

**DESCRIPTIF DE L'ACTIVITÉ**

<b>Objectifs</b>	<ul style="list-style-type: none"> <li>- Caractériser un mouvement rectiligne uniforme ou non uniforme</li> <li>- Exploiter une chronophotographie d'un système en mouvement et représenter des vecteurs vitesse</li> <li>- Décrire la variation du vecteur vitesse</li> <li>- Représenter les positions successives et les vecteurs vitesse d'un système modélisé par un point lors d'un mouvement à l'aide d'un langage de programmation</li> </ul>	
<b>Niveau concerné</b>	Seconde	
<b>Programme</b>	<b>Notion et contenus</b>	<b>Capacités exigibles Activités expérimentales support</b>
	Description du mouvement d'un système par celui d'un point. Position et trajectoire  Vecteur vitesse d'un point Mouvement rectiligne	Décrire le mouvement d'un système par celui d'un point  <b>Capacité numérique</b> : Représenter les positions successives d'un système modélisé par un point lors d'une évolution unidimensionnelle à l'aide d'un langage de programmation  Caractériser un mouvement rectiligne uniforme ou non uniforme  <i>Exploiter une chronophotographie d'un système en mouvement et représenter des vecteurs vitesse</i>  <i>Décrire la variation du vecteur vitesse</i>  <b>Capacité numérique</b> : Représenter des vecteurs vitesse d'un système modélisé par un point lors d'un mouvement à l'aide d'un langage de programmation  <b>Capacité mathématique</b> : Représenter des vecteurs
<b>Place de l'activité dans la progression et pré-requis</b>	Pré-requis Collège : Vitesse (direction, sens, valeur) Pré-requis seconde : Choisir un référentiel pour décrire le mouvement d'un système Mouvements rectilignes, uniformes, non-uniformes	
<b>Compétences évaluées</b>	Cette activité expérimentale permet d'évaluer les compétences <ul style="list-style-type: none"> <li>● <b>S'approprier (APP)</b> Comprendre les informations issues d'un langage de programmation</li> <li>● <b>Réaliser (REA)</b> Compléter un langage de programmation</li> <li>● <b>Valider (VAL)</b> Exploiter des résultats</li> <li>● <b>Attitude (ATT)</b> Être autonome, gérer son temps</li> </ul>	
<b>Éléments d'évaluation</b>	Evaluation formative	
<b>Mise en œuvre de l'activité</b>	Activité préparatoire : Faire le « 1 <sup>er</sup> défi : Etude d'une chronophotographie » à la maison 1 Séance d'1h30 de TP : Correction du 1 <sup>er</sup> défi et 2 <sup>ème</sup> défi à faire en classe Travail en groupe par 2	
<b>Outils numériques utilisés</b>	Logiciel EduPython Logiciel de lecture d'une vidéo (VLC Media Player...)	
<b>Remarques</b>	Lien vidéo : <a href="https://youtu.be/9OeVwaFBkF?t=1646">https://youtu.be/9OeVwaFBkF?t=1646</a> (ou <a href="https://dgxy.link/V4daD">dgxy.link/V4daD</a> )	
<b>Auteur(s)</b>	Solène Valla	

## L'ACTIVITÉ

### Atterrissage sur MARS

Nom, Prénom :	autoéval	eval
Compétences travaillées		
S'approprier : Comprendre les informations issues d'un langage de programmation		
Réaliser : Compléter un langage de programmation		
Valider : Exploiter des résultats		
Attitude : Être autonome, gérer son temps		

#### Objectifs :

- Caractériser un mouvement rectiligne uniforme ou non uniforme
- Exploiter une chronophotographie d'un système en mouvement et représenter des vecteurs vitesse
- Décrire la variation du vecteur vitesse
- Représenter les positions successives et les vecteurs vitesse d'un système modélisé par un point lors d'un mouvement à l'aide d'un langage de programmation



Photo : <https://cnes.fr/fr/mars-la-nouvelle-frontiere>

### CONTEXTE



On parle de plus en plus d'envoyer des Hommes sur Mars. La [Nasa](#) s'y prépare pour un premier voyage habité à l'horizon 2030. Mais dans le secteur privé, le patron de [SpaceX](#) Elon Musk se montre très ambitieux dans ce domaine et envisage l'arrivée des premiers humains sur Mars pour 2024. Ces pionniers poseraient les jalons d'une grande cité martienne qui pourrait accueillir jusqu'à un million de personnes avant la fin du XXI<sup>e</sup> siècle. Du moins, c'est ce qu'il souhaite.

Toutefois, un des freins au développement des voyages et technologies spatiales est le coût important de construction des fusées, qui sont à usage unique. SpaceX est une société qui conçoit, construit et commercialise des fusées en partie réutilisables dans le but de diminuer les coûts. Pour cela, cette société a mis au point les lanceurs Falcon 9, capables de redescendre et d'atterrir sur des plateformes prévues à cet effet. Pour réussir à développer cette technologie de nombreux essais ont été nécessaires.

<https://www.futura-sciences.com/sciences/questions-reponses/colonisation-mars-voyage-vers-mars-temps-faut-il-y-aller-7970/>

### QUESTION

- ➔ Pour ne pas compromettre sa réutilisation, nous cherchons à déterminer si le lanceur atterrit sans dommage sur la plateforme.

### DOCUMENTS

#### Document 1 : Atterrissage du premier étage du lanceur

La vidéo de l'atterrissage du premier étage du lanceur Falcon 9 (Mission NROL-108) est disponible sur le lien suivant : [dgxy.link/V4daD](https://dgxy.link/V4daD)



#### Caractéristiques techniques du premier étage du lanceur Falcon 9 Bloc 5

Longueur	Propulsion	Poussée maximale	Durée de combustion
41,5 m	9 Merlin 1D++	8 227 kN	182 s

On pourra considérer que la valeur absolue de la vitesse verticale du lanceur juste avant l'atterrissage ne doit pas excéder 21,6 km.h<sup>-1</sup>. Au-delà, le lanceur risquerait d'être endommagé, ce qui pourrait compromettre sa réutilisation ultérieure.

## Document 2 : Définition du vecteur vitesse

Pour préciser la direction, le sens et la valeur de la vitesse en un point  $M_i$  on utilise le **vecteur vitesse** noté  $\vec{v}_i$  ( $M_{i-1}$  et  $M_{i+1}$  sont les points avant et après le point auquel on s'intéresse)

Ses caractéristiques sont :

- **Point d'application (= origine)** : le point  $M_i$
- **Direction** : parallèle (ou colinéaire) au vecteur  $\overrightarrow{M_{i-1}M_{i+1}}$
- **Norme** : elle est donnée par  $\vec{v}_i = \frac{\overrightarrow{M_{i-1}M_{i+1}}}{t_{i+1} - t_{i-1}}$
- **Sens** : celui du mouvement

Le vecteur vitesse est tangent à la trajectoire. Dans un repère  $(O,x,y)$ , ses coordonnées sont :  $\vec{v}(v_x, v_y)$

## Document 3 : Méthode pour tracer un vecteur vitesse (ici cas du point 2).

Le vecteur vitesse  $\vec{v}_2$  d'un système au point  $M_2$  entre deux dates  $t_1$  et  $t_3$  a pour expression :  $\vec{v}_2 = \frac{\overrightarrow{M_1M_3}}{t_3 - t_1}$

**Méthode à suivre** pour tracer le vecteur vitesse au point  $M_2$  : (à faire à chaque fois...)

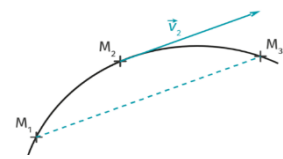
Mesurer la distance entre les points  $M_1$  et  $M_3$  sur la chronophotographie (=  $M_1M_3$ )

Déterminer la distance réelle entre  $M_1$  et  $M_3$  en m (utiliser l'échelle)

Calculer la durée séparant les instants  $t_3$  et  $t_1$  en s

Calculer la valeur de la vitesse au point  $M_2$  en  $m.s^{-1}$  grâce la formule

$$v_2 = \frac{M_1M_3}{t_3 - t_1}$$



Choisir une échelle de représentation (exemple : 1 cm sur le schéma correspond à 5  $m.s^{-1}$  en réalité)

Tracer le vecteur  $\vec{v}_2$ , à partir du point  $M_2$  et parallèle au segment  $M_1M_3$

## Document 4 : Pointage des différentes positions successives du lanceur

Point	$M_1$	$M_2$	$M_3$	$M_4$	$M_5$	$M_6$	$M_7$	$M_8$	$M_9$	$M_{10}$	$M_{11}$	$M_{12}$	$M_{13}$
Temps t (en s)	1689,12	1689,52	1689,92	1690,32	1690,72	1691,12	1691,52	1691,92	1692,32	1692,72	1693,12	1693,76	1695,2
Position x (en m)	0	0	0	0	0	0	0	0	0	0	0	0	0
Position y (en m)	136,965	125,478	113,548	102,945	92,7827	83,9463	75,5516	68,0407	61,4133	55,6696	50,3678	46,2638	41,383

## TRAVAIL A EFFECTUER

### 1<sup>er</sup> défi : Etude d'une chronophotographie

La chronophotographie jointe en annexe permet d'étudier l'atterrissage de la fusée Falcon 9 qui a eu lieu le 19 décembre 2020. Le système « lanceur » est modélisé par un point matériel représentant le sommet du 1<sup>er</sup> étage du lanceur.

🌀 Regarder la vidéo de l'atterrissage du premier lanceur du document 1.

- 1) Dans quel référentiel se place-t-on pour réaliser la chronophotographie de l'annexe ?
- 2) Caractériser la trajectoire du lanceur dans ce référentiel.
- 3) Calculer la valeur du vecteur vitesse (en  $m.s^{-1}$ ) en suivant la méthode (document 3) aux points  $M_3$  et  $M_8$  en détaillant votre raisonnement (attention à l'échelle).
- 4) Représenter ces vecteurs vitesse sur la chronophotographie (en annexe) en précisant l'échelle choisie.
- 5) Caractériser le mouvement du lanceur lors de son atterrissage. Justifier.



Certains logiciels (tracker, pyaviméca, regressi, etc...) permettent de pointer les différentes positions du lanceur (valeurs indiquées dans le document 4).

**Document 5 : Code Python traçant les positions du lanceur au cours de son atterrissage**

Le script se trouve en format texte dans le dossier Echange de la classe

```

1  import matplotlib.pyplot as plt
2  import numpy as np
3
4  ## Données expérimentales
5  t = [1689.12,...] # Liste des dates t du lanceur
6  x = [0,...] # Liste des positions en x du lanceur
7  y = [136.965,...] # A COMPLETER Liste des positions en y du lanceur
8
9  ## Affichage
10 plt.plot(x, y, "g+") # g:green, +: forme des points
11 plt.xlabel("Position x (m)")
12 plt.ylabel() # A COMPLETER
13 plt.title() # A COMPLETER
14
15 ## Calculs des vecteurs vitesse
16 N = len(x) # Nombre de points de mesures
17 vx = np.zeros(N) # Initialisation d'un tableau vide
18 vy = np.zeros(N)
19 for i in range(1, N-1):
20     vx[i]=(x[i+1]-x[i-1])/(t[i+1]-t[i-1])
21     vy[i]= # A COMPLETER
22
23 ## Affichage de la vitesse verticale au point i
24 for i in range(1, N-1):
25     plt.quiver(x[i], y[i], vx[i], vy[i]/5, scale=0.5, scale_units='xy', width=0.005, color='red')
26     print('Point', i+1, '-> vy=', round(vy[i],2), " m/s")
27
28 ## Affichage des axes et des courbes
29 plt.xlim([-0.5, 0.5]) # Limite du Repère orthonormé
30 plt.ylim([35, 140])
31 plt.show()
    
```

6) Faire correspondre à chaque commentaire son bloc d'instruction informatique (A, B, C, D, E ou F) issu du programme Python du document 5.

Commentaire	Bloc d'instruction informatique
Légènder les axes, donner un titre au graphique	
Importer les bibliothèques nécessaires	
Calculer les composantes de la vitesse	
Placer les points (+) en vert (green=g) dans le repère	
Attribuer à x et y les valeurs des coordonnées du lanceur	
Tracer le vecteur vitesse en un point de coordonnées (x[i], y[i]) en rouge	

**APPEL N°1** Appeler le professeur pour vérifier vos réponses ou en cas de difficulté

- 🖨 Dans le dossier Echange de la classe, ouvrir le fichier texte Script y=f(x) \_ Elève, le copier.
- 🖨 Ouvrir le logiciel EduPython dans Bureau/autres raccourcis/Physique-Chimie/Edupython
- 🖨 Copier le script  $y = f(x)$  \_ Elève et l'enregistrer sur le bureau en ajoutant votre nom à la place de Elève (exemple : Script  $y=f(x)$ \_Dupont)
- 🖨 Compléter les valeurs de la ligne 7 avec les valeurs expérimentales.
- 🖨 Compléter les lignes 12 et 13.

### APPEL N°2

 Appeler le professeur pour vérifier ou en cas de difficulté

- 🖨 Lancer le programme Python et visualiser la trajectoire du lanceur.

On souhaite tracer les vecteurs vitesses du lanceur. Les instructions figurant en dessous de ## Calculs des vecteurs vitesse, permettent de les tracer.

- 🖨 Supprimer les # figurant au début des lignes 16 à 31 pour qu'elles ne soient plus considérées comme du texte et deviennent actives.

7) Expliquer à quoi sert la ligne du programme :

$$vx[i]=(x[i+1]-x[i-1])/(t[i+1]-t[i-1])$$

8) A quoi correspond «  $t[i+1]-t[i-1]$  » dans cette formule ?

- 🖨 Par Analogie, compléter la ligne 21 :  $vy[i]=$ (

- 🖨 Fermer le graphe précédent et relancer le programme.

### APPEL N°3

 Appeler le professeur pour vérifier ou en cas de difficulté

- 9) Expliquer pourquoi les valeurs calculées de la coordonnée  $vy$  du vecteur vitesse sont de signe négatif.
- 10) De quelle couleur apparaissent les vecteurs vitesse ? Est-ce cohérent avec le programme ?
- 11) Décrire les variations du vecteur vitesse.
- 12) Est-ce cohérent avec votre réponse à la question 5) ?

### Conclusion :

- 13) Répondre à la problématique.

Questions 3) et 4)

Les positions du lanceur sont représentées toutes les 0,4 s



La double flèche verticale représente la longueur du 1<sup>er</sup> étage du lanceur Falcon 9 Bloc 5.

```
import matplotlib.pyplot as plt
import numpy as np

## Données expérimentales
t =
[1689.12,1689.52,1689.92,1690.32,1690.72,1691.12,1691.52,1691.92,1692.32,1692.72,1693.12,1693.76,1
695.2] # Liste des dates t du lanceur
x = [0,0,0,0,0,0,0,0,0,0,0,0,0] # Liste des positions en x du lanceur
y = [] # A COMPLETER Liste des positions en y du lanceur

## Affichage
plt.plot(x, y, "g+", markersize=4) # b:bleu, +:forme des points, --:relié par des pointillés
plt.xlabel("Position x (m)")
plt.ylabel() # A COMPLETER
plt.title() # A COMPLETER

## Calculs des vecteurs vitesse
#N = len(x) # Nombre de points de mesures
#vx = np.zeros(N)
#vy = np.zeros(N)
#for i in range(1, N-1):
# vx[i]=(x[i+1]-x[i-1])/(t[i+1]-t[i-1])
# vy[i]=() # A COMPLETER

## Affichage de la vitesse verticale au point i
#for i in range(1, N-1):
# plt.quiver(x[i], y[i], vx[i], vy[i]/5, scale=0.5, scale_units='xy', width=0.005, color='red')
# print('Point', i+1, '-> vy=', round(vy[i],2), " m/s")

## Affichage des axes et des courbes
plt.xlim([-0.5, 0.5]) # Limite du Repère orthonormé
plt.ylim([30, 140])
plt.show()
```

**1<sup>er</sup> défi : Etude d'une chronophotographie**

14) Dans quel référentiel se place-t-on pour réaliser la chronophotographie de l'annexe ?

→ On se place dans le référentiel terrestre.

15) Caractériser la trajectoire du lanceur dans ce référentiel.

→ La trajectoire est rectiligne dans ce référentiel

16) Calculer la valeur du vecteur vitesse en suivant la méthode (document 3) aux points M<sub>3</sub> et M<sub>8</sub> en détaillant votre raisonnement (attention à l'échelle).

→ Sur la chronophotographie, on mesure la distance M<sub>2</sub>M<sub>4</sub> = 1,6 cm

→ Sur le schéma, on mesure la longueur de la double flèche correspondant à la longueur du 1<sup>er</sup> étage du lanceur, soit 3,0 cm ↔ 41,5m

En tenant compte de l'échelle, M<sub>2</sub>M<sub>4</sub> = 1,6 cm ↔ ? = 1,6 x 41,5 / 3,0 = 22 m

→ On calcule la valeur de la vitesse au point M<sub>3</sub> en m.s<sup>-1</sup> grâce la formule

$$v_3 = \frac{M_2M_4}{t_4 - t_2} = \frac{22}{1690,32 - 1689,52} = 28 \text{ m.s}^{-1}$$

→ De même pour M<sub>8</sub>, sur la chronophotographie, on mesure la distance M<sub>7</sub>M<sub>9</sub> = 1,0 cm

→ En tenant compte de l'échelle, M<sub>7</sub>M<sub>9</sub> = 1,0 x 41,5 / 3,0 = 14 m

→ On calcule la valeur de la vitesse au point M<sub>8</sub> en m.s<sup>-1</sup> grâce la formule

$$v_8 = \frac{M_7M_9}{t_9 - t_7} = \frac{14}{1692,32 - 1691,52} = 18 \text{ m.s}^{-1}$$

17) Représenter ces vecteurs vitesse sur la chronophotographie (en annexe) en précisant l'échelle choisie.

Echelle des vitesses : 1 cm ↔ 10 m.s<sup>-1</sup>

18) Caractériser le mouvement du lanceur lors de son atterrissage. Justifier.

→ Le mouvement est **rectiligne ralenti** car la trajectoire est une droite et la vitesse diminue.

**2<sup>ème</sup> défi : Utilisation de Python pour représenter et modéliser****Document 5 : Code Python traçant les positions d'un ballon au cours d'un lancer**

Le script se trouve en format texte dans le dossier Echange de la classe

```

1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 ## Données expérimentales
5 t = [1689.12,...] # Liste des dates t du lanceur
6 x = [0,...] # Liste des positions en x du lanceur
7 y = [136.965,...] # A COMPLETER Liste des positions en y du lanceur
8
9 ## Affichage
10 plt.plot(x, y, "g+") # g:green, +: forme des points
11 plt.xlabel("Position x (m)")
12 plt.ylabel() # A COMPLETER
13 plt.title() # A COMPLETER
14
15 ## Calculs des vecteurs vitesse
16 N = len(x) # Nombre de points de mesures
17 vx = np.zeros(N) # Initialisation d'un tableau vide
18 vy = np.zeros(N)
19 for i in range(1, N-1):
20     vx[i]=(x[i+1]-x[i-1])/(t[i+1]-t[i-1])
21     vy[i]= # A COMPLETER

```

Diagramme de blocs :

- Bloc A : lignes 1 et 2 (importations)
- Bloc B : lignes 5, 6 et 7 (données expérimentales)
- Bloc C : ligne 10 (affichage)
- Bloc D : lignes 11, 12 et 13 (affichage)
- Bloc E : lignes 19, 20 et 21 (calculs des vecteurs vitesse)



```

22
23 ## Affichage de la vitesse verticale au point i
24 for i in range(1, N-1):
25     plt.quiver(x[i], y[i], vx[i], vy[i]/5, scale=0.5, scale_units='xy', width=0.005, color='red')
26     print('Point', i+1, '-> vy=', round(vy[i],2), " m/s")
27
28 ## Affichage des axes et des courbes
29 plt.xlim([-0.5, 0.5]) # Limite du Repère orthonormé
30 plt.ylim([35, 140])
31 plt.show()
32

```

Bloc F

19) Faire correspondre à chaque commentaire son bloc d’instruction informatique (A, B, C, D, E ou F) issu du programme Python du document 5.

Commentaire	Bloc d’instruction informatique
Légender les axes, donner un titre au graphique	Bloc D
Importer les bibliothèques nécessaires	Bloc A
Calculer les composantes de la vitesse	Bloc E
Placer les points (+) en vert (green=g) dans le repère	Bloc C
Attribuer à x et y les valeurs des coordonnées du lanceur	Bloc B
Tracer le vecteur vitesse en un point de coordonnées (x[i], y[i]) en rouge	Bloc F

- 📄 Dans le dossier Echange de la classe, ouvrir le fichier texte Script y=f(x) \_ Elève, le copier.
- 📄 Ouvrir le logiciel EduPython dans Bureau/autres raccourcis/Physique-Chimie/Edupython
- 📄 Copier le script y = f(x) \_ Elève et l’enregistrer sur le bureau en ajoutant votre nom à la place de Elève (exemple : Script y=f(x)\_Dupont)
- 📄 Compléter les valeurs de la ligne 7 avec les valeurs expérimentales.
- 📄 Compléter les lignes 12 et 13.

**👋 Appeler le professeur pour vérifier ou en cas de difficultés**

- 📄 Lancer le programme Python et visualiser la trajectoire du lanceur.

On souhaite tracer les vecteurs vitesses du lanceur. Les instructions figurant en dessous de ## Calculs des vecteurs vitesse, permettent de les tracer.

- 📄 Supprimer les # figurant au début des lignes 16 à 32 pour qu’elles ne soient plus considérées comme du texte et deviennent actives.

20) Expliquer à quoi sert la ligne du programme :

$$vx[i]=(x[i+1]-x[i-1])/(t[i+1]-t[i-1])$$

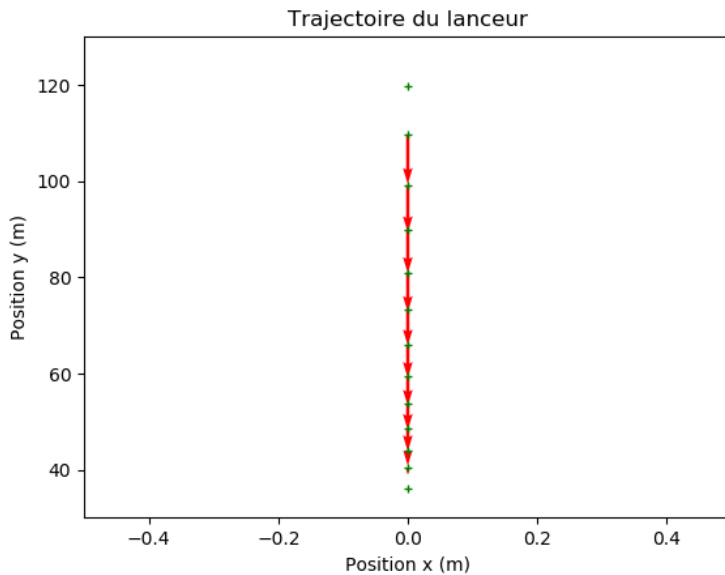
→ A calculer la norme de la composante horizontale de la vitesse pour chaque point i

21) A quoi correspond « t[i+1]-t[i-1] » dans cette formule ?

→ A la durée qui s’écoule entre le point « i-1 » et « i+1 »

📄 Par Analogie, compléter la ligne 21 :  $vy[i]=(y[i+1]-y[i-1])/(t[i+1]-t[i-1])$

- 📄 Fermer le graphe précédent et relancer le programme



Point 2	->	$v_y = -28.17$	m/s
Point 3	->	$v_y = -25.96$	m/s
Point 4	->	$v_y = -23.75$	m/s
Point 5	->	$v_y = -21.54$	m/s
Point 6	->	$v_y = -19.88$	m/s
Point 7	->	$v_y = -17.67$	m/s
Point 8	->	$v_y = -15.46$	m/s
Point 9	->	$v_y = -13.81$	m/s
Point 10	->	$v_y = -9.04$	m/s
Point 11	->	$v_y = -4.32$	m/s

- 22) Expliquer pourquoi les valeurs calculées de la coordonnée  $v_y$  du vecteur vitesse sont de signe négatif.  
 → Les valeurs de la coordonnée  $v_y$  du vecteur vitesse sont de signe négatif car l'axe (oy) est orienté vers le haut et le lanceur a un mouvement descendant.
- 23) De quelle couleur apparaissent les vecteurs vitesse ? Est-ce cohérent avec le programme ?  
 → En rouge, ce qui est cohérent avec le programme car ligne 25, il y a noté « color='red' »
- 24) Décrire les variations du vecteur vitesse.  
 → Le vecteur vitesse ne change pas de direction (verticale) mais sa norme diminue
- 25) Est-ce cohérent avec votre réponse à la question 5) ?  
 → C'est cohérent avec la question 5) car le mouvement est bien rectiligne (direction du vecteur vitesse ne change pas) et ralenti (norme du vecteur vitesse diminue)

Conclusion :

- 26) Répondre à la problématique.  
 La dernière valeur de la vitesse  $v_y$  au point  $M_{11}$  obtenue sur Python est  $4,32 \text{ m}\cdot\text{s}^{-1}$ , soit  $v_y(M_{11}) = 4,32 \times 3,6 = 15,6 \text{ km}\cdot\text{h}^{-1}$  donc cette valeur est inférieure à la vitesse maximale que doit avoir le lanceur avant l'atterrissage  $21,6 \text{ km}\cdot\text{h}^{-1}$ .

ANNEXE au TP

La double flèche verticale représente la longueur du 1<sup>er</sup> étage du lanceur Falcon 9 Bloc



Questions 3) et 4)