ANNEXE

Principe de la méthode de Monte-Carlo

Pour estimer la variabilité de la grandeur z, on va simuler un grand nombre N de calculs de la variable z résultant de $i = 1, \ldots, N$ jeux de variables $x_1, x_2, x_3, \ldots$.

L'incertitude-type sur la grandeur z sera obtenue en calculant l'écart-type sur les valeurs z_i pour l'ensemble des jeux $i = 1, \ldots, N$.

Hypothèses de la simulation de Monte-Carlo

Pour la grandeur x_1 :

- si l'incertitude-type de x_1 est connue, on réalisera un tirage des valeurs de x_1 selon une **loi normale** d'écart-type $u(x_1)$
- si seule la **plage de valeur acceptables** est connue pour la grandeur x_1 alors on réalisera un tirage selon une **loi uniforme** dans l'intervalle $[x_1 - \Delta_{x_1}, x_1 - \Delta_{x_1}]$

De même pour les grandeurs x_2, x_3, etc...

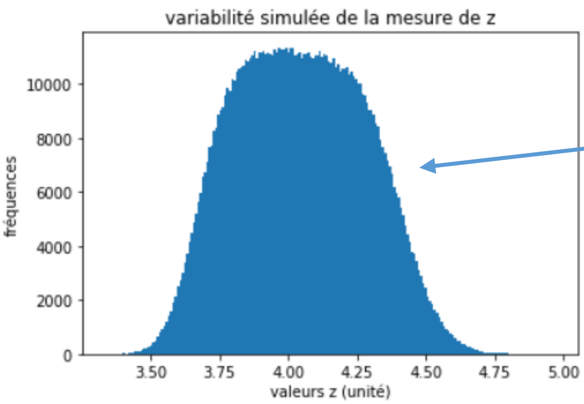
Exemple de Code Python avec tracé d'histogramme

Dans cet exemple, la grandeur z est obtenue par l'expression mathématique suivante :

$$z = f(x_1, x_2, x_3) = \frac{x_1^2}{\cos\left(\frac{x_2}{10}\right) + \sin\left(\frac{x_3}{10}\right)}$$

```python
# importation des modules
import numpy as np
import matplotlib.pyplot as plt
N = 10**6 # nb de tirages

# tirages de la variable x1
x1 , delta_x1 = 2.2, 0.1 # valeur mesurée, demi-largeur de l'intervalle de valeurs acceptables
x1i = np.random.uniform(x1 - delta_x1, x1 + delta_x1, N) # loi uniforme : a, b, N

# tirage de la variable x2
x2, u_x2 = 1.3, 0.42 # valeur mesurée, incertitude-type
x2i = np.random.normal(x2,u_x2,N) # loi normale : moyenne, écart-type, N

# tirage de la variable x3
x3, u_x3 = 2.1, 0.32 # valeur mesurée, incertitude-type
x3i = np.random.normal(x3,u_x3,N) # loi normale : moyenne, écart-type, N

# calculs des z pour chaque triplet (x1, x2, x3)
def f(x,y,z):
    return x**2/(np.cos(y/10)+np.sin(z/10)) # fonction quelconque
zi = f(x1i, x2i, x3i) # appel à la fonction f qui calcule les N valeurs zi

# résultats
moyZ = np.mean(zi) # moyenne
etypeZ = np.std(zi, ddof = 1) # estimateur non-biaisé de l'écart-type

# tracé des histogrammes
plt.hist(zi, bins = 'rice') # tracé d'histogramme
plt.xlabel('valeurs z (unité)')
plt.ylabel('fréquences')
plt.title('variabilité simulée de la mesure de z')
print("Z : moyenne z = ",moyZ," (unité) incertitude-type u(z) =  ", etypeZ, " (unité)")
```

```
Z : moyenne z =  4.042740627725263  (unité) incertitude-type u(z) =   0.2377378394186036  (unité)
```



REMARQUE :
Dans le cas général, l'histogramme des valeurs calculées n'est pas une courbe gaussienne.