

```

##### Programme Python pour calculer Q et Tpsi d'un régime pseudo-périodique acquis numériquement

#les données sont stockées dans le fichier rlc.csv : 1ere colonne = TEMPS ; 2eme colonne = TENSION

# 1ere étape : lecture du fichier rlc.csv et chargement des 2 colonnes dans deux listes

with open("rlc.csv","r") as f: #Creer un fichier texte avec les donnees

    po=[] #creation du point
    for line in f:
        po+= line.split(';') #une histoire de \t et \n

#f.close()

LX=[] #creation des abscisses
LY=[] #ordonnees
i=0
while i in range(len(po)-1):
    nx=float(po[i].replace(',','.')) #si les decimaux sont ecrits avec , au lieu de .
    LX+= [nx]
    ny=float(po[i+1].replace(',','.'))
    LY+= [ny]
    i+=2

#A ce stade, la liste LX contient les données temporelles et la liste LY contient les données de la tension

#####Fonction de lissage : car la courbe est bruité et cela va nuire à la recherche des maxima locaux...
def lissage(Lx,Ly,p):
    "Fonction qui débruite une courbe par une moyenne glissante
    sur 2P+1 points"
    Lxout=[]
    Lyout=[]
    for i in range(p,len(Lx)-p):
        Lxout.append(Lx[i])
    for i in range(p,len(Ly)-p):
        val=0
        for k in range(2*p):
            val+=Ly[i-p+k]
        Lyout.append(val/2/p)

    return Lxout,Lyout

xx,yy=lissage(Lt,Ly,7) ###On moyenne tous les 7 points par exemple et on stocke dans les listes xx et yy
plt.plot(Lt,Ly) ### On affiche la courbe pour voir si le lissage est suffisant
plt.plot(xx,yy)
plt.show()

#####TRAITEMENTS DES DONNEES#####
# Calcul du facteur de qualité : on recherche les maxima locaux, on calcule le décrement et on en déduit Q
# Calcul de la pseudo-période : écart temporel entre 2 maxima locaux

def calculeQTPSI(Lt,Lu):
    Lm=[]
    s=0
    m=0
    for i in range (len(Lu)-1): #calcul de la moyenne pour recentrer la courbe
        s=s+Lu[i]

```

```

m=s/len(Lu)
for i in range (len(Lu)-1): #recentrage de la courbe
    Lu[i]=Lu[i]-m

Ltt=[]
for i in range (len(Lu)-1): #recherche des maxima locaux et stockage de leurs index
    if Lu[i]>=Lu[i-1] and Lu[i]>=Lu[i+1]:
        Lm.append(Lu[i])
        Ltt.append(Lt[i]) # on stocke les instants des maxima locaux
d=0
for i in range (len(Lm)-2): #calcul du décrement par log(bosse k / bosse k+1)
    d=d+log(Lm[i]/Lm[i+1])
d=d/(len(Lm)-1) #on fait la moyenne
Q=sqrt(pi**2/d**2+1/4) #on en déduit le facteur de qualité

Lecart=[] #liste pour stocker les écarts temporels
for i in range(Ltt-1):
    Lecart.append(Lecart[i+1]-Lecart[i])
moy=0
for t in Lecart: #on fait la moyenne pour plus de précision
    moy+=t
Tpsi=moy/Len(Lecart)

return Q,Tpsi

print(calculerQTPSI(xx,yy))

```