Utilisation du plug-in CodeRunner pour le calcul des incertitudes en Travaux Pratiques

F. Kany, A. Bourgeois

ENGINEERING SCHOOL
Creating the future together

Plan I

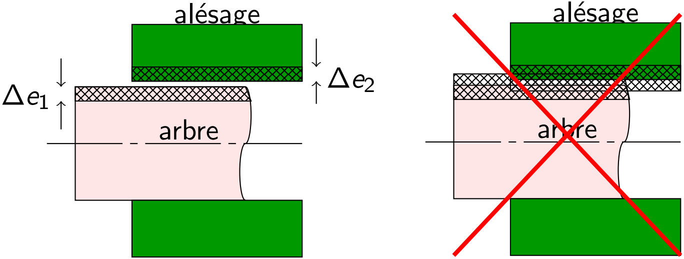
1. Problème de la mesure en TP (physique, chimie, SI, SVT)
2. Smart metrology
3. Comment calculer les incertitudes expérimentales avec Moodle ?
4. Comment valider le résultat d'une mesure avec Moodle ?

Avant :

- Mesure de quantités : x_1, x_2, ...
- Estimation grossière des erreurs de mesure : Δx_1, Δx_2,...
- Calcul d'une valeur souhaitée : $f(x_1, x_2, \dots)$
- Calcul de la propagation d'erreur par la méthode des dérivées partielles :
  $\Delta f = \left|\frac{\partial f}{\partial x_1}\right| .\mathrm{d}x_1 + \left|\frac{\partial f}{\partial x_2}\right| .\mathrm{d}x_2 + \dots$
- Compensation des risques de sous-estimation des Δx_1, Δx_2,...par une augmentation des facteurs de sécurité : $f(x_1, x_2, \dots) \to s_f.f(x_1, x_2, \dots)$ avec s_f safety factor

Problème : appliquée à l'industrie, cette méthode conduit à :

- une mise au rebut de pièces qui auraient pu être validées (gaspillage, surcoût)



- une surestimation du facteur de sécurité
  - $\Rightarrow$ surcoût : matière première, production, maintenance préventive
  - $\Rightarrow$ nouveaux problèmes !

Causes

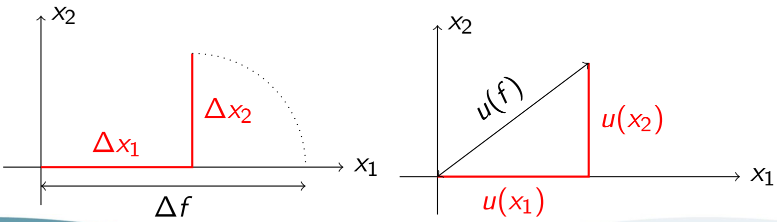
- Méthode des dérivées partielles pas justifiée d'un point de vue statistique
- La propagation d'erreur sur-estime l'erreur sur Δf

Exemple : $f(x_1, x_2) = x_1 \pm x_2$

$$\Delta f = \left|\frac{\partial f}{\partial x_1}\right|.\Delta x_1 + \left|\frac{\partial f}{\partial x_2}\right|.\Delta x_2 = 1.\Delta x_1 + 1.\Delta x_2$$

alors que statistiquement :

$$u(f) = \sqrt{(u(x_1))^2 + (u(x_2))^2} < \Delta f$$



Smart metrology

- Joint Committee for Guides in Metrology (BIPM), Evaluation of measurement data — Guide to the expression of uncertainty in measurement, 2008, `https://www.bipm.org/utils/common/documents/jcgm/JCGM1002008E.pdf`
- ISO 9001 :2015 requirements for a quality management system `https://www.iso.org/standard/62085.htm`

erreur	$\Rightarrow$	incertitude
propagation des erreurs	$\Rightarrow$	composition des incertitudes

Nouveau paradigme II

Analyse statistique fine (incertitude = écart-type).

Cas simples :

Relation	Incertitude (à l'ordre 1)
$f(x) = \lambda.x$ avec λ constante	$u(f) = \lvert\lambda\rvert.u(x)$
$f(x_1, x_2) = x_1 + x_2$ ou $f(x_1, x_2) = x_1 - x_2$	$u(f)^2 = u(x_1)^2 + u(x_2)^2$
$f(x_1, x_2) = x_1.x_2$ ou $f(x_1, x_2) = x_1/x_2$	$\left(\frac{u(f)}{f}\right)^2 = \left(\frac{u(x_1)}{x_1}\right)^2 + \left(\frac{u(x_2)}{x_2}\right)^2$
$f(x_1, x_2) = \lambda.x_1^a.x_2^b$ avec λ constante	$\left(\frac{u(f)}{f}\right)^2 = a^2.\left(\frac{u(x_1)}{x_1}\right)^2 + b^2.\left(\frac{u(x_2)}{x_2}\right)^2$

Nouveau paradigme III

Difficulté :
dans ces calculs, x_1 et x_2 doivent être des VAR indépendantes.
Exemple :
arête d'un carré : $a \pm u(a)$
surface du carré ? $a^2 \pm u(a^2)$

Relation	Incertitude (à l'ordre 1)
$f(x_1, x_2) = x_1.x_2$	$\left(\frac{u(f)}{f}\right)^2 = \left(\frac{u(x_1)}{x_1}\right)^2 + \left(\frac{u(x_2)}{x_2}\right)^2$ $u(a \times a) \neq a.\sqrt{2.\left(\frac{u(a)}{a}\right)^2} = \sqrt{2}.u(a)$
$f(x_1, x_2) = \lambda.x_1^a.x_2^b$ avec λ constante	$\left(\frac{u(f)}{f}\right)^2 = a^2.\left(\frac{u(x_1)}{x_1}\right)^2 + b^2.\left(\frac{u(x_2)}{x_2}\right)^2$ $u(a^2) = a^2.\sqrt{4.\left(\frac{u(a)}{a}\right)^2} = 2.u(a)$

Nouveau paradigme IV

En dehors des cas simples :

- quand c'est possible : se ramener aux formules "simples"
  Ex calorimétrie :
  - $T_f = \frac{m_1.c_{p1}^{mass}.T_{i1}+m_2.c_{p2}^{mass}.T_{i2}}{m_1.c_{p1}^{mass}+m_2.c_{p2}^{mass}}$
  - incertitudes :
    $u(T_{i1})$, $u(T_{i2})$, $u(m_1)$, $u(m_2)$, $u(c_{p1}^{mass})$ et $u(c_{p2}^{mass})$
  - composition des incertitudes $u(T_f)$: cauchemar !
    $\Rightarrow$ décourage les élèves
- cas général : pas de formule !

Alternative à l'utilisation des formules I

Simulation de Monte-Carlo :

- on simule le tirage aléatoire de
  N valeurs de x_1,
  N valeurs de x_2, ...
- on calcule les N valeurs de $y = f(x_1, x_2, \dots)$
- on calcule la moyenne $\bar{y}$
  et l'écart-type σ_{N-1} des N valeurs de y

Intérêts :

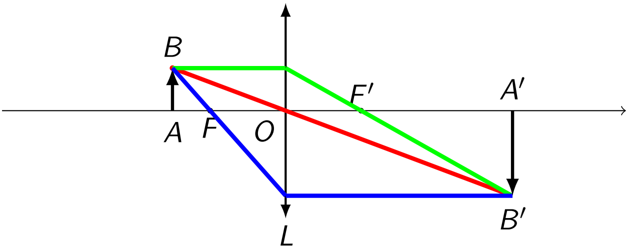
- code simple (toujours la même structure)
- code rapide (σ_{N-1} avec 2 chiffres significatifs avec $N = 1000$)
- code + précis (formules "simples" seulement valables à l'ordre 1)

Alternative à l'utilisation des formules II

Types de tirage aléatoire :

- Incertitude de type A
  incertitude liée à la non-répétabilité de *n mesures* de x :
  $u(x) = \frac{\sigma_{n-1}}{\sqrt{n}}$ avec $\sigma_{n-1} = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})}{n-1}}$ (intervalle de confiance à 68%)
  $\Rightarrow$ loi normale (Gaussienne)
- Incertitude de type B
  incertitude liée à la précision de l'instrument lors d'une mesure de x :
  $u(x) = \frac{\Delta}{\sqrt{3}}$ avec Δ la demi-largeur de l'intervalle $[x_{min}, x_{max}]$
  i.e. $\Delta = \frac{x_{max} - x_{min}}{2}$.
  $\Rightarrow$ loi uniforme

Exemple 1



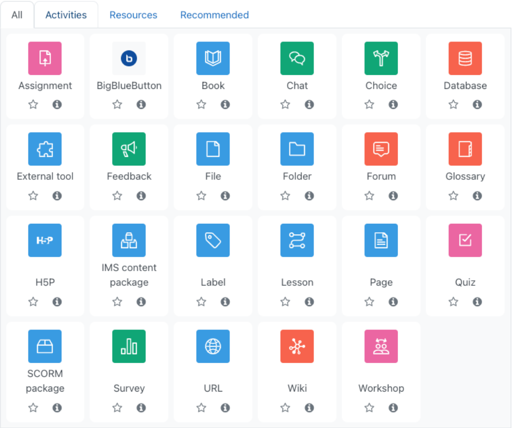
1. mesurer $\overline{OA}$;
2. estimer $u(\overline{OA})$
   (incertitude de type B liée à la graduation du banc)
3. mesurer $\overline{OA'}$;
4. estimer $u(\overline{OA'})$
   (incertitude de type B liée à la mise au point + graduation du banc)
5. calculer f' d'après la formule de Descartes $\frac{1}{\overline{OA'}} - \frac{1}{\overline{OA}} = \frac{1}{f'}$
6. calculer $u(f')$

Sans Moodle I

- $\overline{OA} \in [\overline{OA}_{min}, \overline{OA}_{max}]$
- $\Delta = \frac{1}{2}.(\overline{OA}_{max} - \overline{OA}_{min})$ (demi-étendue de l'intervalle)
- $u(\overline{OA}) = \frac{\Delta}{\sqrt{3}}$ (écart-type de la loi uniforme)
- $u(\frac{1}{\overline{OA}}) = \frac{1}{\overline{OA}}.\frac{u(\overline{OA})}{\overline{OA}} = \frac{\Delta}{\sqrt{3}.\overline{OA}^2}$
- $\overline{OA'} \in [\overline{OA'}_{min}, \overline{OA'}_{max}]$
- $\ldots \Rightarrow u(\frac{1}{\overline{OA'}}) = \frac{\Delta'}{\sqrt{3}.\overline{OA'}^2}$
- $\frac{1}{f'} = \frac{1}{\overline{OA'}} - \frac{1}{\overline{OA}}$
- $u(\frac{1}{f'}) = \sqrt{\left(u(\frac{1}{\overline{OA}})\right)^2 + \left(u(\frac{1}{\overline{OA'}})\right)^2}$
- $u(f') = f'^2.u(\frac{1}{f'}) = f'^2.\sqrt{\frac{\Delta^2}{3.\overline{OA}^4} + \frac{\Delta'^2}{3.\overline{OA'}^4}}$

Utilisation de Moodle I

Création d'un "Quiz" ("Test")

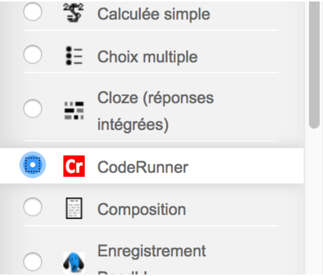


Utilisation de Moodle II

1. onglet **Paramètres**
   - Note
     - Nombre de tentatives autorisées : illimité
     - Méthode d'évaluation : note la plus haute
   - Option de relecture
     - Immédiatement & Plus tard
       - La tentative
       - Points

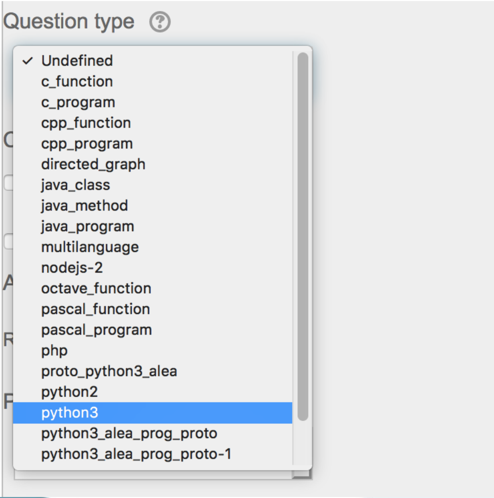
Utilisation de Moodle III

2. onglet **Banques de questions**
   - Catégories / Ajouter une catégorie : “Monte-Carlo”
3. onglet **Questions**
   3.1 Ajout d’une question
   3.2 Type de question : Code Runner



Utilisation de Moodle IV

3.3 CodeRunner question type :



Utilisation de Moodle V

3.4 Généraux :

- Catégorie : Monte-Carlo
- Texte de la question : *à remplir*

```
On mesure la position de l'objet \(\overline{OA}\);
on estime cette valeur entre \(x_{min}\) et \(x_{max}\).

On mesure la position de l'image \(\overline{OA'}\);
on estime cette valeur entre \(x'_{min}\) et \(x'_{max}\).

Ecrire une fonction
conjugaison(x_min, x_max, x_prime_min, x_prime_max)
qui renvoie l'estimation de \(f'\)
et son incertitude \(u(f')\).
```

3.5 Réponse pré-remplie : *à remplir*

```
import numpy.random as rd
import numpy as np

def conjugaison(x_min, x_max, x_prime_min, x_prime_max):
    N = 1000 #nbre de simulations
    x = ...
    x_prime = ...
    f_prime = ...
    f_prime_estime = ...
    u_f_prime = ...
    return f_prime_estime, u_f_prime
```

3.6 Test cases

3.6.1 Test case 1

```
f,u = conjugaison(-1.1,-0.9,0.9,1.1)
print(abs((f-0.5)/f)<5e-2 and abs(u-0.0206)/u<1e-1)
```

3.6.2 Sortie attendue

```
True
```

Utilisation de Moodle VII

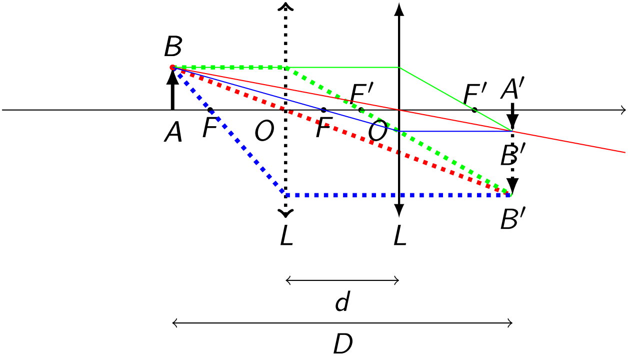
Solution :

```
import numpy.random as rd
import numpy as np

def conjugaison(x_min, x_max, x_prime_min, x_prime_max):
    N = 1000 #nbre de simulations
    x = rd.uniform(x_min,x_max,N) #type B
    x_prime = rd.uniform(x_prime_min,x_prime_max,N) #type
    f_prime = x*x_prime/(x-x_prime)
    f_prime_estime = np.average(f_prime)
    u_f_prime = np.std(f_prime,ddof=1)
    return f_prime_estime, u_f_prime
```

Exemple 2

Méthode de Bessel



1. mesurer d et D, estimer $u(d)$ et $u(D)$
2. calculer f' d'après la formule de Bessel : $f' = \frac{D^2 - d^2}{4.D}$
3. calculer $u(f')$

Exemple 2 : sans Moodle

- $D \in [D_{min}, D_{max}]$
- $\Delta = \frac{1}{2}.(D_{max} - D_{min})$ (demi-étendue de l'intervalle)
- $u(D) = \frac{\Delta}{\sqrt{3}}$
- $d \in [d_{min}, d_{max}]$
- $\delta = \frac{1}{2}.(d_{max} - d_{min})$ (demi-étendue de l'intervalle)
- $u(d) = \frac{\delta}{\sqrt{3}}$
- $u(D^2) = 2.D.u(D)\,;\; u(d^2) = 2.d.u(d)$
- $u(D^2 - d^2) = \sqrt{4.D^2.(u(D))^2 + 4.d^2.(u(d))^2}$
- $\frac{u(4.D)}{4D} = \frac{u(D)}{D}$
- $\frac{u(f')}{f'} \neq \sqrt{\left(\frac{u(D^2-d^2)}{D^2-d^2}\right)^2 + \left(\frac{u(D)}{D}\right)^2}$

```python
import numpy.random as rd
import numpy as np

def Bessel(d_min, d_max, D_min, D_max):
    N = 1000 #nbre de simulations
    d = rd.uniform(d_min,d_max,N)
    D = rd.uniform(D_min,D_max,N)
    f_prime = (D**2-d**2)/(4*D)
    f_prime_estime = np.average(f_prime)
    u_f_prime = np.std(f_prime,ddof=1)
    return f_prime_estime, u_f_prime
```

Exemple 3

Mesure de g.

Chute libre d'une hauteur h, pendant une durée t : $g = \frac{2.h}{t^2}$
$h = 1$ m, $\frac{u(h)}{h} = 1\%$, $t = 0,452$ s, $u(t) = \frac{2}{30} = 0,067$ s.
Calculer $u(g)$.

Sans Moodle
$g = \frac{2.h}{t^2}$ de la forme $g = \lambda.h.t^{-2}$

$$u(g) = g.\sqrt{\left(\frac{u(h)}{h}\right)^2 + 4.\left(\frac{u(t)}{t}\right)^2} = 2,9 \text{ m/s}^2$$

Exemple 3 : avec Moodle

```
import numpy.random as rd
import numpy as np

def pesanteur(h, u_h, t, u_t):
    N = 1000 #nbre de simulations
    h = rd.normal(h, u_h, N)
    t = rd.normal(t, u_t, N)
    g = 2*h/t**2
    g_estime = np.average(g)
    u_g = np.std(g,ddof=1)
    return g_estime, u_g
```

$u(g) = 3,5$ m.s^{-2}

$+$ précis que formule sans Moodle !

Utilisation des résultats de la simulation I

Ce n'est pas parce que Moodle fait tout qu'il ne faut rien faire !
Garder 2 chiffres significatifs
Exemples :

- $x = 1,00 \pm 0,14$; $y = 1,00 \pm 0,14$;
  $x + y = 2,00 \pm \sqrt{0,14^2 + 0,14^2} = 2,00 \pm 0,20$
  (contre-exemple : $x = 1,0 \pm 0,1$; $y = 1,0 \pm 0,1$;
  $x + y = 2,0 \pm \sqrt{0,1^2 + 0,1^2} = 2,0 \pm 0,1$
  erreur d'arrondi importante).

Utilisation des résultats de la simulation II

- On mesure une longueur L de 10,0 cm avec une règle graduée en millimètre.
  On a $L \in [9,95; 10,05]$ cm (car $\Delta L = 0,5$ mm=0,05 cm).
  Incertitude de type B
  $\Rightarrow L = 10,00 \pm \frac{0{,}05}{\sqrt{3}} = 10,00 \pm 0,0288$ cm.
  On écrit : $10,000 \pm 0,029$ cm (en gardant 2 chiffres significatifs pour l'incertitude).

Utilisation des résultats de la simulation III

- On mesure un angle $A = 82^\circ$ avec $u(\bar{A}) = 1^\circ$
  (où 1° correspond à l'incertitude-type,
  la division de la demi-étendue par $\sqrt{3}$ a déjà été faite).
  Pour écrire $\sin(A)$, on calcule :
  $\sin(A) = 0,990268\ldots$; $\sin(84^\circ) - \sin(83^\circ) = 2 \times 0.024$.
  D'où : $\sin(A) = 0,9903 \pm 0,024$.
  On voit qu'il faut 4 chiffres significatifs pour $\sin(A)$ (pour être cohérent avec les 2 chiffres significatifs de l'incertitude) alors qu'il y avait seulement 2 chiffres significatifs pour A.

Validation de la mesure I

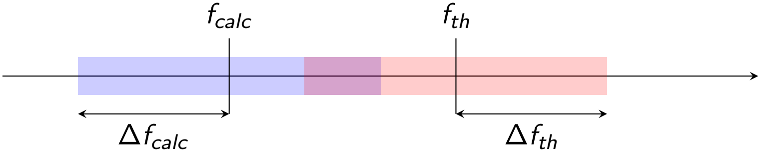
Avant

1. écart relatif à la valeur attendue
   $\varepsilon = \frac{|f_{calc} - f_{th}|}{f_{th}} < 5\% \Rightarrow$ OK
   Ne repose sur RIEN.
   Contre-exemples :
   - $T_{fusion\ exp} = 1\,°C$; $T_{fusion\ th} = 0\,°C$
     $\varepsilon = \infty$
   - $T_{fusion\ exp} = 274$ K ; $T_{fusion\ th} = 273$ K
     ε fini mais était censé ne pas dépendre des unités !
   - $T_{ebullition\ exp} = 101\,°C$; $T_{ebullition\ th} = 100\,°C$
     $\varepsilon = 1\% < 5\%$: validé
     (alors que c'est la même erreur systématique que pour la fusion)
   - équilibre pont de Wheatstone : $u_{exp} = 1$ mV ; $u_{th} = 0$ mV
     $\varepsilon = \infty$ quel que soit le système d'unités !

   ε : AUCUN SENS.

Validation de la mesure II

2. recouvrement des intervallles de confiance



Pas justifié d'un point de vue statistique

La bonne méthode

L'écart normalisé (z-score)

Comparaison d'une valeur expérimentale et d'une valeur théorique	Comparaison de 2 valeurs expérimentales
$z = \frac{\lvert f_{calc} - f_{th} \rvert}{u(f_{calc})}$	$z = \frac{\lvert f_{calc\,1} - f_{calc\,2} \rvert}{\sqrt{u(f_{calc\,1})^2 + u(f_{calc\,2})^2}}$
$\mathcal{H}_0 : \mathbb{E}(f_{calc}) = f_{th}$	$\mathcal{H}_0 : \mathbb{E}(f_{calc\,1}) = \mathbb{E}(f_{calc\,2})$
$\mathcal{H}_1 : \mathbb{E}(f_{calc}) \neq f_{th}$	$\mathcal{H}_1 : \mathbb{E}(f_{calc\,1}) \neq \mathbb{E}(f_{calc\,2})$

Validation de la mesure IV

$$\text{Hypothèses : } \mathcal{H}_0 \text{ et } z \sim \mathcal{N}(0,1)$$
$$\boxed{\mathbb{P}(z > 2) \leq 5\%}$$

Si $z > 2$, il y a moins de 5% de chance que l'écart entre les deux valeurs soit du aux fluctuations statistiques lors du tirage des N valeurs mesurées (qui ont permis de calculer la moyenne f_{calc} et l'écart-type $u(f_{calc})$).

Si $z > 2$: on rejette $\mathcal{H}_0$
(cette procédure de rejet de l'hypothèse $\mathcal{H}_0$ est fiable à 95% ;
il y a 5% de risque de rejeter à tort $\mathcal{H}_0$,
i.e. de valider à tort $\mathcal{H}_1$ alors que les 2 valeurs sont compatibles).

Validation de la mesure V

```
import scipy.stats as stat

alpha = 0.05 #on fixe le risque
p = 1-alpha/2
print(stat.norm.ppf(p)) #on trouve le z limite: 1.9599

z = 2 #on fixe la limite normalisée
cumul = stat.norm.cdf(z)
print((1-cumul)*2*100) #on trouve le risque 4,55%

survie = stat.norm.sf(z)
print(survie*2*100) #on trouve le risque 4,55%
```

(N.B. $\mathbb{P}(z > 5) < \frac{1}{1\,744\,278}$)

Séance de TP

Utilisation de Moodle :

- **Avant la séance**
  - Diffusion du sujet
  - Travail préparatoire (auto-grading)
- **Pendant la séance**
  - Utilisation de CodeRunner pour calcul de $u(f)$ et validation de la mesure $\frac{f-f_{th}}{u(f)} < 2$
- **Après la séance**
  - Dépôt des comptes-rendus
  - Notation des copies

Point de vue des élèves

Feedback : 385 élèves de 1$^{\text{ère}}$ année cycle préparatoire.

- travail préparatoire
  - nombre de tentatives illimitées
  - toujours la même méthode (changer 1 ligne de code)
  5 pts faciles assurés
- CodeRunner au sein de Moodle :
  pas besoin d'installer une distribution Python sur PC
- Possibilité de copier-coller le résultat de la simulation dans le compte-rendu en restant dans Moodle :
  pas besoin de jongler entre 2 environnements de travail