

Arriverez-vous à poser le rover Perseverance sur Mars ?

Le rover Perseverance s'est posé avec succès sur la planète Mars le 18 février 2021 au fond du cratère Jezero. Lors de la dernière étape de son atterrissage, le rover Perseverance était soutenu par 3 câbles à l'étage de descente nommé "Sky Crane" (grue du ciel). L'étage de descente contrôlait sa trajectoire grâce à des rétrofusées.

Un programme Python a été écrit pour simuler cet atterrissage. Il permet de piloter l'étage de descente en allumant ou en éteignant ses rétrofusées. Cependant, il manque quelques lignes au programme pour qu'il soit complet...

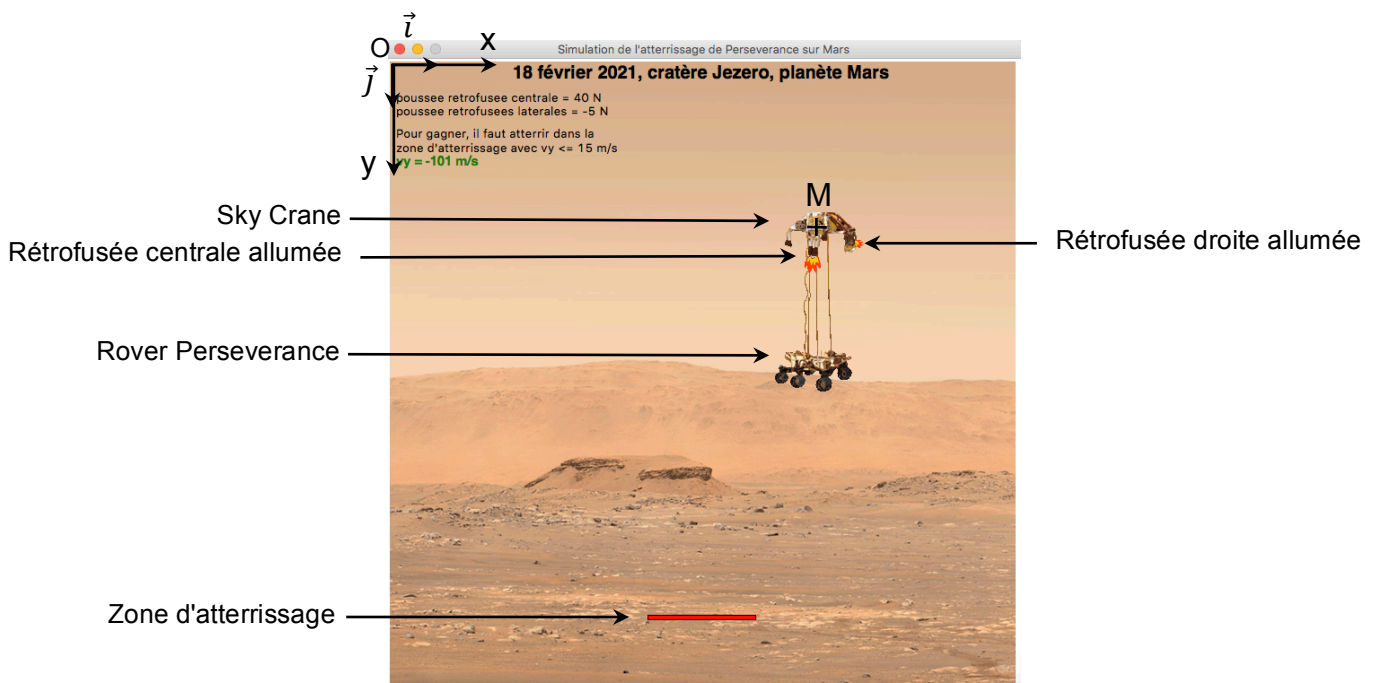
Votre objectif est de compléter le programme pour pouvoir poser en douceur Perseverance sur Mars !

On considère dans cette étude la dernière étape de l'atterrissage de Perseverance. Le système étudié, de masse m , est le rover soutenu à l'étage de descente. Le mouvement du centre de masse du système, noté M , est étudié dans le repère $(O ; \vec{i} ; \vec{j})$ (voir schéma ci-dessous).

L'étage de descente contrôle sa trajectoire grâce à 3 rétrofusées : gauche, centrale et droite. Les 3 rétrofusées peuvent être allumées en même temps. On notera \vec{F}_g , \vec{F}_c et \vec{F}_d les forces de poussée exercées respectivement par les rétrofusées gauche, centrale et droite.

Le vecteur vitesse initiale \vec{v}_0 du système est vertical orienté vers le bas et a pour valeur 100 m.s^{-1} .

La position initiale du système a pour coordonnées $(500 ; 0)$.

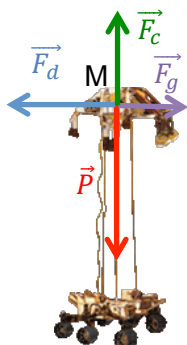


1. Définir le référentiel d'étude.

Le référentiel d'étude est le référentiel martien supposé galiléen (surface martienne muni du repère $(O ; \vec{i} ; \vec{j})$ et d'un repère de temps).

2. Déterminer les forces exercées sur le système. Les représenter sur un schéma.

Les forces exercées sur le système sont : le poids \vec{P} et les forces de poussée \vec{F}_g , \vec{F}_c et \vec{F}_d .



3. Un extrait du programme Python est donné ci-dessous. Identifier la valeur de l'intensité de la pesanteur martienne et de la force de poussée maximale des rétrofusées gauche, centrale et droite.

```
32 from tkinter import *
33 import time
34
35 gravity = 3.8
36 central_thrust_max = 73000.
37 lateral_thrust_max = 9000.
38 vmax = 15.
```

L'intensité de la pesanteur est représentée par la variable *gravity*, de valeur $3,8 \text{ m.s}^{-2}$.

La force de poussée maximale de la rétrofusée centrale est représentée par la variable *central_thrust_max*, de valeur 73000 N.

La force de poussée maximale des rétrofusées gauche et droite est représentée par la variable *lateral_thrust_max*, de valeur 9000 N.

4. Dans un référentiel donné, la somme vectorielle des forces exercées sur un système de masse m constante est reliée de façon approchée au vecteur variation de vitesse $\Delta \vec{v}$ du centre de masse M pendant une durée très courte Δt par :

$$\sum \vec{F} = m \frac{\Delta \vec{v}}{\Delta t}$$

Déterminer les coordonnées dans le repère $(O ; \vec{i} ; \vec{j})$ du vecteur variation de vitesse $\Delta \vec{v}$ du centre de masse M pendant une durée très courte Δt .

$$\sum \vec{F} = m \frac{\Delta \vec{v}}{\Delta t}$$

$$\Delta \vec{v} = \frac{\Delta t}{m} \times \sum \vec{F}$$

$$\Delta \vec{v} \begin{cases} \Delta v_x = \frac{1}{m} \times (F_g - F_d) \times \Delta t \\ \Delta v_y = \left(g - \frac{F_c}{m} \right) \times \Delta t \end{cases}$$

5. On donne ci-dessous un deuxième extrait du programme. Repérer la ligne donnant le calcul de la coordonnée Δv_x . Donner la masse du système {rover + étage de descente}.

```
90 now = time.time()
91 delta_t = now - self.time
92 if central_rocket.allume == True:
93     self.central_thrust = central_thrust_max
94 if left_rocket.allume == True:
95     self.left_thrust = lateral_thrust_max
96 if right_rocket.allume == True:
97     self.right_thrust = lateral_thrust_max
98 if central_rocket.allume == False:
99     self.central_thrust = 0
100 if left_rocket.allume == False:
101     self.left_thrust = 0
102 if right_rocket.allume == False:
103     self.right_thrust = 0
104 delta_vx = delta_t * (self.left_thrust - self.right_thrust)/1825.
105 delta_vy = #A COMPLETER
106 self.vx = self.vx + delta_vx
107 self.vy = self.vy + delta_vy
108 delta_x = #A COMPLETER
109 delta_y = #A COMPLETER
110 self.x = self.x + delta_x
111 self.y = self.y + delta_y
112 jeu.canevas.move(self.widget_lander, delta_x, delta_y)
```

La coordonnée Δv_x est calculée à la ligne 104 : c'est la variable *delta_vx*.

La masse du système est de 1825 kg.

6. En vous inspirant de la syntaxe de la ligne donnant le calcul de la coordonnée Δv_x , compléter la ligne permettant de calculer la coordonnée Δv_y .

*delta_vy = delta_t * (gravity - self.central_thrust / 1825.)*

7. Déterminer les coordonnées dans le repère $(O ; \vec{i} ; \vec{j})$ du vecteur variation de position $\overrightarrow{\Delta OM}$ du centre de masse M pendant une durée très courte Δt en fonction du vecteur vitesse \vec{v} .

$$\vec{v} = \frac{\overrightarrow{\Delta OM}}{\Delta t} \text{ donc } \overrightarrow{\Delta OM} = \vec{v} \times \Delta t \text{ Les coordonnées de } \overrightarrow{\Delta OM} \text{ sont } \overrightarrow{\Delta OM} \begin{cases} \Delta x = v_x \times \Delta t \\ \Delta y = v_y \times \Delta t \end{cases}$$

8. Repérer dans le programme les lignes correspondant au calcul des coordonnées du vecteur vitesse \vec{v} puis compléter les lignes permettant de calculer les coordonnées Δx et Δy .
Les coordonnées v_x et v_y sont calculées respectivement aux lignes 106 et 107 : ce sont les variables $self.vx$ et $self.vy$.
`delta_x = delta_t * self.vx`
`delta_y = delta_t * self.vy`

Vous avez complété le programme Python !
Votre objectif désormais : poser Perseverance en douceur ($v_y < 15$ m/s)
dans la zone d'atterrissage.