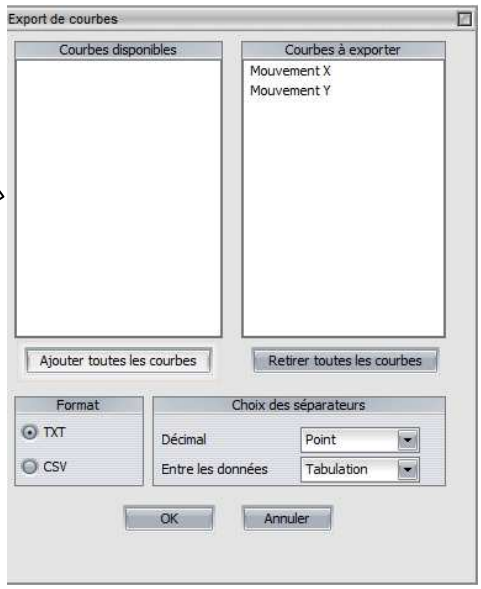
Tracé de vecteurs vitesse sur Python à partir d'un pointage vidéo

1. Acquisition et export du tableau de données

Avec Latis Pro :

Procéder au pointage de la séquence vidéo à étudier.
Dans ***Fichier***, ***Exportation***, ajouter les courbes Mouvement x et Mouvement Y.

Exporter sous format txt en choisissant comme séparateur :
- ***Point*** pour ***Décimal***
- ***Tabulation*** pour ***Entre les données***



Ligne à supprimer dans le programme pour conserver uniquement les valeurs

parabole_Latis Pro.txt - Bloc-notes

Fichier Edition Format Affichage ?

Temps	Mouvement X	Temps	Mouvement Y
0	-0.0130224034452742	0	-0.000705241778661271
0.030303	0.131034040408668	0.030303	0.168552785100039
0.060606	0.266616575800614	0.060606	0.320885009290869
0.090909	0.410673019654556	0.090909	0.456291430793829
0.121212	0.554729463508498	0.121212	0.566309148264984

Tracé de vecteurs vitesse sur Python à partir d'un pointage vidéo

2. Exécution du code Python avec récupération des données

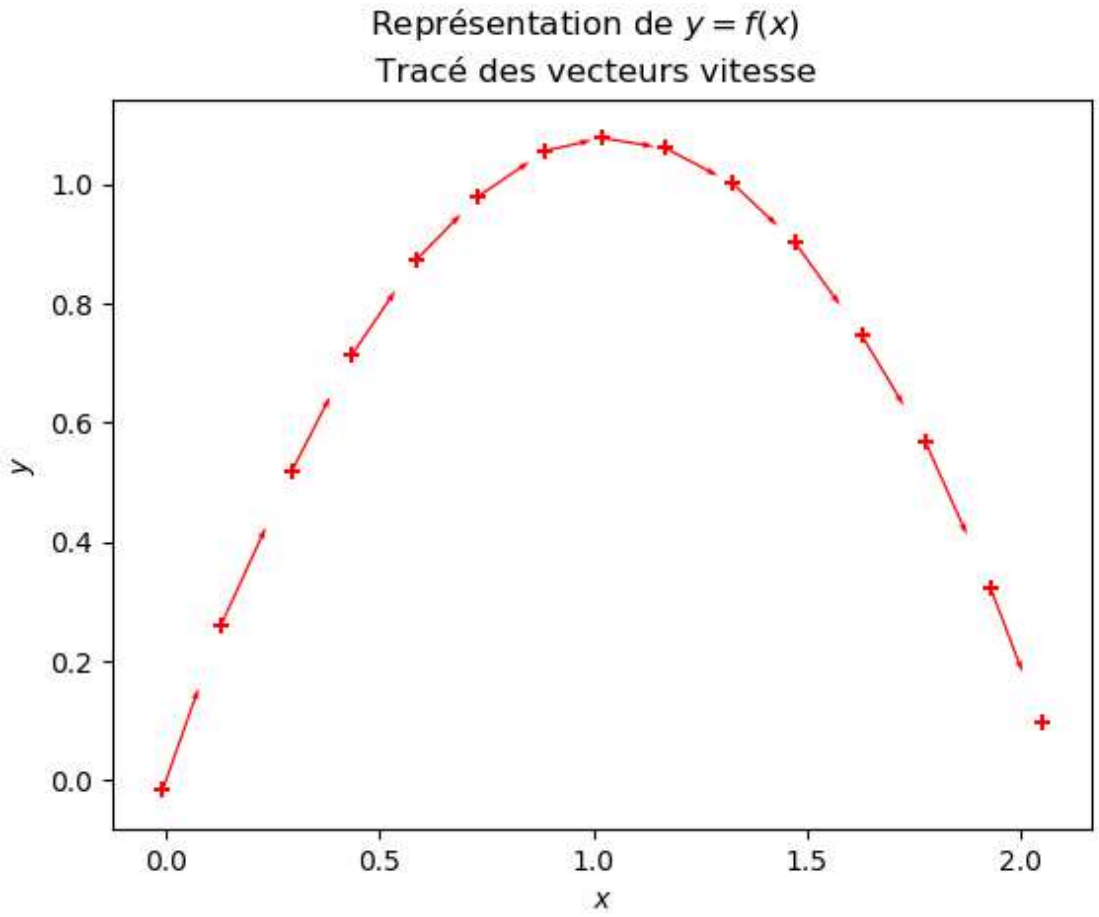
En utilisant un environnement Python adapté (**Pyzo** par exemple), ouvrir le fichier **Vecteurs vitesse .py**

ATTENTION : il faut que les fichiers .txt et .py soient dans le même dossier.

```
import matplotlib.pyplot as plt # pour les graphiques
import numpy as np # numpy pour l'importation des donnees en format txt

# importation des donnees txt obtenues apres pointage en supprimant la premiere ligne dans le fichier texte (obtenu apres le pointage du mouvement parabolique sur Latis Pro)
lines = open('parabole_Latis_Pro.txt').readlines() # ouverture du fichier texte
open('data.txt', 'w').writelines(lines[1:]) #création d'un nouveau fichier texte sans la première ligne
data = np.loadtxt('data.txt')# importation du nouveau fichier texte pour récupérer les valeurs det, x et y dans un tableau

t = data[:,0] # selection de la premiere colonne
x = data[:,1] # selection de la deuxieme colonne
y = data[:,3] # selection de la quatrieme colonne


# creation de listes vides pour les composantes des vitesses
vx = []
vy = []

# boucle pour calcul de vx et vy et construction de fleches a partir de la methode quiver de la fonction pyplot (legendes sans texte)
for i in range(0,len(t)-1) :
vx.append((x[i+1]-x[i])/(t[i+1]-t[i]))
vy.append((y[i+1]-y[i])/(t[i+1]-t[i]))

plt.quiver(x[i],y[i], vx[i], vy[i],color = 'r',width=0.005,scale=20,units='xy',angles='xy')
# méthode pour tracer des vecteurs

# plt.annotate('', xy = (x[i]+vx[i-1]/20, y[i]+vy[i-1]/20), xytext = (x[i], y[i]), arrowprops = {'color' : 'r','width': 1, 'headwidth': 3})
# autre méthode pour tracer des vecteurs : xytext donne les coordonnees du debut de la fleche, xy donne les coordonnnees de la pointe de la fleche

# plt.arrow(x[i],y[i], vx[i]/20, vy[i]/20,length_includes_head=True,color = 'r',width=0.005)
# 3ème méthode pour tracer des vecteurs

# afficher points avec croix rouges. Inserer texte (titre, nom des axes,…)
plt.figure(1)
plt.scatter(x, y, c = 'red', marker = '+')
plt.suptitle (" Représentation de $y=f(x)$ " )
plt.title (" Tracé des vecteurs vitesse " )
plt.xlabel("$x$")
plt.ylabel("$y$")
plt.show()
```

Tracé de vecteurs vitesse sur Python à partir d'un pointage vidéo

Exécuter le script pour obtenir la représentation graphique de y=f(x) et le tracé des vecteurs sur Pyzo (raccourci **ctrl+ E**):



Quelques liens utiles :

- Pour installer Pyzo et les modules nécessaires (matplotlib, numpy,..) : http://maths.spip.ac-rouen.fr/IMG/pdf/pyzo-miniconda-windows.pdf

- Pour comprendre le langage Python : https://www.chimsoft.com/python

- Créer graphique scientifique avec Python : http://apprendre-python.com/page-creer-graphiques-scientifiques-python-apprendre

- Pour manipuler un fichier txt avec numpy (sélection de colonnes) : https://www.science-emergence.com/Articles/S%C3%A9lectionner-certaines-colonnes-dun-fichier-de-donn%C3%A9es-avec-numpy-de-python/