

TP 3 - RÉOLUTION NUMÉRIQUE D'ÉQUATIONS

**Exercice 1 - Résolution numérique d'équations**

1. Implémenter la méthode de recherche par dichotomie.
2. Implémenter la méthode de la sécante.
3. Implémenter la méthode de Newton.
4. Tester les méthodes pour trouver la (ou les) racine(s) de la fonction

$$g(x) = \sin(x) + x \cos(x)$$

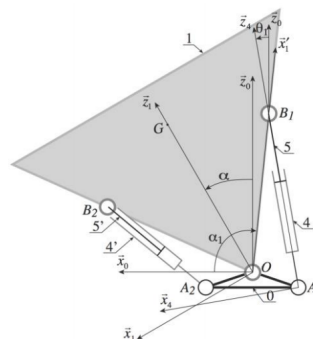
entre -1 et 2.5.

Vous pourrez commencer par tracer la fonction.

```
import numpy as np
import matplotlib.pyplot as plt
x = np.linspace(-1,2.5,100)
plt.plot(x,np.sin(x)+x*np.cos(x))
```

**Exercice 2 - Application : inclinaison de Clever**

On s'intéresse au mouvement de la cabine du véhicule à trois roues Clever qui s'incline à l'image d'une moto pour prendre un virage.



$$\begin{aligned} \alpha &= (\vec{z}_0, \vec{z}_1) \\ \alpha_1 &= (\vec{x}_0, \vec{x}_1) \\ \theta_1 &= (\vec{z}_0, \vec{z}_2) \\ \overline{OA_1} &= -a\vec{x}_0 - b\vec{z}_0 \\ \overline{A_1B_1} &= \lambda_1\vec{z}_1 \\ \overline{OB_1} &= L\vec{x}_1 \end{aligned}$$

Pour piloter le mécanisme, il est nécessaire de connaître la relation entre l'angle d'inclinaison  $\alpha$  de la cabine et les longueurs  $\lambda_1$  et  $\lambda_2$  des vérins. L'étude géométrique permet d'obtenir

$$\lambda_1(\alpha) = \sqrt{(L \cos(\alpha - 130^\circ) + a)^2 + (L \sin(\alpha - 130^\circ) - b)^2}$$

avec  $\alpha \in [-50, 50]$ ,  $a = 0.14$  m,  $b = 0.046$  m et  $L = 0.49$  m.

L'objectif est de déterminer l'angle  $\alpha$  pour une valeur d'élongation  $\lambda_1$  donnée.

1. Implémenter la fonction `lambda1(alpha, a=0.14, b=0.046, L=0.49)`.
2. Écrire une fonction `lambda1_p(alpha, a=0.14, b=0.046, L=0.49)` qui renvoie la dérivée de  $\lambda_1$  en  $\alpha$ .
3. Déterminer  $\alpha_0$  et  $\alpha_1$  tels que  $\lambda_1(\alpha_0) = 0.4$  m et  $\lambda_1(\alpha_1) = 0.55$  m. Vous pouvez utiliser une ou plusieurs des méthodes de recherche de racine implémentées à l'exercice 1.
4. Comparer les résultats à ceux qu'on obtient avec les fonctions `optimize.bisect` et `optimize.newton` du module `scipy`.
5. Écrire le code permettant de tracer la fonction  $\alpha(\lambda_1)$  pour  $\lambda_1 \in [0.4, 0.55]$ .
6. Il est possible de mesurer le temps mis par un programme (ou une fonction) en se servant de la bibliothèque `time` : pour cela on importe la bibliothèque à l'aide par exemple de `import time as tm` puis, on évalue la durée d'exécution d'une suite d'instructions à l'aide de la séquence suivante :

```
debut=tm.perf_counter()
...
fin=tm.perf_counter()
temps=fin-debut
```

Testez et comparez les différents temps d'exécution pour les méthodes programmées et celles de `scipy`. Il pourra être pertinent de répéter les différents tests plusieurs fois et de calculer une moyenne.