

INCERTITUDES DE MESURE

1. DISPERSION DES RÉSULTATS D'UNE MESURE

1.1 Pourquoi donner des incertitudes ?

la **métrologie** (science de la mesure) rassemble les connaissances permettant de donner un sens à la mesure ; elle permet :

- >>> la caractérisation quantitative des propriétés d'un objet
- >>> la comparaison d'une même propriété pour deux objets
- >>> la validation ou non d'un modèle théorique

tout passe par l'attribution d'une valeur numérique aux grandeurs intervenant : **on mesure une grandeur X**

MAIS... la **dispersion** des résultats est inévitable

⇒ le résultat d'une mesure ne se résume pas à une seule valeur mais à un **ensemble de valeurs** numériques traduisant la **variabilité** de la mesure

la prise en compte des incertitudes permet de répondre à certaines questions :



cette pièce va-t-elle s'emboîter dans un mécanisme ?



ce cycliste s'est-il dopé ?



la théorie des ondes gravitationnelles est-elle vérifiée ?

sans évaluation des incertitudes, pas de prise de décision possible !

1.2 Incertitude-type

on réduit (perte d'information...) le résultat de la mesure à :

>>> la **valeur mesurée x**

>>> l'**incertitude-type**, notée $u(X)$, qui est une évaluation de l'écart

type σ de la distribution, défini ainsi : $\sigma = \sqrt{(x - \bar{x})^2}$

σ est l'analogie de la valeur efficace d'un signal alternatif, d'autant plus grande que le signal s'écarte de 0

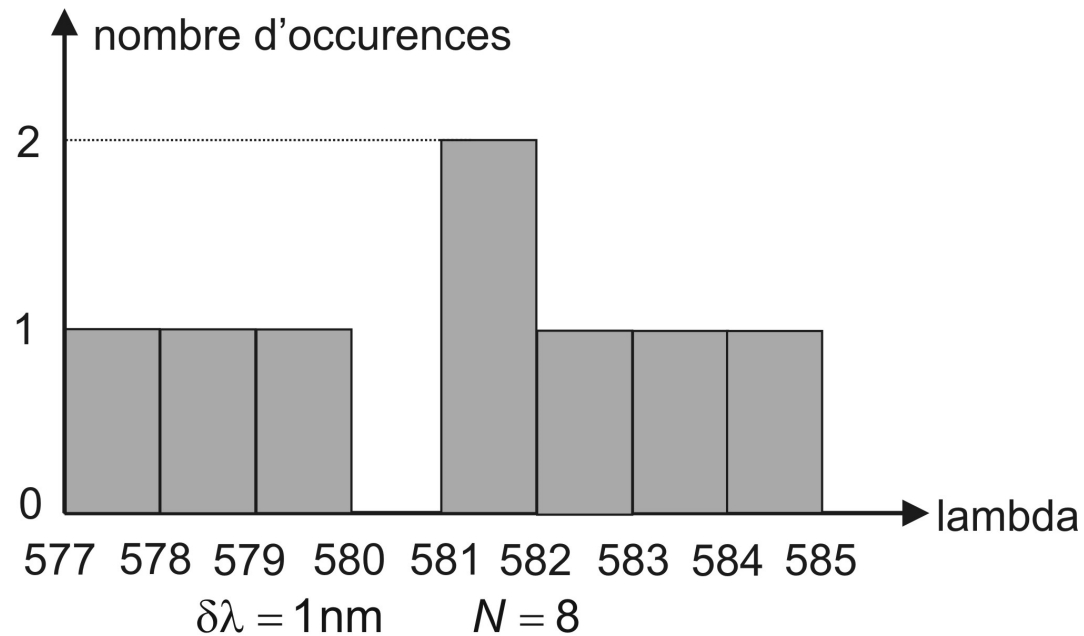
incertitude élargie et intervalle de confiance ne sont pas au programme

1.3 Densité de probabilité

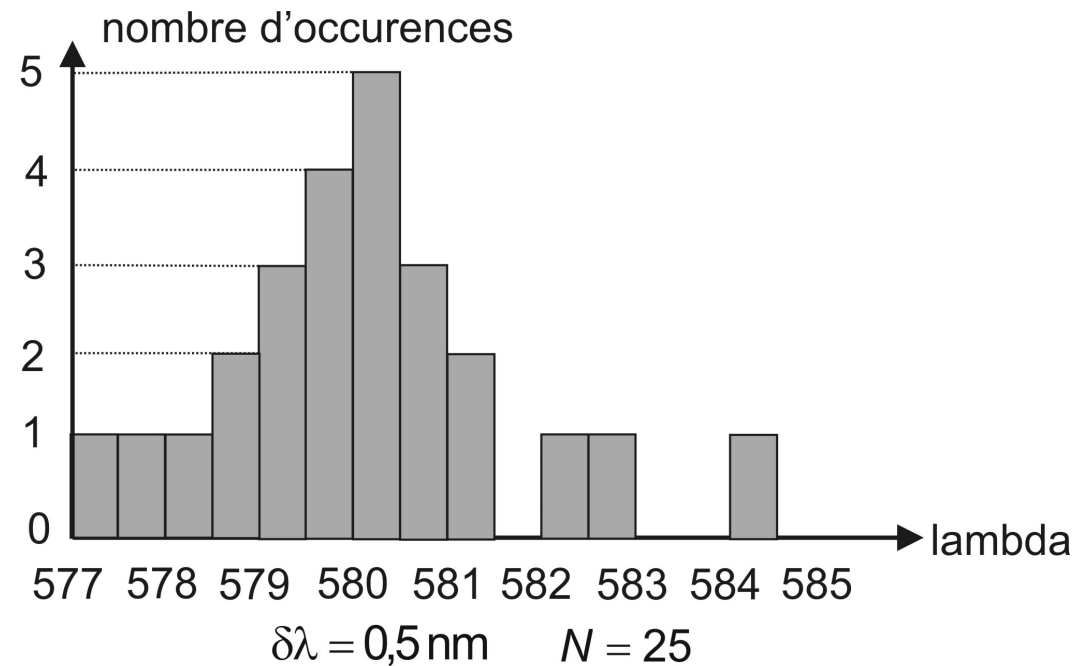
exemple : $N = 8$ groupes de TP mesurent une longueur d'onde :

n° groupe de TP	1	2	3	4	5	6	7	8
lambda (nm)	581,1	583,1	577,6	581,6	579,2	584,7	582,8	578,9

histogramme : on reporte le nombre de mesures donnant un résultat compris entre x_i et $x_i + \delta x$



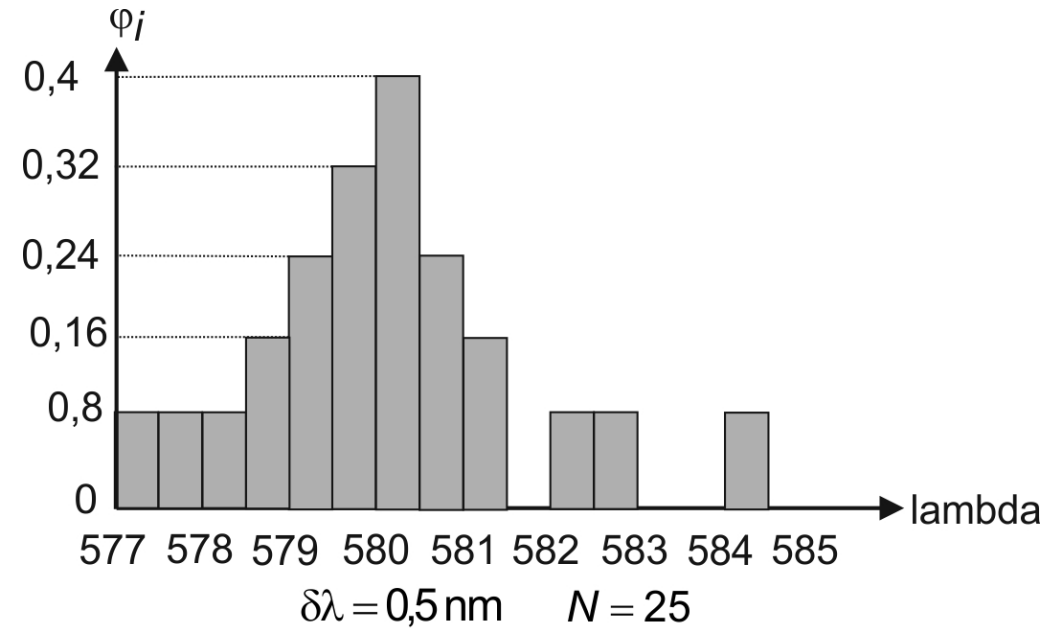
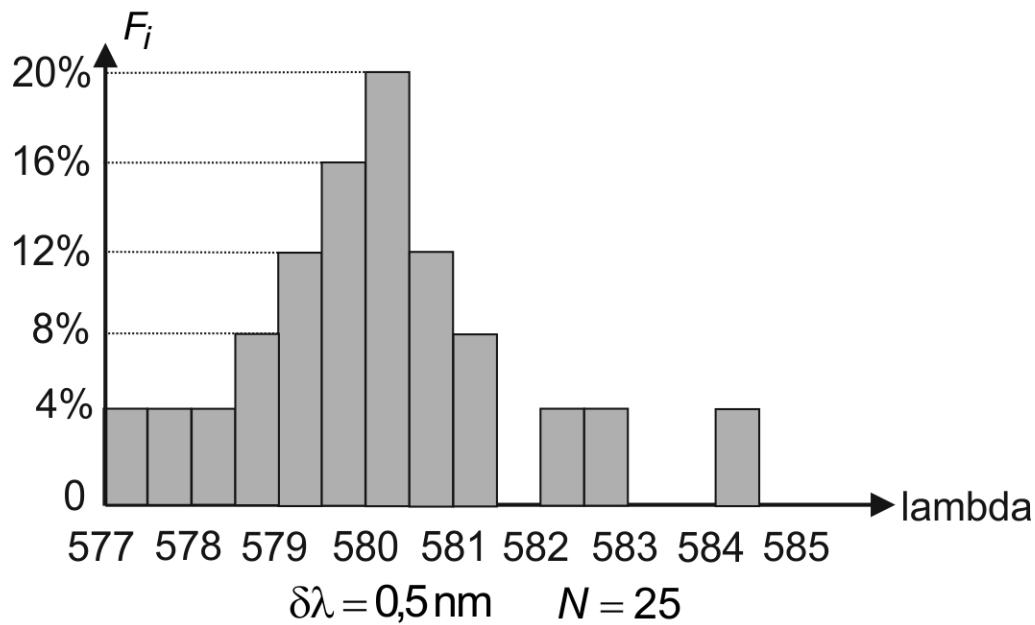
recommençons avec 25 groupes et des intervalles plus fins :



fréquence d'apparition de la valeur x_i (à δx près) :

$$F_i = \frac{n_i = \text{nombre de mesures dans } [x_i, x_i + \delta x]}{N}$$

$$\Rightarrow \sum_{i=1}^N F_i = 1 \quad \text{et} \quad \bar{x} = \frac{\sum_{i=1}^N n_i x_i}{N} = \sum_{i=1}^N F_i x_i \quad (\text{valeur moyenne})$$



si $\phi_i = \frac{F_i}{\delta x}$, la fréquence d'apparition de x_i est $F_i = \phi_i \cdot \delta x$, soit l'aire du rectangle associé

F_i est la probabilité d'obtenir x_i à δx près dans le cas d'un très grand nombre de mesures

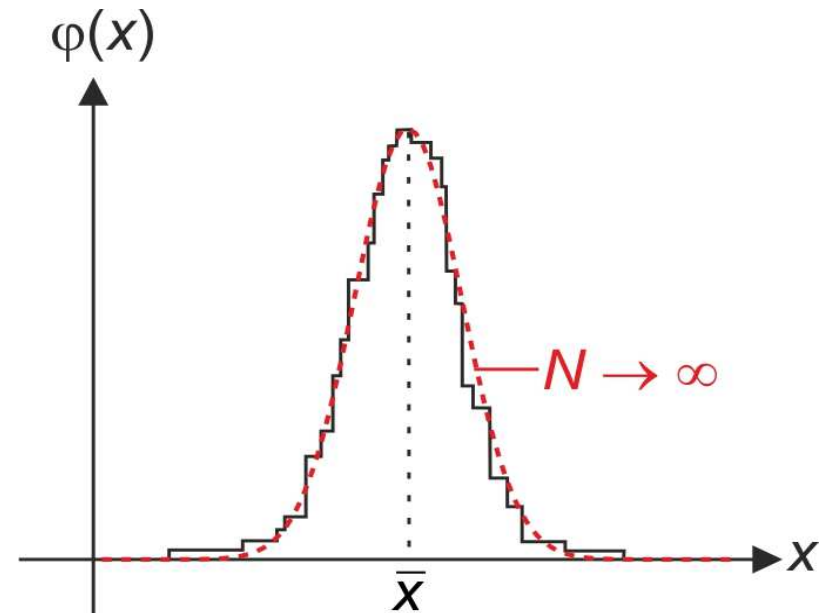
distribution limite :

$$\delta x \rightarrow 0$$

$$N \rightarrow \infty$$

l'aire sous la courbe correspondant à un rectangle de largeur infinitésimale dx et de hauteur $\varphi(x)$: probabilité que le résultat de la mesure soit compris entre x et $x + dx$

$\varphi(x)$ est une densité de probabilité



$dp = \varphi(x)dx$ est la probabilité que le résultat de la mesure de la

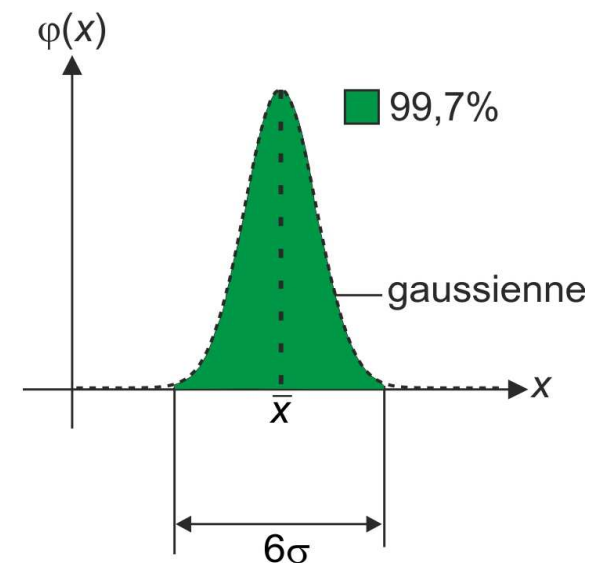
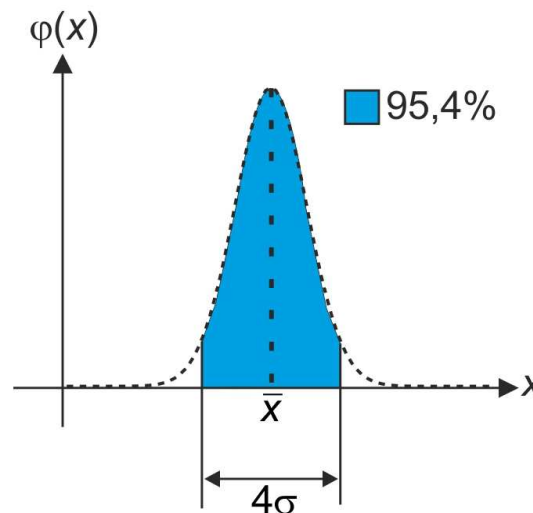
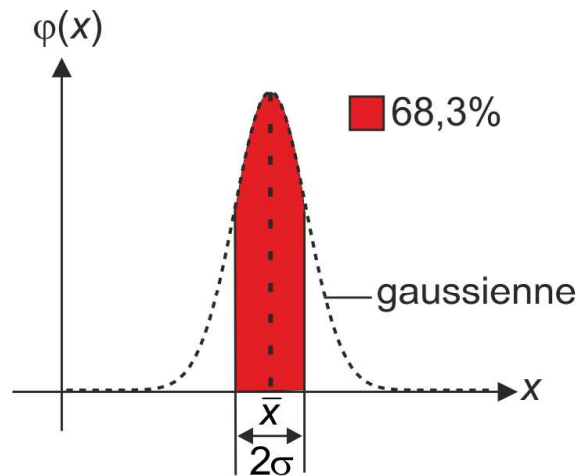
grandeur X se trouve dans l'intervalle $[x, x + dx] \Rightarrow \int_{-\infty}^{+\infty} \varphi(x)dx = 1$

$\bar{x} = \int_{-\infty}^{+\infty} x \cdot \varphi(x)dx$ valeur moyenne de X : c'est le « résultat » de la mesure

$\sigma = \sqrt{\overline{(x - \bar{x})^2}}$ est l'écart-type de la distribution, $V = \sigma^2 = \overline{(x - \bar{x})^2}$ sa variance

loi normale (gaussienne)

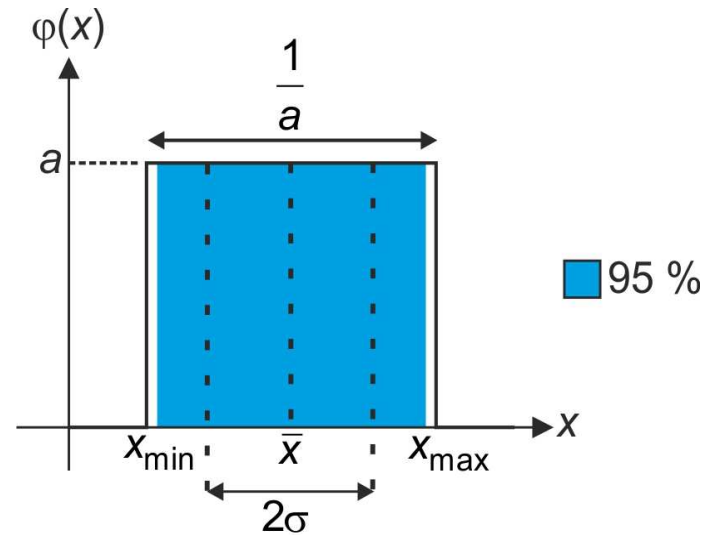
$$\varphi(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\bar{x})^2}{2\sigma^2}}$$



on a une probabilité de **95%** que le résultat de la mesure soit compris dans l'intervalle $[\bar{x} - 2\sigma, \bar{x} + 2\sigma]$

théorème central limite : si l'incertitude sur X est la somme de M contributions aléatoires indépendantes de même densité de probabilité, la densité de probabilité de X se rapproche de la loi normale lorsque M augmente

loi uniforme (rectangulaire)



si Δ est la **demi**-largeur de l'intervalle $[x_{\min}, x_{\max}]$: $x_{\max} - x_{\min} = 2\Delta$

calcul :

$$\sigma = \frac{\Delta}{\sqrt{3}}$$

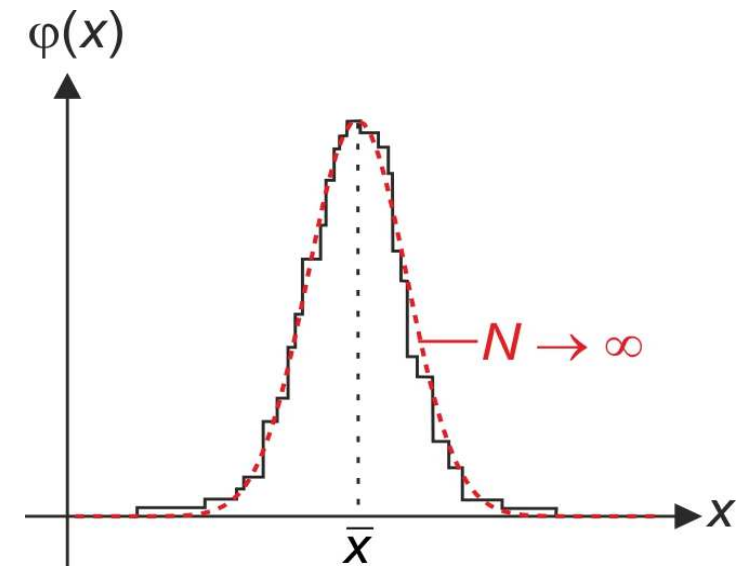
2. ÉVALUATION DES INCERTITUDES DE TYPE A

2.1 Traitement d'une série de N mesures indépendantes de la grandeur X

N est **fini** $\Rightarrow \bar{x}$ reste inconnue : différentes séries de N mesures de X donnent des valeurs moyennes x^* différentes (réparties autour de la valeur cherchée $\bar{x}$) et des écarts-type différents (répartis autour de σ)

cf. propagation des incertitudes : les carrés des écarts-types se somment

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i \Rightarrow \sigma_{\bar{x}} = \frac{1}{N} \sqrt{\sum_{i=1}^N \sigma^2} = \frac{\sigma \sqrt{N}}{N} = \frac{\sigma}{\sqrt{N}}$$



l'écart-type $\sigma_{\bar{x}}$ sur la moyenne est lié à l'écart-type σ de la distribution par la formule $\sigma_{\bar{x}} = \frac{\sigma}{\sqrt{N}}$

on retrouve ainsi que $\sigma_{\bar{x}} \xrightarrow{N \rightarrow \infty} 0$: on n'aurait aucune incertitude sur la valeur recherchée $\bar{x}$ avec un nombre infini de mesures...

l'incertitude décroît donc « lentement » : en $\frac{1}{\sqrt{N}}$

en pratique : « petit » nombre de mesures (de l'ordre de la dizaine)

dans le cas où l'on a réalisé N mesures de la même grandeur donnant des résultats x_i , $i \in \llbracket 1, N \rrbracket$, on montre que les meilleurs **estimateurs** de $\bar{x}$ et de σ sont respectivement :

$$x^* = \frac{1}{N} \sum_{i=1}^N x_i \quad \sigma^* = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - x^*)^2} = \sqrt{\frac{1}{N-1} \left[\sum_{i=1}^N x_i^2 - \frac{1}{N} \left(\sum_{i=1}^N x_i \right)^2 \right]}$$

écart-type *expérimental de la distribution* (nécessite au moins deux mesures, infini si $N = 1$)

x^* et σ^* s'identifient bien à $\bar{x}$ et σ quand $N \rightarrow \infty$

le résultat de la mesure est x^* qui estime $\bar{x}$

l'incertitude-type sur la mesure (sur la moyenne) est $u(\bar{x}) = \frac{\sigma^*}{\sqrt{N}}$

exemple : le dosage en ions Fe^{2+} d'une même solution est réalisé simultanément par 9 groupes d'élèves, avec les résultats suivants (en $10^{-2} \cdot \text{mol} \cdot \text{L}^{-1}$) pour $c = [\text{Fe}^{2+}]$

groupe	1	2	3	4	5	6	7	8	9
résultat	1,024	1,028	0,975	1,031	0,854	1,100	0,921	0,945	0,821

$$c^* = \frac{1}{9} \sum_{i=1}^9 c_i = 0,9666 \cdot 10^{-2} \text{ mol} \cdot \text{L}^{-1}$$

$$\sigma^* = \sqrt{\frac{1}{8} \left[\sum_{i=1}^9 c_i^2 - \frac{1}{9} \left(\sum_{i=1}^9 c_i \right)^2 \right]} = 0,0905 \cdot 10^{-2} \text{ mol} \cdot \text{L}^{-1}$$

$$u(c^*) = \frac{\sigma^*}{\sqrt{9}} = 0,0302 \cdot 10^{-2} \text{ mol} \cdot \text{L}^{-1}$$

le résultat de la mesure est : $c = 0,967 \cdot 10^{-2} \text{ mol} \cdot \text{L}^{-1}$

$$u(c) = 0,030 \cdot 10^{-2} \text{ mol} \cdot \text{L}^{-1}$$

2.2 Python : moyenne et incertitude-type expérimentaux d'une liste de mesures

```
import numpy as np

def calcul_mu_sigma(L): # renvoie la moyenne et l'écart-type
    sur la moyenne d'un tableau 1D de valeurs (ou d'une liste)

    N = len(L)
    somme = 0
    for mes in L:
        somme += mes
    mu = somme/N # moyenne
    somme = 0
    for mes in L:
        somme += (mes-mu)**2
    sigma_exp = (somme/(N-1))**0.5 # écart-type de la
distribution
    sigma_moy = sigma_exp/(N)**0.5 # écart-type de la moyenne
    return (mu,sigma_moy)

>>> concentrations = np.array([1.024,1.028,0.975,1.031,
0.854,1.100,0.921,0.945,0.821])*1E-2
>>> calcul_mu_sigma(concentrations)
(0.009665555555555557, 0.0003016211342760572)
```



avec les fonctions Python :



```
>>> np.mean(concentrations)
# valeur moyenne d'un tableau ou d'une liste de N valeurs

0.0096655555555555574
```

```
>>> np.std(concentrations, ddof = 1)
# écart-type du tableau de valeurs « standard deviation »
# ddof : « delta degrees of freedom », indique qu'il faut
diviser par N-ddof et pas N... »
```

```
0.00090486340282817154
```

```
>>> np.std(concentrations, ddof = 1) / (len(conc)) ** 0.5
# écart-type sur la mesure de concentration (sur la
moyenne)
```

```
0.0003016211342760572
```

3. ÉVALUATION DES INCERTITUDES DE TYPE B

3.1 Analyse des causes d'incertitude sur une mesure unique

une **unique** mesure de X fournit la valeur numérique x

⇒ on s'appuie sur toutes les informations disponibles, et sur le bon sens, pour déterminer l'incertitude-type $u(X)$ de la mesure

(1) le constructeur d'un appareil de mesure (multimètre, pipette...) a effectué une étude statistique et fournit une *incertitude-type* (sans préciser le type de distribution)
⇒ on considère que la distribution est gaussienne

(2) le plus fréquent : le constructeur fournit une incertitude « constructeur » : Δ_{constr} sans autre information (souvent notée $\pm\Delta_{\text{constr}}$ et appelée « tolérance »)

Le résultat de la mesure est x

on suppose alors que la distribution est *rectangulaire uniforme* :

équiprobabilité de trouver la valeur de X dans $[x - \Delta_{\text{constr}}, x + \Delta_{\text{constr}}]$, et une probabilité nulle hors de cet intervalle

on en déduit alors que l'incertitude-type vaut : $u(X) = \frac{\Delta_{\text{constr}}}{\sqrt{3}}$

(3) si on ne peut qu'encadrer les valeurs possibles de la mesure :

$$X \in [X_{\min}, X_{\max}]$$

on suppose également que la densité de probabilité est rectangulaire

$$\Rightarrow \begin{aligned} X^* &= \frac{X_{\max} + X_{\min}}{2} \text{ est la valeur mesurée} \\ u(X) &= \frac{X_{\max} - X_{\min}}{2\sqrt{3}} \text{ est l'incertitude-type de la mesure} \end{aligned}$$

par exemple :

>>> une plage de positions d'une lentille sur un banc optique telle que l'image d'un objet sur un écran à une distance fixe de l'objet soit vue nette

>>> une graduation pour la valeur lue sur une règle

>>> l'épaisseur de la trace sur un écran d'oscilloscope analogique

3.2 Exemples

exemple 1 : on lit sur un conducteur ohmique la couleur des bagues : marron (1), noir (0), rouge (2), or (5%)

$$R = (10 \times 1 + 0) \cdot 10^2 \, \Omega = 1 \, \text{k}\Omega \quad u(R) = \frac{1}{\sqrt{3}} \times \frac{5}{100} \times 1000 = 28,9 \, \Omega$$

exemple 2 : un thermomètre indique $121,84 \, ^\circ \text{C}$: sa *résolution* en température est de $0,01 \, ^\circ \text{C}$ (les indications en $^\circ \text{C}$ se font avec deux chiffres après la virgule)

pas de $\Delta_{\text{constr}} \Rightarrow$ on considère que $T_{\text{min}} = 121,835 \, ^\circ \text{C}$ et $T_{\text{max}} = 121,845 \, ^\circ \text{C}$

$$\Rightarrow u(T) = \frac{1}{2\sqrt{3}} \times 0,01 = 2,89 \cdot 10^{-3} \, ^\circ \text{C}$$

exemple 3 : on lit sur un vernier un angle compris entre $152^\circ 27'$ et $152^\circ 29'$

$$\Rightarrow \theta = 152^\circ 28' \quad u(\theta) = \frac{1}{\sqrt{3}} = 0,58'$$

exemple 4 : focométrie: méthode d'autocollimation ; l'image est nette pour une distance entre l'objet et la lentille convergente dans l'intervalle $[9,8 \, \text{cm} ; 11,0 \, \text{cm}]$

$$f' = \frac{9,8 + 11,0}{2} = 10,40 \, \text{cm} \quad u(f') = \frac{11,0 - 9,8}{2\sqrt{3}} = 0,35 \, \text{cm}$$

exemple 5 : on mesure avec un multimètre (dont on donne ci-contre un extrait de la notice) une tension efficace W de **1.54297** V pour une fréquence de 7 kHz sur le calibre 2 V

d : un digit, ici **0.00000** !

AC VOLTS (True RMS)				
Range	Accuracy(±)			
	45Hz - 5kHz	5kHz - 10kHz	10kHz-20kHz	20kHz-50kHz
200mV	0.2% + 150d	0.2% + 250d	3% + 500d	-
2V	0.2% + 150d	0.2% + 250d	0.5% + 300d	3% + 1500d
20V	0.2% + 150d	0.2% + 250d	0.5% + 300d	2% + 1000d
200V	0.2% + 150d	0.2% + 250d	0.5% + 500d	-
750V	0.2% + 150d	-	-	-

$$\Delta_{\text{constr}} = \frac{0,2}{100} \times 1,54297 + 250 \times 0,00001 = 0,00309 + 0,00250 = 0,00559 \text{ V}$$

$$u(W) = \frac{0,00559}{\sqrt{3}} = 3,23 \cdot 10^{-3} \text{ V}$$

4. PROPAGATION DES INCERTITUDES

X n'est pas directement accessible par la mesure mais lié à des mesures *indépendantes* de *deux grandeurs elles-mêmes indépendantes* X_1 et X_2

valeurs mesurées	$\bar{x}_1$	$\bar{x}_2$
incertitudes-type	$u(X_1)$	$u(X_2)$

$$X = f(X_1, X_2)$$

4.1 Méthode analytique

valable si **les incertitudes sont *suffisamment petites***

$$x = f(x_1, x_2) \simeq f(\bar{x}_1, \bar{x}_2) + (x_1 - \bar{x}_1) \frac{\partial f}{\partial X_1}(\bar{x}_1, \bar{x}_2) + (x_2 - \bar{x}_2) \frac{\partial f}{\partial X_2}(\bar{x}_1, \bar{x}_2)$$

$$\Rightarrow \bar{x} = f(\bar{x}_1, \bar{x}_2) + \overline{(x_1 - \bar{x}_1)} \frac{\partial f}{\partial X_1}(\bar{x}_1, \bar{x}_2) + \overline{(x_2 - \bar{x}_2)} \frac{\partial f}{\partial X_2}(\bar{x}_1, \bar{x}_2) = f(\bar{x}_1, \bar{x}_2)$$

$$\bar{x} = f(\bar{x}_1, \bar{x}_2)$$

$$u^2(X) = \overline{(x - \bar{x})^2} = \overline{[x - f(\bar{x}_1, \bar{x}_2)]^2}$$

$$u^2(X) = \overline{(x_1 - \bar{x}_1)^2} \left[\frac{\partial f}{\partial X_1}(\bar{x}_1, \bar{x}_2) \right]^2 + \overline{(x_2 - \bar{x}_2)^2} \left[\frac{\partial f}{\partial X_2}(\bar{x}_1, \bar{x}_2) \right]^2$$

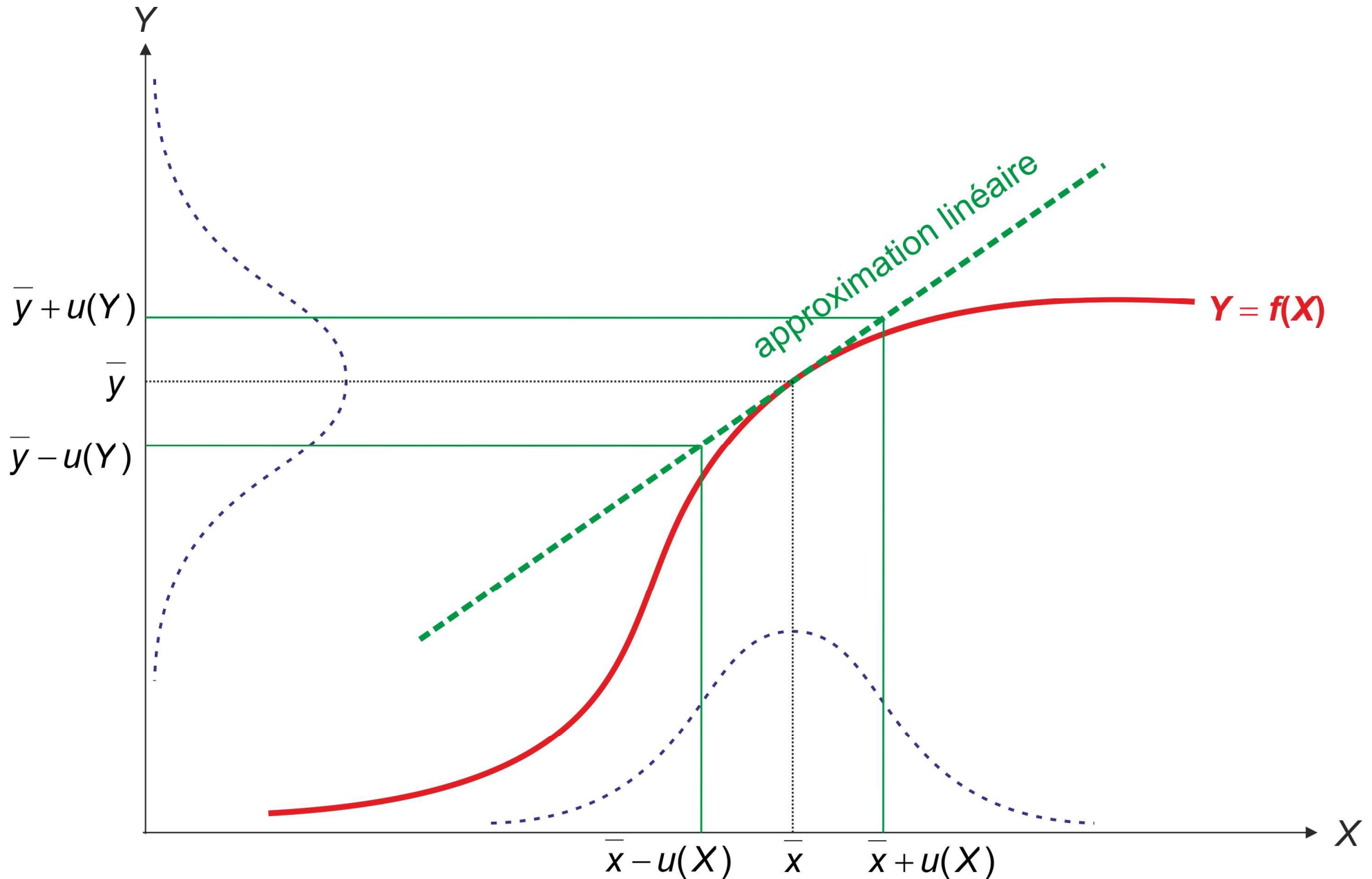
$$+ 2 \underbrace{\overline{(x_1 - \bar{x}_1)(x_2 - \bar{x}_2)}}_{0 \text{ (corrélation nulle)}} \frac{\partial f}{\partial X_1}(\bar{x}_1, \bar{x}_2) \frac{\partial f}{\partial X_2}(\bar{x}_1, \bar{x}_2)$$

$$\Rightarrow u(X) = \sqrt{u^2(X_1) \left[\frac{\partial f}{\partial X_1}(\bar{x}_1, \bar{x}_2) \right]^2 + u^2(X_2) \left[\frac{\partial f}{\partial X_2}(\bar{x}_1, \bar{x}_2) \right]^2}$$

$$u(X) = \sqrt{\sum_k \left[\frac{\partial f}{\partial X_k} \right]^2 u^2(X_k)} \quad \text{pour un nombre quelconque de variables}$$

$$X = X_1 \pm X_2 \Rightarrow u(X) = \sqrt{u^2(X_1) + u^2(X_2)}$$

$$X = X_1^\alpha X_2^\beta \Rightarrow \frac{u(X)}{|X|} = \sqrt{\alpha^2 \left(\frac{u(X_1)}{X_1} \right)^2 + \beta^2 \left(\frac{u(X_2)}{X_2} \right)^2}$$



l'approximation linéaire peut ne pas être vérifiée si $u(X)$ est trop grand...

4.2 Simulation par la méthode de Monte-Carlo

on a toujours $X = f(X_1, X_2)$

X_1 et X_2 indépendantes

valeurs mesurées

$\overline{x_1}$

$\overline{x_2}$

incertitudes-type

$u(X_1)$

$u(X_2)$

distribution uniforme ou gaussienne

principe :

- 1) on génère *aléatoirement* $N \gg 1$ mesures de X_1 et X_2 notées x_{1i} et x_{2i} pour $i \in \llbracket 1, N \rrbracket$
 >>> de valeurs moyennes $\overline{x_1}$ et $\overline{x_2}$
 >>> d'écart-type $u(X_1)$ et $u(X_2)$
 >>> distribuées selon une loi uniforme ou gaussienne
- 2) on calcule $x_i = f(x_{1i}, x_{2i})$ pour $i \in \llbracket 1, N \rrbracket$
- 3) on en déduit la valeur moyenne $\overline{x}$ (le « résultat ») et l'écart-type $u(X)$ des x_i (son incertitude-type)
- 4) on peut tracer l'histogramme des valeurs x_i (*a priori* ni uniforme ni gaussien)

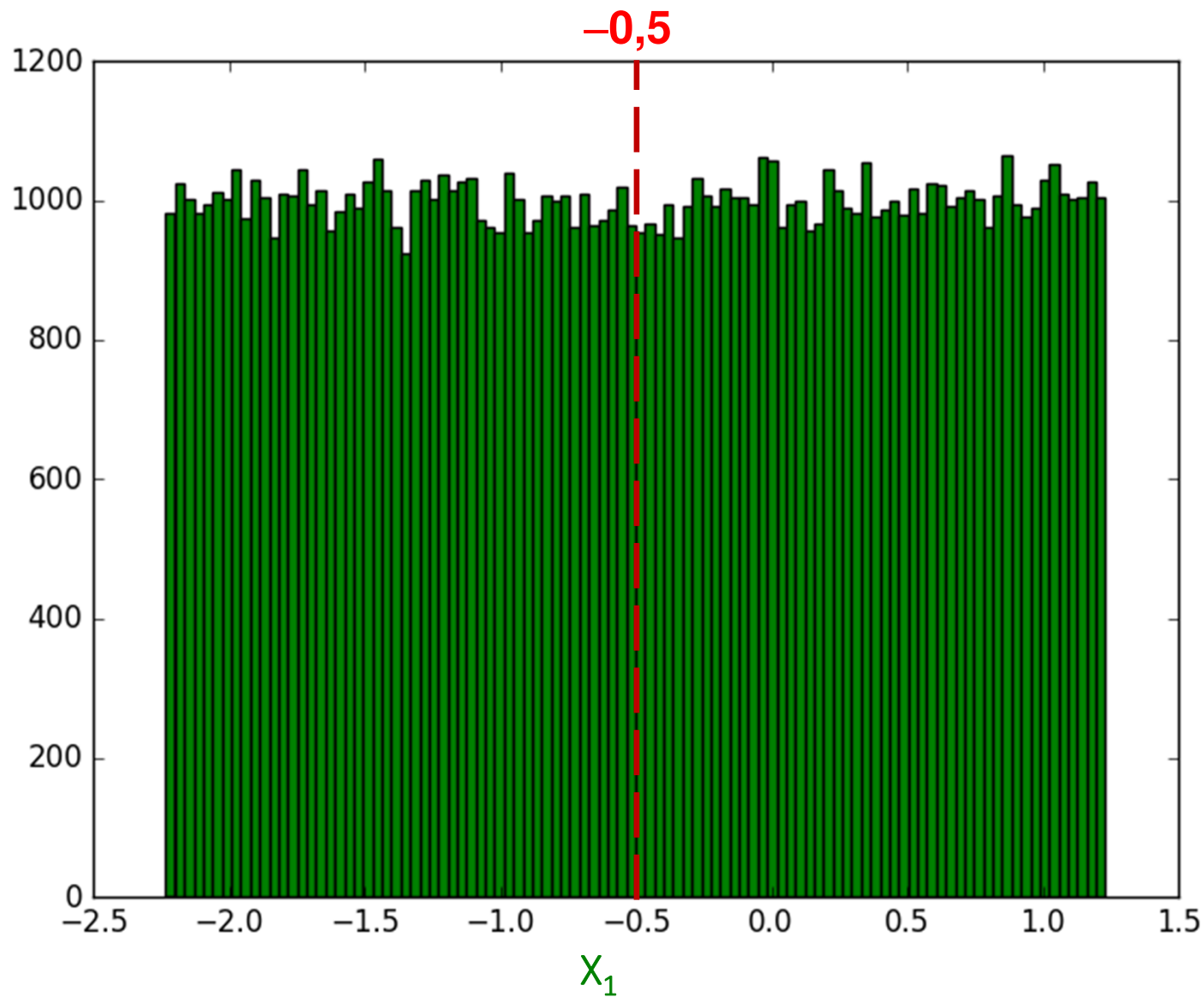
génération d'une distribution **uniforme** sous Python :

```
import numpy as np
import numpy.random as rd # bibliothèque random de numpy
from matplotlib import pyplot as plt # graphiques

moy1,sigma1 = -0.5,1 # valeur moyenne et écart-type de X1
xmin1,xmax1 = moy1-sigma1*3**.5,moy1+sigma1*3**.5 # on calcule
les bornes de l'intervalle (facteur racine de 3 entre le demi-
intervalle et l'écart-type)
N = 100000
X1 = rd.uniform(xmin1,xmax1,N) # tire aléatoirement N nombres
autour de moy avec un écart-type sigma (renvoie un tableau 1D)
plt.hist(X1,100,color='g') # le nombre d'intervalles de
l'histogramme est fixé à 100
plt.show()

>>> np.mean(X1) # moyenne du tableau de valeurs
-0.498551962647#on vérifie que la valeur moyenne est bien
proche de -0.5
>>> np.std(X1, 0, ddof = 1) # écart-type du tableau de valeurs
0.999595207002 # on vérifie que l'écart-type est bien proche
de 1
```





100 000 tirages avec une loi uniforme de valeur moyenne $-0,5$ et d'écart-type 1

EXERCICE :

soient trois variables aléatoires X_1 , X_2 et X_3 de distributions rectangulaires, de valeurs moyennes respectives $-0,5$, 0 et $0,5$, et de même écart-type $\sigma = 1$

déterminer la valeur moyenne et l'écart-type des distribution suivantes :

$$>>> X = X_1 + X_3$$

$$>>> X = X_1 + X_2 + X_3$$

comparer à la méthode analytique

tracer à chaque fois les histogrammes de X et commenter

SOLUTION :

```

N = 100000
moy2,sigma2 = 0,1 # valeur moyenne et écart-type de X2
xmin2,xmax2 = moy2-sigma2*3**.5,moy2+sigma2*3**.5
X2 = rd.uniform(xmin2,xmax2,N)
moy3,sigma3 = 0.5,1 # valeur moyenne et écart-type de X3
xmin3,xmax3 = moy3-sigma3*3**.5,moy3+sigma3*3**.5
X3 = rd.uniform(xmin3,xmax3,N)

X = X1 + X3
plt.figure()
plt.hist(X,100,color='blue')
plt.show()

>>> np.mean(X)
-0.0001412135666561349 # valeur attendue : 0
>>> np.std(X, ddof = 1)
1.4218455694130963 # valeur attendue : racine de 2

```

```
X = X1 + X2 + X3
plt.figure()
plt.hist(X, 100, color='red')
plt.show()
```

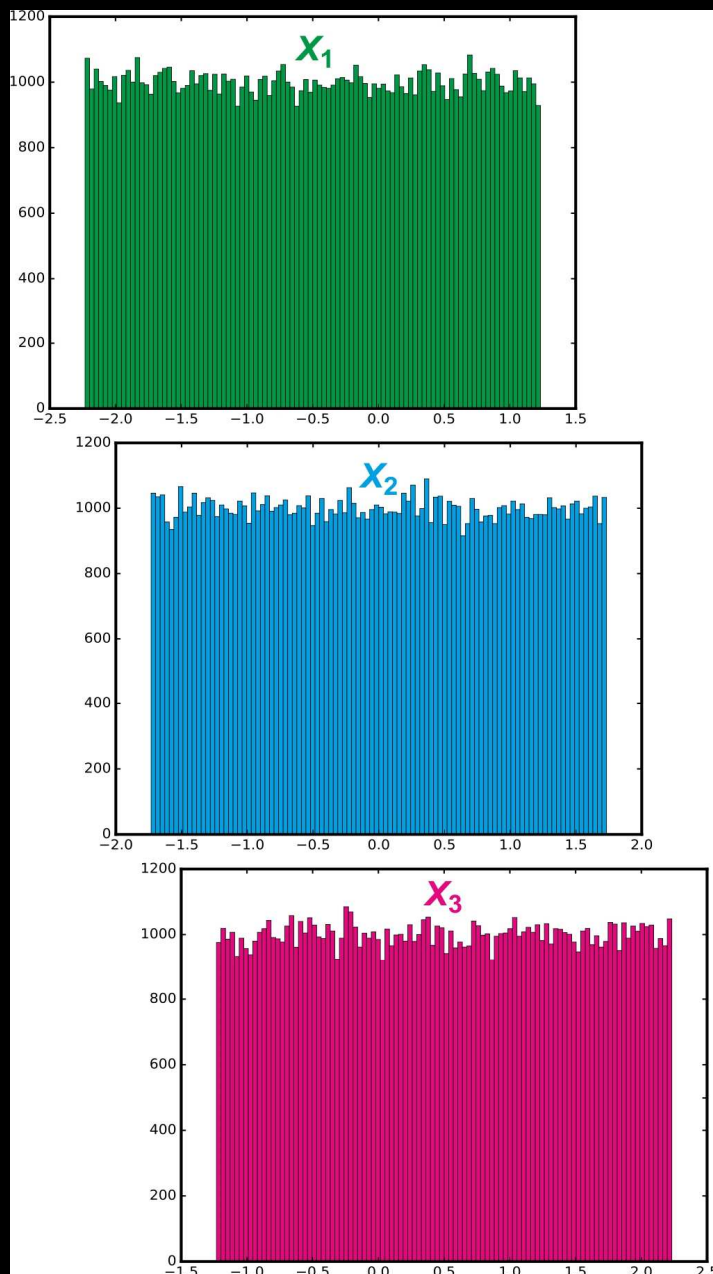
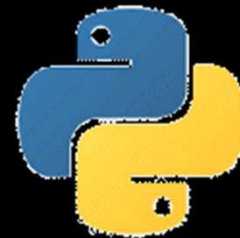
```
>>> np.mean(X)
-0.0018297122263692652 # valeur attendue : 0
>>> np.std(X, ddof = 1)
1.7419180911613585 valeur attendue : racine de 3
```

méthode analytique :

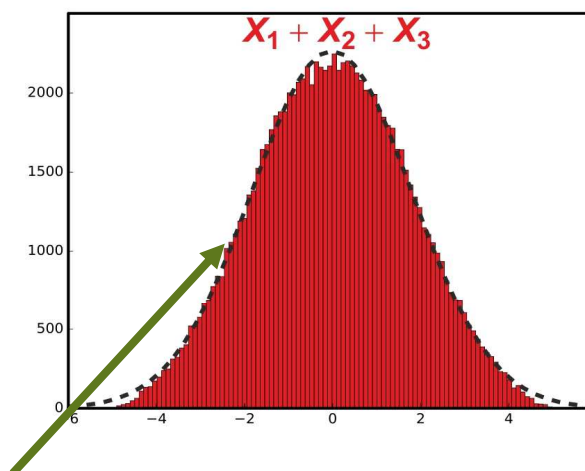
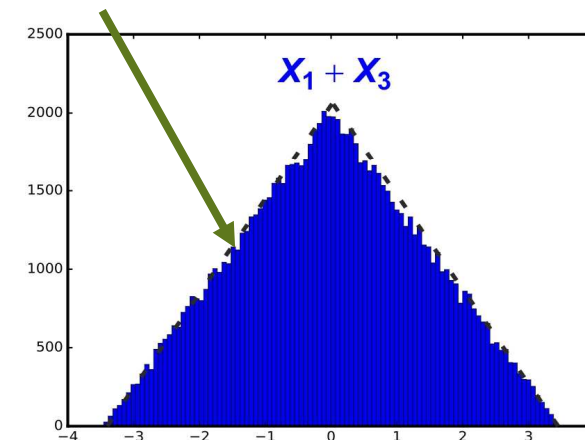
$$X = X_1 + X_3 \Rightarrow \begin{cases} \bar{x} = \bar{x}_1 + \bar{x}_3 = -0,5 + 0,5 = 0 \\ u(X) = \sqrt{u^2(X_1) + u^2(X_3)} = \sqrt{2} \simeq 1,414 \end{cases}$$

$$X = X_1 + X_2 + X_3 \Rightarrow \begin{cases} \bar{x} = \bar{x}_1 + \bar{x}_2 + \bar{x}_3 = -0,5 + 0 + 0,5 = 0 \\ u(X) = \sqrt{u^2(X_1) + u^2(X_2) + u^2(X_2)} = \sqrt{3} \simeq 1,732 \end{cases}$$

histogrammes de $X_1 + X_3$ et $X_1 + X_2 + X_3$



$\approx$ triangle



$\approx$ gaussienne (théorème centrale limite)

génération d'une distribution gaussienne sous Python :

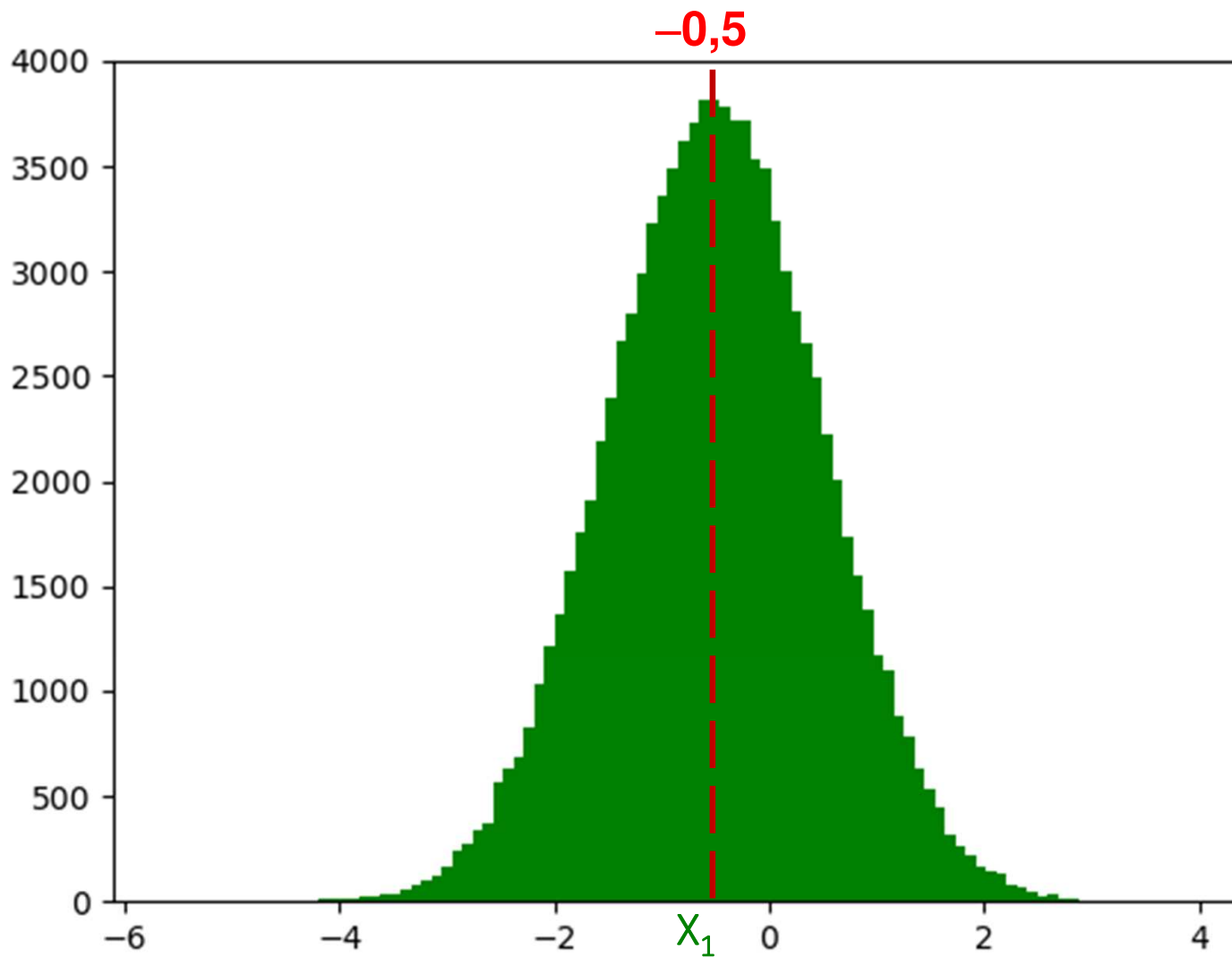
```
import numpy as np
import numpy.random as rd # bibliothèque random de numpy
from matplotlib import pyplot as plt # graphiques

moy1,sigma1 = -0.5,1 # valeur moyenne et écart-type de X1
N = 100000
X1 = rd.normal(moy1,sigma1,N) # tire aléatoirement N nombres
autour de moy avec un écart-type sigma (renvoie un tableau 1D)

plt.hist(X1,100,color='g') # le nombre d'intervalles de
l'histogramme est fixé à 100
plt.show()

>>> np.mean(X1) # moyenne du tableau de valeurs
-0.5015615042397337#on vérifie que la valeur moyenne est bien
proche de -0.5
>>> np.std(X1, 0, ddof = 1) # écart-type du tableau de valeurs
1.0018271205185607 #on vérifie que l'écart-type est bien
proche de 1
```





100 000 tirages avec une loi gaussienne de valeur moyenne $-0,5$ et d'écart-type 1

EXERCICE : fréquence d'oscillations libres non amorties

facteur de qualité : distribution uniforme $Q = 4,99$ $u(Q) = 0,84$

pseudo-période : distribution uniforme $T = 990 \mu\text{s}$ $u(T) = 120 \mu\text{s}$

calculer la valeur et l'incertitude-type de la fréquence $f_0 = \frac{1}{T \sqrt{1 - \frac{1}{4Q^2}}}$

>>> par la méthode analytique

>>> par la méthode de Monte-Carlo

>>> comparer les résultats et interpréter

SOLUTION :

méthode analytique : $f_0 = \frac{1}{T \sqrt{1 - \frac{1}{4Q^2}}} = 1015,2 \text{ Hz}$

$$q = 1 - \frac{1}{4Q^2} = 0,9900 \Rightarrow dq = \frac{1}{2Q^3} dQ \Rightarrow u(q) = \frac{u(Q)}{2Q^3} = 3,38 \cdot 10^{-3}$$

$$f_0 = \frac{1}{T \sqrt{q}} \Rightarrow \ln(f_0) = -\ln(T) - \frac{1}{2} \ln(q) \Rightarrow \frac{df_0}{f_0} = -\frac{dT}{T} - \frac{1}{2} \frac{dq}{q}$$

$$\Rightarrow \frac{u(f_0)}{f_0} = \sqrt{\left[\frac{u(T)}{T} \right]^2 + \frac{1}{4} \left[\frac{u(q)}{q} \right]^2} = 0,121$$

$$u(f_0) = 123 \text{ Hz}$$

méthode de Monte-Carlo :



```
import numpy as np
from matplotlib import pyplot as plt
import numpy.random as rd
```

```
N = 1000000
```

```
def f0(Q,T):
    return 1/(T*(1-1/(4*Q**2))**0.5)
```

```
Q,u_Q = 4.99,0.84
```

```
T,u_T = 990E-6,120E-6
```

```
N = 100000
```

```
Q_sim = np.random.uniform(Q-u_Q*3**.5,Q+u_Q*3**.5,N) # valeur
simulées de Q
```

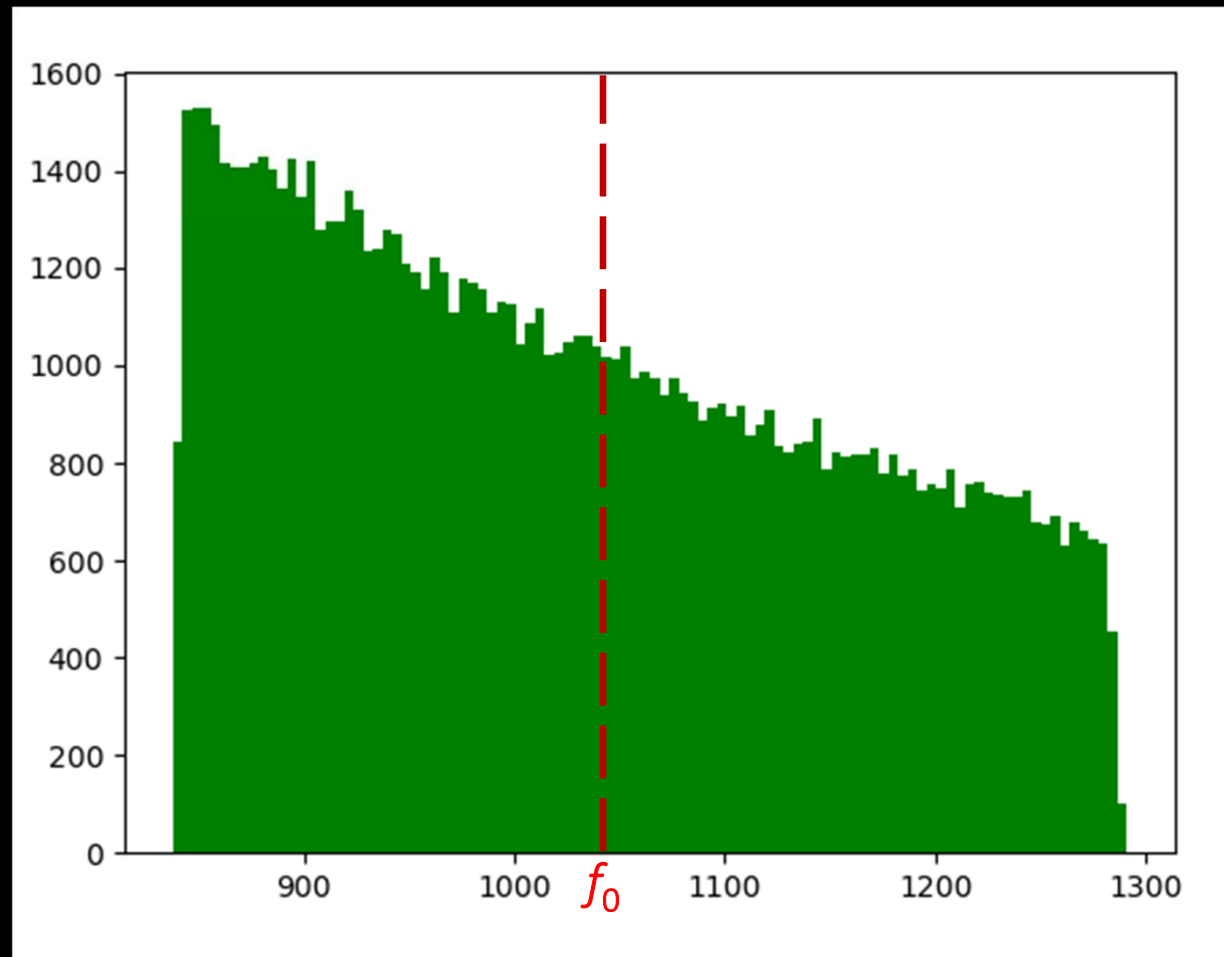
```
T_sim = np.random.uniform(T-u_T*3**.5,T+u_T*3**.5,N) # valeur
simulées de T
```

```
f0_sim = f0(Q_sim,T_sim) # valeur simulées de f0
```

```
f0_moy = np.mean(f0_sim) # moyenne de ces valeurs
```

```
u_f0 = np.std(f0_sim, ddof = 1) # écart-type de ces valeurs
```

```
plt.figure() # histogramme
plt.hist(f0_sim, 100, color='g')
plt.show()
```



la distribution **n'est pas uniforme** (la fonction $f(Q, T)$ n'est pas linéaire et les incertitudes relatives sur Q et T sont trop grandes)

les bornes ne permettent pas le calcul de la moyenne ni de l'écart-type...

```
>>> f0_moy  
1030.898167012851
```

au lieu de 1015,2 (méthode analytique)

```
>>> u_f0  
126.91683180054731
```

au lieu de 123 (méthode analytique)



la méthode analytique est une approximation, généralement très bonne, sauf quand les incertitudes relatives sont grandes

quand elle n'est pas applicable, moyenne et écart-type d'une distribution ne suffisent plus pour caractériser une mesure

5. PRÉSENTATION DES RÉSULTATS

5.1 Effet de la propagation des incertitudes sur le résultat présenté

pour déterminer le nombre de chiffres « significatifs » sur $X = f(X_1, X_2, \dots, X_k, \dots)$ connaissant les incertitudes-type sur $X_1, X_2, \dots$, **il n'y a pas à appliquer de règles particulières** : il suffit de calculer l'incertitude-type sur le résultat en utilisant la propagation des incertitudes

attention ! Lorsque $Y = f(X)$, le nombre de chiffres significatifs de y peut être supérieur à celui de x

exemple : angle X dont la mesure donne $x = 7,3^\circ$ avec $u(X) = 1,1^\circ$ (2 chiffres significatifs) ; quelle est l'incertitude sur $Y = \cos(X)$?

$$Y = \cos(X) \Rightarrow dY = -\sin(X)dX$$

$$\Rightarrow u(Y) = |\sin(X)| \cdot u(X) = \sin\left(7,3 \times \frac{\pi}{180}\right) \times 1,1 \times \frac{\pi}{180} = 2,4 \cdot 10^{-3}$$

comme $\cos(7,3^\circ) = 0,9919$, on a au moins 3 chiffres significatifs sur Y

EXERCICE

utiliser un script Python pour obtenir l'histogramme, la moyenne et l'écart-type de $Y = \cos(X)$ pour une distribution gaussienne de X de moyenne $7,3^\circ$ et d'écart-type $1,1^\circ$:

conclure

SOLUTION :

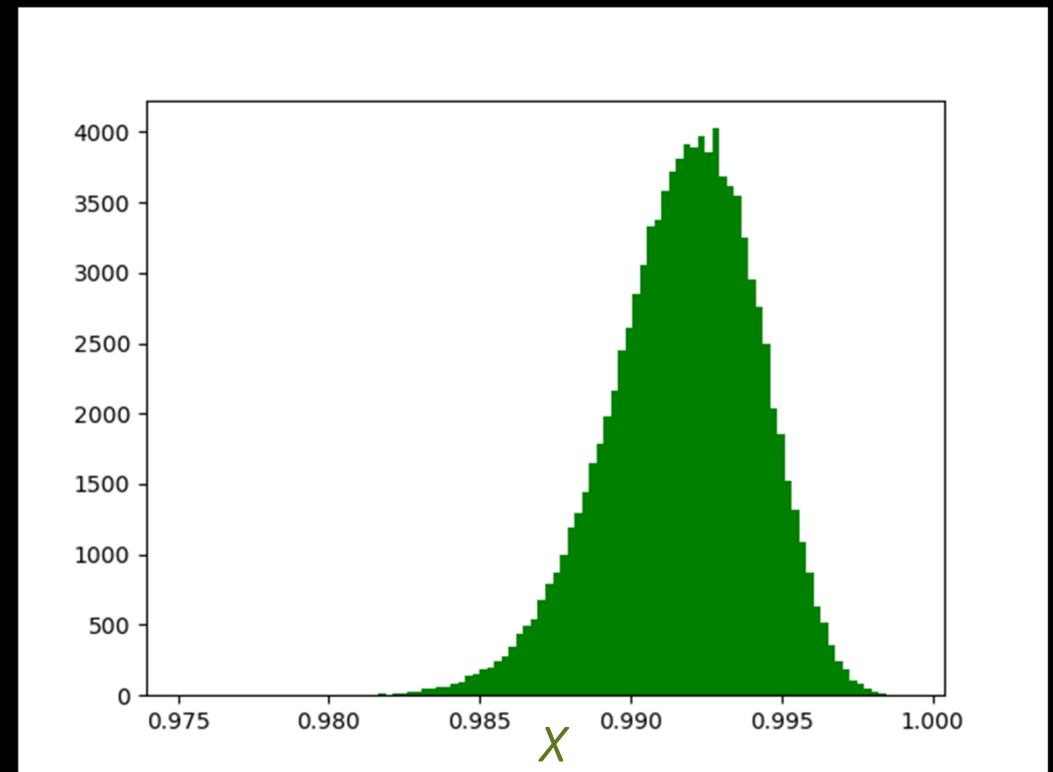


```
def f(X):
    return np.cos(X*np.pi/180)
N = 100000
X_sim = np.random.normal(7.3,1.1,N) # valeur simulées de X
Y_sim = f(X_sim) # valeur simulées de Y = cos(X)
Y_moy = np.mean(Y_sim) # moyenne de ces valeurs
u_Y = np.std(Y_sim, ddof = 1) # écart-type de ces valeurs
plt.figure() # histogramme
plt.hist(Y_sim,100,color='g')
plt.show()
```

```
>>> Y_moy
0.9917228516461807
```

```
>>> u_Y
0.0024478357595842937
```

au moins trois chiffres significatifs sur Y...



5.2 Nombre de chiffres pour l'incertitude et le résultat

les incertitudes sont elles-mêmes entachées d'incertitudes
 $\Rightarrow$ un seul chiffre pour l'incertitude-type $u(X)$?

non ! : problème d'arrondi qui va se propager lors des calculs utilisant $u(X)$

exemple :

$$x_1 = 2,50 \quad x_2 = 2,68$$

$$u(x_1) = 0,11 \quad u(x_2) = 0,13$$

$$X = X_2 - X_1 \quad u(X) = \sqrt{u^2(X_1) + u^2(X_2)} = \sqrt{0,11^2 + 0,13^2} \simeq 0,17$$

en arrondissant à un seul chiffre : $x = 0,2 \quad u(X) = 0,2$

si on ne gardait d'emblée qu'un seul chiffre :

$$x_1 = 2,5 \quad x_2 = 2,7$$

$$u(x_1) = 0,1 \quad u(x_2) = 0,1$$

$$\Rightarrow u(X) = \sqrt{0,1^2 + 0,1^2} \simeq 0,14$$

$x = 0,2 \quad u(X) = 0,1$

l'incertitude est divisée par 2 du fait de la troncature !

on conservera *deux* chiffres pour l'incertitude-type $u(X)$

les chiffres significatifs ne sont pas des chiffres dont on est sûr, mais des chiffres qui apportent des informations sur la mesure dont le résultat peut alors être réutilisé, comparé avec d'autres...

exemple de présentation de résultats :

$$f' = 10,40 \text{ cm} \quad u(f') = 0,35 \text{ cm}$$


EXERCICE : résolution d'un instrument et incertitudes

la mesure avec un instrument d'une grandeur X est distribuée selon une gaussienne mesurée de valeur moyenne $x = 20,1$ et d'écart-type $\sigma = 1,0$

cependant, l'affichage ne produit que des valeurs entières arrondies à l'entier le plus proche de la grandeur mesurée : 18, 19, 20, 21, 22, etc. (la résolution de l'affichage est égale à 1)

**on lit $N = 100\,000$ mesures de la grandeur sur l'affichage
l'incertitude-type sur la *moyenne* de ces valeurs peut-elle être inférieure à la résolution (ici égale à 1) ?**

sous numpy, on prend l'arrondi de g avec `np.round(g)`

SOLUTION :



```
N = 100000
```

```
mu, sigma = 20.1, 1 # paramètres de la gaussienne
```

```
X = rd.normal(mu, sigma, N)
```

```
moyenne_X = np.mean(X)
```

```
ecart_type_X = np.std(X, ddof = 1) / np.sqrt(N) # écart-type sur la  
MOYENNE
```

```
plt.figure('histogramme de X')
```

```
plt.hist(X, 200, color='g')
```

```
plt.show()
```

```
Y = np.round(X)
```

```
moyenne_Y = np.mean(Y)
```

```
ecart_type_Y = np.std(Y, ddof = 1) / np.sqrt(N) # écart-type sur  
la MOYENNE
```

```
plt.figure('histogramme de Y')
```

```
plt.hist(Y, 200, color='r')
```

```
plt.show()
```

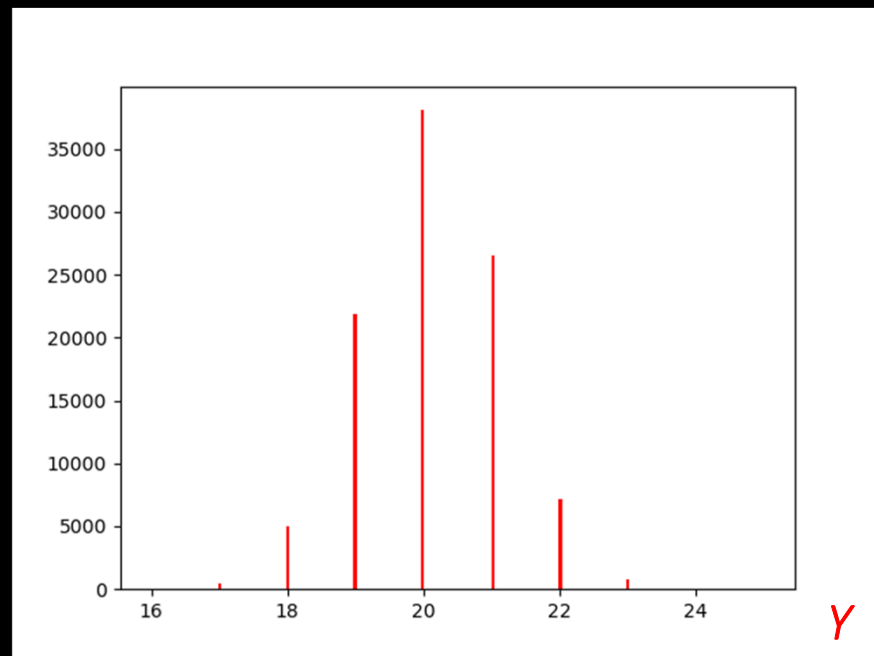
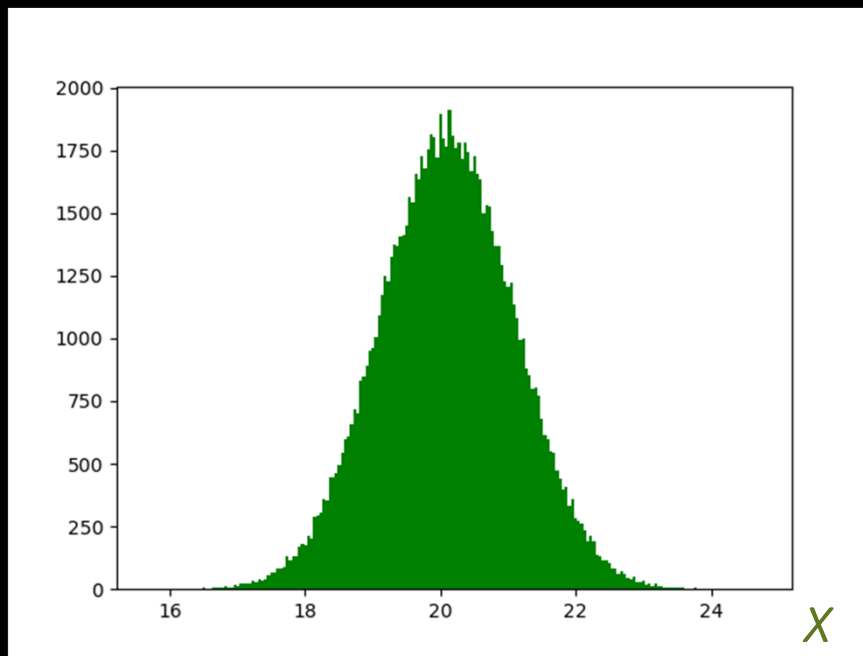
X

```
>>> moyenne_X,ecart_type_X
20.1008660259205 0.003147226633651741
```

```
>>> moyenne_Y,ecart_type_Y
20.10121 0.0032808799125737605
```



l'incertitude est (ici très) inférieure à la résolution



(l'écart-type σ est suffisant pour dissymétriser l'histogramme de Y par rapport à $Y = 20$)

6. ÉCARTS NORMALISÉS (Z-SCORE) / PRISE DE DÉCISION

6.1 Comparaison entre une valeur mesurée et une valeur de référence

$$Z = \frac{X - X_{\text{réf}}}{u(X)} : \text{Z-score (algébrique)}$$

le Z-score est aussi appelé écart-normalisé car :

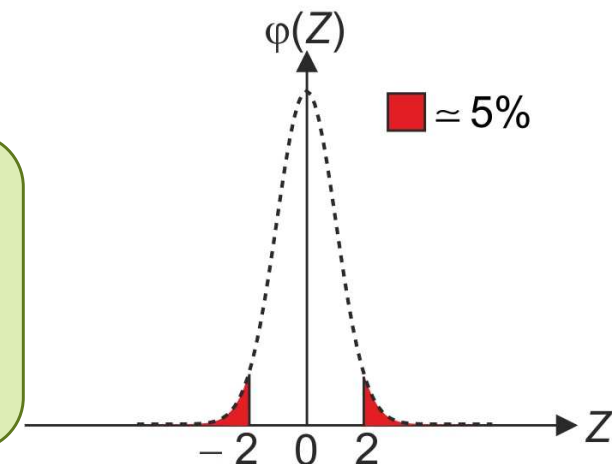
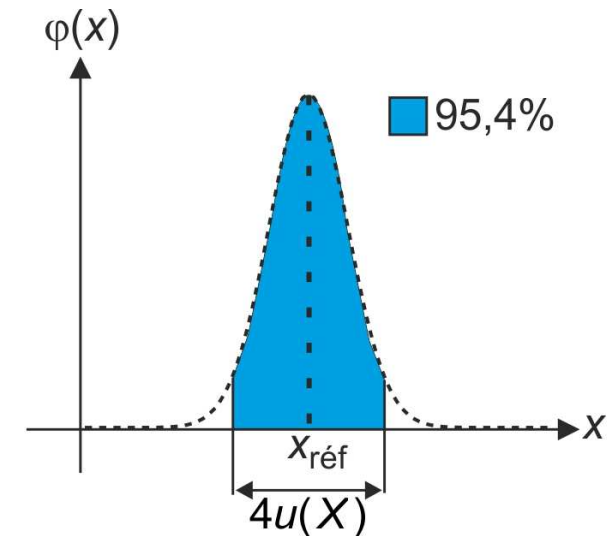
- sa valeur moyenne est $\bar{Z} = \frac{\bar{X} - X_{\text{réf}}}{u(X)} = 0$
- son écart-type est $u(Z) = \frac{u(X - X_{\text{réf}})}{u(X)} = \frac{u(X)}{u(X)} = 1$

si Z suit une loi normale, la probabilité d'avoir

$|Z| > 2$ vaut 5%.

en prenant un critère à 5%, c'est-à-dire si $|Z| = \left| \frac{X - X_{\text{réf}}}{u(X)} \right| > 2$,

on rejette l'hypothèse que la mesure est compatible avec la valeur théorique



il faut définitivement abandonner la « gourance » : calcul de $\left| \frac{X^* - X_{\text{réf}}}{X_{\text{réf}}} \right|$,
 qui consiste à donner en % de la valeur théorique l'écart entre valeur mesurée et
 valeur théorique

aucun critère de décision ne peut être établi à partir de cette grandeur

l'hypothèse d'une loi normale peut ne pas être vérifiée, néanmoins on montre
 dans tous les cas (inégalité de Bienaymé-Tchebychev) que la probabilité
 d'obtenir un $|Z|$ supérieur à quelques incertitudes-type est très faible

6.2 Comparaison de deux mesures

on pose $Z = \frac{X_1 - X_2}{\sqrt{u^2(X_1) + u^2(X_2)}}$ et on compare $|Z|$ à 2 afin de décider si les

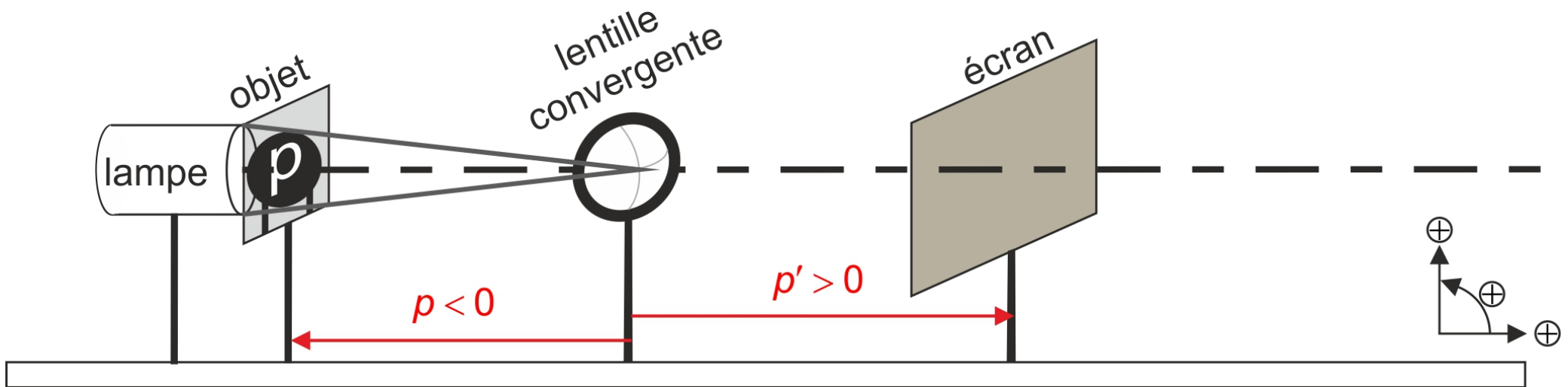
deux mesures sont compatibles ou non

7. INCERTITUDES MIXTES

Exemple : distance focale d'une lentille convergente

on fait varier la position p de l'objet par rapport à la lentille (incertitude négligée)
et on déplace l'écran afin d'obtenir l'image la plus nette possible

on repère l'image par rapport à la lentille à l'aide de la distance p'
l'incertitude provient de la plage de netteté d'amplitude $2\Delta p'$ (on suppose une distribution uniforme)



les mesures obtenues :

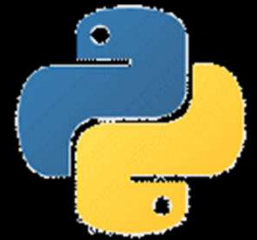
p (cm)	-23,5	-24	-25	-27	-30	-34	-40	-45	-55
p' (cm)	139,5	122,5	104,6	77,3	60,7	48,7	40,1	36,2	31,5
$\Delta p'$ (cm)	3	2	1	0,5	0,5	0,3	0,2	0,2	0,2

on dispose donc de 9 valeurs de la distance focale : $f' = \frac{pp'}{p - p'}$

$$\frac{1}{f'} = \frac{1}{p'} - \frac{1}{p} \Rightarrow -\frac{df'}{f'^2} = -\frac{dp'}{p'^2} \Rightarrow u(f') = \left(\frac{f'}{p'}\right)^2 u(p') = \frac{1}{\sqrt{3}} \left(\frac{f'}{p'}\right)^2 \Delta p'$$

quelle valeur de f' présenter et avec quelle incertitude-type ?

(1) on évalue la dispersion des mesures grâce à distribution des valeurs de f' (sans prendre en compte les incertitudes-type sur chaque mesure (de type B))



```
P = np.array([-23.5, -24, -25, -27, -30, -34, -40, -45, -55]) * 1E-2
Pp =
np.array([139.5, 122.5, 104.6, 77.3, 60.7, 48.7, 40.1, 36.2, 31.5]) * 1E-2
sigma_Pp = np.array([3, 2, 1, 0.5, 0.5, 0.3, 0.2, 0.2, 0.2]) * 1E-2 / 3 ** .5

def fp(p, pp): # calcul de la distance focale
    return p * pp / (p - pp)

Fp = fp(P, Pp) # tableau des distances focales

>>> calcul_mu_sigma(Fp) # fonction créée précédemment

(0.20064736446152062, 0.00017781409869729436)
```

(2) on prend la moyenne arithmétique des mesures et l'incertitude-type de cette moyenne :



$$f' = \frac{1}{N} \sum_{i=1}^N f'_i \quad u(f') = \frac{\sqrt{\sum_{i=1}^N u^2(f'_i)}}{N} = \frac{\sqrt{\sum_{i=1}^N \frac{1}{w_i}}}{N}$$

avec $w_i = \frac{1}{u^2(f'_i)}$ inverse de la variance

on trouve :

(**0.20064**736446152062, **0.00010**664878963662461)



au lieu de **0,00017** : on a affiné l'incertitude-type sur la focale...

(3) hors-programme mais le plus rigoureux : on pondère par les inverses des variances, donnant plus de poids aux valeurs de focales les moins entachées d'incertitudes



$$f' = \frac{\sum_{i=1}^N w_i f'_i}{\sum_{i=1}^N w_i} \quad u(f') = \frac{1}{\sqrt{\sum_{i=1}^N w_i}}$$

avec $w_i = \frac{1}{u^2(f'_i)}$ inverse de la variance

on trouve :

(**0.200678**36305879627, **9.4**151482137413259**e-05**)



au lieu de **0,00010** : on a encore affiné l'incertitude-type sur la focale...

8. AJUSTEMENT LINÉAIRE OU AFFINE

8.1 Ajustement affine (« régression linéaire ») / méthode des moindres carrés

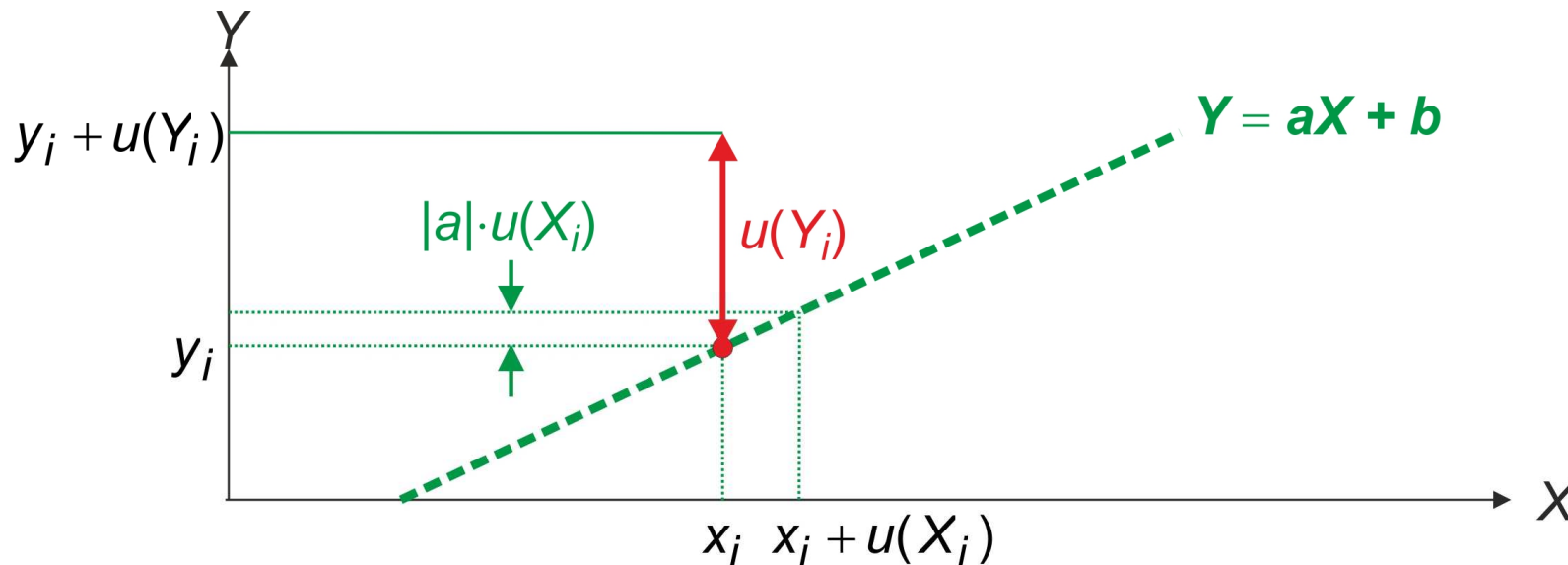
la relation théorique $Y = f(X) = aX + b$ est-elle bien vérifiée ?

quelles sont alors les valeurs expérimentales de a et b ? les incertitudes sur ces coefficients ?

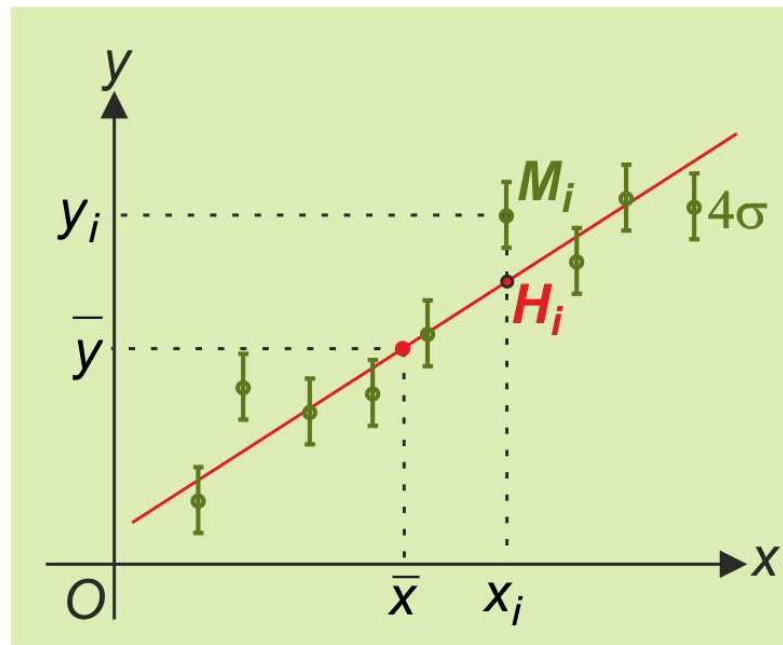
on mesure (x_i, y_i) $i \in \llbracket 1, n \rrbracket$ avec $(u(X_i), u(Y_i))$ $i \in \llbracket 1, n \rrbracket$

les hypothèses de la régression linéaire ordinaire sont :

— l'incertitude-type sur X est négligeable, soit $|a| \cdot u(X_i) \ll u(Y_i)$



— les incertitudes-type $u(Y_1) = u(Y_2) = \dots = u(Y_n) = \sigma$ sont toutes identiques et Y_i suit une loi normale



on représente alors les « bâtonnets » d'incertitude de taille 4σ autour des points $M_i(x_i, y_i)$ $i \in \llbracket 1, n \rrbracket$

si H_i est le projeté selon Oy de M_i sur une droite quelconque d'équation $Y = aX + b$,

on note $\mathbf{S} = \sum_{i=1}^n [y_i - f(x_i)]^2 = \sum_{i=1}^n [y_i - (ax_i + b)]^2$ la somme des carrés des distances $H_i M_i$

on cherche a et b tels que $\chi^2 = \frac{1}{\sigma^2} \sum_{i=1}^N [y_i - f(x_i)]^2 = \frac{S}{\sigma^2}$ est minimal

$$\Rightarrow \begin{cases} a = \frac{\overline{xy} - \bar{x} \bar{y}}{\overline{x^2} - \bar{x}^2} \\ b = \bar{y} - a\bar{x} \end{cases} \quad \text{avec } \bar{x} = \frac{1}{N} \sum_{i=1}^N x_i \text{ valeur moyenne (non pondérée)}$$

remarque : méthode analytique (hors-programme)

on montre :

$$u(a) = \frac{\sigma}{\sqrt{N[\overline{x^2} - \bar{x}^2]}}$$

$$u(b) = \sigma \sqrt{\frac{\bar{x}^2}{N[\overline{x^2} - \bar{x}^2]}}$$

pour contrôler si les mesures sont bien compatibles avec une loi affine, on utilise les résidus $\varepsilon_i = y_i - (ax_i + b)$ qui ne doivent pas être supérieurs en valeur absolue à 2σ

remarque : si σ est inconnu, on peut l'évaluer à partir des N mesures, *en supposant la*

loi affine : on montre que σ est estimé par $\sigma^* = \sqrt{\frac{S}{N-2}}$

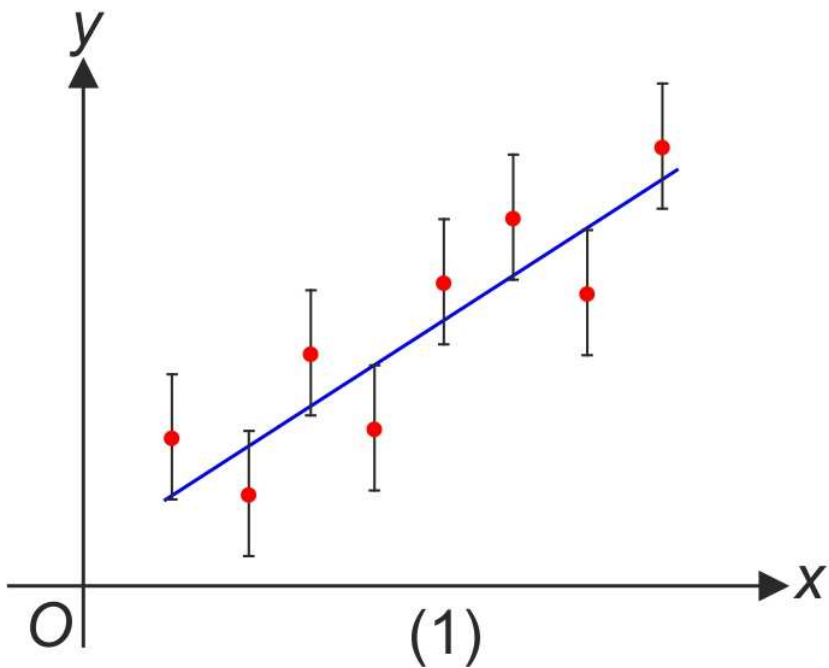
aucun test de conformité ne peut alors être appliqué puisque l'on a supposé *a priori* la loi vérifiée

quand on demande de vérifier qu'une loi est affine à l'aide de résultats expérimentaux et que les incertitudes de mesures ne sont pas données, cette vérification est en réalité impossible

l'usage est de postuler que la loi affine théorique est bien vérifiée par l'expérience si le coefficient de corrélation linéaire au carré est très proche de 1

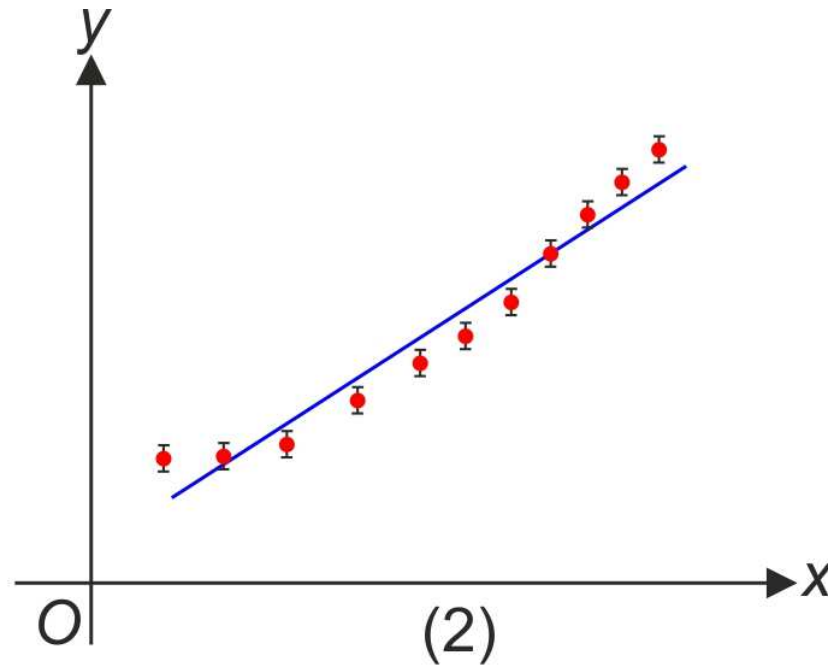
cette démarche n'a aucun fondement théorique

en effet, ce coefficient (très utile pour faire des statistiques) **ne prend pas en compte les incertitudes de mesure** ; il est excellent dès qu'il y a une corrélation, même non linéaire entre Y et X , ce qui est pratiquement toujours le cas dans les sciences expérimentales



$$r^2 = 0,87$$

résidus < 2, hypothèse « loi affine » vérifiée



$$r^2 = 0,99$$

hypothèse « loi affine » NON vérifiée

8.2 Simulation par la méthode de Monte-Carlo

on mesure (x_i, y_i) $i \in \llbracket 1, n \rrbracket$ avec $u(Y_i) = \sigma$ pour $i \in \llbracket 1, n \rrbracket$

- (1) on simule N jeux de mesures $(x_i^j = x_i, y_i^j = y_i + \varepsilon_i^j)$ avec $j \in \llbracket 1, N \rrbracket$
où ε_i suit la distribution choisie centrée sur 0 et d'incertitude-type σ
- (2) on effectue une régression linéaire sur chaque jeu de mesures et on obtient N valeurs des coefficients de la droite d'ajustement (a_j, b_j) avec $j \in \llbracket 1, N \rrbracket$
- (3) on en déduit la valeur de l'écart-type sur a et b

Entrées :

Lx et **Ly** : tableaux 1D `np.array([..., ...])` valeurs des x_i et y_i
sigma : nombre flottant incertitude-type sur les y_i (supposée la même pour tout i)

fig : booléen (défaut `fig = True` : on trace et une figure avec le nuage de points expérimentaux et la droite d'ajustement affine, et une figure représentant les résidus normalisés)

Print : valeurs de la pente a , de $u(a)$, de l'ordonnée à l'origine b et de $u(b)$

Returns : **a_MC**, **b_MC** : `np.array([..., ...])`, `np.array([..., ...])` les tableaux des pentes et des ordonnées à l'origine pour traitement ultérieur



```
def AjustAff_MC(Lx, Ly, sigma, fig = True):
    Lx, Ly = np.asarray(Lx), np.asarray(Ly) # si on a oublié
    que Lx et Ly sont des tableaux numpy, pas des listes
    N_MC = 100000 # nombre d'expériences simulées
    a_MC, b_MC = np.zeros(N_MC), np.zeros(N_MC) # initialisation
    des listes dans lesquelles on va stocker les pentes et ordonnées à
    l'origine simulées
    for j in range(N_MC): # pour chaque expérience simulée
        # on simule la mesure de Y pour toutes les valeurs de X
        Ly_MC = Ly + rd.uniform(-sigma*3**0.5, sigma*3**0.5, size
        = len(Lx))
        # on effectue l'ajustement affine pour cette expérience simulée
        p = np.polyfit(Lx, Ly_MC, 1)
        # on stocke les valeurs des paramètres d'ajustement dans les listes
        a_MC et b_MC
        a_MC[j] = p[0]
        b_MC[j] = p[1]
    a_moy = np.mean(a_MC)
    u_a = np.std(a_MC, ddof = 1)
    b_moy = np.mean(b_MC)
    u_b = np.std(b_MC, ddof = 1)
```



```

print('pente a = ',a_moy)
print('incertitude-type : u(a) = ',u_a)
print('ordonnée à l\'origine b = ',b_moy)
print('incertitude-type : u(b) = ', u_b)

if fig == True: # optionnel : tracés
    X = np.array([np.min(Lx),np.max(Lx)])
    Y = a_moy*X+b_moy
    plt.figure('points et ajustement affine')
    plt.errorbar(Lx, Ly, yerr =
2*sigma,fmt='or',ecolor = 'black',elinewidth = 3,label =
'points expérimentaux')
    plt.xlabel('$X$')
    plt.ylabel('$Y$')
    plt.plot(X,Y, label = 'ajustement affine',color
= 'blue')
    plt.legend(fontsize = 15)
    plt.show()

```



```

    Z = (Ly-(a_moy*Lx+b_moy))/(sigma)
    plt.figure('résidus normalisés ajustement affine Monte
Carlo')
    plt.plot(Lx, Z, 'o', color = 'red')
    marge = abs(np.max(Lx)-np.min(Lx))/100
    plt.fill_between([np.min(Lx), np.max(Lx)], y1 = -2, y2 =
2, color = 'olive', alpha = 0.4)
    plt.xlim(np.min(Lx)-marge, np.max(Lx)+marge)
    plt.ylim(-3, 3)
    plt.xlabel('$X$')
    plt.ylabel('résidus normalisés de $Y$' )
    plt.grid()
    plt.show()

    return (a_MC, b_MC)

```

8.3 Ajustement linéaire (et pas affine)

si le modèle physique est $Y = aX$, il n'est pas justifié de chercher a et b tels que $Y = aX + b$

plus généralement, lorsque l'on cherche à vérifier expérimentalement un modèle physique, on ne doit pas introduire plus de paramètres que celui du modèle !

c'est une erreur très fréquente commise par les étudiants lors des TIPE

on reprend les calculs : $a^* = \frac{\overline{xy}}{\overline{x^2}}$, et on adapte le code Python

$$\text{remarque : } u(a) = \frac{\sigma}{\sqrt{N \overline{x^2}}}$$

```
def linfit(Lx, Ly): # calcul de la pente a = moy(xy)/moy(x**x) de la
droite d'ajustement linéaire, remplace polyfit
```

```
    N = len(Lx)
```

```
    xy, x2 = 0,0
```

```
    for i in range(N):
```

```
        xy = xy + Lx[i]*Ly[i]
```

```
        x2 = x2 + Lx[i]**2
```

```
    return xy/x2
```



```
def AjustLin_MC(Lx, Ly, sigma, fig = True):
```

```
    Lx, Ly = np.asarray(Lx), np.asarray(Ly) # si on a oublié
que Lx et Ly sont des tableaux numpy, pas des listes
```

```
    N_MC = 100000
```

```
    a_MC = np.zeros(N_MC)
```

```
    for j in range(N_MC):
```

```
        Ly_MC = Ly + rd.uniform(-sigma*3**0.5, sigma*3**0.5, size
= len(Lx))
```

```
        a_MC[j] = linfit(Lx, Ly_MC)
```

```
    a_moy = np.mean(a_MC)
```

```
    u_a = np.std(a_MC, ddof = 1)
```

```
    print('pente a = ', a_moy)
```

```
    print('incertitude-type : u(a) = ', u_a)
```



```

if fig == True:
    X = np.array([np.min(Lx), np.max(Lx)])
    Y = a_moy*X
    plt.figure('points et ajustement linéaire')
    plt.errorbar(Lx, Ly, yerr = 2*sigma, fmt='or', ecolor =
'black', elinewidth = 3, label = 'points expérimentaux')
    plt.xlabel('$X$')
    plt.ylabel('$Y$')
    plt.plot(X, Y, label = 'ajustement linéaire', color =
'blue')
    plt.legend(fontsize = 15)
    plt.show()

```



```

        Z = (Ly-(a_moy*Lx))/(sigma)
        plt.figure('résidus normalisés ajustement linéaire
Monte Carlo')
        plt.plot(Lx, Z, 'o', color = 'red')
        marge = abs(np.max(Lx)-np.min(Lx))/100
        plt.fill_between([np.min(Lx), np.max(Lx)], y1 = -2, y2
= 2, color = 'olive', alpha = 0.4)
        plt.xlim(np.min(Lx)-marge, np.max(Lx)+marge)
        plt.ylim(-3, 3)
        plt.xlabel('$X$')
        plt.ylabel('résidus normalisés de $Y$')
        plt.grid()
        plt.show()

    return (a_MC)

```

8.4 Exemple : nombre n de traits par mm d'un réseau

on mesure la déviation sous incidence normale dans l'ordre 1 pour Hg
théorie : la raie de longueur d'onde λ se trouve dans la direction faisant l'angle i'

par rapport à la lumière incidente sur le réseau, avec $\underbrace{\sin i'}_Y = \underbrace{n}_a \cdot \underbrace{\lambda}_X$

$\lambda(\text{nm})$	couleur	$i'(^{\circ})$	$n = \frac{\sin i'}{\lambda}$ (traits/mm)	$u(n)$
404,7	violet intense	13,35	570,5	$\sigma = 0,57$
407,8	violet	13,45	570,4	0,57
435,8	indigo	14,37	569,5	0,53
491,6	bleu-vert	16,27	569,9	0,46
546,1	vert	18,17	571,0	0,41
577,0	jaune 1	19,23	570,8	0,39
579,1	jaune 2	19,27	569,9	0,39
623,4	rouge	20,80	569,6	0,36

détermination de $u(n)$

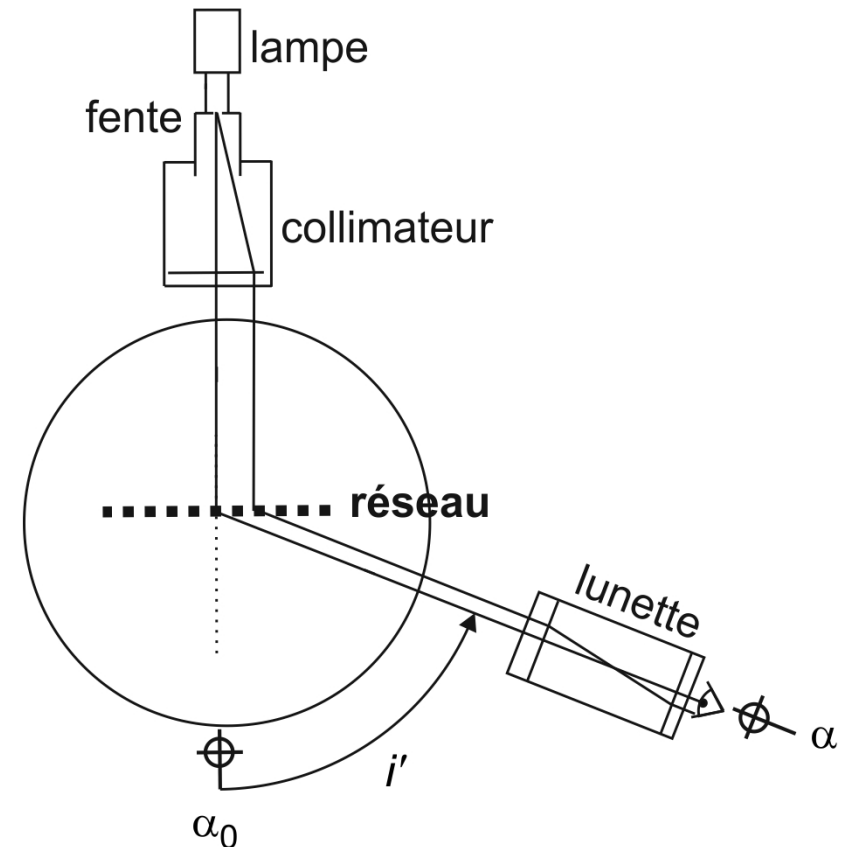
les angles α sont mesurés au vernier avec :

$$\Delta\alpha = 1' \Rightarrow u(\alpha) = \frac{1}{60} \times \frac{\pi}{180} \times \frac{1}{\sqrt{3}} = 1,68 \cdot 10^{-4} \text{ rad}$$

$$i' = \alpha - \alpha_0 \Rightarrow u(i') = \sqrt{u^2(\alpha_0) + u^2(\alpha)} = \sqrt{2} u(\alpha) = 2,37 \cdot 10^{-4} \text{ rad}$$

λ ordre 1 ordre 0

$$dn = \frac{\cos i'}{\lambda} di' \Rightarrow u(n) = \frac{\cos i'}{\lambda} u(i')$$





$$\underbrace{\sin i'}_Y = \underbrace{n}_a \cdot \underbrace{\lambda}_X$$

$$u(Y) = \cos i' \cdot u(i')$$

```
Lx = np.array([404.7, 407.8, 435.8, 491.6, 546.1, 577, 579.1, 623.4])*1E-9 # les longueurs d'onde
```

```
L = np.array([13.35, 13.45, 14.37, 16.27, 18.17, 19.23, 19.27, 20.8])*np.pi/180 # les angles transformés en radians
```

```
Ly = np.sin(L) # on calcule leur sinus
```

```
sigma_angle = 2.37E-4 # incertitude-type sur les angles
sigma = np.cos(L[0])*sigma_angle # toutes les incertitudes-type sur Y sont supposées être les mêmes (on prend ici la plus grande des valeurs sigma = 0.57)
```

```
AjustAff_MC(Lx, Ly, sigma)
```

```
AjustLin_MC(Lx, Ly, sigma)
```

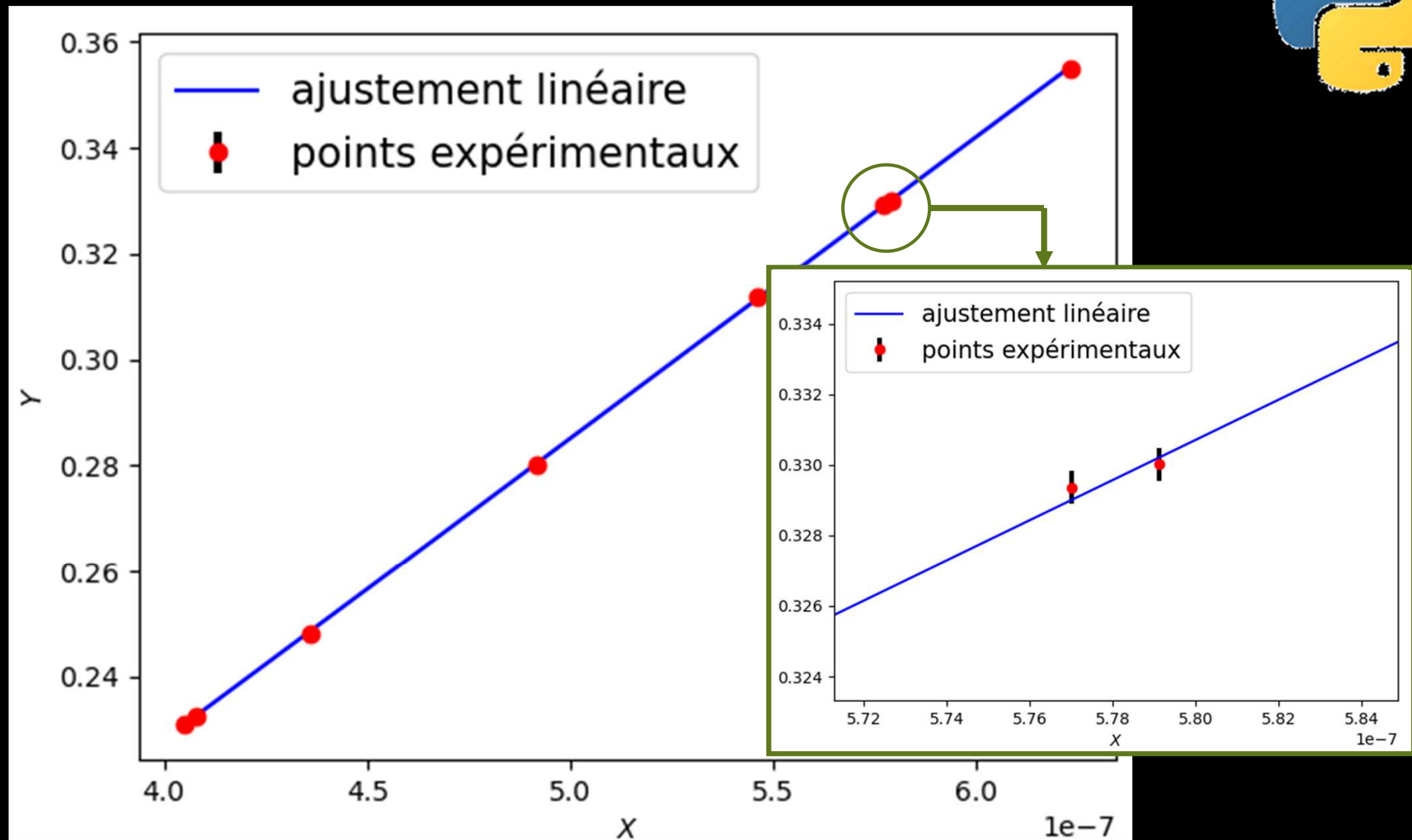
résultats des ajustements affine et linéaire par simulation de Monte-Carlo

	a	$u(a)$	b	$u(b)$
affine	570027	1028	9E-5	5E-4
linéaire	570200	159		

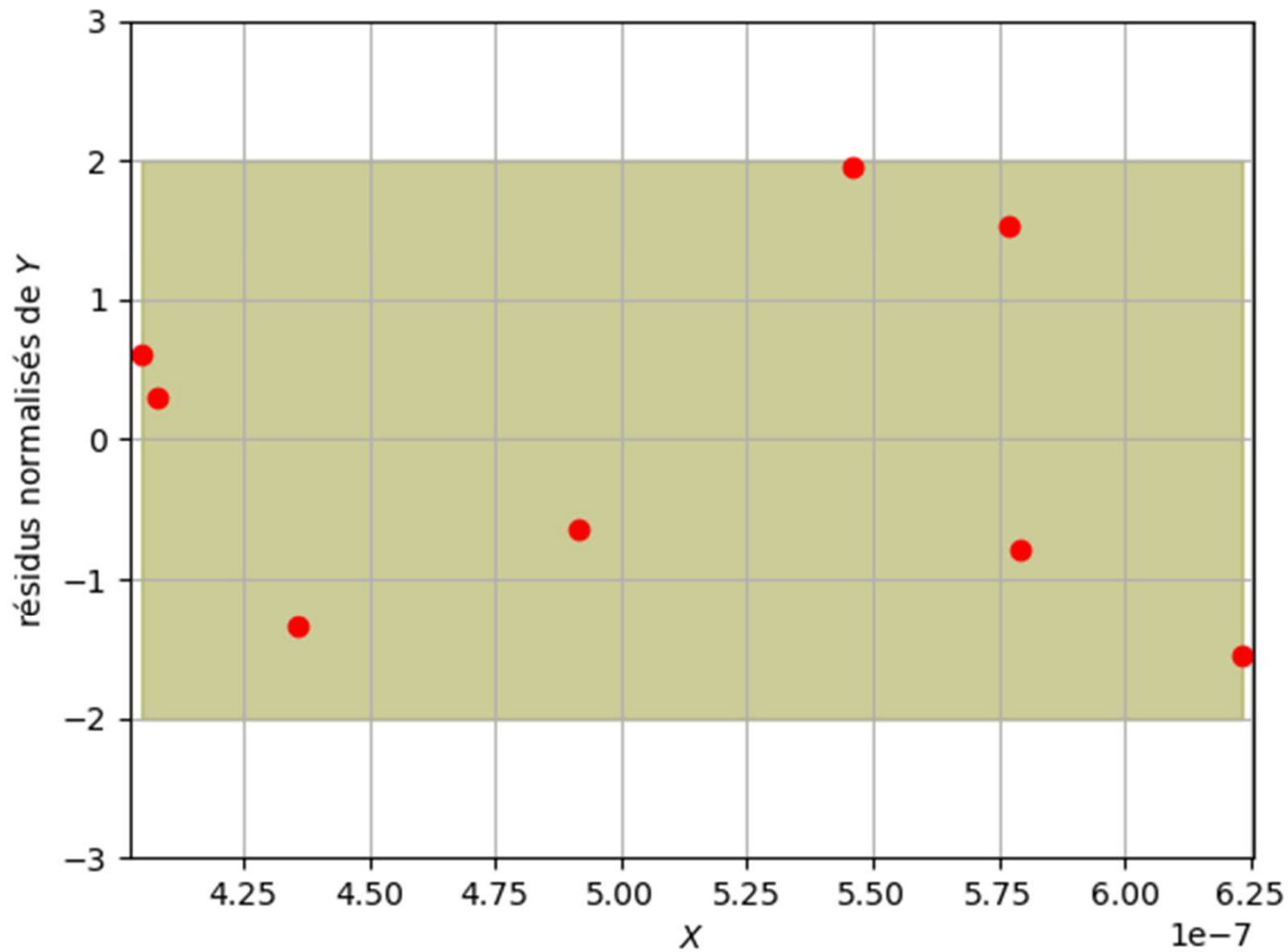
ajustement affine : $b = 0$ est bien compatible avec les valeurs trouvées, mais l'incertitude sur a est bien plus grande que pour l'ajustement linéaire

conclusion :
l'incertitude est surévaluée quand le modèle n'est pas adapté

AjustLin_MC (Lx, Ly, sigma)



“visuel” mais les barres d’incertitudes sont trop petites



le tracé des Z-scores est plus parlant ; le modèle linéaire est bien vérifié ici

9. CONCLUSION

la méthode de Monte-Carlo est considérée comme «l'avenir des incertitudes »

elle a été ajoutée au GUM

(Guide to the expression of uncertainty in Measurement)

elle permet de transmettre les informations du mesurage sans perte d'information
(sur la nature des distributions notamment)

on peut ainsi par exemple tenir compte de la *corrélation* entre les coefficients a et b d'un ajustement linéaire lorsque l'on souhaite déterminer l'incertitude sur $Y = aX + b$ due à une incertitude sur X ... (cf. capacités numériques 2)

