

### Incertitude sur la teneur en cuivre d'un laiton

```
❶ u_A = 0.02*A ..... # incertitude-type sur l'absorbance
u_C = 0.01/np.sqrt(3) ..... # incertitude-type sur les concentrations en masse en mg/L

# Méthode Monte Carlo
N = 10000 ..... #Nombre de simulations souhaitées
t_sim = [] ..... #Création d'une liste vide qui stockera les N valeurs de teneur en cuivre dans le laiton

for i in range(N):
    A_sim=[] ..... #Création d'une boucle for permettant de générer les N expériences virtuelles
    C_sim=[] ..... #Création d'une liste vide qui stockera les valeurs simulées d'absorbance'
    for k in range(len(C)):
        A_sim_k = rd.normal(A[k],u_A) ..... #Création d'une liste vide qui stockera les valeurs simulées de concentrations en masse
        C_sim_k = rd.normal(C[k],u_C) ..... #Création d'une boucle pour simuler une série de couples (C,A) autour de chaque couple expérimental
        A_sim.append(A_sim_k) ..... #Génération d'une valeur de A aléatoire autour de la valeur expérimentale
        C_sim.append(C_sim_k) ..... #Génération d'une valeur de C aléatoire autour de la valeur expérimentale
        A_sim.append(A_sim_k) ..... #Stockage de la valeur aléatoire de A dans la liste
        C_sim.append(C_sim_k) ..... #Stockage de la valeur aléatoire de C dans la liste
    a_sim, b_sim = np.polyfit(C_sim,A_sim,1) ..... #Régression linéaire avec les valeurs de C et de A simulées
    Cm = (A0-b_sim)/a_sim ..... #concentration en masse en cuivre de 50
    t_sim_valeur=100*Cm*0.1/maiton ..... #teneur en cuivre dans le laiton simulée
    t_sim.append(t_sim_valeur) ..... #stockage de la valeur simulée de t dans la liste

#Affichage du résultat
u_t_sim = np.std(t_sim,ddof=1) ..... #incertitude-type = écart-type des N valeurs de g simulées
print("l'incertitude sur la teneur en cuivre du laiton est:",u_t_sim)

... l'incertitude sur la teneur en cuivre du laiton est: 0.5634123242408127
```