

**TD INFO05 – CORRECTION  
LES GRAPHIQUES  
LES BIBLIOTHÈQUES NUMPY ET MATPLOTLIB.PYPLOT****# condition\_gauss.py**

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

# initialisation
ANGLE = np.arange(-90, 90.1, 0.1)

# construction des tableaux contenant les angles en radian et les
sinus des angles
ANGLERAD = ANGLE/180*m.pi
SIN = np.sin(ANGLERAD)

# tracé du graphique
plt.figure()
plt.plot(ANGLE, ANGLERAD, label="i", color="red", linewidth=2, linestyle="--")
plt.plot(ANGLE, SIN, label="sin(i)", color="blue", linewidth=2)
plt.axis(xmin=-90, xmax=90, ymin=-2, ymax=2)
plt.xticks(np.arange(-90, 100, 10))
plt.yticks(np.arange(-2, 2.5, 0.5))
plt.title("Vérification de la condition de Gauss")
plt.legend(loc="lower right")
plt.xlabel("angle en °")
plt.axhline(color='black')
plt.axvline(color='black')
plt.grid(True)
plt.show()
```

**# courbe.py**

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

Thêta = np.linspace(-m.pi, m.pi, 1000)
X = (np.sin(Thêta))**3
Y = np.cos(Thêta) - (np.cos(Thêta))**4

plt.figure()
plt.plot(X, Y)
plt.axis("equal")
plt.show()
```

**# spectre\_lycopène.py**

```
import matplotlib.pyplot as plt

# ouverture du fichier
f = open("lycopène.txt")

# lecture du fichier
Lignes = f.readlines()

# fermeture du fichier
f.close()

# initialisation des listes
Lambda = []
Abs = []

# lecture automatique du fichier à partir de la 12e ligne
# Valeurs[0] ne contient rien
# on utilise la méthode replace pour remplacer les "," par des "."
for ligne in Lignes[11:]:
    # on remplace les , par des .
    ligne = ligne.replace(",", ".")
    # on coupe aux tabulations
    Valeurs = ligne.split("\t")
    # on transforme en flottant et on stocke
    Lambda.append(float(Valeurs[1]))
    # on transforme en flottant et on stocke
    Abs.append(float(Valeurs[2]))

# tracé du graphique
plt.figure()
plt.plot(Lambda, Abs)
plt.title("Spectre d'absorption du lycopène")
plt.xlabel("lambda (nm)")
plt.ylabel("absorbance")
plt.axis(xmin=Lambda[0], xmax=Lambda[-1])
plt.show()

# recherche des maxima

def maxima(Abs):
    Lmax = []
    for k in range(1, len(Lambda)-1):
        if Abs[k]>Abs[k-1] and Abs[k]>Abs[k+1]:
            Lmax.append(Lambda[k])
    return Lmax

pic = maxima(Abs)
print(pic)
```