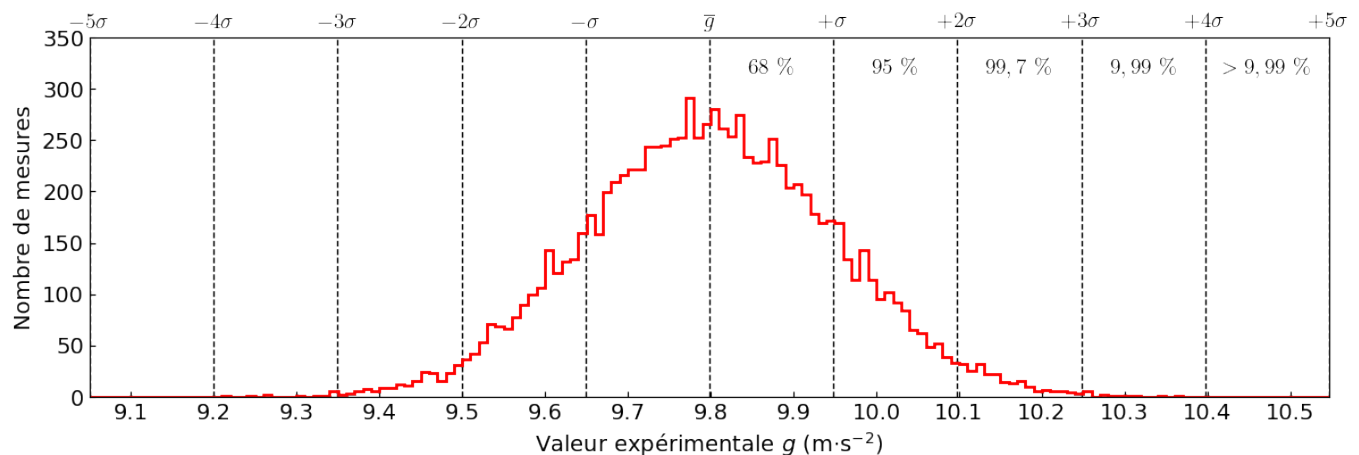
En effet, parmi deux distributions ayant même valeur moyenne $\bar{x}$ et même l'incertitude-type $u(x)$, nous aurons « plus confiance » en celle réalisée avec $N = 10^6$ mesures que celle réalisée avec seulement $N = 10$ mesures. Il est donc attendu que l'incertitude-type sur la valeur moyenne diminue avec le nombre de mesures qui augmente.

Application

Imaginons un TP où l'objectif est de mesurer l'accélération de pesanteur g . Le protocole consiste à mesurer le poids P d'une masse m à l'aide d'un dynamomètre. Le dynamomètre permet d'obtenir la valeur de P et une balance permet d'obtenir la valeur de m . L'accélération de pesanteur est alors définie par :

$$P = mg \quad \Rightarrow \quad g = \frac{P}{m}$$

La même protocole est répété un grand nombre de fois (N) pour différentes masses ou par différents étudiants. On obtient ainsi N valeurs de la pesanteur, notées $\{g_i\}$ pour $i \in \llbracket 1, N \rrbracket$. Traçons les résultats sous forme d'histogramme.



On obtient une courbe en cloche (appelée une **courbe de Gauss** ou **loi normale**) centrée autour de $\bar{g}$ et d'étalement typique σ . Les pourcentages indiquent que : 68 % des mesures tombent dans l'intervalle $\bar{g} \pm \sigma$, 95 % des mesures tombent dans l'intervalle $\bar{g} \pm 2\sigma$, etc.

L'analyse des $N = 10\,000$ données donne :

$$\boxed{\bar{g} = 9,7980 \text{ m} \cdot \text{s}^{-2}} \quad \text{et} \quad \sigma = 0,1490 \text{ m} \cdot \text{s}^{-2}$$

On en déduit alors l'incertitude-type sur chacune des mesures $u(g_i)$ ainsi que sur la valeur moyenne $u(\bar{g})$.

$$u(g_i) = \sigma = 0,1490 \text{ m} \cdot \text{s}^{-2} \quad \text{et} \quad \boxed{u(\bar{g}) = \frac{\sigma}{\sqrt{N}} = 0,0015 \text{ m} \cdot \text{s}^{-2}}$$

2) Autre approche – évaluation de type B

Bien souvent, la mesure ne présente pas de variabilité ou seule une unique mesure peut être réalisée (par soucis de temps). Dans ce cas, il n'est pas possible de réaliser un traitement des incertitudes de type A.

L'expérimentateur doit alors réaliser une seule mesure, et estimer la plus petite plage dans laquelle il est certain de trouver la valeur recherchée. On note $\bar{x}$ la valeur centrale de cette plage et Δ sa demi-largeur. Autrement dit, l'expérimentateur est certain que la valeur réelle est dans l'intervalle $\bar{x} \pm \Delta$.

Propriété :

En supposant une densité de probabilité uniforme dans l'intervalle $\bar{x} \pm \Delta$ et nulle hors de cet intervalle, l'incertitude-type sur la mesure vaut :

$$u(\bar{x}) = \frac{\Delta}{\sqrt{3}}$$

Application

Reprenons l'expérience précédente mais du point de vue d'un étudiant faisant une mesure unique. Ce dernier possède une masse m dont il veut connaître la valeur à l'aide d'une balance. Cette mesure ne présente aucune variabilité, la balance affiche toujours 500,0 g.

Étant donné la précision de la balance, on pose :

$$\boxed{\bar{m} = 500,0 \text{ g}} \quad \text{et} \quad \Delta = 0,05 \text{ g}$$

En effet, si la masse était telle que $m > 500,05 \text{ g}$, alors la balance aurait arrondi le résultat à 500,1 g. Même chose si $m < 499,95 \text{ g}$ qui aurait été arrondi à 499,9 g. L'expérimentateur est donc certain que la valeur réelle est dans l'intervalle $\bar{m} \pm \Delta$, mais au sein de cet intervalle, la densité de probabilité est uniforme. Il y a autant de chance que le résultat soit 500,00 g ou 500,04 g.

On en déduit alors l'incertitude-type sur la valeur moyenne $u(\bar{m})$.

$$\boxed{u(\bar{m}) = \frac{\Delta}{\sqrt{3}} = 0,029 \text{ g}}$$

3) Composition d'incertitudes-types

Soit une grandeur y qui s'écrit comme une somme, une différence, un produit ou un quotient de deux grandeurs x_1 et x_2 . L'incertitude-type $u(y)$ vaut :

$$\begin{array}{llll} y = x_1 + x_2 & \text{ou} & y = x_1 - x_2 & \Rightarrow & u(y)^2 = u(x_1)^2 + u(x_2)^2 \\ y = x_1 \times x_2 & \text{ou} & y = \frac{x_1}{x_2} & \Rightarrow & \left(\frac{u(y)}{y}\right)^2 = \left(\frac{u(x_1)}{x_1}\right)^2 + \left(\frac{u(x_2)}{x_2}\right)^2 \end{array}$$

Remarque :

La grandeur $\frac{u(x)}{x}$ s'appelle l'**incertitude-type relative**.

Application

Poursuivons l'application de la partie I.2. Imaginons qu'un raisonnement similaire est conduit l'étudiant à déterminer $\bar{P}$ et $u(\bar{P})$ par un traitement de type B des incertitudes sur une mesure unique.

$$\begin{cases} \bar{m} = 500,0 \text{ g} \\ u(\bar{m}) = 0,029 \text{ g} \end{cases} \quad \text{et} \quad \begin{cases} \bar{P} = 4,900 \text{ N} \\ u(\bar{P}) = 0,058 \text{ N} \end{cases}$$

Nous pouvons alors en déduire $\bar{g}$ et $u(\bar{g})$ grâce à la formule de composition d'incertitudes-types.

$$\boxed{\bar{g} = \frac{\bar{P}}{\bar{m}} = 9,800 \text{ m} \cdot \text{s}^{-2}} \quad \text{et} \quad \left(\frac{u(\bar{g})}{\bar{g}}\right)^2 = \left(\frac{u(\bar{P})}{\bar{P}}\right)^2 + \left(\frac{u(\bar{m})}{\bar{m}}\right)^2$$

Nous pouvons soit déterminer rigoureusement $u(\bar{g})$ grâce à l'expression ci-dessus, soit remarquer que :

$$\frac{u(\bar{P})}{\bar{P}} = 1,18 \times 10^{-2} \quad \gg \quad \frac{u(\bar{m})}{\bar{m}} = 5,8 \times 10^{-5}$$

C'est l'avantage des incertitude-type relative, elles peuvent se comparer entre elles. Après simplification, on en déduit :

$$\frac{u(\bar{g})}{\bar{g}} \simeq \frac{u(\bar{P})}{\bar{P}} \Rightarrow \boxed{u(\bar{g}) = 0,116 \text{ m} \cdot \text{s}^{-2}}$$

Lorsque y ne s'écrit pas comme une somme, une différence, un produit ou un quotient de deux grandeurs x_1 et x_2 , nous allons effectuer une **simulation de Monte-Carlo** pour déterminer $u(y)$. L'objectif de cette simulation est de :

- simuler le tirage aléatoire d'un grand nombre (N) de valeur de $\{x_{1,i}\}$ et $\{x_{2,i}\}$ selon une loi normale ;
- calculer les valeurs de $\{y_i\}$ correspondantes ;
- calculer la valeur moyenne $\bar{y}$ et l'écart-type σ_y de cette distribution.

Autrement dit, nous donnons au programme le paramètres $\bar{x}_1$, $\bar{x}_2$, $u(\bar{x}_1)$, $u(\bar{x}_2)$ et nous lui demandons de simuler N expériences avec ces paramètres expérimentaux. Nous pouvons ainsi effectuer un traitement de type A de la série de « mesures ».

Application

Voici un code traitant du même exemple que précédemment. Une simulation de Monte-Carlo n'est pas nécessaire dans ce cas puisque l'on connaît la formule de propagation des incertitudes-types dans le cas d'un quotient, mais la méthode s'applique à n'importe quelle expression de y (il suffit de changer la ligne 17).

En revanche, le résultat de la simulation doit être en accord avec le calcul littéral précédent, ce qui est bien le cas ici !

```
1 import numpy as np
2
3 # PARAMÈTRES EXPÉRIMENTAUX -----
4
5 m_moy = 500.0 * 1e-3      # Valeur moyenne de m
6 u_m = 0.029 * 1e-3       # Incertitude-type de m
7
8 P_moy = 4.900             # Valeur moyenne de P
9 u_P = 0.058              # Incertitude-type de P
10
11 # SIMULATION DE MONTE-CARLO -----
12
13 N = 1000000              # Nombre de tirages
14 m_MC = np.random.normal(m_moy, u_m, N)    # Tirage des {m_i}
15 P_MC = np.random.normal(P_moy, u_P, N)    # Tirage de {P_i}
16
17 g_MC = P_MC / m_MC       # Calcul des {g_i}
18 g_MC_moy = np.mean(g_MC) # Valeur moyenne des {g_i}
19 u_g_MC = np.std(g_MC, ddof=1) # Écart-type des {g_i}
20
```

```

21 # AFFICHAGE DES RÉSULTATS -----
22
23 print("Valeur moyenne : g = {:.3f} m/s2".format(g_MC_moy))
24 print("Incertitude-type : u(g) = {:.3f} m/s2".format(u_g_MC))

```

```

1 Valeur moyenne : g = 9.800 m/s2
2 Incertitude-type : u(g) = 0.116 m/s2

```

4) Écriture du résultat d'une mesure

Convention d'écriture : une incertitude-type s'écrit toujours avec 1 ou 2 chiffres significatifs, et le résultat de la mesure doit s'arrêter à la même décimale que l'incertitude-type.

Application

Poursuivons l'application de la partie I.3. Le résultat de la mesure g à l'aide d'une mesure unique s'écrit (en choisissant 2 chiffre significatifs pour l'incertitude-type) :

$$g = (9,80 \pm 0,12) \text{ m} \cdot \text{s}^{-2}$$

5) Compatibilité de deux mesures

Soit deux mesures x_1 et x_2 d'incertitude-type respective $u(x_1)$ et $u(x_2)$. On appelle **écart normalisé** la grandeur :

$$E_N = \frac{y}{u(y)} \quad \text{avec : } y = x_1 - x_2 \quad \Rightarrow \quad E_N = \frac{x_1 - x_2}{\sqrt{u(x_1)^2 + u(x_2)^2}}$$

Propriété :

Les deux mesures sont compatibles entre elles si : $|E_N| < 2$

Application

Terminons l'application de la partie I.3. La valeur trouvée par l'étudiant est-elle compatible avec la valeur tabulée à Montpellier ?

$$\begin{cases} g_{exp} = 9,80 \text{ m} \cdot \text{s}^{-2} \\ u(g_{exp}) = 0,12 \text{ m} \cdot \text{s}^{-2} \end{cases} \quad \text{et} \quad \begin{cases} g_{tab} = 9,806 \text{ m} \cdot \text{s}^{-2} \\ u(g_{tab}) = 0,005 \text{ m} \cdot \text{s}^{-2} \end{cases} \quad \Rightarrow \quad E_N = \frac{g_{tab} - g_{exp}}{\sqrt{u(g_{tab})^2 + u(g_{exp})^2}} = 0,05 < 2$$

Les mesures sont compatibles !

II) Régression affine

1) Obtention des paramètres de régression

Dans cette partie, nous allons chercher à vérifier si deux variables x et y sont reliées par une fonction affine, c'est-à-dire par la relation : $y = ax + b$, avec a et b des constantes à déterminer. Pour cela, nous allons réaliser une régression affine, que appelée plus communément (par abus de langage) une **régression linéaire**.

Il est nécessaire d'afficher simultanément les points expérimentaux et la droite de régression. Pour valider *qualitativement* le modèle affine, les points expérimentaux doivent être « proches » de la droite de régression et « aléatoirement répartis » au-dessus et en-dessous de la droite.

Application

Restons sur l'exemple de la détermination de g . Un étudiant dispose de plusieurs masses $\{m_i\}$ allant de 10 à 500 g. Il mesure au dynamomètre les valeurs de $\{P_i\}$ correspondantes. Puisque $P = mg$, une régression linéaire est envisageable avec :

$$y = ax + b \quad \text{avec :} \quad y = P \quad x = m \quad a = g \quad \text{et} \quad b = 0$$

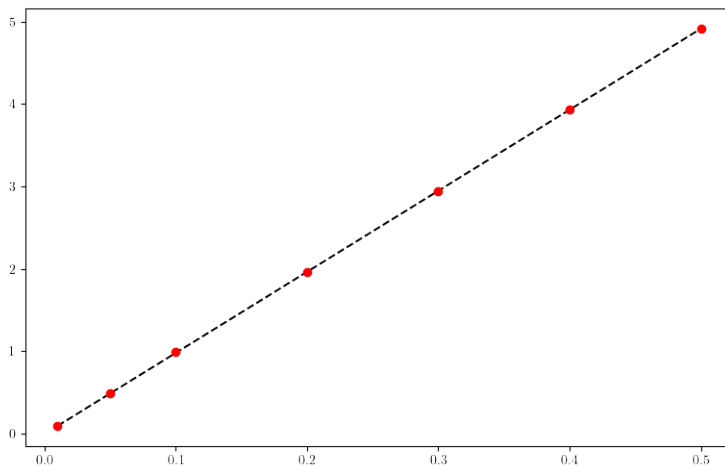
Voici un exemple de code minimaliste, en utilisant la fonction `polyfit`. Au vu des résultats de la simulation, le modèle affine est validé *qualitativement*.

Remarque :

La fonction `polyfit` (seule fonction au programme de CPGE) ne permet pas de fixer $b = 0$. On se contentera alors de dire que le b obtenu par régression est « suffisamment petit » pour pouvoir être négligé.

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 # MESURES EXPÉRIMENTALES -----
5
6 m = [10, 50, 100, 200, 300, 400, 500]          # Valeurs de m en g
7 m = np.array(m) * 1e-3                          # Conversion en array + en kg
8
9 P = [0.09, 0.49, 0.99, 1.96, 2.94, 3.93, 4.92]  # Valeurs de P en N
10 P = np.array(P)                                # Conversion en array
11
12 # RÉGRESSION AFFINE -----
13
14 x = m
15 y = P
16 a, b = np.polyfit(x, y, 1) # Régression affine y = a*x + b
17
18 # AFFICHAGE DES RÉSULTATS -----
19
20 print("Paramètres de régression : g = {:.3f} m/s2 (et b = {:.3f} N)".format(a, b))
21
22 plt.plot([min(x), max(x)], [a*min(x)+b, a*max(x)+b], 'k--') # Droite de régression
23 plt.plot(m, P, "ro")                                         # Données expérimentales
24 plt.show()
```

```
1 Paramètres de régression : g = 9.838 m/s2 (et b = -0.004 N)
```



2) Validation du modèle

On suppose que x est mesuré avec une très grande précision, donc seule l'incertitude-type $u(y)$ sur y joue un rôle pertinent. Si en pratique c'est l'inverse qui est vérifié, tracer x en fonction de y au lieu de y en fonction de x .

Afin de valider *quantitativement* le modèle affine, il faut calculer l'écart normalisé entre les points de mesure et la droite de régression (où aucune incertitude-type n'est associée aux paramètres de régression), et vérifier que $|E_N| < 2$ pour tous les points.

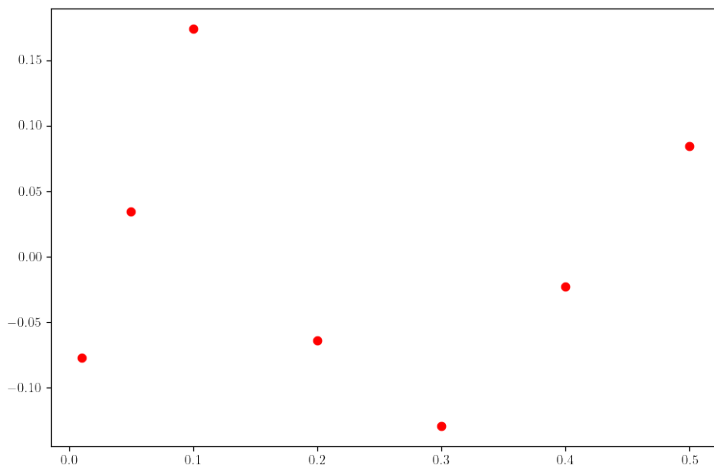
$$E_N(i) = \frac{y_i - (ax_i + b)}{u(y_i)} < 2 \quad \forall i$$

Application

Reprenons le résultat du code précédent et ajoutons l'incertitude-type $u(\overline{P}) = 0,058$ N déterminée à la partie I.3.

Dans cet exemple, le modèle affine est bien validé!

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 m = [10, 50, 100, 200, 300, 400, 500]          # Valeurs de m en g
5 m = np.array(m) * 1e-3                          # Conversion en array + en kg
6
7 P = [0.09, 0.49, 0.99, 1.96, 2.94, 3.93, 4.92]  # Valeurs de P en N
8 P = np.array(P)                                 # Conversion en array
9 u_P = 0.058                                     # Incertitude-type sur P
10
11 a, b = np.polyfit(m, P, 1)                     # Régression affine y = a*x + b
12
13 EN = (P-(a*m+b))/u_P                           # Écart normalisé
14
15 # AFFICHAGE DES RÉSULTATS -----
16
17 plt.plot(m, EN, "ro")
18 plt.show()
```



3) Incertitude-types sur les paramètres de régression

Il est finalement possible d'obtenir les incertitudes-type $u(a)$ et $u(b)$ des paramètres de régression a et b . Nous allons procéder à une simulation de Monte-Carlo, en suivant le protocole suivant :

- générer un tableau numpy des $\{y_i\}$ où chaque valeur est générée aléatoirement selon une loi normale de moyenne y_i et d'écart-type $u(y_i)$;
- effectuer une régression affine pour obtenir une valeur de a et b ;
- recommencer le processus un grand nombre de fois (N) pour obtenir une distribution de valeurs de $\{a_n\}$ et $\{b_n\}$;
- calculer les valeurs moyennes $\bar{a}$ et $\bar{b}$ ainsi que les écarts-type $u(\bar{a})$ et $u(\bar{b})$ de ces distributions.

Application

Reprenons les applications numériques précédentes.

```
1 import numpy as np
2
3 # MESURES EXPÉRIMENTALES -----
4
5 m = [10, 50, 100, 200, 300, 400, 500]          # Valeurs de m en g
6 m = np.array(m) * 1e-3                          # Conversion en array + en kg
7
8 P = [0.09, 0.49, 0.99, 1.96, 2.94, 3.93, 4.92]  # Valeurs de P en N
9 P = np.array(P)                                 # Conversion en array
10 u_P = 0.058                                     # Incertitude-type sur P
11
12 # SIMULATION DE MONTE-CARLO -----
```

```

13
14 N = 100000 # Nombre de tirages
15 a_MC = [] # Création d'une liste vide pour les {a_n}
16 b_MC = [] # Création d'une liste vide pour les {b_n}
17
18 for n in range(N):
19     P_sim = [np.random.normal(P_i, u_P) for P_i in P] # Tirage aléatoire de P
20     a, b = np.polyfit(m, P_sim, 1) # Régression affine
21     a_MC.append(a) # Ajout de la valeur de a
22     b_MC.append(b) # Ajout de la valeur de b
23
24 a_MC_moy = np.mean(a_MC) # Valeur moyenne des {a_n}
25 u_a_MC = np.std(a_MC, ddof=1) # Écart-type des {a_n}
26 b_MC_moy = np.mean(b_MC) # Valeur moyenne des {b_n}
27 u_b_MC = np.std(b_MC, ddof=1) # Écart-type des {b_n}
28
29 # AFFICHAGE DES RÉSULTATS -----
30
31 print("Valeur moyenne : a = {:.3f} m/s2".format(a_MC_moy))
32 print("Incertitude-type : u(a) = {:.3f} m/s2".format(u_a_MC))
33 print("Valeur moyenne : b = {:.3f} N".format(b_MC_moy))
34 print("Incertitude-type : u(b) = {:.3f} N".format(u_b_MC))

```

```

1 Valeur moyenne : a = 9.838 m/s2
2 Incertitude-type : u(a) = 0.128 m/s2
3 Valeur moyenne : b = -0.004 N
4 Incertitude-type : u(b) = 0.036 N

```

On en déduit donc, en faisant attention à mettre le bon nombre de chiffres significatifs :

$$g = (9,84 \pm 0,13) \text{ m} \cdot \text{s}^{-2}$$