

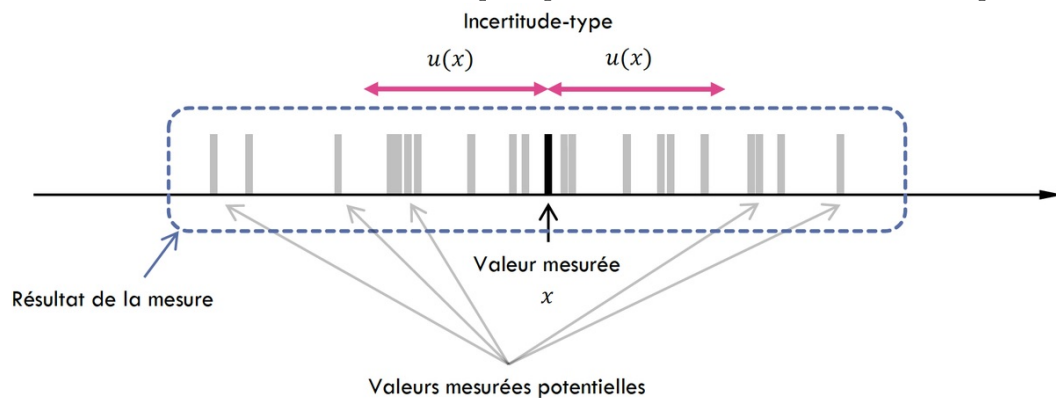
# intro des incertitudes

September 21, 2024

## 1 Incertitude type d'une mesure directe ; illustration sur les angles d'incidence et de réfraction sur une interface plane plexiglas vers air

On va introduire et illustrer la notion d'incertitude type. Pour cela on prend un cas de mesure d'angle effectué dans ce premier TP. On prend par exemple une mesure estimée comme centrée sur 40 degrés et de 1/2 largeur 1 degré.

L'incertitude type est la grandeur statistique qui va illustrer la dispersion des mesures obtenues dans l'expérience réalisée si on la reproduit à l'identique un grand nombre de fois. Elle s'identifie avec l'écart type standard de la distribution des mesures sur l'intervalle obtenu par la mesure. On utilisera alors systématiquement cette notion d'incertitude type pour caractériser toutes les mesures auquel permettent d'avoir accès les expériences



réalisées.

Comme il est impossible de réaliser concrètement un très grand nombre de réalisations expérimentales, On va les simuler par un programme informatique, et donc en python.

```
[8]: #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 contrôlées
      #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
```

On fait l'hypothèse que chaque réalisation expérimentale va donner une valeur d'angle située dans

l'intervalle estimé comme contenant les mesures et que la probabilité qu'un tirage se situe en un point particulier de l'intervalle est uniforme (hypothèse forte mais systématiquement effectuée).

La détermination de l'incertitude par la répétition de l'expérience un grand nombre de fois, et par l'exploitation des outils statistiques décrivant la distribution des résultats est appelée évaluation de l'incertitude de type A.

La simulation numérique du tirage aléatoire (dite simulation Monte Carlo), donne les résultats suivants.

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

angle=40      #valeur de l'angle
l_angle=1     #demi largeur de l'intervalle entourant la valeur mesurée d'angle
N=10000      # nombre de tirage aléatoire utilisé

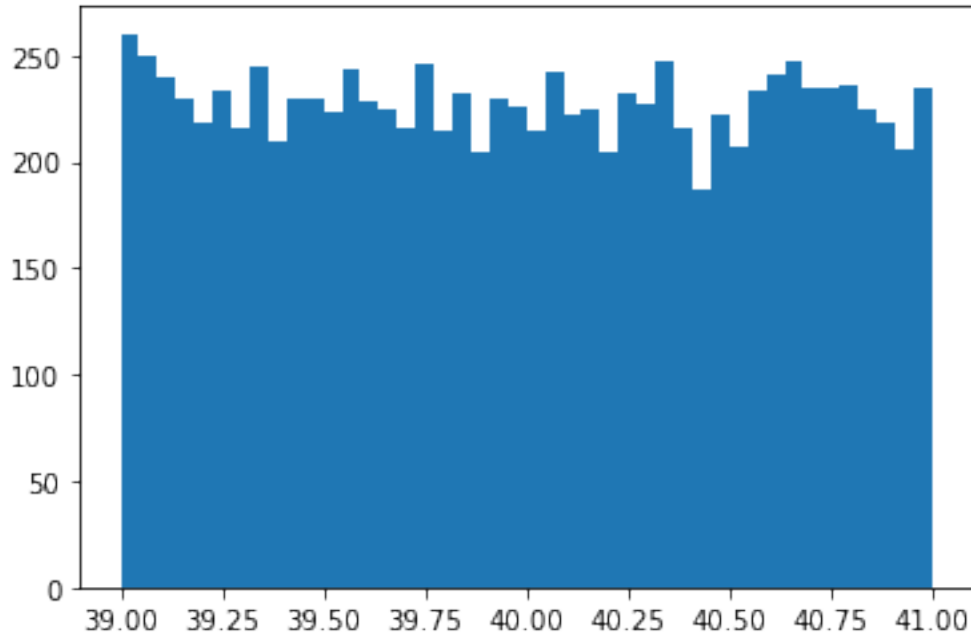
angle_MC=angle+rd.uniform(-l_angle,l_angle,N) # on crée une liste de 10000
      ↪ tirages aléatoires dans l'intervalle de mesure

#on peut alors réaliser un histogramme des tirages réalisés.
pl.hist(angle_MC,bins='rice')
#et en demander l'affichage.
pl.show()

#on calcule la valeur moyenne sur le tirage aléatoire réalisé
moyenne_angle=np.average(angle_MC)
#on affiche cette valeur moyenne
print("la valeur moyenne sur les tirages réalisés est égale à ",moyenne_angle)

#on calcule l'écart type sur le tirage aléatoire réalisé. Evaluation de type A
      ↪ de l'incertitude
u_angle=np.std(angle_MC,ddof=1)
#on affiche l'incertitude associée
print("l'incertitude associée à la mesure est égale à ",u_angle)

#on compare alors l'incertitude à la valeur numérique théorique pour une
      ↪ répartition uniforme.
#Evaluation de type B de l'incertitude
u_angle_unif=l_angle/np.sqrt(3)
print("l'incertitude théorique pour une distribution uniforme est égale à
      ↪ ",u_angle_unif)
```



la valeur moyenne sur les tirages réalisés est égale à 39.9928428669586  
 l'incertitude associée à la mesure est égale à 0.5817159377396836  
 l'incertitude théorique pour une distribution uniforme est égale à 0.5773502691896258

**Conclusion :** On retiendra que pour une mesure “a” effectuée à l’aide d’un système gradué, par exemple ici, le rapporteur pour lequel on peut estimer que la valeur mesurée est située sur un intervalle de demi largeur “l/2”, l’incertitude type “u(a)” sera exprimable directement par la relation théorique :

$$u(a) = \frac{\ell}{2\sqrt{3}}$$

On procède ainsi à une évaluation de type B de l’incertitude sur la mesure effectuée

On va maintenant s’intéresser à la loi de Snell-Descartes, qui affirme que l’angle d’incidence  $i_0$ , pour un faisceau arrivant sur l’interface depuis le plexiglas d’indice optique n, et l’angle de réfraction  $i_1$  dans l’air d’indice supposé unitaire sont reliés par :

$$\sin(i_1) = n \sin(i_0)$$

Pour obtenir une évaluation de n, on va exploiter à nouveau une simulation numérique de la réalisation d’un grand nombre de mesures avec la même configuration expérimentale.

```
[12]: #on crée une liste de 10000 tirages aléatoires uniformément répartis sur
      ↪ l'intervalle étudié
      # pour l'angle d'incidence i0.
      angle_inciden=30
      l_angle=1
      N=10000
```

```

angle_inciden_MC=angle_inciden+rd.uniform(-l_angle,l_angle,N)

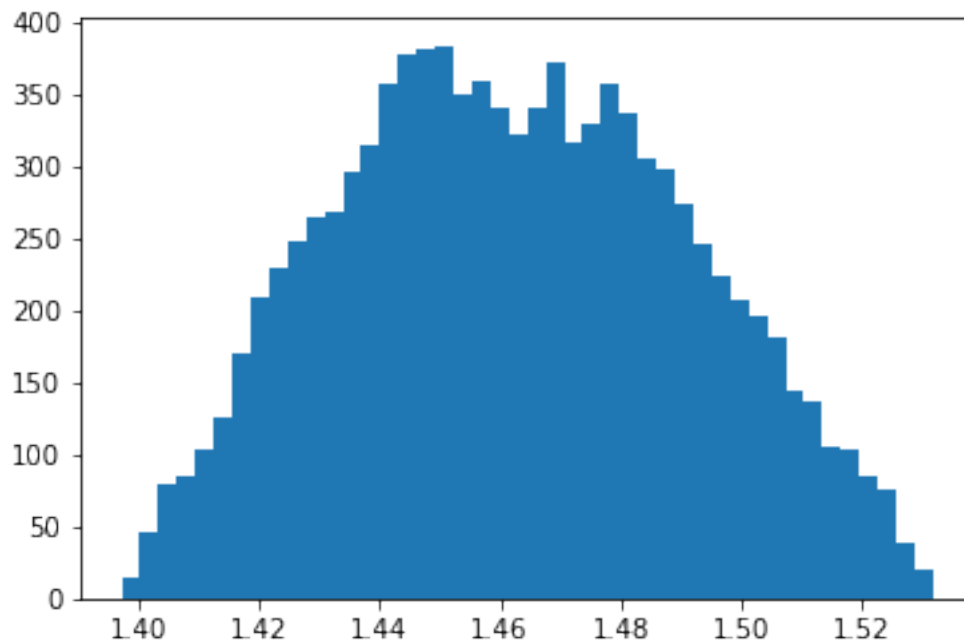
#on fait de même pour l'angle de réfraction i1.
angle_refrac=47
l_angle=1
N=10000
angle_refrac_MC=angle_refrac+rd.uniform(-l_angle,l_angle,N)

#on evalue alors la valeur de l'indice optique pour tous les tirages aléatoires
↳ effectués
# on notera qu'il faut convertir les angles en radian pour exploiter np.sin
n_MC=np.sin(np.pi/180*angle_refrac_MC)/np.sin(np.pi/180*angle_inciden_MC)

#On peut alors réaliser un histogramme des valeurs aléatoires calculées pour
↳ l'indice
pl.hist(n_MC,bins='rice')
#et en demander l'affichage.
pl.show()

#On évalue la moyenne, et l'incertitude type pour l'indice optique
moyenne_indice=np.average(n_MC)
u_indice=np.std(n_MC,ddof=1)
#on affiche cette valeur moyenne et cette incertitude
print("la valeur moyenne de l'indice est égale à ",moyenne_indice)
print("l'incertitude sur l'indice est égale à ",u_indice)

```



la valeur moyenne de l'indice est égale à 1.4624937463091567  
l'incertitude sur l'indice est égale à 0.0291150669642338

La mesure de l'indice optique obtenue avec la simulation de type Monte Carlo donne alors - n=1,51 avec une incertitude type u(n)=0,04 (lorsqu'on donne l'incertitude avec un seul chiffre) - n=1,506 avec une incertitude type u(n)=0,033 (lorsqu'on donne l'incertitude avec deux chiffres)

On évalue alors la compatibilité de ce résultat expérimental avec la valeur tabulée pour le plexiglas,  $n_{plexiglas}=1,50$ , en évaluant l'écart normalisé, encore appelé Z-score, entre la valeur expérimentale (n=1,5?? ; u(n)=0,0??) et la valeur théorique, en exploitant la définition :

$$Z_{score} = \left| \frac{n - n_{plexiglas}}{u(n)} \right|$$

```
[13]: n_plexiglas=1.5
Z=np.abs((moyenne_indice-n_plexiglas)/u_indice)
if Z<=2:
    print("le Z score est de",Z,", il est inférieur ou égale à 2")
    print("la mesure effectuée est cohérente avec la valeur théorique tabulée")
else :
    print("le Z score est de",Z,", il est supérieur à 2")
    print("la mesure effectuée n'est pas cohérente avec la valeur théorique,
    ↳tabulée")
```

le Z score est de 1.2882077082947312 , il est inférieur ou égale à 2  
la mesure effectuée est cohérente avec la valeur théorique tabulée

```
[ ]:
```