Fiche-méthode sur les incertitudes

I Mesure d'une grandeur physique

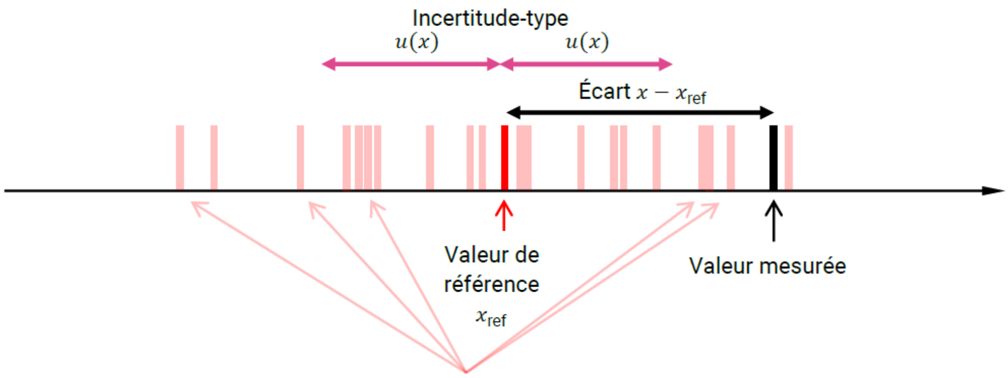
1.) Définitions

On veut mesurer la valeur x d'une grandeur physique X, appelée mesurande.
Mesurage : Ensemble des opérations ayant pour but de déterminer la valeur d'une grandeur.
Grandeur d'influence : Grandeur qui n'est pas le mesurande, mais qui a un effet sur le mesurage.

La répétition de la mesure conduit généralement à un ensemble de mesures différentes, ce qu'on appelle la variabilité de la mesure.
Cette variabilité peut provenir de plusieurs aspects :
- le choix de la méthode de mesure (règle ou pied à coulisse pour mesurer une longueur) ;
- les variations de l'environnement (évolution de la température en fonction de l'heure) ;
- les instruments de mesure ;
- l'expérimentateur (le plus important).

La variabilité de la mesure est quantifiée par l'incertitude-type u(x) (uncertainty = incertitude).
Le résultat sera alors donné sous la forme : $x \pm u(x)$
L'incertitude relative est donnée par $\frac{u(x)}{x}$

Comparaison à une valeur de référence x_{ref} :
Ecart normalisé (ou z-score) : $E_N = \frac{|x - x_{ref}|}{u(x)}$ Erreur relative $= \left|\frac{x - x_{ref}}{x_{ref}}\right|$ en %
Si l'écart normalisé est inférieur à 2, la mesure est conforme à la valeur de référence.

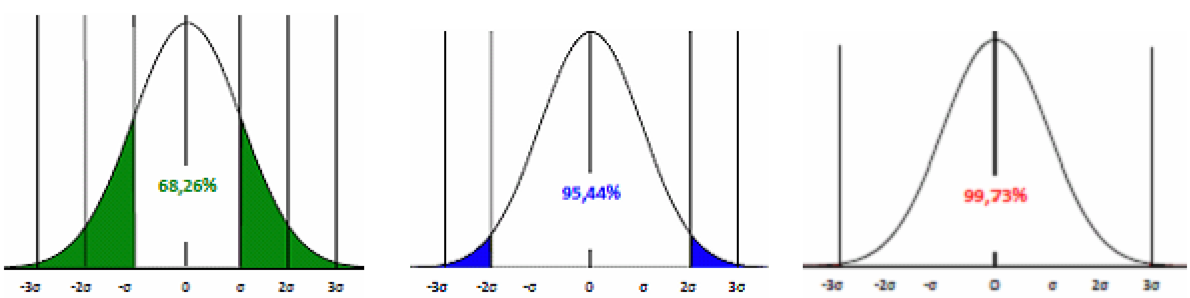


2.) Intervalle de confiance

Si on répète une mesure un très grand nombre de fois, la distribution de l'ensemble des mesures tend généralement vers une loi particulière appelée loi normale (ou gaussienne), que l'on caractérise par :

- sa valeur moyenne $\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$ (fonction mean de numpy sous python)

- son écart-type $\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2}$ ou déviation standard, noté aussi σ_n (fonction np.std(x) de numpy sous python)



- on a 68 % de chances que la valeur soit dans l'intervalle $\left[\bar{x}-\sigma, \bar{x}+\sigma\right]$
- on a 95 % de chances que la valeur soit dans l'intervalle $\left[\bar{x}-2\sigma, \bar{x}+2\sigma\right]$ On parle d'incertitude élargie.

II Evaluation d'une incertitude-type sur une mesure directe

1.) Evaluation de type A (évaluation statistique)

On effectue alors N mesures (N > 8) dans les mêmes conditions pour chiffrer la dispersion des valeurs autour de la valeur moyenne.

$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$ Moyenne expérimentale (ou estimée) (fonction np.mean(x) sous python)

$u(x) = \frac{s(x)}{\sqrt{N}}$ Incertitude-type avec $s(x) = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2}$ Ecart type expérimental (ou estimé)

(noté aussi σ_{n-1}) (fonction np.std(x, ddof=1) de numpy sous python) s(x) > σ

2.) Evaluation de type B

Si l'on n'observe pas la variabilité de la mesure, **on estime la plus petite plage dans laquelle l'expérimentateur est certain de trouver la valeur recherchée :**

On note $\bar{x}$ la valeur centrale de la plage et Δ sa demi-largeur, qui est la précision de la mesure.

- Pour un instrument gradué (par exemple, une règle), Δ correspond à une demi-graduation.
- Pour un instrument de mesure, elle est donnée par la notice constructeur et correspond à la tolérance.
- Pour une méthode de mesure, on la détermine expérimentalement (par exemple, la latitude de mise au point sur un banc d'optique).

$\bar{x}$ est la valeur centrale de la plage de mesure.
$u(x) = \frac{\Delta}{\sqrt{3}}$ où Δ est la précision de la mesure, ou demi-largeur.

Rq: Pour une incertitude élargie à 95 %, $x \in \left[\bar{x} - \frac{2\Delta}{\sqrt{3}}, \bar{x} + \frac{2\Delta}{\sqrt{3}}\right]$. Or 2/√3 = 1,1 ≈ 1 donc $x \in [\bar{x} - \Delta, \bar{x} + \Delta]$.

III Evaluation d'une incertitude-type composée sur une mesure indirecte.

La valeur de la grandeur z est obtenue à partir d'une formule faisant intervenir différentes grandeurs mesurées x et y.

Incertitude-type composée de type somme

Si z = x + y ou z = x – y, alors $u(z) = \sqrt{u(x)^2 + u(y)^2}$

Incertitude-type composée de type produit

Si $z = x \times y$ ou $z = \frac{x}{y}$, alors $\frac{u(z)}{z} = \sqrt{\left(\frac{u(x)}{x}\right)^2 + \left(\frac{u(y)}{y}\right)^2}$

Dans les cas plus compliqués, on utilise un algorithme basé sur la méthode de Monte-Carlo. On peut aussi utiliser un logiciel type Gum_MC.

IV Méthode Monte-Carlo.

1.) Principe

On cherche à estimer une grandeur y donnée par $y=f(x_1, x_2...x_N)$, f étant une fonction connue.
Chaque mesure x_i est caractérisée par sa valeur et son incertitude-type.

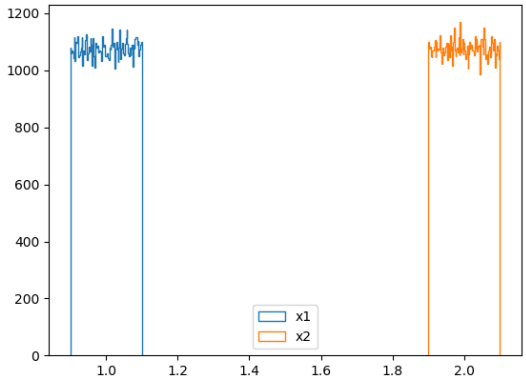
Pour estimer l'incertitude-type de y, il faut :
- fixer un nombre N de simulations à réaliser;
- pour k entre 1 et N, réaliser un tirage aléatoire pour chaque x_k , utiliser les valeurs de ce tirage et la fonction f pour calculer une valeur y_k et la sauvegarder.

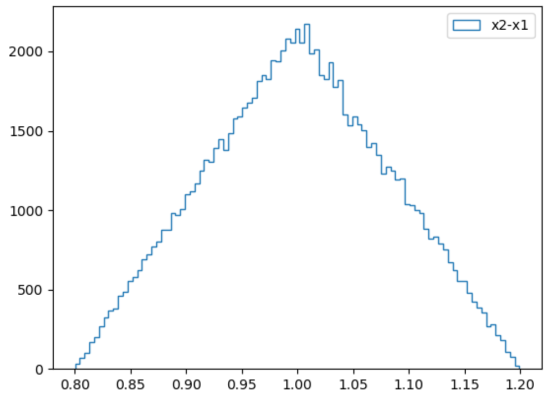
L'incertitude-type de y est l'écart-type estimé de la distribution des y_k.
La moyenne des y_k permet de retrouver la valeur de y.

On prend en général une loi de probabilité de distribution uniforme, plus rarement une loi normale.

2.) Incertitude sur une somme ou une différence

```python
import numpy as np
import matplotlib.pyplot as plt

##### Incertitude somme #####

# Entrez les deux valeurs

x1 =1
x2 = 2

# Entrez les demi-intervalle de précision des deux valeurs

Delta1=0.1
Delta2=0.1

# Entrez le nombre de simulation que vous voulez effectuer

N = 100000

# Simulation

resx=[]
resy=[]
resd=[]

for i in range(0,N):

    x=np.random.uniform(x1-Delta1,x1+Delta1)
    resx.append(x)

    y=np.random.uniform(x2-Delta2,x2+Delta2)
    resy.append(y)

    resd.append(y-x)

plt.figure(1)
plt.hist(resx,label='x1', bins='rice', histtype = 'step')
plt.hist(resy,label='x2', bins='rice', histtype = 'step')
plt.legend()

plt.figure(2)
plt.hist(resd,label='x2-x1', bins='rice', histtype = 'step')
plt.legend()

plt.show()

# Calcul et affichage moyenne et écart type

moy = np.mean(resd)
std = np.std(resd,ddof=1)

# Utilisation de la formule du programme

ux1=Delta1/np.sqrt(3)
ux2=Delta2/np.sqrt(3)
usomme=np.sqrt(ux1**2+ux2**2)

print('Moyenne =', moy)
print('Ecart type =', std)
print('incertitude de type somme avec la formule =',usomme)
```






```
# Moyenne = 0.999755106501
# Ecart type = 0.0817091559719
# incertitude de type somme avec la formule = 0.0816496580928
```

<u>3.) Incertitude sur un produit ou un rapport</u>

```
# Entrez les deux valeurs

lamb = 0.008
f = 40000

#  Entrez les demi-intervalle de précision des deux valeurs

Deltalamb=0.001
Deltaf=1000

# Entrez le nombre de simulation que vous voulez effectuer

N = 100000

# Simulation

resx=[]
resy=[]
resd=[]

for i in range(0,N):

    x=np.random.uniform(lamb-Deltalamb,lamb+Deltalamb)
    resx.append(x)

    y=np.random.uniform(f-Deltaf,f+Deltaf)
    resy.append(y)

    resd.append(x*y)

plt.figure(1)
plt.hist(resx,label='x1', bins='rice', histtype = 'step')
plt.legend()

plt.figure(2)
plt.hist(resy,label='x2', bins='rice', histtype = 'step')
plt.legend()

plt.figure(3)
plt.hist(resd,label='x1*x2', bins='rice', histtype = 'step')
plt.legend()

plt.show()

# Calcul et affichage moyenne et écart type

moy = np.mean(resd)
std = np.std(resd,ddof=1)

# Utilisation de la formule du programme
uf=Deltaf/np.sqrt(3)
ulamb=Deltalamb/np.sqrt(3)
uProd=f*lamb*np.sqrt((uf/f)**2+(ulamb/lamb)**2)


print('Moyenne =', moy)
print('Ecart type =', std)
print('incertitude de type produit =',uProd)

# Moyenne = 320.184768341
# Ecart type = 23.5700821406
# incertitude de type produit = 23.5513623102
```