Mathématiques et physique — Traitement des incertitudes

On souhaite déterminer une évaluation expérimentale x_{exp} d'une grandeur x.

1 — Évaluation de type A de l'incertitude type (série de N mesures)

On fait une approche statistique, à partir de N mesures x_i indépendantes.
L'estimation de la grandeur mesurée est la moyenne des valeurs mesurées :

$$\boxed{x_{\text{exp}} = \overline{x} = \frac{1}{N}\sum_{i=1}^{N} x_i}\ .$$

L'incertitude-type associée à la moyenne est donnée par

$$\boxed{u(x) = \frac{\sigma(x)}{\sqrt{N}} \quad \text{avec} \quad \sigma(x) = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i-\overline{x})^2}.}$$

- σ est l'écart-type de la distribution des N mesures.
- Plus on réalise de mesures (N grand), plus l'incertitude-type est réduite.

Avec python, si X est une série de mesures sous la forme d'un `array` :
- la moyenne est donnée par `X.mean()` ;
- l'écart-type est donné[a] par `X.std(ddof = 1)`.

a. L'instruction ddof = 1 utilise $N-1$ au lieu de N dans le calcul de $\sigma(X)$.

2 — Évaluation de type B de l'incertitude type (mesure unique)

L'incertitude-type est donnée par

$$\boxed{u(x) = \frac{\Delta}{\sqrt{3}} \quad \text{avec} \quad \Delta : \ \text{demi-largeur de l'intervalle de mesure.}}$$

- La notice de l'appareil fournit une incertitude sous la forme $\pm\Delta$: on considère une loi de distribution uniforme avec un niveau de confiance égal à 100 %.
- Les multimètres indiquent l'incertitude-type sous la forme $n\%\text{L} + m\text{UR}$ (soit $n\%$ de la valeur lue, ajoutée de m fois la puissance de 10 du dernier chiffre affiché).

3 — Présentation du résultat : niveau de confiance

L'incertitude type $u(x)$ est donnée avec un **niveau de confiance de 68 %**.

- Ce résultat suppose une distribution des mesures qui suit la loi normale ; 68 % des valeurs mesurées sont comprises dans l'intervalle $[\overline{x}-u(x), \overline{x}+u(x)]$. Le résultat est noté $x = \overline{x} \pm u$.
- Le niveau de confiance donne la probabilité que la valeur vraie (inconnue) de la grandeur mesurée se trouve dans l'intervalle défini par l'incertitude de mesure.
- Le **niveau de confiance à 95 %** est donné par **l'incertitude élargie** : $\boxed{\Delta x = 2u(x)}$.

L'incertitude-type est donnée avec **un seul chiffre significatif**.
Le dernier chiffre significatif de x_{exp} doit avoir la même puissance de 10 que l'incertitude-type.
On arrondit toujours au chiffre supérieur.

Exemples : $\lambda = 589 \pm 2$ nm ; $T = 12{,}4 \pm 0{,}2$ s.

4 — Comparaison à une valeur de référence

On compare une valeur mesurée x_{mes}, d'incertitude-type $u(x_{\text{mes}})$ à une valeur de référence $x_{\text{réf}}$ d'incertitude-type $u(x_{\text{réf}})$ en calculant l'écart normalisé (appelé aussi z-score)

$$z = \frac{|x_{\text{mes}} - x_{\text{réf}}|}{\sqrt{u^2(x_{\text{mes}}) + u^2(x_{\text{réf}})}}.$$

$z \leqslant 2$ la mesure est jugée compatible avec la valeur de référence.

$z > 2$ la mesure est jugée incompatible avec la valeur de référence.

➤ Si l'incertitude-type de la valeur de référence est inconnue, on calcule $\boxed{z = \frac{|x_{\text{mes}} - x_{\text{réf}}|}{u(x_{\text{mes}})}}$.

5 — Incertitudes-type composées

5.1 Calcul direct à partir des valeurs mesurées

On souhaite déterminer l'incertitude-type sur une variable y_{calc}, calculée à partir d'une loi $y_{\text{calc}} = f(x_1, x_2, \ldots)$ faisant intervenir des grandeurs x_1, x_2, ... dont on connaît les incertitudes-type.

Le principe de la propagation des incertitudes est basé sur la relation $u(y_{\text{calc}}) = \sqrt{\left(\frac{\partial f}{\partial x_1}\right)^2 u^2(x_1) + \left(\frac{\partial f}{\partial x_2}\right)^2 u^2(x_2) + \cdots}$.

somme ou différence	$y_{\text{calc}} = x_1 \pm x_2$	$u(y_{\text{calc}}) = \sqrt{u^2(x_1) + u^2(x_2)}$
produit ou quotient	$y_{\text{calc}} = a x_1 x_2$ ou $y_{\text{calc}} = a\frac{x_1}{x_2}$	$\frac{u(y_{\text{calc}})}{\lvert x_{\text{calc}}\rvert} = \sqrt{\left(\frac{u(x_1)}{x_1}\right)^2 + \left(\frac{u(x_2)}{x_2}\right)^2}$

5.2 Calcul à partir d'une mesure, par la méthode de Monte-Carlo

On souhaite évaluer une grandeur $y = f(x_1, x_2)$ fonction des grandeurs mesurées x_1 et x_2, à partir d'un jeu de valeurs mesurées $x_{1,\text{mes}}$, $x_{2,\text{mes}}$, dont on connaît les intervalles de précision : $[x_{1,\text{mes}} - \Delta x_1, x_{1,\text{mes}} + \Delta x_1]$ et $[x_{2,\text{mes}} - \Delta x_2, x_{2,\text{mes}} + \Delta x_2]$. On utilise l'outil informatique pour générer aléatoirement $N \gg 1$ séries de grandeurs (x_1, x_2) à partir d'une loi de probabilité fournie On utilise généralement une loi de probabilité uniforme pour générer les variables x_i dont la précision est estimée (graduations de lecture par exemple). Un tirage de la variable x sur l'intervalle $[a, b]$ est réalisée par

```
np.random.uniform(a,b)
```

Dans le cas où l'incertitude-type `sigma` sur la grandeur x est fournie, on peut utiliser une loi distribution normale (gaussienne) autour de la valeur moyenne `x_m` par

```
np.random.normal(x_m,sigma)
```

On calcule ensuite $y = f(x_1, x_2)$ pour chaque couple de variables généré. On obtient alors une série de N valeurs de y (liste `Y`), dont on peut calculer la moyenne et l'incertitude type :

```
Y_m = Y.mean()
u_Y = Y.std(ddof=1)
```

Exemple

Considérons un filtre RC dont on veut mesurer la fréquence de coupure $f_0 = \frac{1}{2\pi RC}$.
On mesure les valeurs de R et de C avec un multimètre, dont la notice précise les incertitudes $u(R)$ et $u(C)$.
On mesure $R_{\text{mes}} = 10{,}47\ \text{k}\Omega$ avec $u(R) = 10\ \Omega$ et $C_{\text{mes}} = 95{,}8\ \text{nF}$ avec $u(C) = 0{,}4\ \text{nF}$.
Le principe de la méthode de Monte-Carlo pour estimer ω_0 et son incertitude-type consiste à :

1. tirer un grand nombre N de valeurs aléatoires de R à partir d'une loi normale de moyenne R_{mes} et d'écart-type $u(R)$;
2. tirer N valeurs aléatoires de C à partir d'une loi normale de moyenne C_{mes} et d'écart-type $u(C)$;
3. pour chaque tirage, calculer la valeur $f = \frac{1}{2\pi RC}$. On obtient donc une série de N valeurs de f prenant en compte les variabilités de R et C ;
4. la valeur moyenne donne l'estimation de la fréquence $f_0 = \langle f \rangle$; l'écart-type donne l'estimation de l'incertitude-type sur la fréquence.

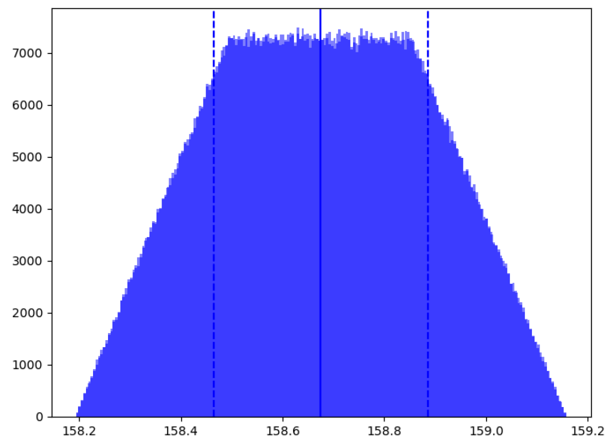
```python
import numpy as np

# Nombre de tirages aléatoires
N = int(1e6)

# valeurs de R et C avec leur incertitude-type
R = 10.47e3
u_R = 20
Delta_R = u_R/2
C = 95.8e-9
u_C = 0.4e-9
Delta_C = u_C/2

# tirage des valeurs de R et C
dist_R = np.random.uniform(R-Delta_R,R+Delta_R,N)
dist_C = np.random.uniform(C-Delta_C,C+Delta_C,N)

# génération de la liste des fréquences calculées pour chaque tirage
dist_F = 1/(2*np.pi*dist_R*dist_C)

# estimation de f_0 et de l'incertitude sur la fréquence

f_0 = dist_F.mean()
u_F = dist_F.std(ddof = 1)

print("f_0 = ",f_0,"u_F = ",u_F)
```

On obtient

```
f_0 = 158.67512119317848 u_F = 0.21045864243814957
```

On conserve le résultat $f_0 = 158{,}7 \pm 0{,}3$ Hz.
Affichons l'histogramme des valeurs de la fréquence obtenues, en indiquant la moyenne f_0 et les valeurs $f_0 \pm u(f)$:

```python
plt.hist(dist_F,bins= 'rice',color='blue',alpha = .5)
plt.axvline(f_0, color='blue')
plt.axvline(f_0-uF,linestyle='-', color='blue')
plt.axvline(f_0+uF,linestyle='-', color='blue')
plt.title( "$f_0$ = (m:.4 ± u:.1) Hz".format(m=f_0,u=uF))
plt.show()
```



6 — Régression linéaire

6.1 Réalisation avec python

À partir de deux listes X et Y, l'instruction np.polyfit(X,Y,1) permet de réaliser une régression linéaire, retournant un polynôme de degré 1.

p = np.polyfit(X,Y,1) effectue une régression linéaire sur les listes X et Y.
Elle retourne le couple (a, b) tel que la droite $y = ax + b$ passe « au plus près » des points (x_i, y_i).
On obtient a = p[0] et b = p[1]

➤ On peut évaluer le polynôme

6.2 Validation d'un modèle linéaire

Pour qu'un modèle linéaire soit valide il faut :

— que les points ne suivent pas une tendance clairement non linéaire ;

— que la droite de régression soit à une distance inférieure à deux fois la barre d'incertitude-type.

Soient X et Y deux listes de même longueur. Si pour chaque valeur y_i on dispose de l'incertitude Δy_i, on construit la liste Y_erreur, ce qui permet d'afficher les barres d'erreurs sur la représentation graphique avec

```
plt.errorbar(X, Y, yerr = Y\_erreur)
plt.show()
```

➤ Si on dispose des incertitudes Δx_i, on peut aussi afficher de même les barres d'erreur selon x en ajoutant à l'instruction python `xerr = X_erreur`

➤ Si l'incertitude u_Y est la même pour tous les points de mesures, on ajoute `yerr = u_Y`.

Tracé des résidus

Pour chaque point expérimental (x_i, y_i), le résidu est l'écart entre de point et le point correspondant de la droite de régression $y = ax + b$, soit $\Delta = y_i - (ax_i + b)$.

On peut définir le **résidu normalisé** $\mathrm{RN} = \dfrac{y_i - (ax_i + b)}{u(y_i)}$ où $u(y_i)$ est l'incertitude-type de la mesure. Les points tels que $-2 \leqslant \mathrm{RN} \leqslant 2$ sont compatibles la loi affine.