

```
# -*- coding: utf-8 -*-
```

```
"""
```

```
Created on Sun Jul 20 20:55:56 2014
```

```
@author: Julien
```

```
"""
```

```
# comptage du nombre de ligne
```

```
fichier = open('tramway.txt','r') # Ouverture d'un fichier en lecture:
```

```
lecture = fichier.readlines()
```

```
nb_lignes = len(lecture)
```

```
fichier.close() # Fermeture du fichier
```

```
# extraction des données utiles
```

```
fichier = open('tramway.txt','r')
```

```
fichier.readline() # saut d'une ligne (non prise en compte des intitulés temps,...)
```

```
# initialisation des listes
```

```
temps=[]
```

```
acceleration=[]
```

```
accel_lat=[]
```

```
for i in range(nb_lignes-2):
```

```
    ligne=fichier.readline() # lecture d'une ligne
```

```
    ligne=ligne.rstrip("\n\r") # suppression retour chariot
```

```
    ligne=ligne.replace(",",".") # changement , en .
```

```
    ligne_data=ligne.split("\t") # découpage aux tabulations
```

```
# Création des listes
```

```
    temps.append(ligne_data[1])
```

```
    acceleration.append(ligne_data[5]) # extraction acceleration longitudinal (selon y)
```

```
    accel_lat.append(ligne_data[3]) # extraction acceleration latéral (selon x)
```

```
    i=i+1 # compteur boucle for
```

```
fichier.close() # Fermeture du fichier
```

```
# conversion en flottant et en unité SI
```

```
for k in range(len(temps)):
```

```
    temps[k]=float(temps[k])
```

```
    acceleration[k]=float(acceleration[k])*9.81 # m.s-2
```

```
    accel_lat[k]=float(accel_lat[k])*9.81
```

```
# détermination de la vitesse par intégration
```

```
# méthode des rectangles
```

```
pas_temps=temps[1]-temps[0] # calcul du pas temporel
```

```
vitesse_rec=[0] # vitesse initiale
```

```
for i in range (len(temps)-1):
```

```
    vitesse_rec=vitesse_rec+[vitesse_rec[i]+acceleration[i]*pas_temps*3.6] # 3.6 :
```

```
    conversion en km/h
```

```
# méthode des trapezes
```

```
pas_temps=temps[1]-temps[0] # calcul du pas temporel
```

```
vitesse_trap=[0] # vitesse initiale
```

```
for i in range (len(temps)-1):
```

```
    vitesse_trap=vitesse_trap+[vitesse_trap[i]+0.5*(acceleration[i+1]+acceleration[i])*pas_temps
```

```
    *3.6] # 3.6 : conversion en km/h
```

```
# détermination de la position par intégration
```

```
# méthode des rectangles
```

```
position=[0] # position initiale
```

```
for i in range (len(temps)-1):
```

```
    position=position+[position[i]+vitesse_rec[i]*pas_temps/3.6]
```

```
# tracé des courbes
```

```
def affiche(L):
```

```
    import matplotlib.pyplot as plt
```

```
    plt.plot(temps,L)
```

```

plt.xlabel("temps (en s)")
plt.show()
return('courbe affichée')

# Calcul du maximum
def maxi(L,i,j):
    """calcul la valeur maximum d'une liste L
    entre deux indices i et j"""
    if j==i:
        return L[i]
    else:
        k=(i+j)//2
        x=maxi(L,i,k)
        y=maxi(L,k+1,j)
        if x>y:
            return x
        else:
            return y

# Calcul de la moyenne et de l'écart-type
def moy_ecart(L,ti,tj):
    # à tester pour ti=8 et tj=12
    pas_temps=temps[1]-temps[0] # calcul du pas temporel
    # détermination des indices correspondant au temps choisi
    for p in range(len(temps)):
        if temps[p]<ti+pas_temps and temps[p]>ti-pas_temps:
            i=p
        if temps[p]<tj+pas_temps and temps[p]>tj-pas_temps:
            j=p

    som=0
    nb=0
    som_ecart=0
    # calcul de la moyenne
    for k in range(i,j-1):
        som=som+L[k]
        nb=nb+1
    moy=som/nb
    # calcul de l'écart-type
    for k in range(i,j-1):
        som_ecart=(moy-L[k])**2
    from math import sqrt
    ecart=sqrt(som_ecart/nb)
    return(moy,ecart)

# moyenne glissante
def filtre_mg(L,n):
    mg=[]
    for p in range(len(L)):
        mg_k=0
        for k in range(n):
            mg_k=mg_k+(L[p-k])/n
        mg=mg+[mg_k]
    return mg

# tracé de toutes les courbes
def affiche_tout():
    import matplotlib.pyplot as plt
    plt.subplot(311)
    plt.plot(temps,acceleration)
    plt.xlabel("temps (en s)")
    plt.subplot(312)
    plt.plot(temps,vitesse_trap)
    plt.xlabel("temps (en s)")
    plt.subplot(313)
    plt.plot(temps,position)
    plt.xlabel("temps (en s)")
    plt.show()
    return('courbes affichées')

```

