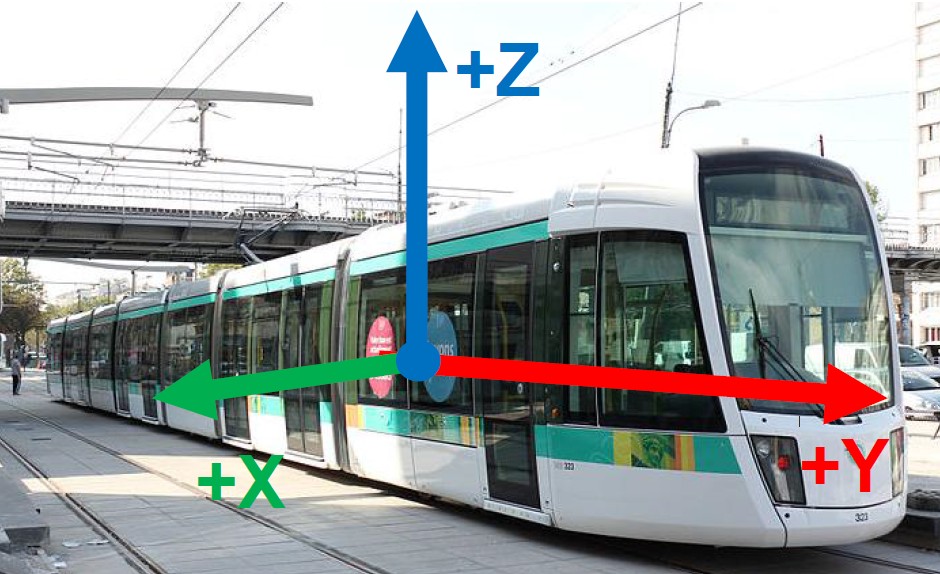
TD Info n°6

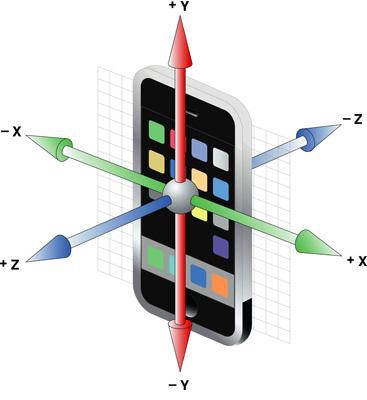
TRAITEMENT DES DONNÉES D’UN ACCÉLÉROMÈTRE



Déterminer le mouvement (trajectoire et déplacement) d’un tramway à partir de mesures réalisées par un accéléromètre.

Les mesures ont été réalisées à l’aide d’un téléphone portable. Le téléphone portable est tenu par un passager de la façon suivante :

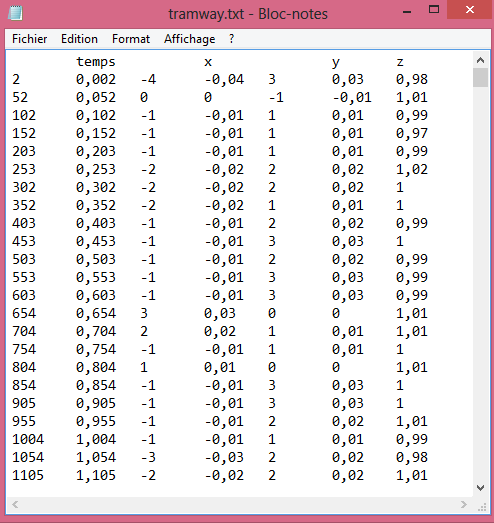
* le vecteur +Y orienté dans le sens du mouvement



* le vecteur +Z s’oppose à la pesanteur
* le vecteur +X est par conséquent orienté « latéral droit »

# Tracer des accélérations

Le capteur du téléphone portable fournit un fichier texte de points de mesure organisé de la manière suivante :



la première ligne indique le nom de mesure des colonnes

* la première colonne est le temps en millisecondes
* la deuxième colonne est le temps en secondes
* la troisième colonne est l’accélération latérale (selon x) en centièmes de g (accélération de pesanteur, on prendra *g* = *−*9.81*m · s−*2)
* la quatrième colonne est l’accélération latérale (selon x) en g
* la cinquième colonne est l’accélération longitudinale (selon y ) en centièmes de g
* la sixième colonne est l’accélération longitudinale (selon y ) en g
* la septième colonne est l’accélération selon z en g

L’objectif de cette partie est de tracer les accélérations longitudinale et latérale mesurées pour en déduire la trajectoire du tramway.

L’extraction des mesures dans des listes est réalisée de la manière suivante :

*# 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]) *# extraction temps*

acceleration.append(ligne\_data[5]) *# extraction acceleration longitudinal (selon y)*

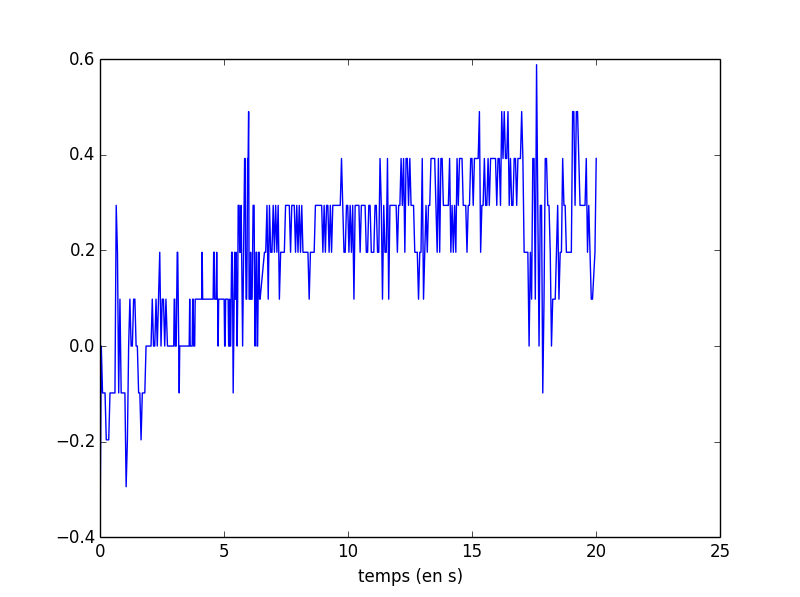
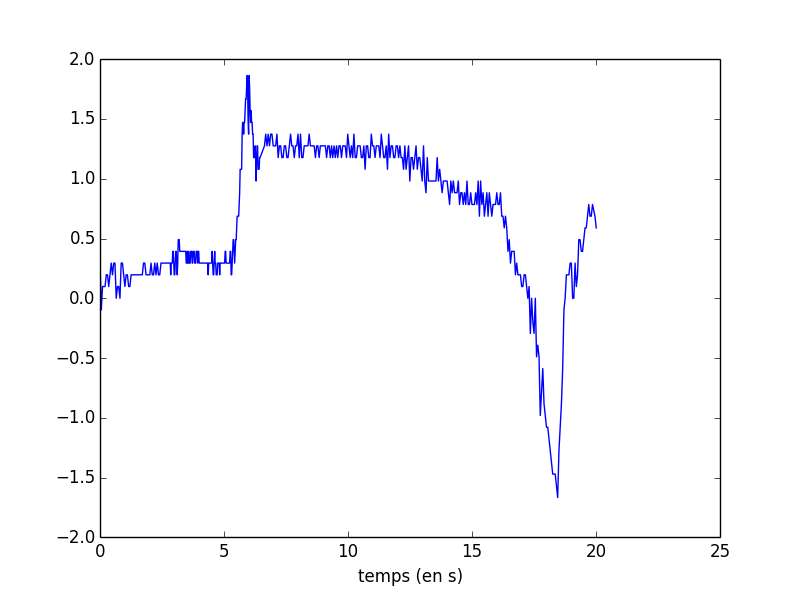
accel\_lat.append(ligne\_data[3]) # *extraction acceleration latéral (selon x)*

fichier.close() *# Fermeture du fichier*

**Question 1 -** Compléter le programme qui convertit les éléments des listes temps, acceleration et accel\_lat en flottant et en *m · s−*2.

**Question 2 -** Écrire, dans le cadre réponse suivant, une fonction affiche(L) qui prend en argument une liste L et qui permet d’afficher les éléments de cette listes en fonction de la liste temps (on supposera que ces deux listes ont le même nombre d’éléments).

Les tracés des accélérations longitudinale (selon x) et latérale (selon y) donnent :



accélération longitudinale en *m · s−*2 accélération latérale en *m · s−*2

**Question 3 -** En déduire quelle trajectoire simple a effectué le tramway

# Détermination de l'accélération maximale

L’objectif de cette partie est de déterminer l’accélération longitudinale maximale en réalisant deux fonctions récursives qui donnent l’élément maximal suivant les méthodes suivantes :

* la fonction maxi1 comparera son premier élément au maximum du reste de la liste.
* la fonction maxi2 comparera le maximum de la première "moitié" de la liste au maximum de la deuxième "moitié"

**Question 4 -** En respectant la méthode imposée dans les objectifs, compléter, dans le cadre réponse, la fonction **récursive** maxi1(L) prenant une liste L en entrée et retournant le plus grand des éléments de la liste. Il ne doit pas y avoir d’effet de bord.

Rappel : L[n :p] désigne la liste des éléments (L[n],L[n+1],...,L[p-1])

**Question 5 -** On note n la longueur de la liste L. Déterminer la relation de récurrence entre la complexité au rang n, noté C(n), et la complexité au rang n-1, C(n-1). En déduire l’ordre de la complexité de la fonction maxi1(L) (linéaire, exponantielle, logarithmique...).

* une comparaison ou une affectation comptera comme une opération
* un appel à la fonction return comptera comme une opération
* une affectation comptera comme une opération

Rappel : n//2 désigne le quotient de la division euclidienne de n par 2

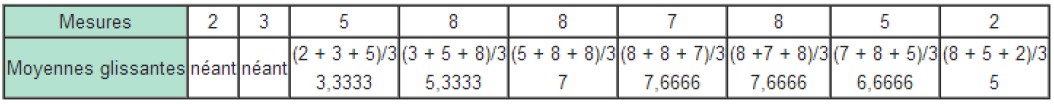
**Question 6 -** En respectant la méthode imposée dans les objectifs, compléter, dans le cadre réponse, la fonction **récursive** maxi2(L) prenant une liste L en entrée et retournant le plus grand des éléments de la liste. Il ne doit pas y avoir d’effet de bord.

**Question 7 -** On note n la longueur de la liste L, on supposera que *n* = 2*k* . Déterminer la relation de récurrence entre la complexité C(2*k* ) et C(2*k−*1). En déduire l’ordre de la complexité de la fonction maxi2(L) par rapport à n. Comparer à la complexité de maxi1(L).

# Filtrage des mesures

Le signal de l’accélération étant bruité, on souhaite le filtrer à l’aide d’une moyenne glissante. C’est une moyenne qui au lieu d’être calculée sur l’ensemble des n valeurs d’un échantillonnage, est calculée tour à tour sur chaque sous-ensemble de N valeurs consécutives (N ≤ n) ; le sous-ensemble utilisé pour calculer chaque moyenne "glisse" sur l’ensemble des données. On appelle N, l’ordre de la moyenne glissante.

Par exemple, le tableau suivant montre les moyennes mobiles simples sur 3 valeurs, pour une série de 9 mesures.



Une formule permettant de calculer une moyenne mobile est :

**Question 8 -** Compléter, dans la cadre réponse, la fonction filtre\_mg(L,N) qui prend en argument une liste L ainsi que l’ordre de la moyenne glissante N et qui retourne la liste L filtrée. **Il ne doit pas y avoir d’effet de bord**.

# Détermination de la distance parcourue

Rappel de la méthode des rectangles pour intégrer une fonction:

On approxime un calcul intégral par un somme discrète:



Où l'aire sous la courbe pour chaque intervalle est approximée par l'aire d'un rectangle de coté et de hauteur

**Question 9 -** Compléter, dans le cadre réponse, le programme qui permet de construire la liste vitesse à partir de la liste accélération en utilisant la méthode des rectangles. La liste vitesse, donnera la vitesse du tramway pour chaque pas de temps.

**Question 10 -** Compléter, dans le cadre réponse, le programme qui permet de construire la liste position à partir de la liste vitesse en utilisant la méthode des rectangles. La liste position, donnera la position du tramway pour chaque pas de temps.

Le tracé de la liste position en mètres donne :

