TP2 autocollimation

September 21, 2024

1 Incertitude type composée pour la mesure d'une distance le long du banc d'optique.

Pour exploiter la méthode d'autocollimation, on doit estimer la distance séparant l'objet de la lentille, le long d'un banc d'optique gradué.

Si on note X_{objet} la position de l'objet et X_L la position de la lentille le long du banc d'optique, la distance focale f' de la lentille est évaluée avec la relation : $f' = X_L - X_{objet}$

Pour obtenir une évaluation de f', on doit donc déterminer la valeur mesurée f'_{mes} de f' ainsi que l'incertitude type $u(f')$ à partir des valeurs mesurées directement pour X_{objet} la position de l'objet obtenue avec l'incertitude type $u\left(X_{objet}\right)$ et X_L la position de la lentille connue avec l'incertitude type $u\left(X_L\right)$.

Dans cette situation où on évalue une grandeur à partir de la combinaison de plusieurs mesures, on dit qu'il faut réaliser la composition des incertitudes pour obtenir l'incertitude type sur la grandeur. Pour la formule de type somme-différence qui fait le lien entre f', X_{objet} et X_L, la formule théorique de composition des incertitudes est la suivante :

estimation de la différence :$f' = X_{objet} - X_L$ et estimation de l'incertitude : $u^2(f') = u^2\left(X_{objet}\right) + u^2\left(X_L\right)$

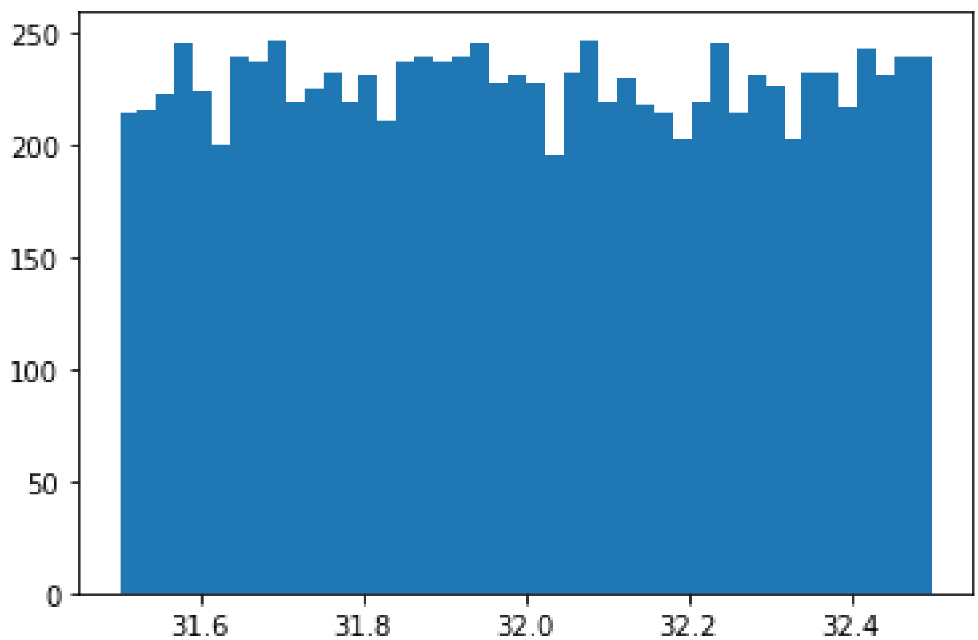
On se propose de la vérifier par simulation de Monte Carlo.

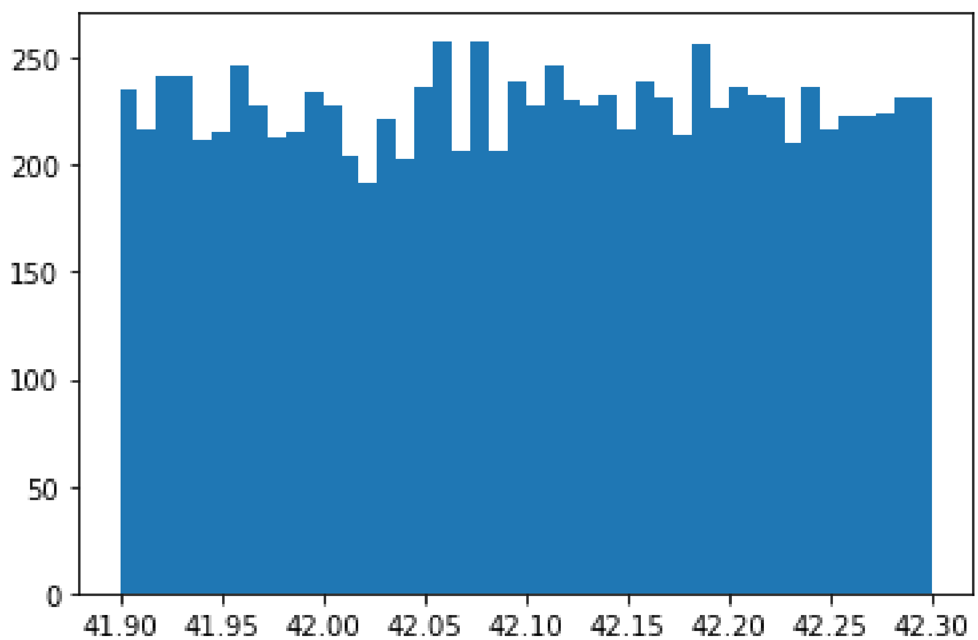
1.1 Evaluation de la distance focale de la lentille convergente par la méthode d'autocollimation

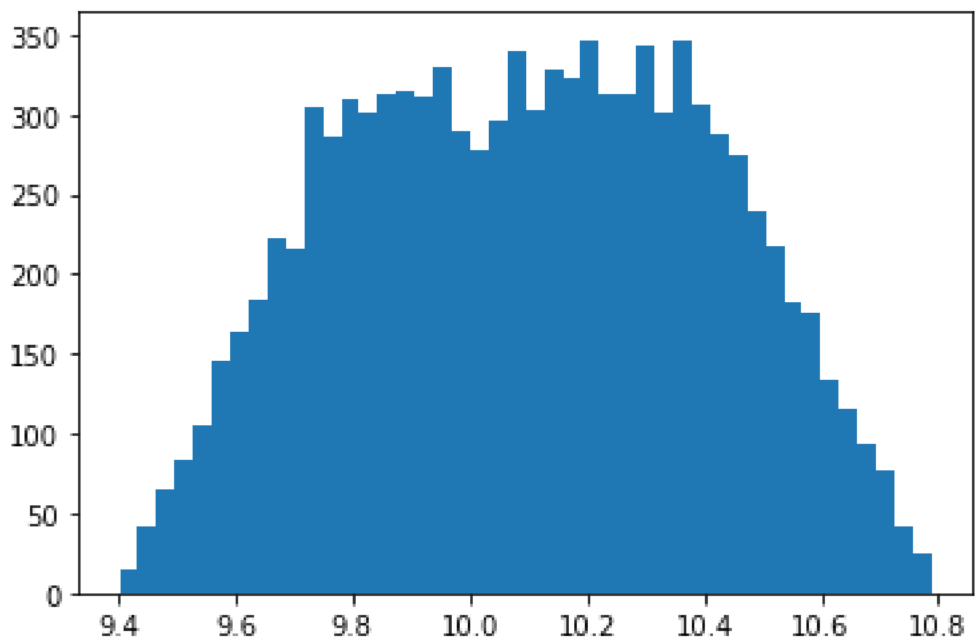
[1]:
```python
#on commence classiquement par importer la library numpy (sous l'alias np) pour␣
↪le calcul numérique
#son sous module numpy.random pour effectuer les tirages aléatoires selon des␣
↪lois bien controlées
#et la library matplotlib.pyplot (sous l'alias pl) pour la réalisation de␣
↪graphique.
import numpy as np
import numpy.random as rd
import matplotlib.pyplot as pl
```

[3]:
```python
#on créée une liste de 10000 tirages aléatoires uniformément répartis sur␣
↪l'intervalle étudié.
#ce qu'on appelle généralement simulation Monte-Carlo.
```

```
N=10000     # nombre de tirage aléatoire utilisé

#Liste des valeurs pour X(objet)
X_O=32.0    #position de l'objet le long du banc en cm
l_XO=0.5   #1/2largeur de l'intervalle de mesure de la position de l'objet en cm
X_O_MC=X_O+rd.uniform(-l_XO,l_XO,N) #tirage Monte Carlo sur la position de l'objet

#Liste des valeurs pour X(Lentille)
X_L=42.1    #position de la lentille de long du banc en cm
l_XL=0.2   #1/2 largeur de l'intervalle de mesure de la position de la lentille en cm
X_L_MC=X_L+rd.uniform(-l_XL,l_XL,N) #tirage Monte Carlo sur la position de la lentille

#Liste des valeurs obtenues par la simulation Monte Carlo pour f' en cm
f_MC=X_L_MC-X_O_MC

#on peut alors réaliser les histogrammes pour visualiser les distributions.
pl.figure(1)
pl.hist(X_O_MC,bins='rice')
pl.figure(2)
pl.hist(X_L_MC,bins='rice')
pl.figure(3)
pl.hist(f_MC,bins='rice')
#et en demander l'affichage.
pl.show()

#on évalue alors la distance focale pour cette expérience en cm
moyenne_f=np.average(f_MC)
print("la valeur moyenne sur les tirages réalisés est égale à ",moyenne_f," cm")
#on évalue alors l'inceritude type
u_f=np.std(f_MC,ddof=1)
#on affiche l'incertitude associée
print("l'incertitude associée à la mesure est égale à ",u_f,' cm')

#on compare alors l'incertitude à la valeur numérique théorique.
#Evaluation de type B de l'incertitude
u_XO=l_XO/np.sqrt(3)
u_XL=l_XL/np.sqrt(3)
u_f_theo=np.sqrt(u_XO**2+u_XL**2)
print("l'incertitude théorique sur la distance focale est ",u_f_theo,'cm')
```


250
200
150
100
50
0
31.6
31.8
32.0
32.2
32.4


250
200
150
100
50
0
41.90
41.95
42.00
42.05
42.10
42.15
42.20
42.25
42.30




```
la valeur moyenne sur les tirages réalisés est égale à  10.099849838097798  cm
l'incertitude associée à la mesure est égale à  0.31001327367181497  cm
l'incertitude théorique sur la distance focale est  0.3109126351029605 cm
```

Conclusion : On retiendra qu'on peut utiliser la formule de propagation des incertitude pour évaluer l'incertitude associée à une grandeur s'exprimant à l'aide d'une relation de type somme-différence en fonction des paramètres expérimentaux directement mesurés

Pour Y s'exprimant par la formule générale de type somme-différence $Y = a * X_1 + b * X_2$**, l'incertitude type composée est exprimée par la relation** $u^2(Y) = a^2 * u^2(X_1) + b^2 * u^2(X_2)$

1.2 Evaluation de la distance focale de la lentille divergente par la méthode d'autocollimation

Pour la lentille divergente, on met en place la méthode d'autocolimation pour le doublet ce qui donne.

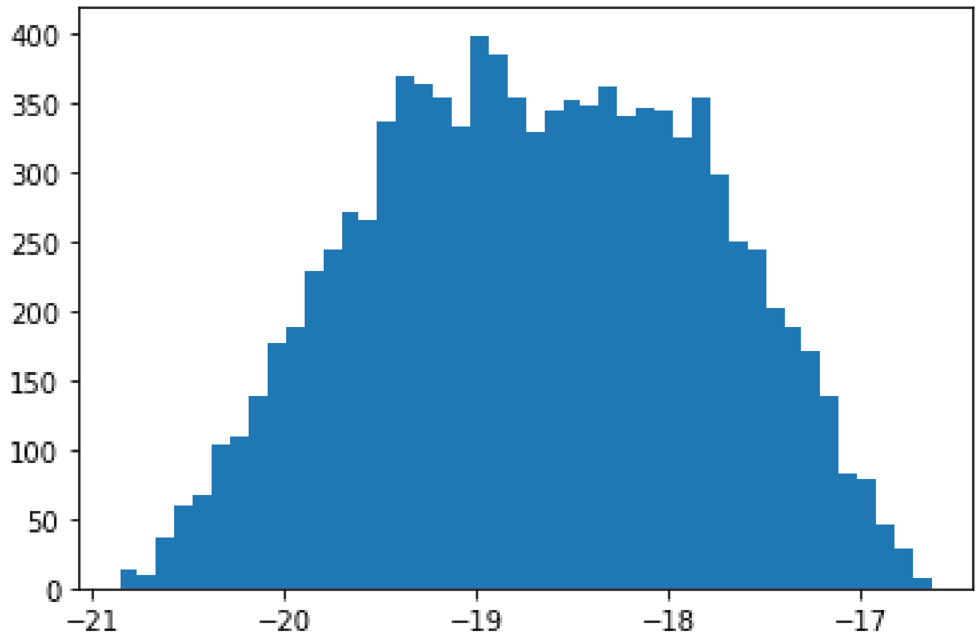
[4]:
```python
#pour l'objet
X_Od=32.0              #position de l'objet en cm
l_XOd=0.5              #demi largeur d'intervalle en cm
u_XOd=l_XOd/np.sqrt(3)#incertitude type en cm

#pour le doublet de lentille
X_Ld=54.0              #poisition du doublet en cm
l_Ld=0.2              #demi largeur d'intervalle en cm
u_Ld=l_Ld/np.sqrt(3) #incertitude type en cm
```

```
#pour la distance focale en cm
f_d=X_Ld-X_Od
u_f_d=np.sqrt(u_XOd**2+u_Ld**2)

print ("la distance focale obtenue pour le doublet est alors", f_d," cm")
print("l'incertitude associée est", u_f_d," cm")
```

```
la distance focale obtenue pour le doublet est alors 22.0  cm
l'incertitude associée est 0.3109126351029605  cm
```

On obtient une estimation de la distance focale de la lentille divergente par la relation $V_{doublet} = V_{conv} + V_{div}$ ce qui se traduit par $f'_{div} = \frac{f'_{doublet} * f'_{conv}}{f'_{conv} - f'_{doublet}}$ qui n'est pas une formule simple. On va devoir composer les incertitudes avec la méthode Monte Carlo.

[6]:
```
#on a déjà créer les listes de tirages aléatoires pour la lentille convergente.
#on crée maintenant les listes des positionspour le doublet
X_Ld_MC=X_Ld+rd.uniform(-l_Ld,l_Ld,N)

#Liste des valeurs pour la distance focale de la lentille divergente en cm
f_div_MC=(X_Ld_MC-X_O_MC)*(X_L_MC-X_O_MC)/((X_L_MC-X_O_MC)-(X_Ld_MC-X_O_MC))

pl.hist(f_div_MC,bins='rice')
pl.show()

#on évalue alors la distance focale pour cette expérience
moyenne_f_div=np.average(f_div_MC)
print("la valeur moyenne sur les tirages réalisés est égale à ",moyenne_f_div,"␣
 ↪cm")
#on évalue alors l'inceritude type
u_f_div=np.std(f_div_MC,ddof=1)
#on affiche l'incertitude associée
print("l'incertitude associée à la mesure est égale à ",u_f_div," cm")
```




```
la valeur moyenne sur les tirages réalisés est égale à  -18.684552353626625  cm
l'incertitude associée à la mesure est égale à  0.8736499737023273  cm
```

On pet observer qu'on obtient bien une distance focale négative ce qui est logique puisque la lentille est divergente. On peut noter aussi l'allure de la distribution des valeurs aléatoire qui n'est plus du tout uniforme mais se centre de plus en plus sur la valeur moyenne. On peut noter enfin que l'incertitude sur cette valeur est grande.

[ ]: