

POURQUOI DONNER DES INCERTITUDES ?

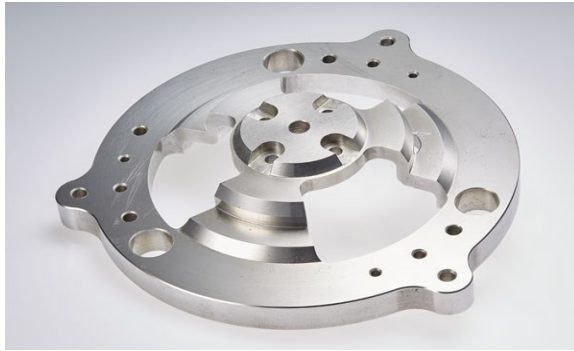
La métrologie (science de la mesure) rassemble les connaissances permettant de donner un sens à la mesure. Elle permet :

- >>> La caractérisation quantitative des propriétés d'un objet
- >>> La comparaison d'une même propriété pour deux objets
- >>> La confrontation entre la théorie et l'expérience
- >>> La validation ou non du modèle retenu

Tout cela 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



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

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



Ce cycliste s'est-il dopé ?



La théorie est-elle vérifiée ?

INCERTITUDE-TYPE

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

>>> **valeur mesurée x**

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

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

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

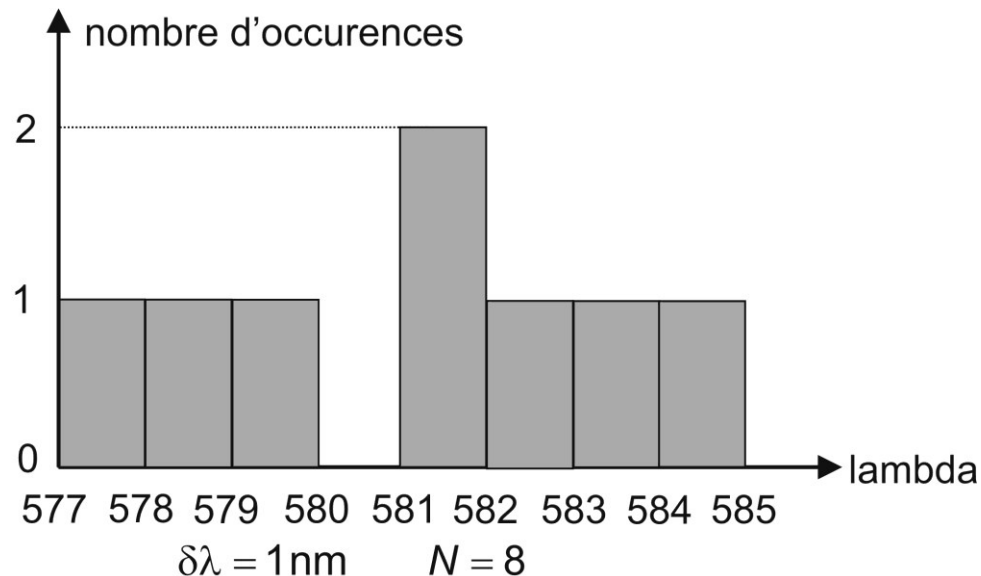
Nouveau programme :
incertitude élargie
intervalle de confiance

DENSITÉ DE PROBABILITÉ

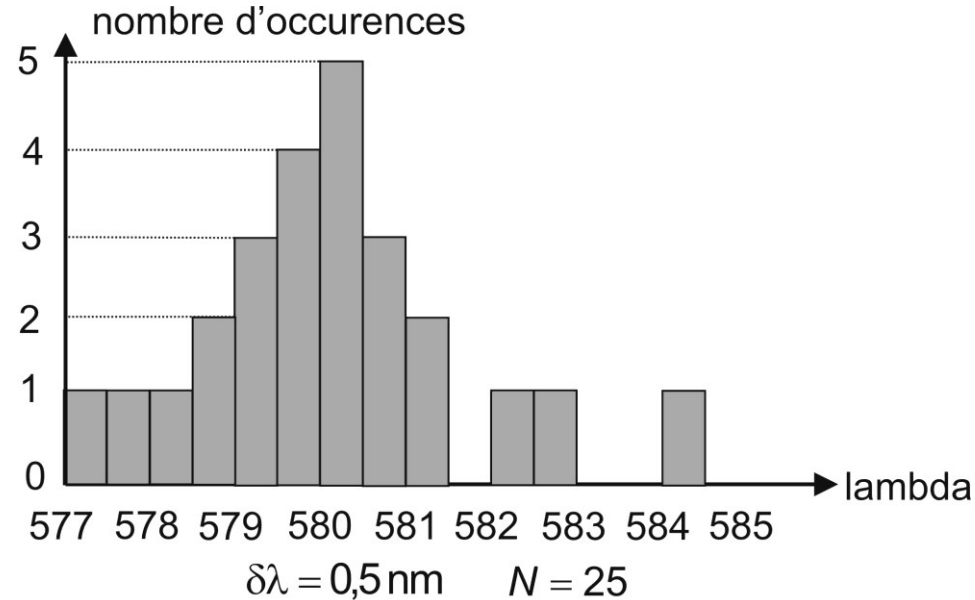
$N = 8$ groupes de TP mesurent une longueur d'onde :

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 $x_i + \delta x$



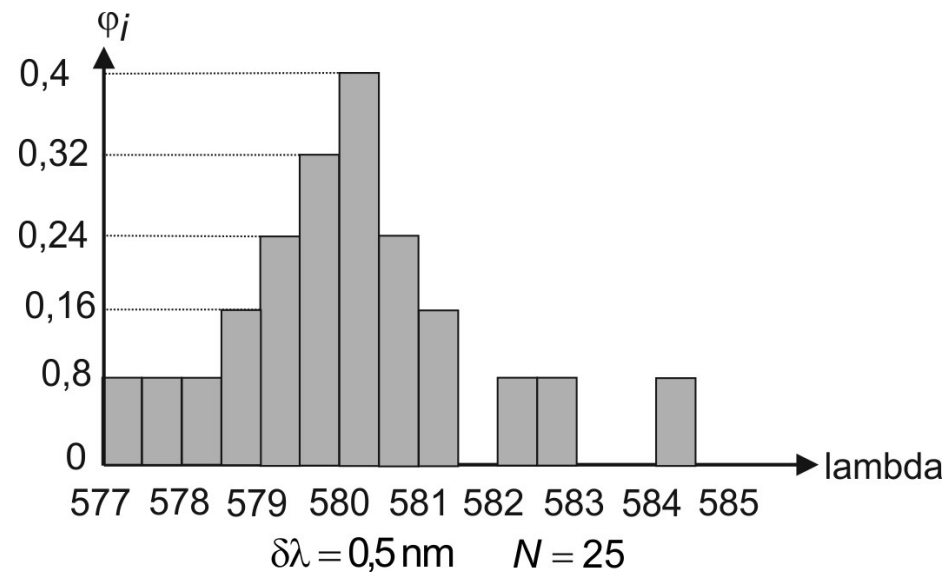
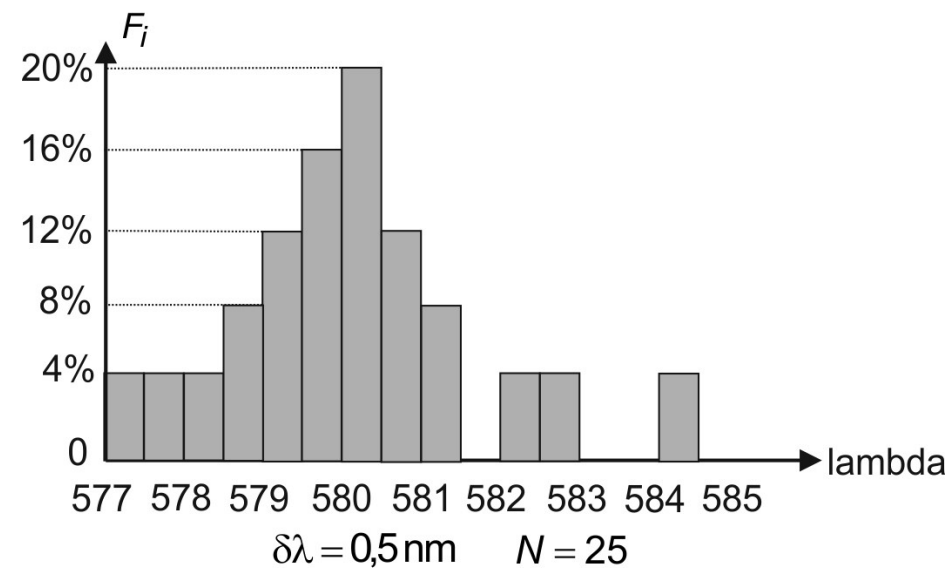
Recommençons avec 25 groupes et des intervalles plus fins :



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

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

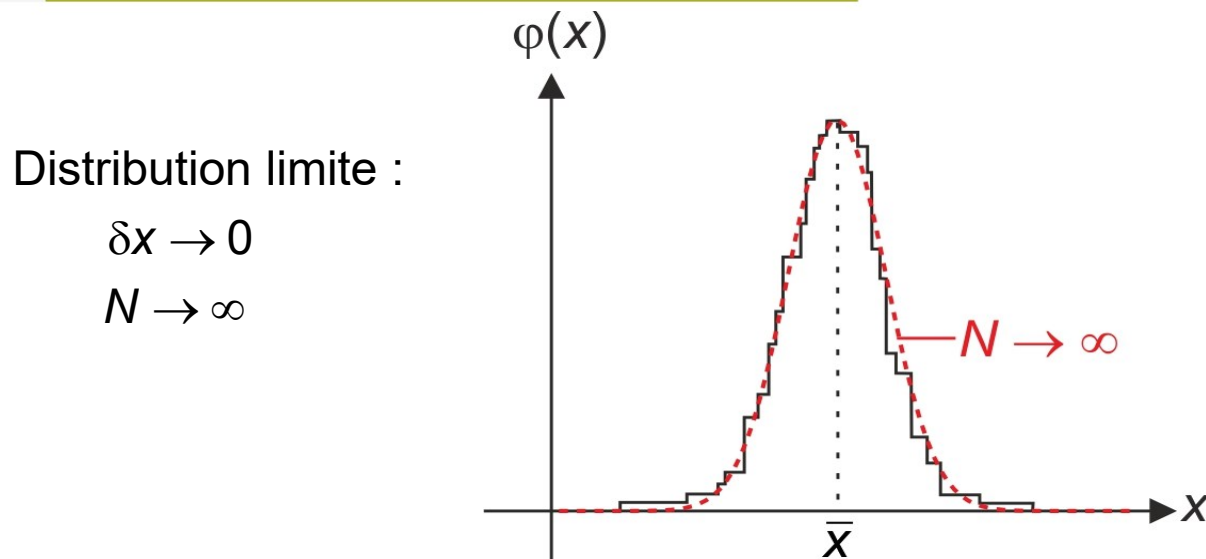
$$\Rightarrow \sum_{i=1}^N F_i = 1 \quad \text{et} \quad \bar{x} = \frac{\sum_{i=1}^N n_i x_i}{N} = \sum_{i=1}^N F_i x_i \quad (\text{valeur moyenne})$$



Si $\phi_j = \frac{F_j}{\delta x}$, la fréquence d'apparition de x_j est donc $F_j = \phi_j \cdot \delta x$, soit l'aire du

rectangle associé

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



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

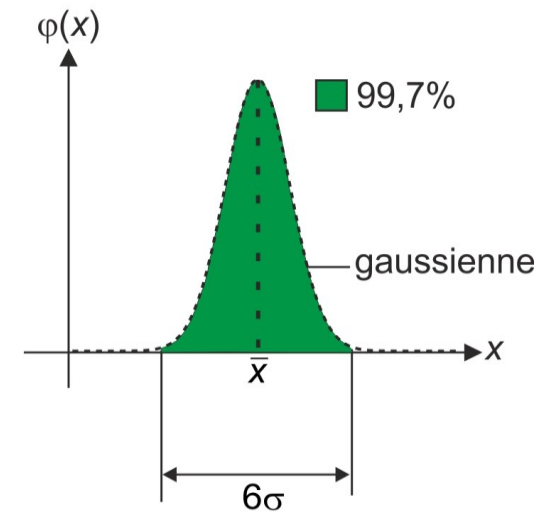
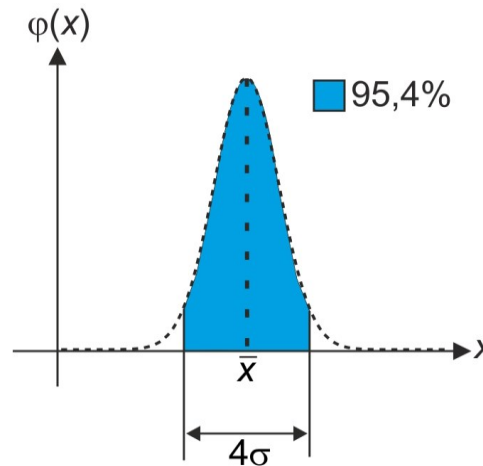
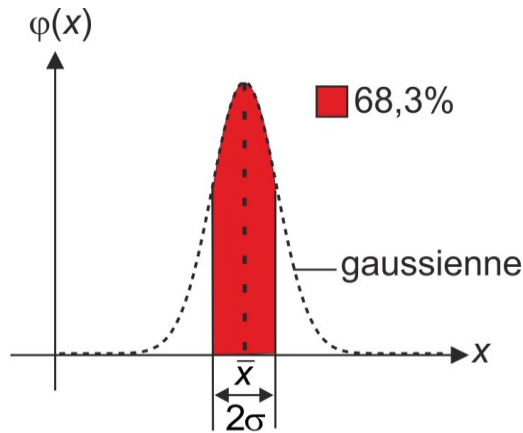
Densité de probabilité $\varphi(x)$:

$dp = \varphi(x)dx$ est la probabilité que le résultat de la mesure de la grandeur X se trouve dans l'intervalle $[x, x + dx]$

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

$\sigma = \sqrt{(x - \bar{x})^2}$ écart-type

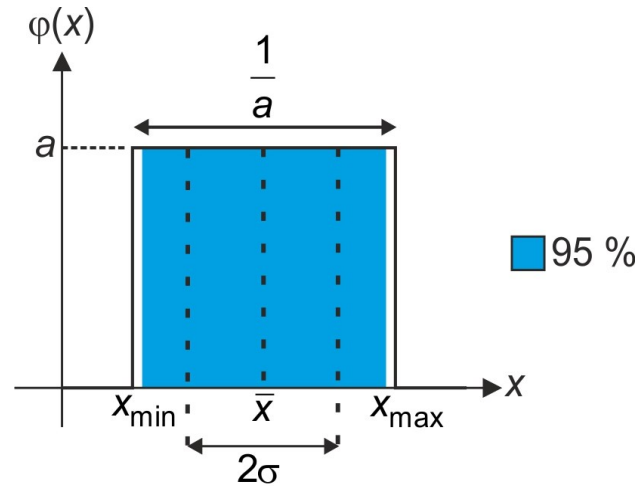
Loi normale (gaussienne) $\varphi(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\bar{x})^2}{2\sigma^2}}$



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)

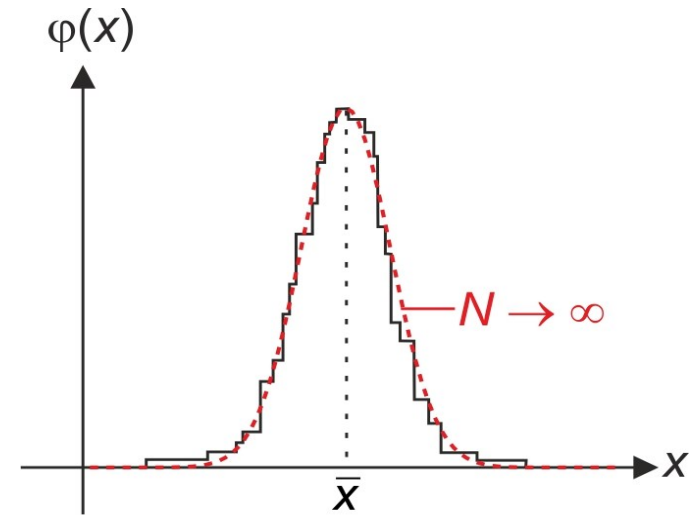


Si Δ est la demi-largeur de l'intervalle $[x_{\min}, x_{\max}]$: $x_{\max} - x_{\min} = 2\Delta$

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

ÉVALUATION DES INCERTITUDES DE TYPE A

Le nombre de mesures N est fini $\Rightarrow \bar{x}$ reste inconnue : différentes séries de mesures portant sur la même grandeur X vont donner des valeurs moyennes x^* différentes (réparties autour de \bar{x}) et des écarts-type différents répartis autour de σ



Propagation des incertitudes :
$$\sigma_{\bar{x}} = \frac{1}{N} \sqrt{\sum_{i=1}^N \sigma^2} = \frac{\sigma\sqrt{N}}{N} = \frac{\sigma}{\sqrt{N}}$$

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 « lentement » en $\frac{1}{\sqrt{N}}$

En pratique : « petit » nombre de mesures

Dans le cas où l'on a réalisé N mesures de la même grandeur donnant des résultats $x_i, i \in 1, N$, 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)

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 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

$$c^* = \frac{1}{9} \sum_{i=1}^9 c_i = 0,9666 \cdot 10^{-2} \text{ mol} \cdot \text{L}^{-1} \quad \sigma^* = \sqrt{\frac{1}{8} \left[\sum_{i=1}^9 c_i^2 - \frac{1}{9} \left(\sum_{i=1}^9 c_i \right)^2 \right]} = 0,0905 \cdot 10^{-2} \text{ mol} \cdot \text{L}^{-1}$$

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

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

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

Moyenne et incertitude-type expérimentaux d'une liste de mesures



```
import numpy as np

def calcul_mu_sigma(liste):
    N = len(liste)
    somme = 0
    for mes in liste:
        somme += mes
    mu = somme/N # moyenne
    somme_sigma_exp = 0
    for mes in liste:
        somme_sigma_exp += (mes-mu)**2
    sigma_exp = (somme_sigma_exp/(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)

>>> liste = 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(liste)
(0.009665555555555557, 0.0003016211342760572)
```

Avec les fonctions Python :



```
>>> np.mean(liste) # valeur moyenne
```

```
0.009665555555555574
```

```
>>> np.std(liste, 0, ddof = 1) # écart-type de la liste de  
valeurs
```

```
0.00090486340282817154
```

ÉVALUATION DES INCERTITUDES DE TYPE B

Si on n'effectue qu'une seule mesure de X qui fournit la valeur numérique x , on s'appuie sur toutes les informations disponibles, voire sur le bon sens... pour déterminer l'incertitude-type $u(X)$ de la mesure

(1) Le constructeur d'un appareil de mesure (multimètre, pipette, etc.) fournit une *incertitude-type* (sans préciser selon quel type de distribution sont réparties les mesures). *On considère alors que la distribution est gaussienne*

(2) Le constructeur fournit une incertitude « constructeur » $\Delta_{\text{constructeur}}$ sans autre information (souvent notée $\pm\Delta_{\text{constructeur}}$). Le résultat de la mesure est x

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

équiprobabilité de trouver la valeur de X dans $[x - \Delta_{\text{constructeur}}, x + \Delta_{\text{constructeur}}]$

et une probabilité nulle hors de cet intervalle

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

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

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

Comme précédemment, on suppose la densité de probabilité rectangulaire

$$\Rightarrow \mathbf{x}^* = \frac{\mathbf{x}_{\max} + \mathbf{x}_{\min}}{2} \quad \text{est la valeur mesurée}$$

$$\mathbf{u}(\mathbf{X}) = \frac{\mathbf{x}_{\max} - \mathbf{x}_{\min}}{2\sqrt{3}} \quad \text{est l'incertitude-type de la mesure}$$

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

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** de température de $0,01^\circ \text{ C}$ (les indications en $^\circ \text{ C}$ se font avec deux chiffres après la virgule)

Pas de $\Delta_{\text{constructeur}}$

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

Exemple 3 : On lit sur un vernier un angle compris entre $152^\circ 27'$ et $152^\circ 29'$.

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

Exemple 4 : focométrie: méthode d'autocollimation. L'image est nette pour une

distance entre l'objet et la lentille convergente dans l'intervalle $[9,8 \text{ cm} ; 11,0 \text{ cm}]$

$$f' = \frac{9,8 + 11,0}{2} = 10,4 \text{ cm} \quad u(f') = \frac{11,0 - 9,8}{2\sqrt{3}} = 0,35 \text{ cm}$$

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

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	-	-	-

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

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

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	$u(X_1)$	$u(X_2)$

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

Méthode analytique . Si les incertitudes sont *suffisamment petites* :

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

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

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

$$u^2(X) = \overline{(x - \bar{x})^2} = \overline{[x - f(x_1, x_2)]^2}$$

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

$$+ 2 \underbrace{\overline{(x_1 - \bar{x}_1)(x_2 - \bar{x}_2)}}_{0 \text{ (corr\u00e9lation nulle)}} \frac{\partial f}{\partial X_1}(\bar{x}_1, \bar{x}_2) \frac{\partial f}{\partial X_2}(\bar{x}_1, \bar{x}_2)$$

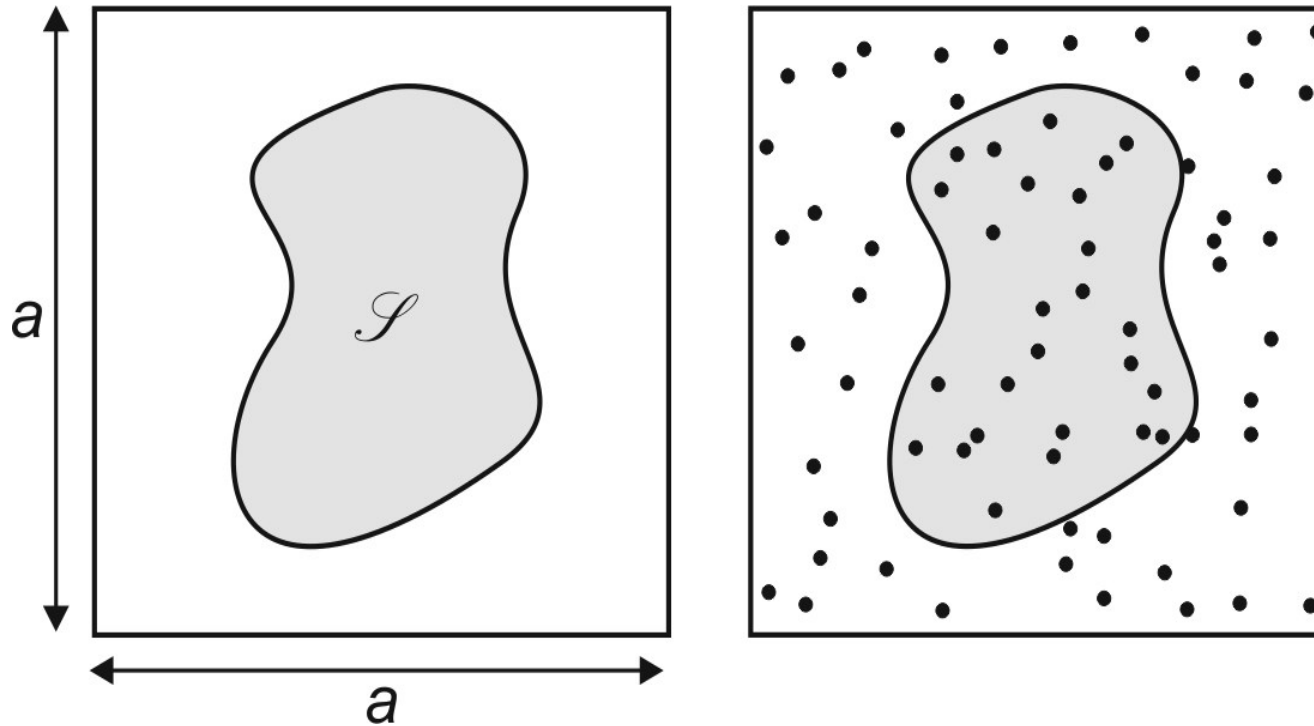
$$\Rightarrow u(X) = \sqrt{u^2(X_1) \left[\frac{\partial f}{\partial X_1}(\bar{x}_1, \bar{x}_2) \right]^2 + u^2(X_2) \left[\frac{\partial f}{\partial X_2}(\bar{x}_1, \bar{x}_2) \right]^2}$$

$$u(X) = \sqrt{\sum_k \left[\frac{\partial f}{\partial X_k} \right]^2 u^2(X_k)}$$

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

$$X = X_1^\alpha X_2^\beta \Rightarrow \frac{u(X)}{|x|} = \sqrt{\alpha^2 \left(\frac{u(X_1)}{x_1} \right)^2 + \beta^2 \left(\frac{u(X_2)}{x_2} \right)^2}$$

Simulation de Monte-Carlo



N boulets tirés sur la surface a^2 avec une **densité de probabilité uniforme**.
 Si n boulets tombent dans le lac, le rapport n / N approche d'autant mieux \mathcal{S} / a^2 que N est grand.

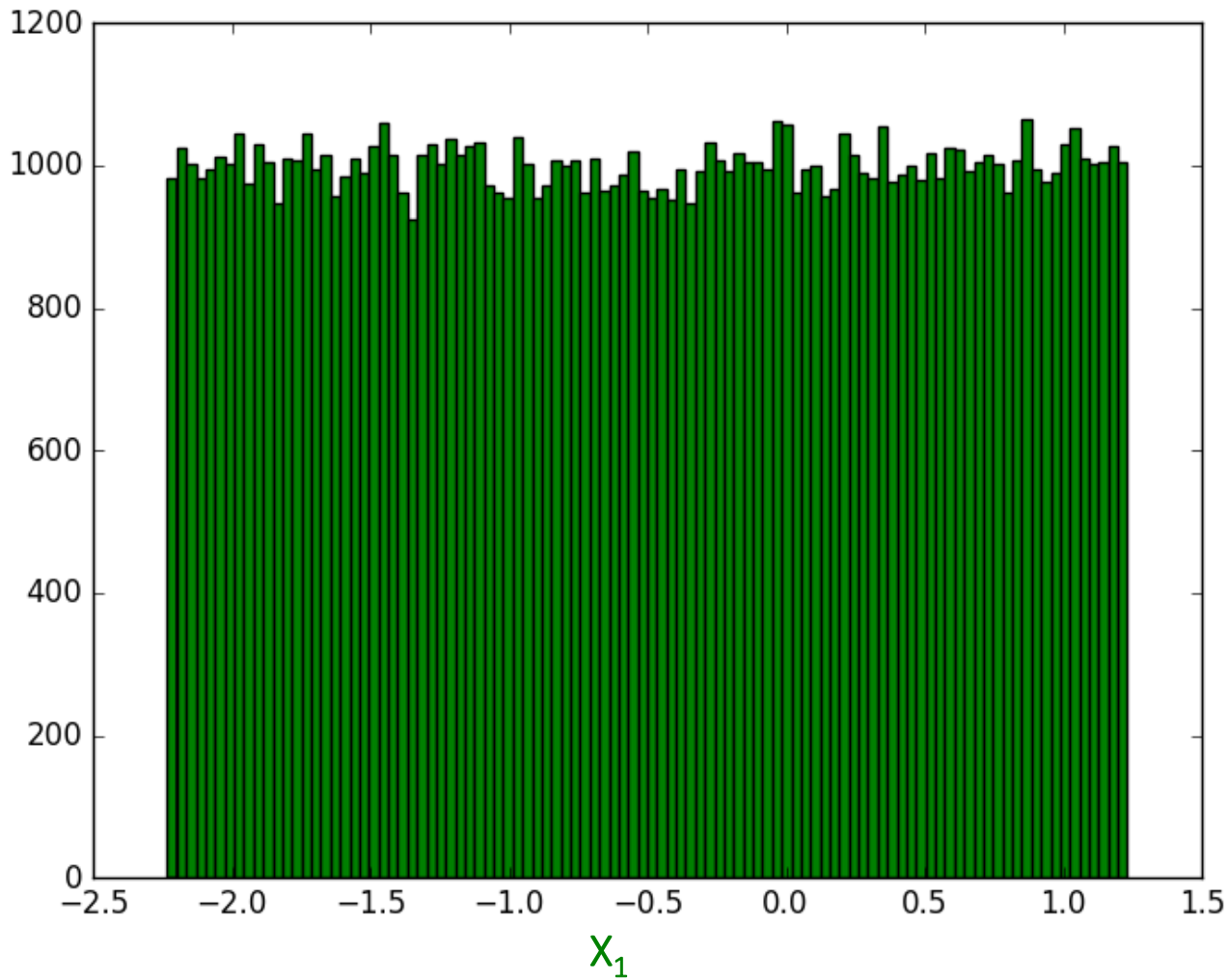
```
import numpy as np #
import numpy.random as rd # bibliothèque random de numpy
from matplotlib import pyplot as plt # graphiques

moy1,sigma1 = -0.5,1
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()

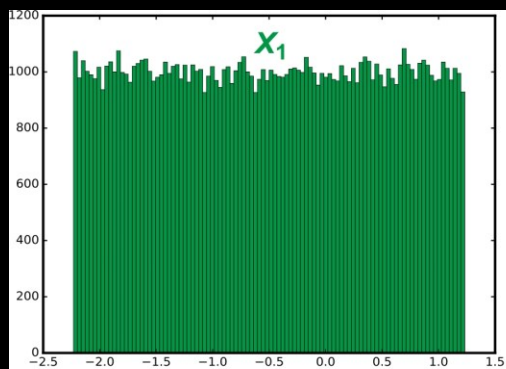
>>> np.mean(X1) # moyenne du tableau de valeurs
-0.498551962647#on vérifie que la valeur moyenne est bien
proche de -0.5
>>> np.std(X1, 0, ddof = 1) # écart-type du tableau de valeurs
0.999595207002 #on vérifie que l'écart-type est bien proche de
1.
```



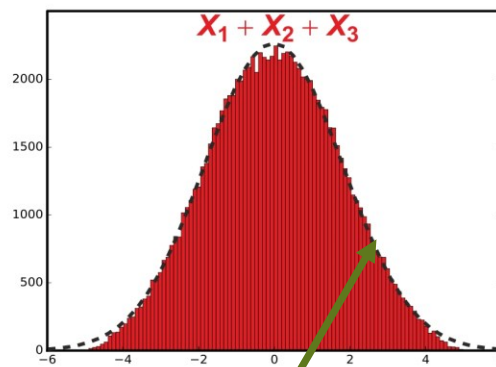
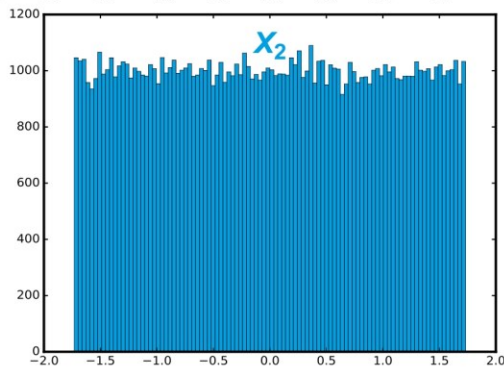
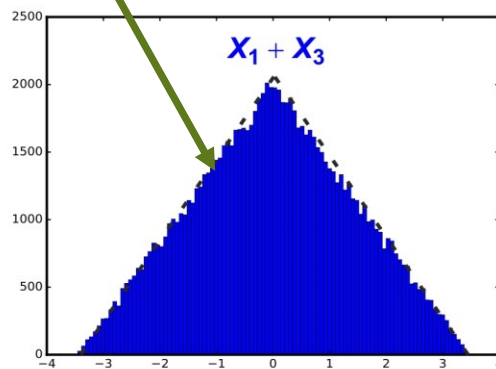


100 000 tirages avec une loi uniforme de valeur moyenne -0,5 et d'écart-type 1

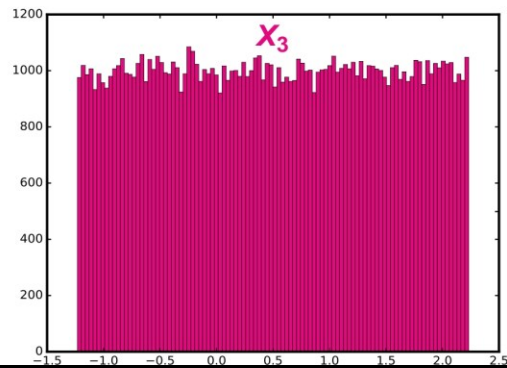
On recommence pour X_2 et X_3 de valeurs moyennes 0 et 0,5 et d'écart-type 1 et on trace les histogrammes de $X_1 + X_3$ et $X_1 + X_2 + X_3$



triangle



gaussienne



EXERCICE



Obtenir à l'aide d'un script Python les histogrammes de X_2 et X_3 de valeurs moyennes 0 et 0,5 et d'écart-type 1, puis les histogrammes de $X_1 + X_3$ et $X_1 + X_2 + X_3$.

Le plus important ! Propagation d'incertitudes avec une simulation de Monte-Carlo



$X = f(X_1, X_2, \dots, X_k, \dots)$ fonction python à écrire

Une distribution de X_k normale est définie par le tableau numpy

```
Xk = ['normal', mu, sigma]
```

Une distribution de X_k uniforme est définie par le tableau numpy :

```
Xk = ['rect', mu, sigma]
```

On génère 1 000 000 tirages aléatoires de X_k obéissant à la distribution choisie, à l'aide de `np.random.normal(Xk[1], Xk[2], N)` pour une loi normale

et de `np.random.uniform(xmin, xmax, N)` pour une loi uniforme
`xmin, xmax`, valeurs extrêmes de la distribution rectangulaire, sont calculés par
`xmin, xmax = Xk[1]-Xk[2]*3**.5, Xk[1]+Xk[2]*3**.5`

La grande valeur $N = 10^6$ est préconisée pour rendre négligeables les fluctuations entre deux tirages différents mais peut être réduite par exemple à 10 000 si les calculs sont trop longs. N est passée en variable globale

Les distributions obtenues (tableaux 1D numpy de type array) sont stockées dans une liste L



On calcule et on renvoie la valeur moyenne et l'écart-type de X. On affiche également son histogramme

```
N = 1000000
```

```
def propagation_monte_carlo(*arg):
    L = []
    for elt in arg:
        if elt[0] == 'normal':
            L.append(rd.normal(elt[1],elt[2],N))
        else:
            xmin,xmax = elt[1]-elt[2]*3**.5,elt[1]+elt[2]*3**.5
            L.append(rd.uniform(xmin,xmax,N))
    X = f(*L) # f est appliquée à tous les éléments de L
    mu = np.mean(X)
    sigma = np.std(X, 0, ddof = 1)
    plt.hist(X,100)
    plt.show()
    return mu,sigma
```

Exemple : fréquence d'oscillations libres non amorties

Facteur de qualité $Q = 4,99$ $u(Q) = 0,84$

Pseudo-période $T = 990 \mu\text{s}$ $u(T) = 120 \mu\text{s}$

Méthode analytique

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

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

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

$$\Rightarrow \frac{u(f_0)}{f_0} = \sqrt{\left[\frac{u(T)}{T}\right]^2 + \frac{1}{4} \left[\frac{u(q)}{q}\right]^2} = 0,121 \quad \boxed{u(f_0) = 123 \text{ Hz}}$$

Simulation de Monte-Carlo



```
def f(T,Q):  
    return 1/(T*(1-1/4/Q**2)**0.5)  
  
T = ['rect',990E-6,120E-6]  
  
Q = ['rect',4.99,0.84]  
  
>>> propagation_monte_carlo(T,Q)  
(1031.0798687795748, 127.23438033510512)
```



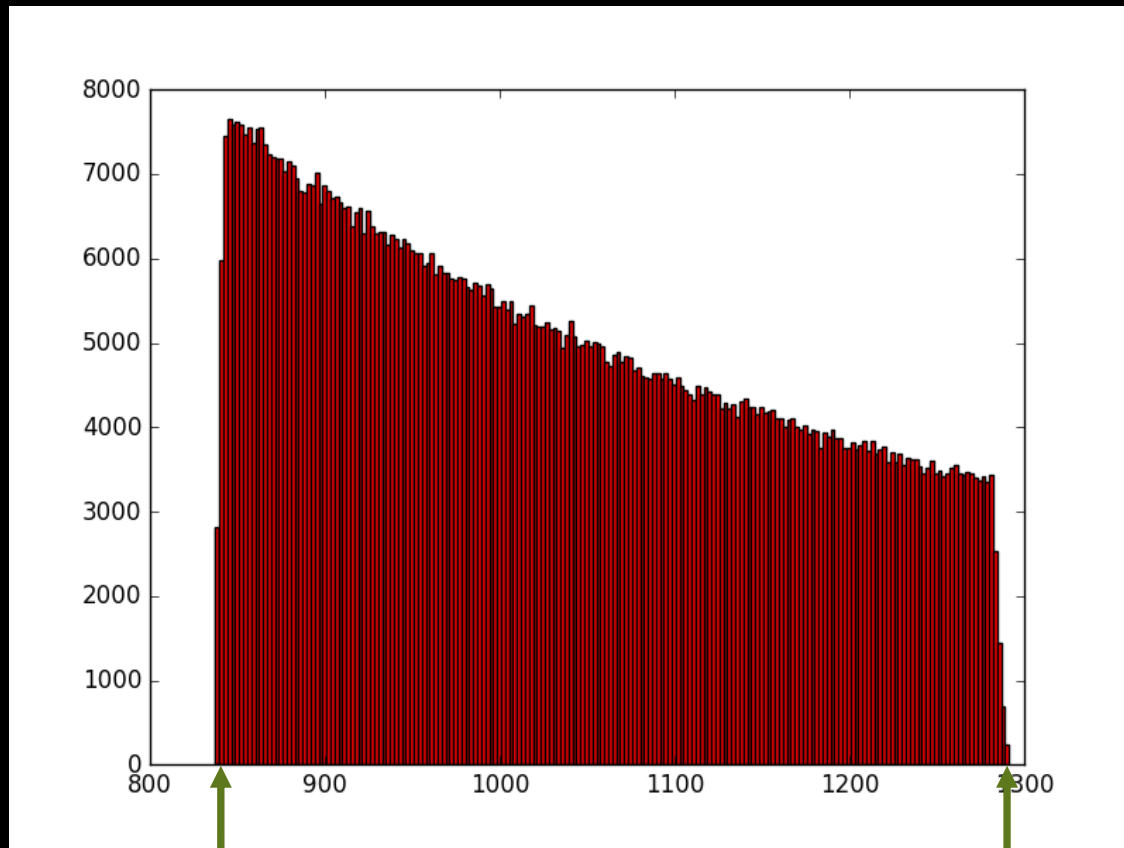
au lieu de 1015,2

au lieu de 123

$$f_0 = 1030 \text{ Hz}$$

$$u(f_0) = 130 \text{ Hz}$$

C'est la limite de la méthode analytique quand les incertitudes relatives sont trop grandes et la fonction f « fortement » non linéaire



837 Hz

1292 Hz

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

PRÉSENTATION DES RÉSULTATS

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 le mesure donne $x = 7,3^\circ$ avec $u(X) = 1,1^\circ$ (2 chiffres significatifs). Incertitude sur $Y = \cos(X)$?

$$u(Y) = \left| \frac{\cos(8,4^\circ) - \cos(6,2^\circ)}{2} \right| = 2,4 \cdot 10^{-3}$$

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

EXERCICE



Utiliser un script python fourni pour obtenir l'histogramme, la moyenne et l'écart-type de Y pour une distribution de X de moyenne $7,3^\circ$ et d'écart-type $1,1^\circ$:

```
>>> uniforme
```

```
>>> gaussienne
```

Conclure

Les incertitudes sont elles-mêmes entachées d'incertitudes

⇒ un seul chiffre pour l'incertitude-type $u(X)$?

Faux ! Cela pose un problème d'arrondi qui va se propager lors des calculs utilisant $u(X)$

Exemple :

$x_1 = 2,50$	$x_2 = 2,68$
$u(x_1) = 0,11$	$u(x_2) = 0,13$

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

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

Si l'on ne gardait d'emblée qu'un seul chiffre : $x_1 = 2,5$ $x_2 = 2,6$
 $u(x_1) = 0,1$ $u(x_2) = 0,1$

⇒ $u(X) = \sqrt{0,1^2 + 0,1^2} \approx 0,14$ $x = 0,2 \quad u(X) = 0,1$

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 :

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


ÉCARTS NORMALISÉS (Z-SCORE) / PRISE DE DÉCISION

Comparaison entre une valeur mesurée et une valeur de référence

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

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

— Sa valeur moyenne est $\bar{Z} = \frac{\bar{X} - X_{\text{réf}}}{u(X)} = 0$

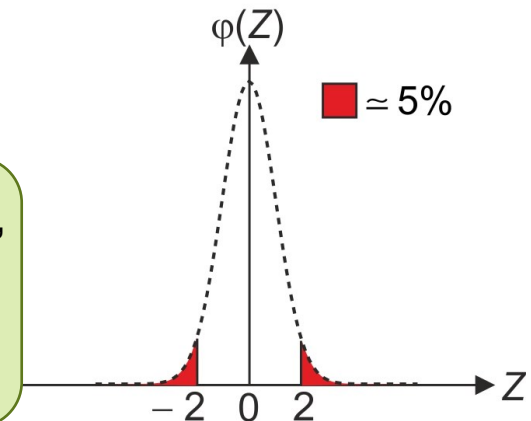
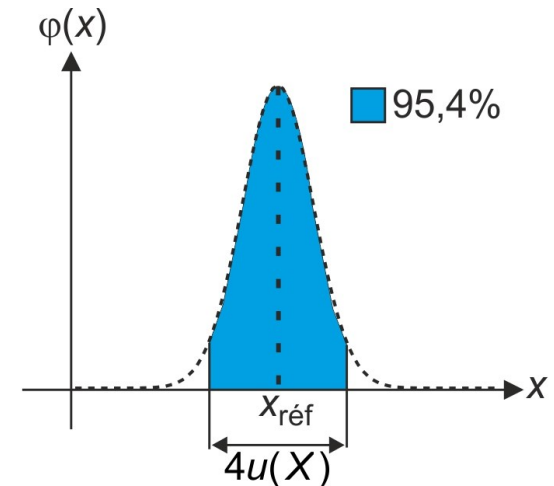
— Son écart-type est $u(Z) = \frac{u(X - X_{\text{réf}})}{u(X)} = \frac{u(X)}{u(X)} = 1$

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



On doit 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

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



EXERCICE

Résolution d'un instrument et incertitudes

Une grandeur X est mesurée par un instrument. La valeur moyenne obtenue est $x = 20,1$ et l'écart-type $\sigma = 1,0$.

Cependant, l'affichage ne produit que des valeurs entières de la grandeur mesurée, égales à sa partie entière.

L'incertitude-type des mesures fournies par l'instrument peut-elle être inférieure à la résolution (ici égale à 1) ?

Sous python, on prend la valeur entière de g avec `np.round(g)`

INCERTITUDES MIXTES

Distance focale d'une lentille convergente

On fait varier la position de la lentille par rapport à l'objet et on déplace l'écran afin d'obtenir l'image la plus nette possible

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 : $f' = \frac{pp'}{p - p'}$

Avec pour chaque valeur une incertitude-type : $u(f') = \left(\frac{f'}{p'}\right)^2 u(p') = \frac{1}{\sqrt{3}} \left(\frac{f'}{p'}\right)^2 \Delta p'$

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):
    return p * pp / (p - pp)

def sigma_fp(fp, pp, sigma_pp):
    return (fp / pp) ** 2 * sigma_pp

Fp = fp(P, Pp)
sigma_Fp = sigma_fp(Fp, Pp, sigma_Pp)

>>> calcul_mu_sigma(Fp)

(0.20064736446152062, 0.00017781409869729436)
```



(2) On prend la moyenne arithmétique des mesures et de leurs incertitudes-type :

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

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

On trouve :

(0.20064736446152062, 0.00010664878963662461)

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

(3) Hors-programme mais le plus rigoureux. On pondère par les inverses des variances, donnant plus de poids aux valeurs de focale 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 :

(0.20067836305879627, 9.4151482137413259e-05)



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

AJUSTEMENT LINÉAIRE OU AFFINE

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

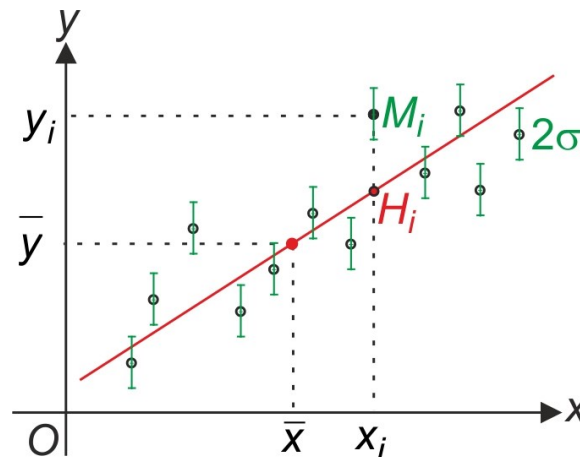
La relation $Y = aX + b$ est-elle bien vérifiée ?

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

On mesure $(x_i, y_i) \quad i \in 1, N$ avec $(u(X_i), u(Y_i)) \quad i \in 1, N$

Les hypothèse de la régression linéaire ordinaire :

- 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



Maximum de vraisemblance : $\chi^2 = \frac{1}{\sigma^2} \sum_{i=1}^N [y_i - f(x_i)]^2 = \frac{S}{\sigma^2}$ est minimal

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

Méthode analytique : on montre (hors-programme) :

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

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



```

def AjustAff(Lx,Ly,sigma,fig = True):
    N = len(Lx)
    if len(Ly) != N:
        return 'listes de longueurs differentes'
    Sx,Sy,Sxx,Syy,Sxy,S = 0,0,0,0,0,0
    for i in range(N):
        Sx += Lx[i]
        Sy += Ly[i]
        Sxx += Lx[i]**2
        Syy += Ly[i]**2
        Sxy += Lx[i]*Ly[i]
    xmoy, ymoy , xxmoy , yymoy xymoy = Sx/N, Sy/N, Sxx/N, Syy/N, Sxy/N
    a = (xymoy-xmoy*ymoy) / (xxmoy-xmoy**2)
    b = ymoy-a*xmoy
    if sigma == 'inconnu':
        for i in range(N):
            S += (Ly[i]-a*Lx[i]-b)**2
        sigma = (S/(N-2))**.5
        sigmaa = ((S/N/(N-2)) / (xxmoy-xmoy**2))**.5
        sigmab = sigmaa*(xxmoy)**.5
    else:
        sigmaa = sigma*1/(N*(xxmoy-xmoy**2))**.5
        sigmab = sigma*(xxmoy/N/(xxmoy-xmoy**2))**.5

```

```
if fig == True:
    X = [Lx[0],Lx[-1]]
    Y = [a*Lx[0]+b,a*Lx[-1]+b]
    plt.figure('points et ajustement affine')
    plt.plot(Lx,Ly, 'o', label = u'points expérimentaux',
color = 'red')
    plt.errorbar(Lx, Ly, yerr = sigma*np.ones(N), fmt = 'm.',ecolor =
'black',elinewidth = 3)
    plt.xlabel('$X$')
    plt.ylabel('$Y$')
    plt.plot(X,Y, "r-", label = u'régression lineaire',color = 'b',alpha =
0.6)
    plt.show()

    Z = (Ly-(a*Lx+b))/sigma # calcul des résidus
    plt.figure('résidus normalisés')
    plt.plot(Lx, Z,'o',color = 'red')
    plt.fill_between([np.min(Lx),np.max(Lx)], y1 = -2, y2 = 2, color =
'green', alpha = .1)
    plt.xlim(.9*np.min(Lx), 1.02*np.max(Lx))
    plt.xlabel('$X$')
    plt.ylabel('résidus normalisés de $Y$' ), plt.ylim(-3,3)
    plt.ticklabel_format(axis = 'x', style = 'sci', scilimits = (0,0))
    plt.grid()
    plt.show()

return [a,b,sigmaa,sigmab]
```



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

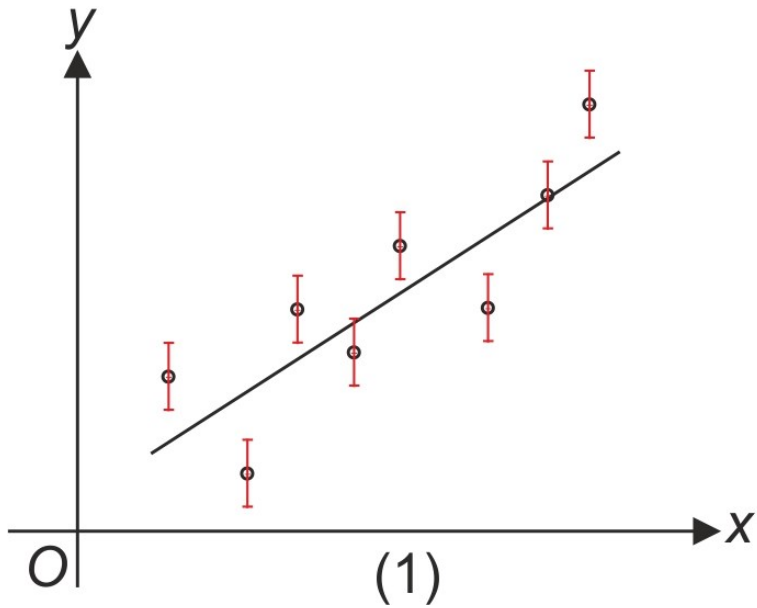
loi affine : on montre que σ est estimé par $\sigma^* = \sqrt{\frac{S}{N-2}}$

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

Quand on demande aux étudiants 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

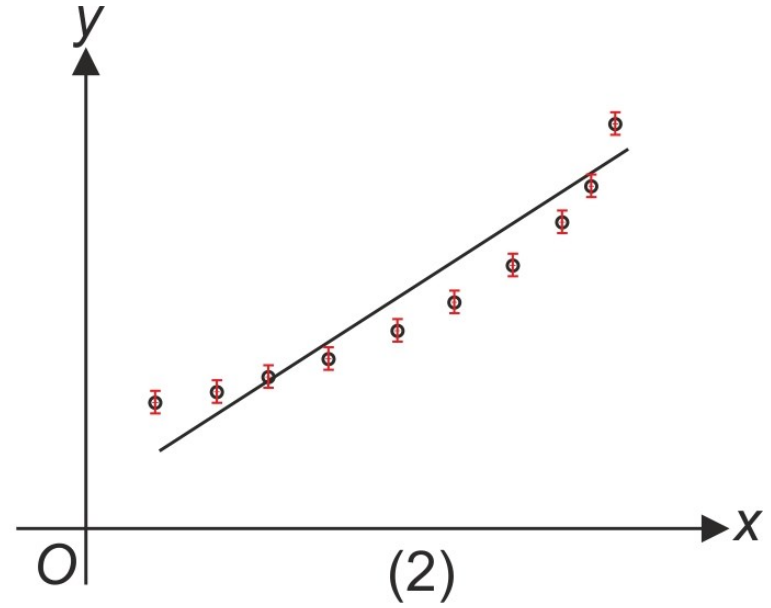
L'usage est de postuler que la loi affine théorique est bien vérifiée par l'expérience si le coefficient de corrélation linéaire au carré est très proche de 1

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

Ajustement affine avec pondération (hors-programme)

Le plus souvent (propagation des incertitudes) : $u(Y_i) = \sigma_i \neq u(Y_j) = \sigma_j$

$$\Rightarrow \chi^2 = \sum_{i=1}^N \left[\frac{y_i - f(x_i)}{\sigma_i} \right]^2 = \sum_{i=1}^N w_i [y_i - f(x_i)]^2$$

$$w_i = \frac{1}{\sigma_i^2} \quad \text{inverse de la variance}$$

$$\bar{g} = \frac{\sum_{i=1}^N w_i g_i}{\sum_{i=1}^N w_i} = \frac{\sum_{i=1}^N w_i g_i}{w} \quad \text{moyenne pondérée, poids « fort » lorsque les incertitudes sont petites}$$

On reprend les calculs...

$$a^* = \frac{\overline{xy} - \bar{x} \bar{y}}{\overline{x^2} - \bar{x}^2} \quad u(a) = \frac{1}{\sqrt{w(\overline{x^2} - \bar{x}^2)}}$$

$$b^* = \bar{y} - a^* \bar{x} \quad u(b) = \sqrt{\frac{\overline{x^2}}{w(\overline{x^2} - \bar{x}^2)}}$$

Aucune difficulté supplémentaire de programmation

Ajustement linéaire

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, avec ou sans pondération... et on adapte le code Python

Sans pondération :

$$a^* = \frac{\overline{xy}}{\overline{x^2}}$$
$$u(a) = \frac{\sigma}{\sqrt{N\overline{x^2}}}$$

Exemple : nombre de traits d'un réseau. Mesure de la déviation sous incidence normale dans l'ordre 1 pour Hg

$\lambda(\text{nm})$	couleur	$i'(^{\circ})$	$n = \frac{\sin i'}{\lambda}$ (traits/mm)	$u(n)$
404,7	violet intense	13,35	570,5	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 $u(n)$

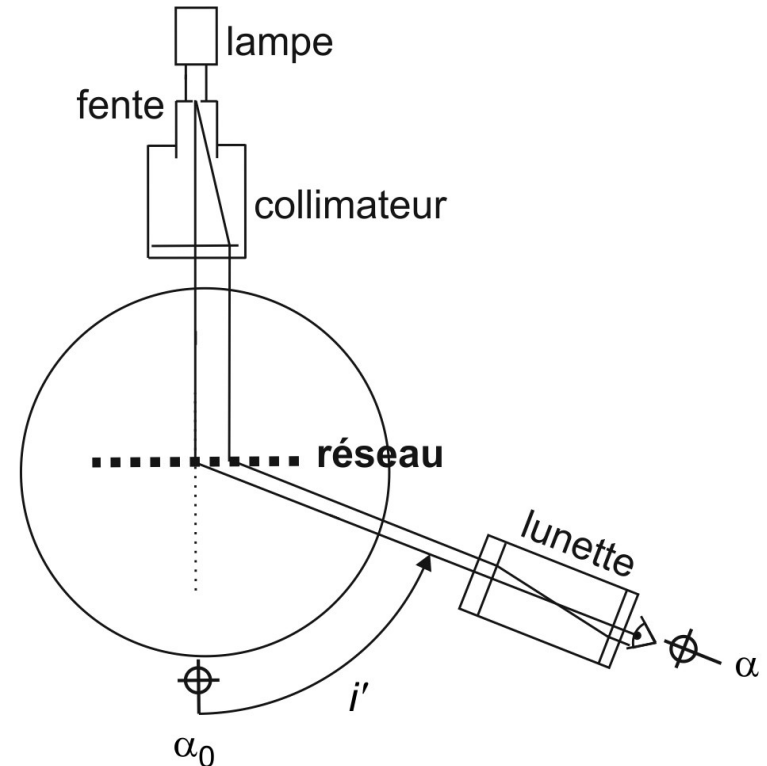
Les angles α sont mesurés au vernier avec :

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

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

λ ordre 1 ordre 0

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



Résultats

	a	$u(a)$	b	$u(b)$
Affine	570027	1028	9E-5	5E-4
Linéaire	570200	159		
Affine Pondéré	570001	1010	2E-4	5E-4
Linéaire Pondéré	570199	156		

Conclusion :

>>> l'incertitude est surévaluée quand le modèle n'est pas adapté
 >>> on peut éviter la pondération

$$\sin i' = n \cdot \lambda$$

Y a 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

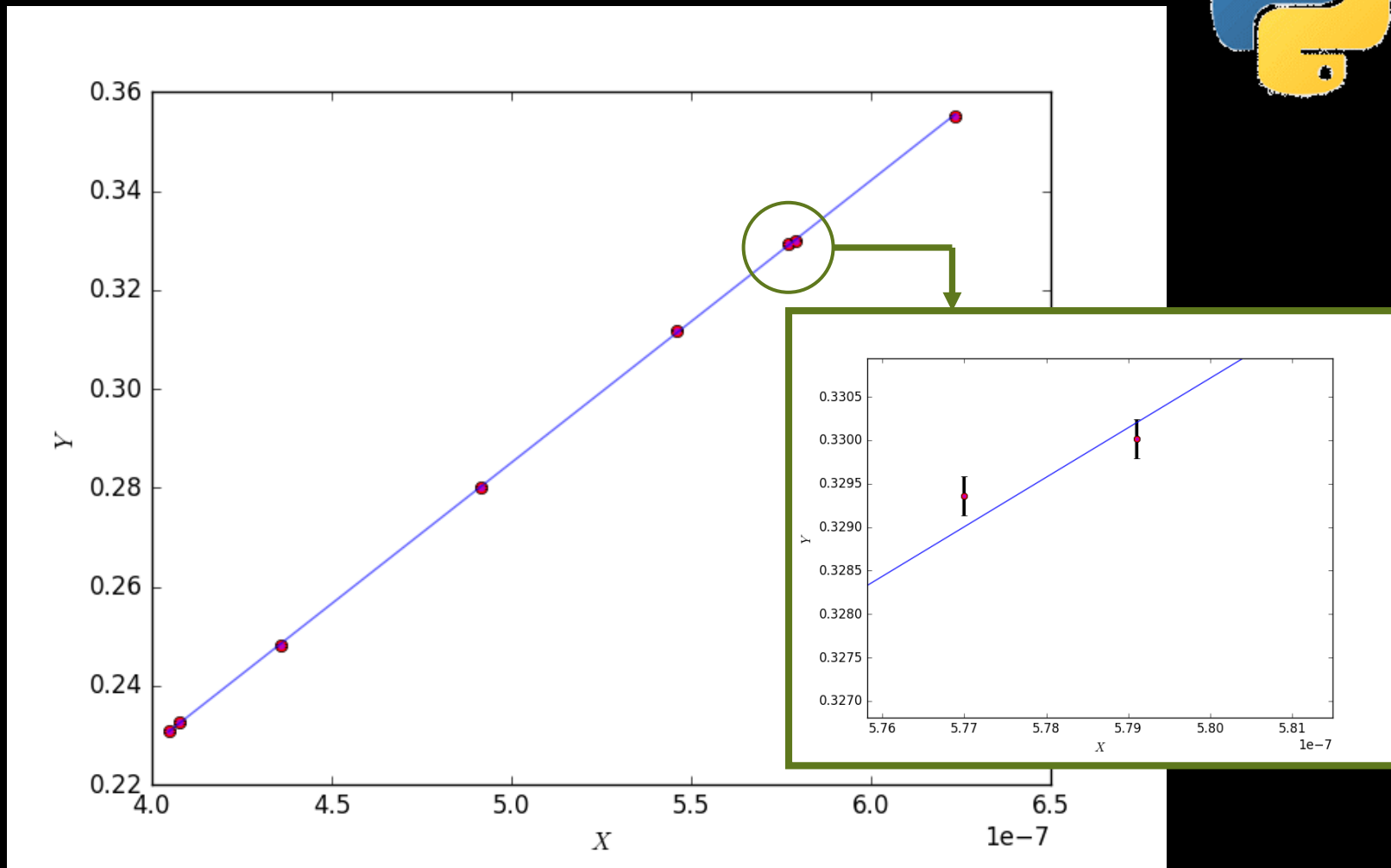
L = np.array([13.35,13.45,14.37,16.27,18.17,19.23,19.27,
              20.8])*np.pi/180 # les angles transformés en radians

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

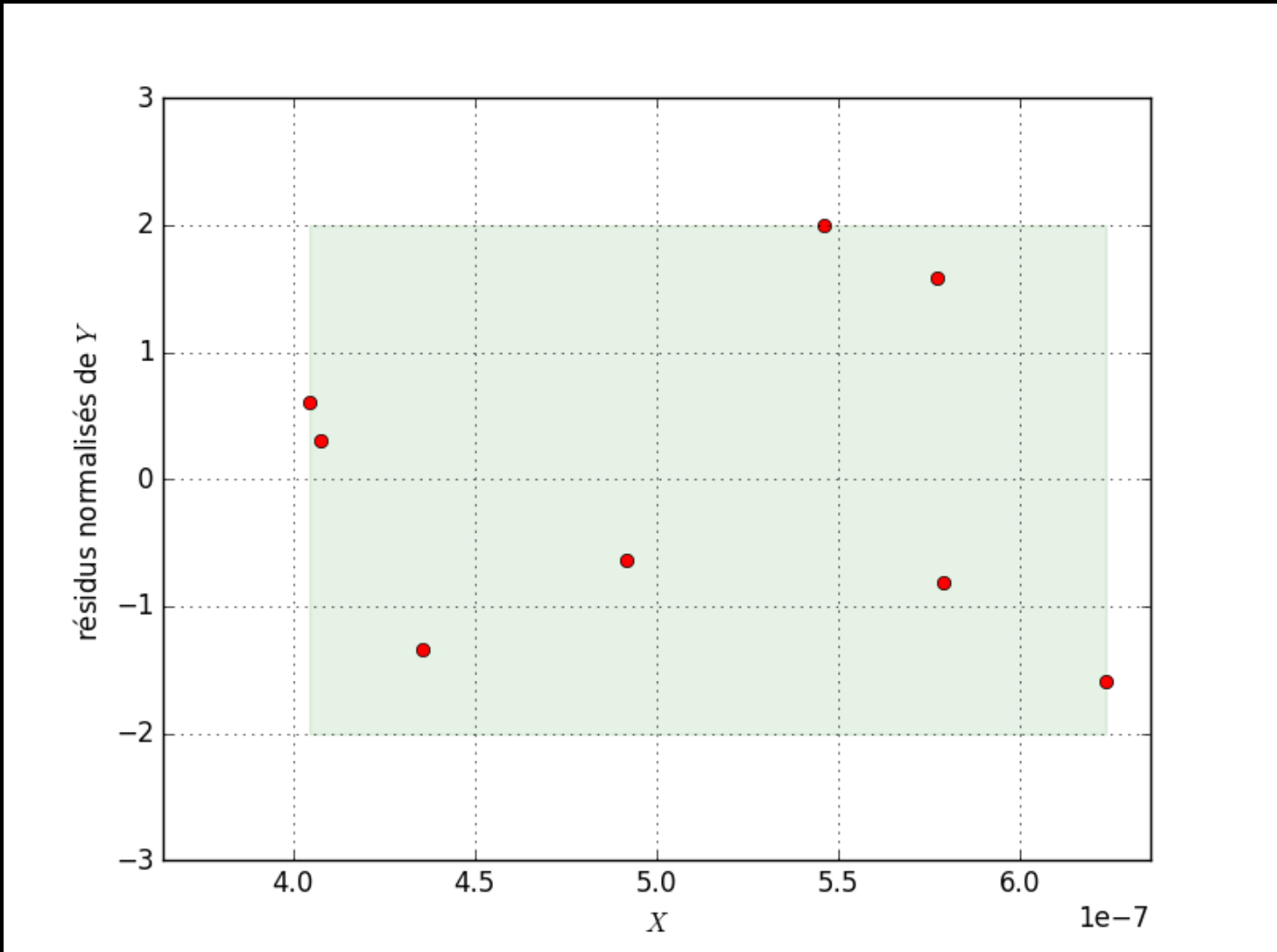
sigma = 2,37E-4 # écart-type sur les angles

sigmaY = np.cos(L)*sigma # propagation des incertitudes sur les
angles vers Y = sin(i') (les incertitudes dépendent de la
longueur d'onde)
```

AjustLinPond (Lx, Ly, sigmaY)



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



Le modèle linéaire est bien vérifié ici

Simulation de Monte-Carlo

On mesure $(x_i, y_i) \quad i \in 1, N$ avec $u(Y_i) = \sigma$ pour $i \in 1, N$

- (1) On calcule a^* et b^* grâce à une régression linéaire. La droite d'ajustement a pour équation $Y_{\text{ajusté}} = a^* X + b^*$
- (2) On simule N jeux de mesures $(x_i^j = x_i, y_i^j = a^* x_i + b^* + \varepsilon_i^j)$ où ε_i suit la distribution choisie centrée sur 0 et d'incertitude-type σ
- (3) On effectue une régression linéaire sur chaque jeu de mesures et on obtient N valeurs des coefficients de la droite d'ajustement $(a_j, b_j) \quad j \in 1, N$
- (4) On en déduit la valeur de l'écart-type sur a^* et b^*


```
def AjustLin_MC(Lx,Ly,sigma):  
    p0 = AjustAff(Lx,Ly,sigma,False)  
    a0,b0 = p0[0],p0[1] # valeurs des coefficients basés sur la  
    réalisation expérimentale  
  
    def ajustement(x):  
        return a0*x+b0  
  
    Ly_ajuste = ajustement(Lx) # valeurs autour desquelles ont  
    simule des mesures  
  
    N = 100000 # nombre d'expériences simulées, pas le nombre de  
    mesures...  
    a_MC, b_MC = np.zeros(N),np.zeros(N) # initialisation des  
    listes dans lesquelles on va stocker les pentes et ordonnées à  
    l'origine simulées  
    for i in range(N): # pour chaque expérience  
# on simule la mesure de Y pour toutes les valeurs de X  
        Ly_MC = Ly_ajuste + rd.normal(0, sigma, size = len(Lx))  
# on reprend l'ajustement affine avec cette série de valeurs de n  
        p = AjustAff(Lx,Ly_MC,sigma,False)  
# on stocke les valeurs des paramètres d'ajustement dans les  
listes a_MC et b_MC  
        a_MC[i] = p[0]  
        b_MC[i] = p[1]
```



```
# les résultats
# pour a
a_moy = np.mean(a_MC)
u_a = np.std(a_MC, ddof = 1)
print("Estimation du paramètre a :")
print("Valeur mesurée : a = {}".format(a_moy))
print("Incertitude-type : u(a) = {}\n".format(u_a))
# pour b
b_moy = np.mean(b_MC)
u_b = np.std(b_MC, ddof = 1)
print("Estimation du paramètre b :")
print("Valeur mesurée : b = {}".format(b_moy))
print("Incertitude-type : u(b) = {}\n".format(u_b))

# les graphiques
plt.figure(figsize = (15, 4))
plt.subplot(1, 2, 1)
plt.hist(a_MC, bins = 'rice', color = 'green')
plt.xlabel(r"Valeurs de $a$ obtenues")
plt.ylabel(r"Nombre d'apparitions")
plt.subplot(1, 2, 2)
plt.hist(b_MC, bins = 'rice', color = 'grey')
plt.xlabel(r"Valeurs de $b$ obtenues")
plt.ylabel(r"Nombre d'apparitions")
plt.show()
```

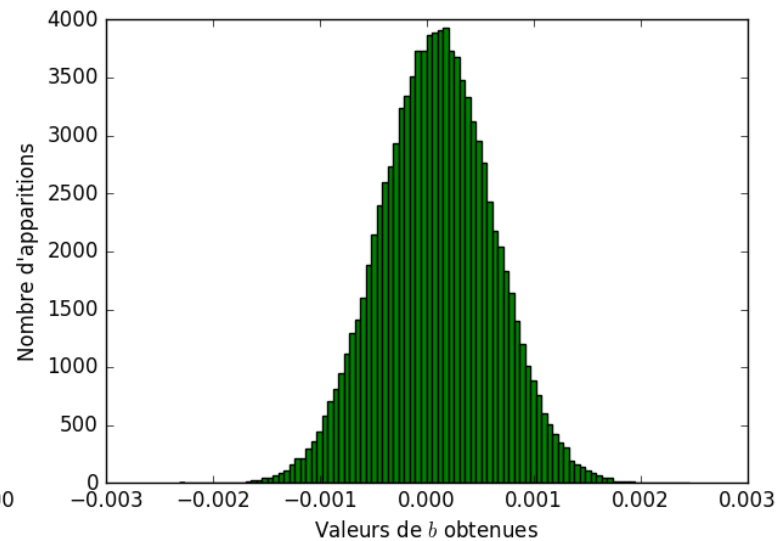
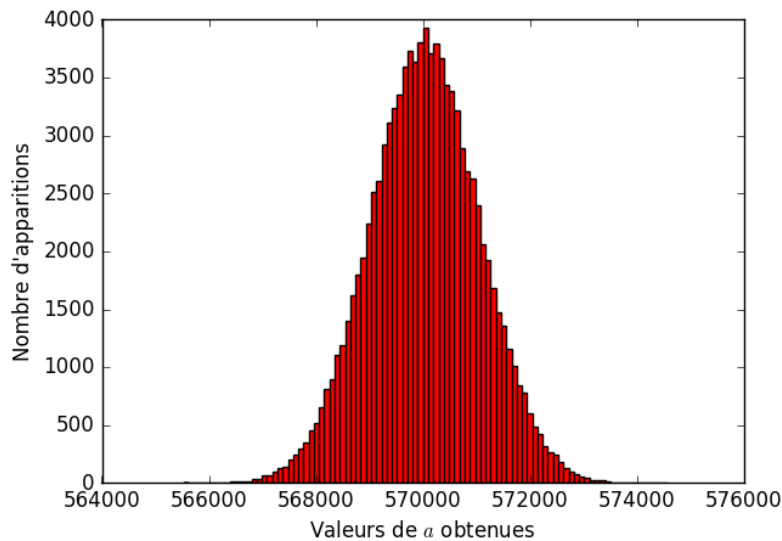


On reprend l'exemple du réseau : ajustement affine sans pondération
 On avait trouvé par la méthode analytique :



$$a = 570027 \quad u(a) = 1028.1241546853,$$

$$b = 9.04e-05, \quad u(b) = 0.000529$$



Estimation du paramètre a :

Valeur mesurée : $a = 570032.687531272$

Incertitude-type : $u(a) = 1027.4654153606768$

Estimation du paramètre b :

Valeur mesurée : $b = 8.710739535013986e-05$

Incertitude-type : $u(b) = 0.000528558911264731$

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 ...

