



MIRAGESIM RACING MISSION 3 (NIVEAU SECONDE)

BIENVENUE !



Bienvenue dans l'écurie Mirage en tant qu'ingénieur d'essais ! Vous allez prendre place à bord d'un Monster truck afin d'étudier le comportement du véhicule lors d'un saut. Vous recevrez ensuite de la part de notre ingénieur télémétrie, les données physiques du véhicule au cours du temps.

CONDUITE

Choisir « **Temps d'échantillonnage : 0.25 s** » puis choisir « **Mission C1** »



COMMANDES UTILISEES



Accélération



Freinage



Activer / désactiver l'acquisition des données pendant tout le mouvement



Changer la Caméra (ou pivotez la souris) pour mieux observer le véhicule



Ralenti. Mode qui vous donne le temps d'ajuster au mieux la trajectoire



Tourner à gauche. Appuyer par petites touches pour ajuster la trajectoire



Tourner à droite. Appuyer par petites touches pour ajuster la trajectoire

TRAVAIL A REALISER

Plans des environs (vue du dessus)



Figure 1 : Vue du dessus du point de départ

Travail à réaliser

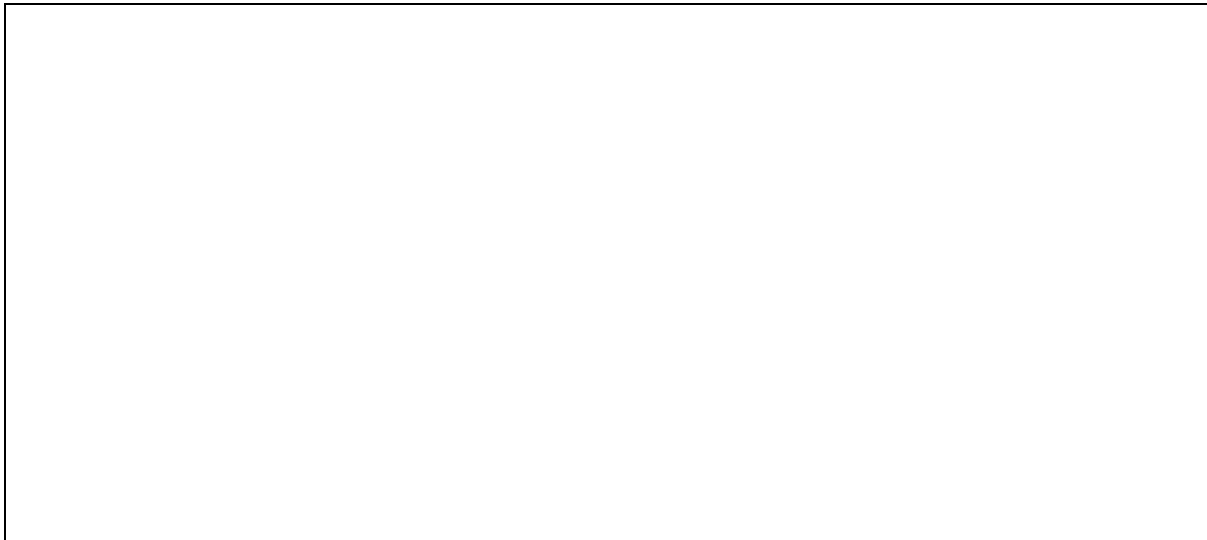
1. En vous déplaçant aux alentours du point de départ en mode acquisition (Touche A), **représenter 2 axes** pour faire **un repère sur la figure 1** Vous avez **le choix entre x, y et z** ;
On représente l'origine à l'un des quatre coins de l'image.

Acquisition du saut à réaliser

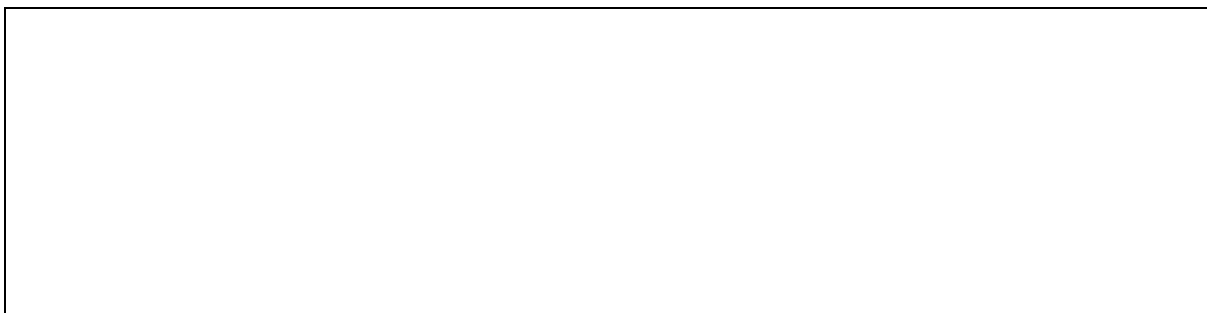
Recharger la mission C1 pour revenir au point de départ.

Utiliser la rampe qui se trouve presque en face du point de départ selon l'axe x. En accélérant en permanence, **réaliser l'acquisition du saut complet sur cette rampe** jusqu'à l'impact final sur le sol. On commence l'acquisition à vitesse nulle en bas de la rampe.

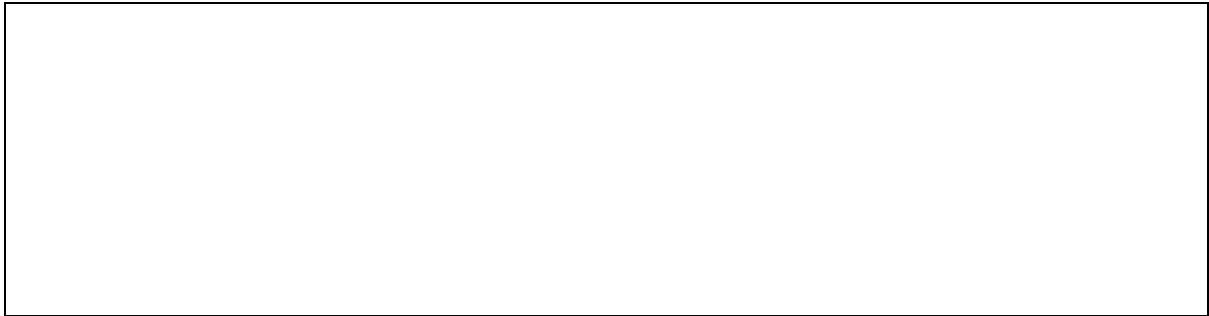
2. Réaliser le graphique des positions successives du véhicule en fonction des axes x et z.
Représenter le graphique obtenu ci-dessous :



3. Sur le graphique précédent, représenter les **forces exercées sur le véhicule** quand celui-ci **est sur la rampe**. *On ne représente les forces de frottement.*
4. Sur le graphique précédent, représenter les **forces exercées sur le véhicule** quand celui-ci ne touche plus la rampe, **pendant son vol**. *On ne représente les forces de frottement.*
5. Modifier le programme pour calculer et afficher les vitesses v_x et v_z dans la console.
Ecrire le code correspondant :



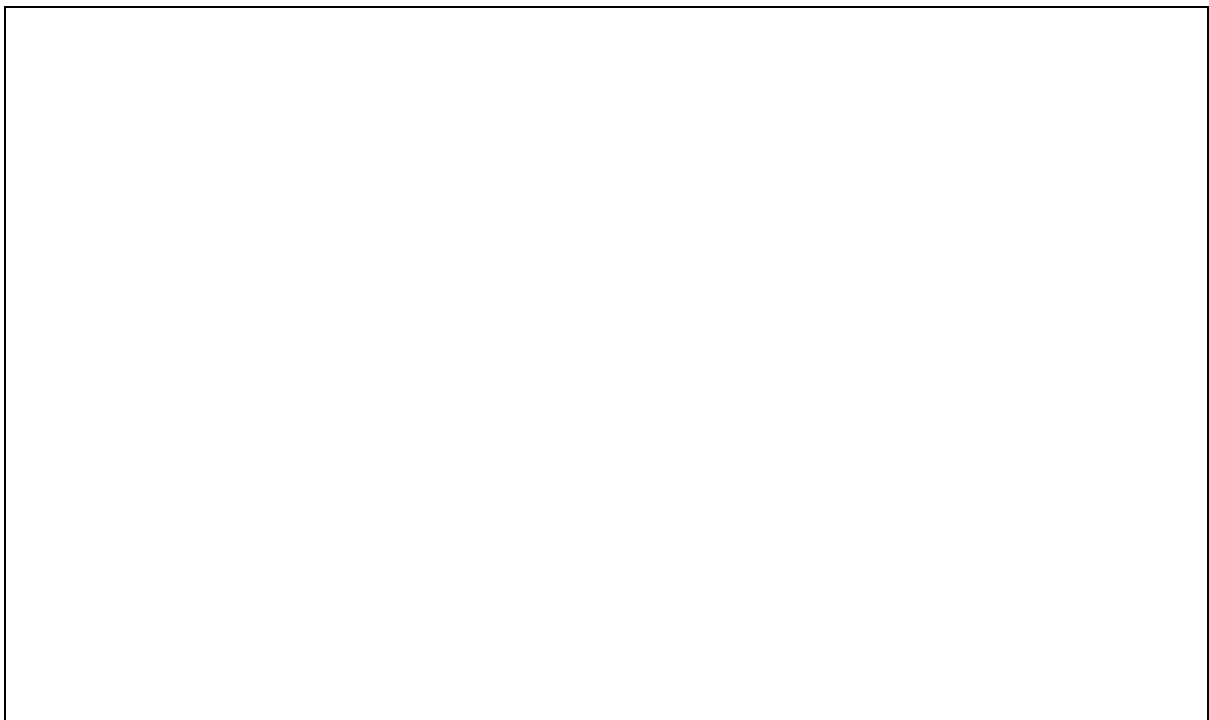
6. Modifier le programme pour calculer et afficher les variations de vitesse $diffvx$ et $diffvz$ dans la console. Ecrire le code correspondant :



7. Modifier le programme pour calculer et afficher les variations de vitesse $VariationVitesse$ utilisant le théorème de Pythagore dans la console. Ecrire le code correspondant :

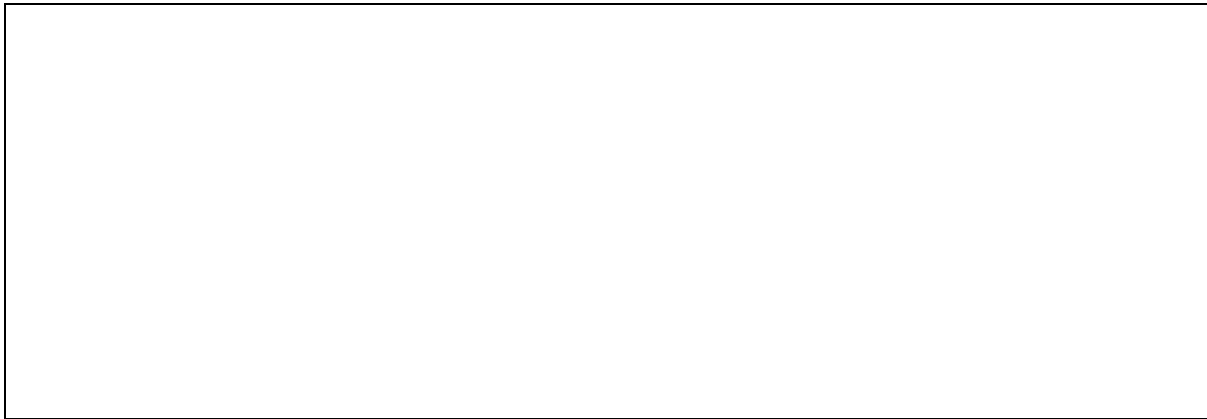


8. Modifier le programme pour afficher sur le graphique **les vecteurs vitesse** selon v_x et v_z . Représenter le graphique obtenu ci-dessous :



Note importante : A partir de ce point, supprimer les vecteurs vitesse du graphique de votre script Python (vous pouvez les garder si vous voulez, mais cela va être un peu confus par la suite ... parce que l'on va rajouter d'autres choses sur le graphique ... c'est comme vous voulez en fait, c'est juste un conseil... Mais si vous arrivez à vous y retrouver ... alors pourquoi pas ! mais bon, je déconseille. Voilà, vous avez tous les éléments en votre possession pour prendre la bonne décision. Bon, on va passer à la question neuf maintenant, parce que j'ai l'impression qu'on est en train de s'égarer un peu en fait ...)

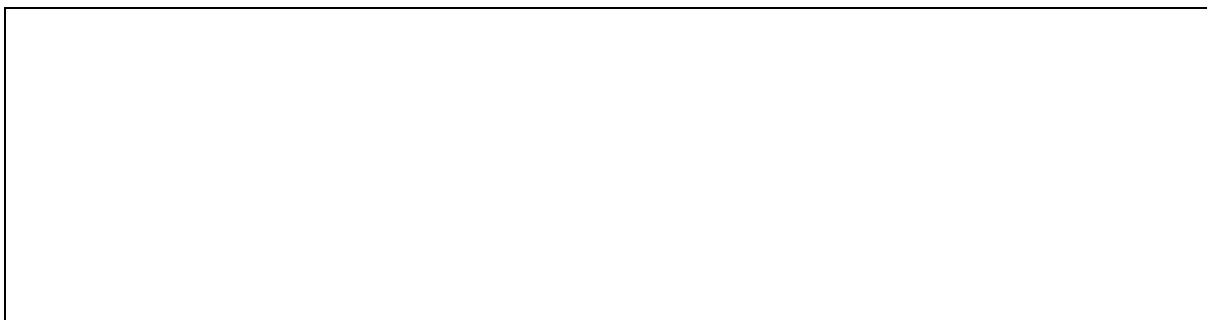
9. Modifier le programme pour afficher sur le graphique **les vecteurs variations de vitesse** selon `diffvx` et `diffvz`. Représenter le graphique obtenu ci-dessous :



10. Observez, sur le graphique ci-dessus, les points pour lesquels le véhicule est en chute libre. Que représentent les vecteurs variations de vitesse par rapport aux forces appliquées sur le véhicule ?



11. Observez attentivement les valeurs de `VariationVitesse` dans la console pour les points pour lesquels le véhicule est en chute libre. Faire la moyenne de ces valeurs qui s'approche de 10 avec votre calculatrice. De quelle valeur bien connue se rapproche-t-on ?



12. Trouver sur la carte le looping (axe y,z) et le mur vertical (axe y,z). Réaliser pour chacun des obstacles des acquisitions avec les mêmes algorithmes légèrement modifiés. Représenter les graphiques obtenus pour chaque obstacle.



DOCUMENT 1 : TRACER UN GRAPHIQUE EN PYTHON AVEC MATPLOTLIB

Charger la librairie pour réaliser les graphiques (déjà importé)

```
import matplotlib.pyplot as plt
```

Ajout d'une grille en arrière-plan

```
plt.grid()
```

Création d'une échelle identique sur les deux axes (indispensable)

```
plt.gca().set_aspect('equal', adjustable='box')
```

Ajout des légendes indispensables au graphique

```
plt.xlabel('à compléter')
plt.ylabel('à compléter')
plt.title('à compléter')
```

Définition des axes et du style des points

```
plt.plot(x, y, 'ro', label='')
```

Affichage du graphique

```
plt.show()
```

DOCUMENT 2 : TRACER DES VECTEURS VITESSE AVEC MATPLOTLIB

Configuration initiale du graphique (voir Doc précédent)

```
plt.grid()
plt.xlabel("à compléter")
plt.ylabel("à compléter")
plt.title("à compléter")
```

Ajout une seule flèche, dont la longueur est basée sur les valeurs de v_x et v_y

```
echelle = 1 #permet de réduire la taille des flèches
plt.arrow(x[1],y[1],echelle*vx[1],echelle*vy[1],head_width=1,
head_length=1)
```

Affichage du graphique

```
plt.plot(x,y,'ro')
plt.show()
```

Régler la longueur des flèches

Si les flèches sont trop longues ou trop petites, ajuster le coefficient `echelle`

PROGRAMME DE LA CLASSE DE SECONDE

Déroulement de la séance :

- L'activité se réalise en demi-groupe lors d'une séance de TP avec un IDE python de votre choix
- Les élèves ont déjà eu une introduction à Python dans le cours précédent
- La séance se déroule en 1h30 pour des élèves très rapides. La plupart des élèves terminent cette activité à la maison. Libre à vous de découper autrement cette proposition d'activité pédagogique en fonction de vos contraintes, ou de rajouter des questions sur les thématiques non abordées du programme de seconde.

Programme :

Décrire un mouvement

- Choisir un référentiel pour décrire le mouvement d'un système.
- **Capacité numérique** : représenter les positions successives d'un système modélisé par un point lors d'une évolution unidimensionnelle ou bidimensionnelle à l'aide d'un langage de programmation.
- Définir le vecteur vitesse moyenne d'un point.
- Approcher le vecteur vitesse d'un point à l'aide du vecteur déplacement.
- Caractériser un mouvement rectiligne uniforme ou non uniforme.
- **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.

Modéliser une action sur un système

- Représenter qualitativement la force modélisant l'action d'un support dans des cas simples relevant de la statique.

Principe d'inertie

- Exploiter le principe d'inertie ou sa contraposée pour en déduire des informations soit sur la nature du mouvement d'un système modélisé par un point matériel, soit sur les forces.