

TD n°2 Ajustement de courbes

Afin de faciliter le TD, on partira du notebook Ajustement déposé sur le github <https://github.com/MatManceau/Lumi2021-2022/TD2>. Les fichiers de données y sont également déposés.

1 Maximum de vraisemblance

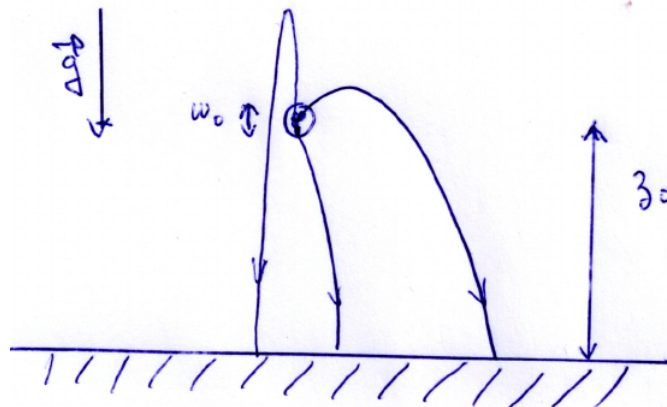
On considère un échantillon de points x_i issus d'une loi de probabilité \mathcal{P} . On rappelle que la vraisemblance (likelihood) se définit comme :

$$\mathcal{L}(\theta) = \prod_i \mathcal{P}(x_i|\theta)$$

Le maximum de vraisemblance est la valeur θ qui maximise \mathcal{L} . Il est souvent indispensable de faire les calculs avec la log de la vraisemblance (et donc de remplacer le produit par une somme).

2 Mesure de la gravité sur de l'antimatière

La matière attire la matière. A priori, c'est aussi le cas pour de l'antimatière. Par contre la question de savoir si de la matière attire ou pas l'antimatière et comment reste ouverte (et pourrait expliquer la dissymétrie matière/antimatière dans l'univers). Pour cela une expérience est en cours de réalisation au CERN. Il s'agit du projet GBAR qui consiste à laisser tomber un anti-atome d'hydrogène lent dans le champ de gravité de la Terre et à mesurer le temps de chute. Dans le meilleur des cas, on espère pouvoir réaliser 1000 fois l'expérience. Le but de ce problème est d'estimer l'incertitude sur la mesure de g à laquelle on s'attend.



A l'instant $t = 0$, la distribution en position des atomes est une gaussienne centrée autour de $z_0 = 30$ cm avec un écart type w_0 de l'ordre de $1 \mu\text{m}$ et une distribution en vitesse gaussienne d'écart type σ_v de l'ordre de 1 m/s centrée en 0. On prendra $g = 9.81 \text{ m.s}^{-2}$ orienté vers le bas. On négligera la dispersion initiale en

position. On peut montrer que le temps de chute d'un atome est donné par :
$$\frac{\sqrt{v_0^2 + 2gz_0} + v_0}{g}$$

1. Simuler un ensemble de $N = 1000$ temps de chute t_i et tracer l'histogramme (`plt.hist` en utilisant l'option `density=True`)

2. Premier estimateur : à partir d'un jeu de données t_i , on peut calculer la moyenne du temps de chute $\langle t_i \rangle$ et ainsi estimer simplement g par la formule :

$$\hat{g}(t_1, \dots, t_N) = \frac{2z_0}{\langle t_i \rangle^2}$$

Implémenter et caractériser cet estimateur (biais, incertitude, convergence). Pour ce, on simulera l'expérience M fois pour calculer l'espérance et l'écart type de l'estimateur.

3. Pour calculer la vraisemblance, il faut calculer $\mathcal{P}(t)$. C'est relativement simple car il existe une relation bijective entre t et la vitesse initiale v_0 . Dans ce cas $\mathcal{P}(t)dt = \mathcal{P}(v)dv$, or $v(t) = gt/2 - z_0/t$, donc :

$$\mathcal{P}(t) = \left(\frac{g}{2} + \frac{z_0}{t^2}\right)\mathcal{P}(v(t))$$

Dans cette formule, $\mathcal{P}(v)$ est donné par une Gaussienne. Comparer graphiquement cette formule à l'histogramme tracé précédemment (on pourra prendre N très grand).

4. Pour simplifier les calculs, on supposera que le seul paramètre inconnu de la loi est g . Pour un jeu de donnée de 1000 points, tracer le log de la vraisemblance en fonction de g .
5. Utiliser la fonction `fmin` du package `scipy.optimize` pour calculer la max de la vraisemblance à l'aide du min de $-\log(\mathcal{L})$.
6. Caractériser l'estimateur de max de la vraisemblance.

3 Ajustement d'une courbe

On considère une courbe $y = f(x, \mathbf{p})$ où les $\mathbf{p} = p_1, \dots, p_n$ sont des paramètres inconnus. Experimentalement, N mesures ont été effectuées pour des abscisses x_1, \dots, x_N . Les valeurs mesurées y_i sont entachées d'un bruit que l'on suppose gaussien de variance σ et non biaisé :

$$y_i = f(x_i, \mathbf{p}) + \epsilon_i$$

L'objectif est de trouver un estimateur des paramètres \mathbf{p} à partir des données (x_i, y_i) . Pour cela, on peut appliquer le principe du maximum de vraisemblance, et on obtient que

$$\hat{\mathbf{p}} = \operatorname{argmin}_{\mathbf{p}} \sum_{i=1}^N (y_i - f(x_i, \mathbf{p}))^2$$

Dans le cas général, les différents paramètres $\mathbf{p} = p_1, \dots, p_n$ sont corrélés. La matrice de corrélation est donnée par :

$$\operatorname{cov}(\mathbf{p}, \mathbf{p}) = \frac{\sigma^2}{2M}$$

avec la matrice M :

$$M_{k,l} = \sum_i \frac{\partial f}{\partial p_k} \Big|_{x_i} \frac{\partial f}{\partial p_l} \Big|_{x_i}$$

4 Fonction python `curve_fit`

On va utiliser la fonction `curve_fit` du package `scipy.optimize`. Cette fonction permet aussi d'obtenir l'incertitude des paramètres sous forme d'une matrice de corrélation. Cette fonction s'utilise de la façon suivante

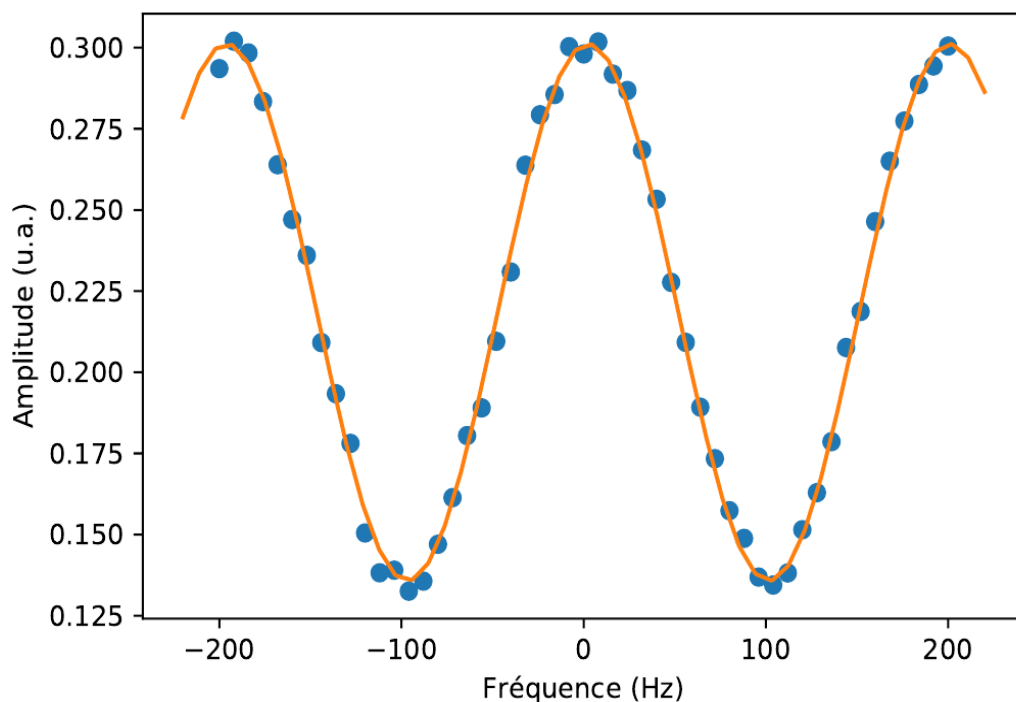
```
p_opt, cor_mat = curve_fit(fonction_de_fit, data_x, data_y, p_ini)
```

où

- `fonction_de_fit(x, p_1, p_2, ..., p_n)` est la fonction de fit. Les variables `p_1, ..., p_n` sont les paramètres de la fonction de fit.
- `data_x` et `data_y` sont les points de mesure.
- `p_ini` sont les paramètres initiaux (sous forme d'une liste/tuple/array).
- `p_opt` seront les paramètres optimaux
- `cor_mat` est la matrice de corrélation entre les paramètres.

5 Ajustement de frange d'interférence

On souhaite ajuster les franges d'un interféromètre atomique. Les données sont dans le fichier `fit_sinus.dat`. La première colonne du fichier (axe x) représente une fréquence en Hz. La seconde colonne représente la population mesurée pour une fréquence donnée. L'objectif est de trouver la position de la frange centrale.



On ajustera par une fonction cosinus avec une amplitude, un décalage vertical, une position centrale et une largeur ajustable (soit quatre paramètres en plus de la fréquence).

1. Écrivez la fonction de fit qui dépend des paramètres ci dessus. On l'appellera `frange(x, ...)`. Tracez la courbe pour x entre ± 220 Hz. On prendra un interfrange de 200 Hz.
2. Chargez et tracez les données. On représentera les données par des points (`plot(..., ..., 'o')`).
3. Calculez les paramètres optimaux. Quelle est la position de la frange centrale? Représentez les points et la courbe comme sur la figure ci-dessus.
4. Quelle est l'incertitude sur la position de la frange centrale?

6 Corrélation entre paramètres

Le fichier `population_france` contient les données de la population de la France entre 2001 et 2015 :

1. Tracez et ajustez les données par une droite $y = ax + b$.
2. Quel est l'incertitude sur b ? Quelle est la signification de b ? Qu'en pensez-vous?
3. Calculez la valeur et l'incertitude de votre fit en $x = 2020$.
4. Trouvez une fonction de fit plus pertinente pour ce problème.