

# Vecteur vitesse d'un point.

## Cadre de l'activité

### Capacité numérique

Capacité numérique : représenter des vecteurs vitesse d'un système modélisé par un point lors d'un mouvement à l'aide d'un langage de programmation.

## I. But de la séance

Nous avons vu que décrire un mouvement consister, une fois le système et le référentiel précisés, à caractériser la trajectoire et à calculer la 'norme' de la vitesse. Il existe un outil mathématique permettant d'exprimer en même temps ces deux caractéristiques : **le vecteur** .

Comment peut-on construire un vecteur vitesse grâce au langage de programmation Python.

Vous pouvez le télécharger [ici](http://www.lommele.com/activite_vitesse.zip) [http://www.lommele.com/activite\_vitesse.zip]

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
# Dlomlele - lyceemangin
from tkinter import *
#####
# Fonctions #
#####
```

```
#####
# PROGRAMME PRINCIPAL #
#####
x=200
y=150
```

```
racine=Tk()
racine.geometry("500x600")
racine.title("Positions")
fond=Canvas(racine, bg='white',width=500,height=500,bd=4)
fond.pack()
rond=fond.create_oval(x,y,x+10,y+10,fill='green')
racine.mainloop()
```

## II.Placer des points dans une fenêtre graphique

### II.I.Plaçons un premier point.

Pour placer un point dans une fenêtre graphique, un code en langage python serait par exemple, le programme ci-contre.

Ce code (vitesse1.py) place un point de rayon 5 pixels, de couleur verte, à une position définie par le couple de coordonnées (200,150)  
 ⇒ Comme dans le TP précédent, amusez-vous à modifier la position du point, la taille du point et la couleur de celui-ci.

Si le point se déplace, il peut être utile de stocker ses positions successives dans une liste, ce qui permet de tracer sa trajectoire. On utilise la syntaxe suivante : Points=[point1.point2, point3,...]. L'avantage des listes, c'est qu'on peut nommer chaque élément qui s'y trouve en utilisant son index (Attention en commençant par 0). Ainsi pour parler du point1 on peut écrire Points[0], pour parler du point 2, on peut écrire Points[1], etc...

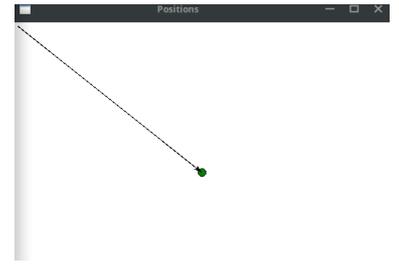
Mais comment décrire chaque point ?

Pour commencer, envisageons le cas d'un tableau ne contenant qu'un point, à la position x=250, y=200 à l'instant t=0. On résume ces données sous la forme [250,200,0]. On écrit alors **Points** = [[250,200,0]].

Il reste à modifier le programme pour qu'il 'comprenne' où sont stockées les valeurs de l'abscisse et de l'ordonnée :

Pour 'lire' la valeur de x du premier point (le seul pour l'instant), on écrit **Points[0][0]**, pour 'lire' la valeur de y du premier point, on écrit **Points[0][1]** et enfin, on écrit **Points[0][2]** pour lire l'instant t de ce point.

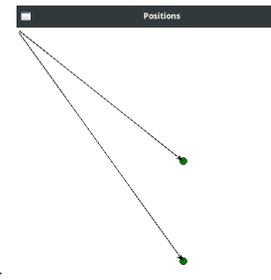
⇒ Ouvrez le programme vitesse2.py. Modifiez-le pour que le point apparaisse aux coordonnées x=20, y=300.



### II.II.Le vecteur position

Le vecteur position prend comme point de départ l'origine du repère et comme point d'arrivée le point que nous venons de placer. ( Attention, dans notre programme, l'origine du repère est en haut à gauche)

⇒ Le programme vitesse3.py ne dessine pas correctement le vecteur position. Modifiez le programme pour avoir le résultat voulu à savoir : [vous pourrez consulter la correction vitesse3\_corr.py après avoir essayé]



### II.III.Plaçons deux points

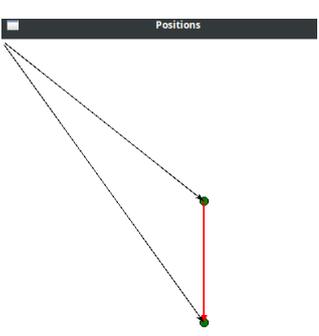
Rajoutons un second point dans la liste. On a maintenant **Points** = [[250,200,0],[250,350,0.2]]

⇒Quelle est la position de ce second point ?

⇒Quel est l'intervalle de temps nécessaire au passage de la position du point 1 à celle du point 2 ?

⇒Complétez le programme vitesse4.py pour faire apparaître les deux points, comme ci-contre.

[vous pourrez consulter la correction vitesse4\_corr.py après avoir essayé].



### II.IV.Vecteur déplacements

Le vecteur déplacement est celui qui représente le passage d'une position à une autre. En mathématiques, il correspond à la différence des deux vecteurs positions.

Pour le tracer dans notre programme il suffit de tracer le vecteur entre la position 1 et la position 2.

⇒Modifiez le programme vitesse5.py pour faire apparaître ce vecteur déplacement en rouge.

[vous pourrez consulter la correction vitesse5\_corr.py après avoir essayé].

### II.V.Plaçons un grand nombre de points

Vous avez constaté, dans les exemples précédents, qu'il suffisait de changer d'indice pour passer d'un point à un autre. Points[0] pour le premier point, Points[1] pour le second... et on généralise Points[2] pour le troisième... etc. Dans le langage de programmation Python, on peut utiliser une boucle pour changer cet indice sans réécrire une ligne par point.

Dans le programme vitesse6.py nous avons écrit la liste de position suivante :

```
Points=[[250, 0.0, 0], [250, 5.0, 1], [250, 20.0, 2], [250, 45.0, 3], [250, 80.0, 4], [250, 125.0, 5], [250, 180.0, 6], [250, 245.0, 7], [250, 320.0, 8], [250, 405.0, 9], [250, 500.0, 10]]
```

⇒ Quelle est la position du premier point ?

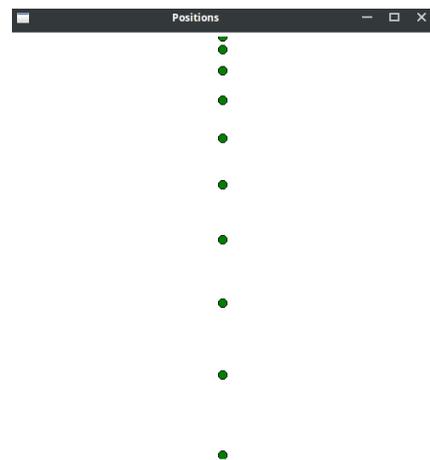
⇒ Quelle est la position du sixième point ?

⇒ Lancez le programme 6 pour observer la trajectoire du point vert.

```
File Edit Format Run Options Window Help
#!/usr/bin/python
# -*- coding: utf-8 -*-
# DLommele - lyceemangin
from tkinter import *
#####
# Fonctions #
#####

#####
# PROGRAMME PRINCIPAL #
#####
Points=[[250, 0.0, 0], [250, 5.0, 1], [250, 20.0, 2], [250, 45.0, 3], [250, 80.0

racine=Tk()
racine.geometry("500x600")
racine.title("Positions")
fond=Canvas(racine, bg='white',width=500,height=500,bd=4)
fond.pack()
for i in range(len(Points)):
    fond.create_oval(Points[i][0]-5,Points[i][1]-5,Points[i][0]+5,Points[i][1]+5
racine.mainloop()
```



### III.Tracé du vecteur vitesse

Le vecteur vitesse en un point M est colinéaire au vecteur déplacement d'un point M au point qui lui succède : il a même direction et même sens. Seule sa norme (sa longueur) change. La norme du vecteur vitesse au point M dépend de la vitesse instantanée du système entre deux points successives. Comment calculer cette vitesse ? Comment représenter le vecteur vitesse ?

Appuyons-nous sur un exemple pour comprendre et traçons le vecteur vitesse au point 7,  $\vec{V}_7$

#### III.I.Première étape : calculer la distance entre un point et son successeur

On rappelle que :

**Points**=[[250, 0.0, 0], [250, 5.0, 1], [250, 20.0, 2], [250, 45.0, 3], [250, 80.0, 4], [250, 125.0, 5], [250, 180.0, 6], [250, 245.0, 7], [250, 320.0, 8], [250, 405.0, 9], [250, 500.0, 10]]

et on fait l'hypothèse que les coordonnées sont données en mètres.

⇒ Quelle est la distance entre les points 7 et 8 ?  $M_7M_8 =$

Nous allons créer une fonction **Distance(i)** calculant la distance entre le  $i^{EME}$  point de la liste et le suivant.

⇒ Rappelez la formule donnant la distance entre deux points  $M_1(x_1,y_1)$  et  $M_2(x_2,y_2)$  vue en cours de mathématiques.

⇒ Ouvrez le programme vitesse7.py Comment la formule donnant la distance entre deux points est-elle écrite dans le programme ?

⇒ Exécutez le programme vitesse7.py. Dans la console, demandez la distance  $M_7M_8$  et vérifiez la valeur trouvée précédemment.

Exemple pour la distance  $M_4M_5$

```
>>> Distance(4)
La distance entre le point 4 et le suivant est : 45.0 m
```

#### III.II.Deuxième étape : calculer la vitesse entre un point et son successeur

Pour déterminer cette vitesse, il faut diviser la distance précédente par l'intervalle de temps nécessaire pour la parcourir.

⇒ Quel intervalle de temps s'est-il écoulé entre ces deux positions ?  $\Delta t =$

⇒ Calculez  $v_7 = M_7M_8 / \Delta t$

Nous allons créer une fonction **Vitesse(i)** calculant la vitesse instantanée du système au  $i^{EME}$  point.

⇒ Exécutez le programme vitesse8.py. Dans la console, demandez les valeurs des vitesses  $v_4, v_5, v_6, v_7$  et  $v_8$  selon l'exemple ci-dessous.

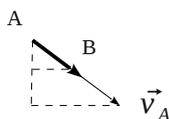
```
>>> Vitesse(4)
La distance entre le point 4 et le suivant est : 45.0 m
La vitesse instantanée au point 4 vaut 45.0 m/s
```

⇒ A la lecture de ces valeurs, comment pouvez-vous décrire le mouvement du système ?

#### III.III.Troisième étape: représenter le vecteur vitesse

Le **vecteur vitesse** au point A est colinéaire au **vecteur déplacement** entre les points A et B.

Pour tracer le vecteur nous allons donc utiliser ces deux points.



⇒ Sur le schéma ci-dessus, situez le **vecteur vitesse** et le **vecteur déplacement**

On appelle k le facteur de colinéarité entre les vecteurs déplacement et vitesse. (par exemple si  $k=2$ , le vecteur vitesse est 2 fois plus grand que le vecteur déplacement).

⇒ D'après le théorème de Thalès, quelle relation existe-t-il entre les coordonnées des points A ( $x_A, y_A$ ) et B ( $x_B, y_B$ ) et celles de l'extrémité du vecteur vitesse (X,Y) ?

⇒ Ouvrez le programme vitesse9.py. Quelles sont les deux lignes calculant ces coordonnées ?

⇒ Exécutez le programme 9 et observez le résultat.

⇒ Modifiez le programme 9 pour qu'il dessine le vecteur vitesse du sixième point.