TP0 - Mesures et incertitudes

La correction est réalisée en utilisant Python et la bibliothèque `numpy` qui contient toutes les fonctions utiles. Certains blocs utilisent les morceaux de code précédents. Il faudra donc les exécuter dans l'ordre, soit en utilisant Jupyter Notebook, soit en rassemblant les morceaux dans un seul fichier `.py`, quitte à faire taire certains `print` à l'aide d'un `#` en début de ligne.

```
import numpy as np
```

1. La valeur moyenne est

   $$R_{\text{num}} \approx 82{,}48\,\Omega.$$

   L'écart type expérimental est

   $$s(R_{\text{num}}) \approx 0{,}377\,\Omega.$$

   Rq : la question des chiffres significatifs sera abordée plus tard. Dans tous les calculs suivants, on utilisera la valeur directement issue du calcul pour éviter de cumuler les erreurs d'arrondissage.
   Rq : Dans le code ci-dessous, on utilise le paramètre `ddof=1` de la fonction `std` qui permet de calculer l'écart-type expérimental ($n-1$ au dénominateur) et non l'écart-type (n au dénominateur).

```
mesures = [81.9, 82.5, 82.3, 82.4, 82.6, 82.6, 83.2, 82.3, 82.9, 82.1]
moyenne        = np.mean(mesures)
ecart_type_exp = np.std(mesures, ddof=1)
print("Moyenne                    =", moyenne, "ohm")
print("Écart-type expérimental =", ecart_type_exp, "ohm")
```

2. Ici on a 10 valeurs expérimentales. L'incertitude-type de type A est

   $$u_A(R_{\text{num}}) = \frac{s(R_{\text{num}})}{\sqrt{10}} \approx 0{,}119\,\Omega.$$

```
ua_num = ecart_type_exp / np.sqrt(len(mesures))
print("u_A(R_num) =", ua_num, "ohm")
```

3. L'ohmmètre analogique est plus fidèle que l'ohmmètre numérique. En effet le premier donne toujours la même valeur, alors que le deuxième donne des valeurs qui fluctuent légèrement. L'amplitude de ces fluctuations est donnée par l'écart-type expérimental $s(R_{\text{num}})$. En revanche l'ohmmètre numérique est plus juste que l'ohmmètre analogique car la valeur moyenne obtenue avec le premier est plus proche du résultat de la mesure réalisée avec la méthode quatre fils ($R = (82{,}529 \pm 0{,}005)\,\Omega$). Cette valeur sert de référence car il est dit dans l'énoncé que l'on suppose qu'elle n'est pas soumise à des erreurs systématiques.
4. On utilise la formule du 2.3 concernant les instruments analogiques et on trouve, puisque $\Delta_{\text{grad}} = 1\,\Omega$:

   $$u_B(R_{\text{num}}) \approx 0{,}289\,\Omega.$$

```
delta_grad = 1
ub_ana = delta_grad / (2 * np.sqrt(3))
print("u_B(R_ana) =", ub_ana, "ohm")
```

5. On dispose de la notice de l'instrument, on peut estimer correctement l'incertitude-type de type B pour une mesure avec l'ohmmètre numérique, en utilisant la formule du 2.3 concernant l'incertitude donnée par le constructeur. On trouve

$$u_B(R_{\text{ana}}) \approx 0{,}486\,\Omega.$$

```
delta_cons = moyenne * 0.9 / 100 + 1 * 0.1
ub_num     = delta_cons / np.sqrt(3)
print("u_B(R_num) =", ub_num, "ohm")
```

6. On utilise la formule de la section 2.4 concernant la combinaison des incertitudes-types de type A et B. L'incertitude-type totale pour la mesure réalisée avec l'ohmmètre numérique est

$$u_{\text{tot}} \approx 0{,}501\,\Omega$$

```
u_tot = np.sqrt(ua_num ** 2 + ub_num ** 2)
print("u_tot(R_num) =", u_tot, "ohm")
```

7. Le résultat de la mesure réalisée avec l'ohmmètre numérique s'écrit :

$$R_{\text{num}} = (82{,}5 \pm 0{,}5)\,\Omega.$$

8. Pour comparer **quantitativement** cette valeur à celle obtenue avec la méthode quatre fils, on utilise l'écart normalisé, défini dans la section 4.1. On trouve

$$E_n = 0{,}058.$$

Cet écart est plus petit que 2 (on a même $E_n \ll 2$), les deux mesures sont donc compatibles.

```
r_ref = 82.529 # méthode quatre fils
u_ref = 0.005
r_num = 82.5   # ohmmètre numérique
u_num = 0.5
ecart_normalise = np.abs(r_ref - r_num) / np.sqrt(u_ref**2 + u_num**2)
print("Écart normalisé =", ecart_normalise)
```

9. On se sert à nouveau de la section 2.3 concernant l'incertitude donnée par le constructeur.

Calibre	**Incertitude-type** (Ω)	**Précision** (%)
$100\,\Omega$	0,01	$\sim 0{,}01$
$10\,\text{k}\Omega$	0,06	$\sim 0{,}07$
$1\,\text{M}\Omega$	6	~ 7

Rq : La précision s'obtient simplement en divisant l'incertitude-type par la valeur mesurée.

```
delta_cons_100 = 0.014 * r_num / 100 + 0.007 * 100 / 100
ub_100         = delta_cons_100 / np.sqrt(3)
delta_cons_10k = 0.014 * r_num / 100 + 0.001 * 1e4 / 100
ub_10k         = delta_cons_10k / np.sqrt(3)
delta_cons_1M  = 0.014 * r_num / 100 + 0.001 * 1e6 / 100
ub_1M          = delta_cons_1M / np.sqrt(3)
print("incertitude-type (calibre 100 ohm) =", ub_100, "ohm")
print("incertitude-type (calibre 10 kohm) =", ub_10k, "ohm")
print("incertitude-type (calibre 1 Mohm)  =", ub_1M, "ohm")
```

10. On constate que la valeur de la précision augmente avec le calibre : la mesure est moins précise pour les calibres élevés que pour les faibles calibres. Pour faire une mesure, il convient d'utiliser le plus petit calibre possible pour réaliser la mesure la plus précise possible.
Attention : plus la valeur de la précision est faible, meilleure est la précision de la mesure !

11. On utilise la loi d'ohm :

$$R = \frac{U}{I} = \frac{12{,}1}{35{,}8 \times 10^{-3}} = 338\,\Omega.$$

Les estimations des incertitudes-types sur U et I se font avec le Doc. 2 et l'incertitude-type sur R se calcule avec la formule de propagation des incertitudes-types dans le cas d'un quotient (section 2.5). Le résultat de cette mesure s'écrit :

$$R = (338 \pm 2)\,\Omega.$$

```
u             = 12.1      # tension en volts
delta_cons_u  = 0.15 * u / 100 + 2 * 0.1e-3
u_u           = delta_cons_u / np.sqrt(3)
i             = 35.8e-3   # intensité en ampères
delta_cons_i  = 1 * i / 100 + 3 * 0.01e-3
u_i           = delta_cons_i / np.sqrt(3)
r             = u / i # résistance en ohms
u_r = r * np.sqrt((u_u / u)**2 + (u_i / i)**2)
print("R = ({:.0f} +/- {:.0f}) ohm".format(r, u_r))
```

12. Les trois premières bagues donnent $R_{\text{fab}} = 330\,\Omega$. La dernière nous indique la valeur de $\Delta_{\text{cons}} = R_{\text{fab}} \times 5\,\% = 16{,}5\,\Omega$, d'où $u(R_{\text{fab}}) = \frac{\Delta_{\text{cons}}}{\sqrt{3}} \approx 9{,}53\,\Omega$. Cette valeur s'écrit donc :

$$R_{\text{fab}} = (3{,}3 \pm 0{,}1) \times 10^2\,\Omega.$$

Rq : l'écriture $R_{\text{fab}} = (33 \pm 1) \times 10^1\,\Omega$ est correcte aussi, mais puisque l'utilisation d'une puissance de dix est nécessaire pour respecter les règles d'écriture d'un résultat, la première a l'avantage de respecter la notation scientifique. On préférera donc la première écriture.
On calcule l'écart normalisé :

$$E_n = \frac{|R - R_{\text{fab}}|}{\sqrt{u^2(R) + u^2(R_{\text{fab}})}} \approx 0{,}8.$$

Celui-ci est plus petit que 2, la valeur mesurée est compatible avec la valeur du fabriquant.

13. Le code ci-dessous permet de simuler les mesures réalisées par la méthode quatre fils avec la fonction `np.random.normal` et de les représenter sous la forme d'un histogramme avec la fonction `plt.hist`.

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

#########################
# PRÉPARATION DES DONNÉES
#########################
x0    = 82.529   # moyenne
sigma = 162e-6   # écart-type
n     = 2000     # nombre de mesures
mesures = np.random.normal(x0, sigma, n)

#######################################
# HISTOGRAMME DES DONNÉES EXPÉRIMENTALES
#######################################
plt.hist(mesures)
plt.title("Méthode quatre fils")
plt.xlabel("Résistance en ohms")
plt.ylabel("Fréquence (%)")
plt.show()
```

14. Pour calculer l'écart-type expérimental, il faut faire attention à la manière dont python calcule l'écart-type d'une série de donnée. Par défaut, c'est la formule avec n au dénominateur qui est utilisée et non $n-1$. Il faut donc multiplier le résultat par $\sqrt{\frac{n}{n-1}}$ (ligne 2), ou spécifier la valeur de l'argument optionnel `ddof=1` qui permet de calculer directement l'écart-type expérimental (ligne 3). Ces deux lignes sont bien sûr parfaitement équivalentes, même si la deuxième est plus concise.

```
print("Moyenne    =", np.mean(mesures))
print("Écart-type =", np.sqrt(n / (n - 1)) * np.std(mesures))
print("Écart-type =", np.std(mesures, ddof=1))
```

15. Pour simuler une erreur systématique, il suffit de changer la valeur centrale de la distribution `x_0`.

Pour simuler un appareil moins fidèle, on doit augmenter la dispersion des valeurs, c'est-à-dire augmenter l'écart-type de la distribution `sigma`.