

TP Appareil photographique.

Première approche sur les incertitudes de mesure et sur le script python

I Première approche sur les incertitudes de mesure

Incertitude de type A : On obtient N valeurs x_k d'une grandeur physique X soit directement par mesure, soit par calcul à partir de mesures d'autres grandeurs.

On calcule la valeur moyenne : $\bar{x} = \langle x \rangle = \frac{1}{N} \sum_{k=1}^N x_k$ (fonction mean de numpy sous python)

On calcule l'écart type expérimental, ou incertitude-type sur une mesure par :

$\sigma_{n-1} = \sqrt{\frac{1}{N-1} \sum_{k=1}^N (x_k - \langle x \rangle)^2}$ noté s(x) sur les calculatrices (fonction std de numpy sous python)

Si on effectue une mesure x_{mes} , l'écart type sur cette mesure est $u(x_{mes}) = s(x)$

On peut comparer cette valeur à une valeur moyenne de référence en calculant l'écart normalisé (ou z-score) :

$$E_N = \frac{|x_{mes} - x_{ref}|}{u(x)}$$

Si ce nombre est assez faible (typiquement en dessous de 2), la mesure est dite conforme à la valeur de référence.

- Si on a effectué N fois la mesure dans les mêmes conditions, on gagne en précision :

L'écart type sur la valeur moyenne, ou incertitude type, est alors : $u(x) = \frac{\sigma_{n-1}}{\sqrt{N}}$

On compare la valeur moyenne à une valeur de référence en calculant l'écart normalisé

$$E_N = \frac{|\langle x \rangle - x_{ref}|}{u(x)}$$

II Première approche sur le script python :

Dans ce premier script on apprend :

- à importer des bibliothèques
- à remplir un tableau numpy
- à effectuer des opérations sur des tableaux numpy
- à calculer la valeur moyenne et l'écart type
- à afficher des résultats dans le shell
- à faire une régression linéaire avec la fonction np.polyfit
- à tracer des courbes avec matplotlib.pyplot

```
##Profondeur de champ
```

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

```
D = np.array([2.95,2.37,1.76,1.2,0.77]) # diamètre du diaphragme en cm  
x_l = 2.5 # position lentille cm  
x_A = np.array([39.5,37.2,32.2,25.3,21.9]) #position objet  
d_min = x_A - x_l  
n=len(D)  
# Etude statistique
```

```
y_exp = d_min / D  
y_ref = 10 # f'/epsilon
```

```
y_moy = np.mean(y_exp) # moyenne  
s=np.std(y_exp,ddof=1)#ecart-type  
u_y = np.std(y_exp,ddof=1)/np.sqrt(n) #incertitude-type
```

```
print('moyenne =',y_moy)  
print('incertitude-type=',u_y)  
z_score = abs((y_moy-y_ref)/u_y)  
print('z_score =',z_score)
```

```
# Regression linéaire  
a,b = np.polyfit(D,d_min,1)  
print("f'/epsilon =",a)  
plt.plot(D,a*D+b, label='modele')
```

```
plt.plot(D,d_min, 'o')  
plt.xlabel('D en cm')  
plt.ylabel('d_min en cm')  
plt.legend()  
plt.show()
```

```
# moyenne = 17.650705657426318  
# ecart-type expérimental = 4.86022212  
# incertitude-type= 2.1735574098000  
# z_score = 3.51990042817878  
# f'/epsilon = 8.48472014074412
```

