Chapitre EXP 03

Modélisation

Sommaire

Version du 9 novembre 2022.

Ce document a pour objectif de fournir des lignes directrices pour le traitement de données expérimentales, en se basant sur un TP de spectrophotométrie. Il s'accompagne d'un script Python accessible en ligne.

I. Données expérimentales

I.1. Rappel du protocole

Cinq solutions de bleu de bromophénol (BBP) sont préparées à partir d'une solution mère de concentration C_0 ($C_0 = 1{,}8 \times 10^{-3}\,\mathrm{mol \cdot L^{-1}}$). Les concentrations des solutions filles sont notées C_i avec i le facteur de dilution par rapport à cette solution. Les solutions filles ont pour concentrations :

- $C_5 = \frac{C_0}{5}$ préparée par dilution de la solution de concentration C_0.
- $C_{125} = \frac{C_0}{125} = \frac{C_5}{25}$ préparée par dilution de la solution de concentration C_5.
- $C_{250} = \frac{C_{125}}{2}$ préparée par dilution de la solution de concentration C_{125}.
- $C_{500} = \frac{C_{250}}{2}$ préparée par dilution de la solution de concentration C_{250}.

- $C_{625} = \frac{C_{125}}{5}$ préparée par dilution de la solution de concentration C_{125}.

L'absorbance de chaque solution est mesurée à une longueur d'onde fixée (592 nm, déterminée au préalable à l'aide d'un spectre). Le blanc a été effectué au préalable avec de l'eau distillée.

I.2. Données et incertitudes

Le résultat des mesures est donc un tableau de la forme :

Concentration	C_5	C_{125}	C_{250}	C_{500}	C_{625}
Absorbance à 592 nm	A_5	A_{125}	A_{250}	A_{500}	A_{625}

Le tracé du spectre pour la solution de concentration C_5 montre une limitation de l'appareil. La valeur A_5 indiquée, de l'ordre de 4,5, n'est pas significative, ce point doit être retiré. En effet une absorbance égale à 4 signifie que l'intensité lumineuse à travers la cuve est 10^4 fois plus faible que l'intensité lumineuse à travers la cuve du blanc. Le capteur qui mesure l'intensité lumineuse ne parvient à effectuer des mesures précises sur une gamme aussi large, donc finit par ne plus détecter le faisceau transmis quand l'absorbance est trop élevée.

Pour les 4 autres couples de valeur, il est essentiel d'évaluer les incertitudes associées, dans l'optique de pouvoir analyser ces valeurs.

L'incertitude sur l'absorbance peut être estimée d'après la notice de l'appareil. Il sera considéré ici que l'incertitude-type sur un mesurage de l'absorbance est :

- $u(A) = 0{,}013$ si $A \leq 0{,}5$,
- $u(A) = 0{,}015$ si $0{,}5 < A \leq 1{,}0$,
- $u(A) = 0{,}020$ si $1{,}0 < A$,

L'incertitude sur la concentration est liée au processus de fabrication par dilution. Par exemple, la solution de concentration C_{125} a été préparée par prélèvement de 2 mL de solution de concentration C_5, transfert dans une fiole jaugée de 50 mL et complément jusqu'au trait de jauge. Ces deux volumes sont des grandeurs expérimentales auxquelles sont associées des incertitudes. Un script Python permet d'estimer cette incertitude par la méthode de Monte-Carlo.

Application 1 Calculer les incertitudes pour chacune de vos valeurs expérimentales (concentration et absorbance), connaissant $C_0 = 1{,}8 \times 10^{-3}\,\text{mol} \cdot \text{L}^{-1}$ et supposant $u(C_0) = 0{,}01 \times C_0$.

I.3. Résultats

Pour préparer la solution de concentration C_5, le volume prélevé de solution mère est $V_m = 10{,}0\,\text{mL}$ et le volume réalisé de solution fille est $V_f = 50{,}0\,\text{mL}$. La relation entre concentrations est :

$$C_5 = C_0 \frac{V_m}{V_f}$$

En supposant $u(V_m) = 0{,}1\,\text{mL}$ et $u(V_f) = 0{,}1\,\text{mL}$ une simulation fournit $C_5 = 3{,}600 \times 10^{-4}\,\text{mol} \cdot \text{L}^{-1}$ et $u(C_5) = 2{,}9 \times 10^{-6}\,\text{mol} \cdot \text{L}^{-1}$.

Le processus est réitéré pour chaque solution, les résultats sont fournis dans le tableau suivant :

Facteur	125	250	500	625
C (mol·L^{-1})	$1{,}44 \times 10^{-5}$	$7{,}19 \times 10^{-6}$	$3{,}60 \times 10^{-6}$	$2{,}88 \times 10^{-6}$
$u(C)$ (mol·L^{-1})	$4{,}2 \times 10^{-7}$	$1{,}2 \times 10^{-7}$	$3{,}6 \times 10^{-8}$	$5{,}1 \times 10^{-8}$
A à 592 nm	1,06	0,51	0,26	0,20
$u(A)$	0,020	0,015	0,013	0,013

Application 2 Un majorant de l'incertitude sur la concentration de la solution fille peut être calculé par la relation :

$$u(C_f) = C_f \times \sqrt{\left(\frac{u(C_m)}{C_m}\right)^2 + \left(\frac{u(V_m)}{V_m}\right)^2 + \left(\frac{u(V_f)}{V_f}\right)^2}$$

Calculer ce majorant et le comparer à la valeur estimée à l'aide du script Python, pour chacune des solutions.

Les données expérimentales consolidées sont donc un ensemble de points $(C\,;A)$, avec pour chacun connaissance des incertitudes $(u(C)\,;u(A))$.

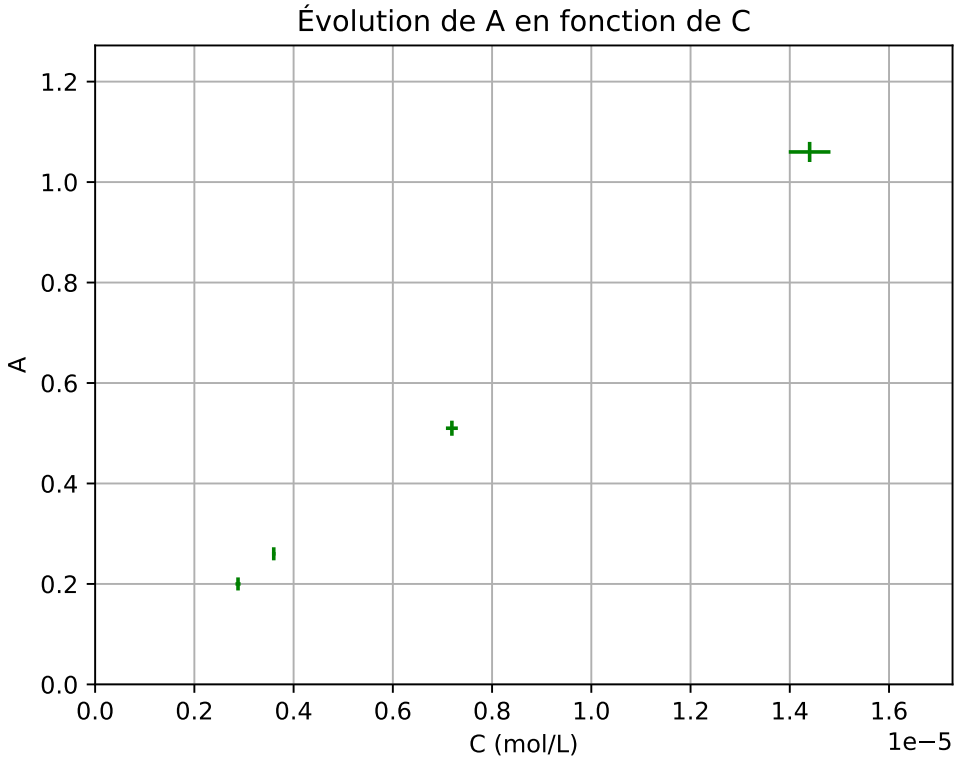
Les extraits de code Python suivants permettent la saisie des données expérimentales, puis le tracé de cet ensemble de points avec incertitudes (le chargement préliminaire des bibliothèques n'est pas indiqué).

Code Python

```
#Données expérimentales
C = np.array([2.88e-6, 3.60e-6, 7.19e-6, 1.44e-5])  # concentration C en mol/L
A = np.array([0.20, 0.26, 0.51, 1.06])       # absorbance A correspondante
u_C = np.array([5.1e-8, 3.6e-8, 1.2e-7, 4.2e-7])
# incertitude-type sur C calculée par simulation Monte-Carlo annexe
u_A = np.array([0.013, 0.013, 0.015, 0.020])
# incertitude-type sur A d'après donnée notice
```

Code Python

```
#Tracé de A en fonction de t
plt.figure(1)
#plt.plot(C, A,'go') # place les points
plt.errorbar(C, A, yerr=u_A, xerr=u_C,linestyle='None', fmt='g') # barres u(A) u(C)
plt.ylim(0, 1.2*np.max(A)) # optionnel : pour bien voir les limites
plt.xlim(0, 1.2*np.max(C)) # optionnel : pour bien voir les limites
plt.xlabel('C (mol/L)')
plt.ylabel('A')
plt.title('Évolution de A en fonction de C')
plt.grid()
plt.show()
```



II. Passage discret ⇒ continu

II.1. Principe de la régression linéaire

Les données expérimentales forment un ensemble fini de points. L'objectif de l'expérimentateur est de pouvoir décrire par une fonction cet ensemble de points, c'est-à-dire d'aboutir à une description continue (prévoir l'absorbance pour n'importe quelle valeur de concentration) à partir de données discrètes (les concentrations utilisées expérimentalement).

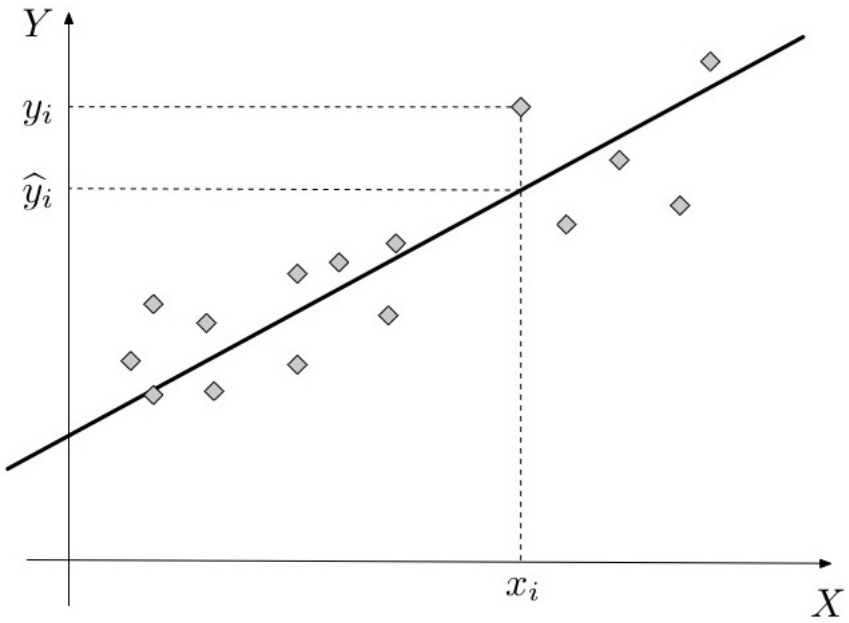
L'ordonnée et l'abscisse sont choisies de manière à ce que la fonction attendue soit de la forme $y = ax + b$. Si le nuage de points ne vérifie visiblement pas cette relation, l'ordonnée est à modifier.

> **Application 3** Supposons qu'une concentration diminue de manière exponentielle avec le temps : $C(t) = C_0 \times \exp(-kt)$. Le tracé de C en fonction de t n'est pas une droite. Proposer un tracé qui soit une droite.

La méthode de la régression linéaire vise à obtenir l'équation de la "meilleure droite" possible. Par "meilleure droite" il faut comprendre "droite qui minimise globalement les écarts entre valeur prédite par la droite et valeur expérimentale".

L'écart entre valeur expérimentale et valeur prédite par la droite, c'est-à-dire la distance y_i-$\hat{y}_i$ sur la figure suivante[1], est appelé résidu. La droite de régression linéaire est celle qui minimise la somme des carrés des résidus (le carré permet de ne pas compenser les résidus positifs par des résidus négatifs), c'est la méthode des moindres carrés ordinaires.

1. Issue de `https ://openclassrooms.com`



Attention !

Un programme de modélisation (calculatrice, tableur, Python, ...) retournera toujours une équation de droite de régression linéaire, même si les points ne forment visiblement pas une droite !

II.2. Simulation Monte-Carlo sur la régression linéaire

Avec un ensemble de points expérimentaux donnés sans incertitudes il est possible d'obtenir une droite de régression linéaire. Néanmoins ce traitement ne tient alors pas compte de la variabilité de la mesure, phénomène essentiel en sciences physiques.

Pour prendre en compte les incertitudes dans la régression linéaire, la stratégie Monte-Carlo consiste à tirer au sort un couple de valeur plausibles (ici C et A) pour chaque point expérimental, d'après la valeur mesurée et l'incertitude fournie. Une fois ce tirage au sort effectué pour l'ensemble des points expérimentaux, une régression linéaire est effectuée.

Ce processus est répété un grand nombre de fois. Chaque régression linéaire fournissant une valeur de pente et une valeur d'ordonnée à l'origine, il est possible de calculer une pente moyenne associée à un écart-type qui fournira l'incertitude-type, et de même pour l'ordonnée à l'origine. C'est l'objet du script suivant.

Code Python

```
# Simulation de régressions linéaires multiples par Monte—Carlo

import numpy.random as rd   # Générateur de nombre aléatoires

N = 10000                   # Nombre de simulations souhaitées
pente = []                  # Création d'une liste pour stocker les pentes
ordorig = []                # Création d'une liste pour stocker les ord à l'orig
nbpts = len(C)              # Nombre de points pour chacune des rég lin
```

```
9
10 for i in range(0,N):        # Boucle for pour les N tirages
11
12     C_sim = C + u_C * rd.normal(0, 1, nbpts) #Génération des valeurs aléatoires de C
13     A_sim = A + u_A * rd.normal(0, 1, nbpts) #Génération des valeurs aléatoires de A
14     reglin_sim = np.polyfit(C_sim,A_sim,1)   #Rég lin de A_sim en fonction de C_sim
15     pente.append(reglin_sim[0])          # Remplissage de la liste des pentes
16     ordorig.append(reglin_sim[1])        # Remplissage de la liste des ord à l'orig
17
18 # Le script en ligne trace ici des histogrammes
19
20 moy_pente = np.average(pente)           # Moyenne des valeurs de pente
21 u_pente = np.std(pente, ddof=1)         # Ecart-type de la série des pentes
22 moy_ordorig = np.average(ordorig)       # Moyenne des valeurs d'ordonnée à l'origine
23 u_ordorig = np.std(ordorig, ddof=1)     # Ecart-type de la série des ord à l'orig
24
25 print('pente = ',  moy_pente)
26 print('u(pente) = ', u_pente)
27 print('ordonnee origine = ',  moy_ordorig)
28 print('u(ordonnee origine) = ', u_ordorig)
```

II.3. Traitement alternatif pour $y = ax$

La stratégie décrite précédemment sera pertinente dès que le modèle recherché sera de la forme $y = ax + b$. Néanmoins ici le modèle attendu, d'après la loi de Beer-Lambert, est de la forme $y = ax$. Numpy, utilisé avec la commande np.polyfit, ne permet pas de forcer l'ordonnée à l'origine nulle. La valeur trouvée pour b étant *a priori* non nulle, il y a un problème de cohérence entre le modèle attendu et la régression linéaire effectuée.

Lorsque la relation attendue est de la forme $y = ax$, une autre stratégie est préconisée. Une fois le tracé y en fonction de x effectué pour détecter d'éventuels points aberrants, le rapport $\frac{y}{x}$ est calculé pour chaque point expérimental. La moyenne fournit la valeur de a, l'écart-type $s(a)$ permet de calculer la valeur de l'incertitude-type $u(a) = \frac{s(a)}{\sqrt{n}}$ avec n le nombre de points expérimentaux (le facteur $\sqrt{n}$ provient du fait qu'il s'agit de l'incertitude-type sur une valeur moyenne).

Ce traitement est mise en oeuvre à l'aide d'une syntaxe rapide (une boucle for est possible, sinon) dans le code Python proposé. Cette version ne tient pas compte des incertitudes sur les valeurs de C et A. Pour tirer au sort des valeurs de C et A tenant compte des incertitudes un script plus avancé est proposé en fin de notebook en ligne (mais l'estimation finale de l'incertitude ne repose que sur les 4 valeurs moyennes par simplicité de code).

Code Python

```
1 # Approche statistique directe
2 k=A/C
3 moy_k = np.average(k)          # Moyenne des valeurs
4 u_k = np.std(k, ddof=1)/np.sqrt(nbpts)       # Incertitude-type
```

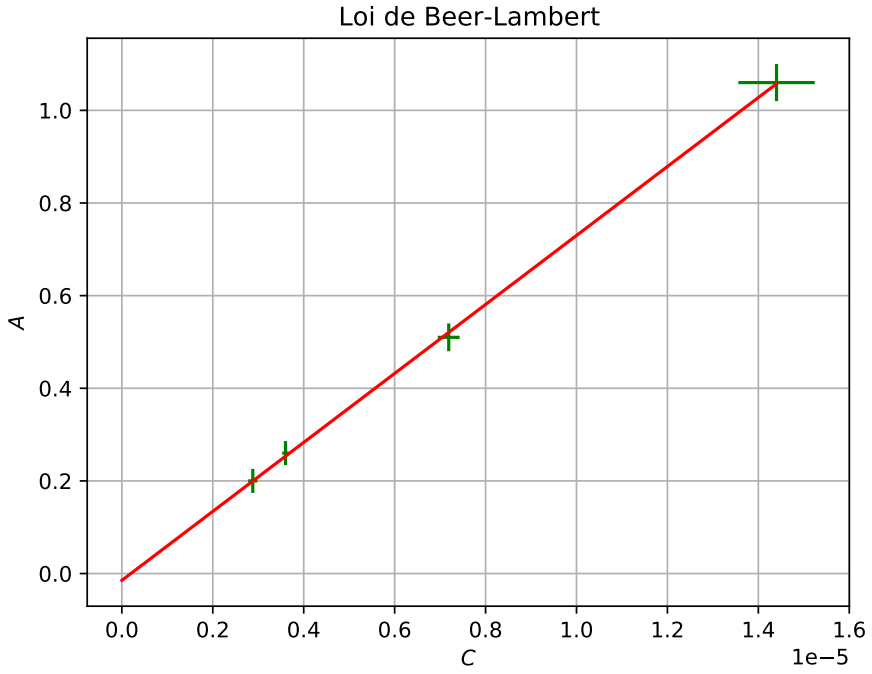
```
print('k = ',  moy_k)
print('u(k) = ', u_k)
```

III. Validation d'un modèle et estimation des paramètres

Lorsqu'un traitement par régression linéaire est pertinent (relation affine $y = ax + b$), deux outils complémentaires sont utilisés pour valider ou non l'expression de la fonction obtenue.

III.1. Tracé de la droite de régression linéaire

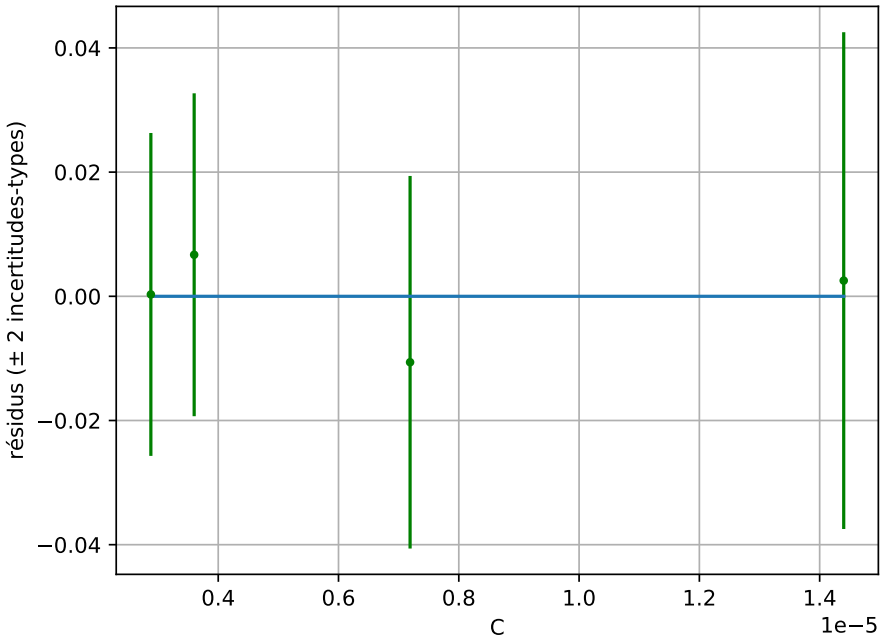
Le premier critère consiste à tracer la droite de régression linéaire, et l'ensemble des points expérimentaux avec leurs incertitudes. La droite de régression linéaire est compatible avec les points expérimentaux si elle passe à l'intérieur des barres d'incertitudes associés à chacun des points. Un code Python est accessible en ligne. Les barres d'incertitudes sont communément prises à deux incertitudes-types (c'est-à-dire $C \pm 2u(C)$ et $A \pm 2u(A)$).

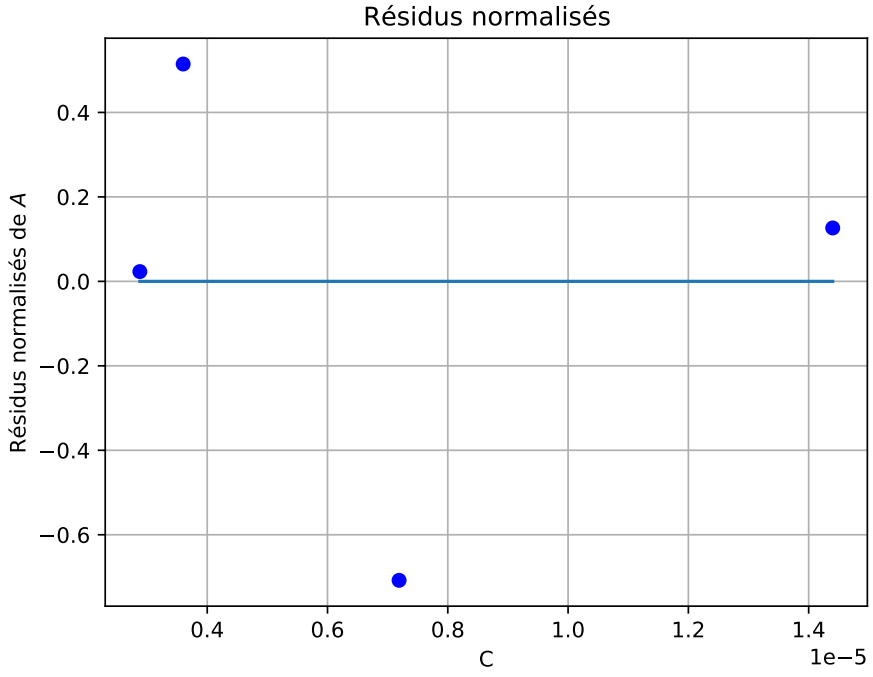


III.2. Tracé des résidus ou des résidus normalisés

Une deuxième méthode consiste à tracer le résidu (l'écart entre la valeur prédite par la droite de régression et la valeur expérimentale) pour chaque point expérimental, et à le comparer à l'incertitude, soit directement, soit en calculant le résidu normalisé (résidu divisé par l'incertitude-type).

Les résidus doivent être aléatoirement dispersés pour que le modèle soit validé. Dans le cas des résidus normalisés, le critère communément admis est que la valeur absolue doit être inférieure à 2 pour que le modèle soit validé en ce point. Un script Python est accessible en ligne.





Corrections

Application 2 Remarque : Il n'est pas exclu d'obtenir $u(C_f)_{\text{simulée}} \approx \frac{u(C_f)_{\text{calculée}}}{\sqrt{3}}$.

Application 3 La relation indiquée implique $\ln(C(t)) = \ln(C_0) - kt$, donc le tracé de $\ln(C(t))$ en fonction du temps t est une droite.