

BACCALAURÉAT GÉNÉRAL**Épreuve pratique de l'enseignement de spécialité physique-chimie
Évaluation des Compétences Expérimentales**

Cette situation d'évaluation fait partie de la banque nationale.

ÉNONCÉ DESTINÉ AU CANDIDAT

NOM :	Prénom :
Centre d'examen :	n° d'inscription :

Cette situation d'évaluation comporte **quatre** pages sur lesquelles le candidat doit consigner ses réponses. Le candidat doit restituer ce document avant de sortir de la salle d'examen.

Le candidat doit agir en autonomie et faire preuve d'initiative tout au long de l'épreuve.

En cas de difficulté, le candidat peut solliciter l'examineur afin de lui permettre de continuer la tâche.

L'examineur peut intervenir à tout moment, s'il le juge utile.

L'usage de calculatrice avec mode examen actif est autorisé. L'usage de calculatrice sans mémoire « type collègue » est autorisé.

CONTEXTE DE LA SITUATION D'ÉVALUATION

Dans le domaine astronautique, un lanceur est une fusée utilisée pour placer un engin spatial en orbite, ou pour l'envoyer dans l'espace interplanétaire. Grâce à lui, l'engin spatial peut atteindre une vitesse suffisamment importante pour être mis en orbite, ou bien échapper à l'attraction terrestre.

La plupart des lanceurs ne sont pas réutilisables, c'est-à-dire que leurs composants ne sont pas récupérés après usage, ce qui augmente leur coût de manière significative. Au cours de l'histoire astronautique il y a eu plusieurs tentatives de mise au point de lanceurs réutilisables. C'est le lanceur Falcon 9 de la société SpaceX qui a atteint le premier, à la fin de l'année 2015, le stade opérationnel. Les coûts des lancements ont ainsi été significativement réduits.

Le but de cette épreuve est d'étudier le mouvement d'un lanceur Falcon lors de son atterrissage.

INFORMATIONS MISES À DISPOSITION DU CANDIDAT

Mission NROL-108

Le 19 décembre 2020, à partir du centre spatial Kennedy, SpaceX a lancé avec succès la mission NROL-108 avec le lanceur Falcon 9. Le premier étage, qui avait déjà été utilisé quatre autres fois, s'est posé avec succès sur la zone d'atterrissage prévue au sol, après un peu plus de huit minutes de vol.

NROL-108 est une mission militaire classifiée visant à lancer un satellite-espion américain sur une orbite basse (environ 540 km d'altitude).

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 poste informatique mis à disposition. **La longueur du lanceur Falcon 9 Bloc 5 est de 41,5 m.**



On pourra considérer que la valeur absolue de la composante verticale de la vitesse du lanceur ne doit pas excéder $20 \text{ km}\cdot\text{h}^{-1}$ juste avant l'atterrissage. Au-delà, le lanceur risquerait d'être endommagé, ce qui pourrait compromettre sa réutilisation ultérieure.

TRAVAIL À EFFECTUER

1. Nature du mouvement du lanceur à l'atterrissage (20 minutes conseillées)



Visionner la vidéo de l'atterrissage du premier lanceur.

1.1. Émettre une hypothèse sur la nature du mouvement du lanceur (trajectoire et vitesse), dans le référentiel terrestre, au cours des dernières secondes précédant son atterrissage.

Je suppose qu'au cours des dernières secondes- dans le référentiel terrestre- le mouvement est rectiligne et ralenti

1.2. Proposer un protocole expérimental pour obtenir les données des différentes positions d'un point du lanceur au cours de son atterrissage.

Faire un pointage des dernières secondes de l'atterrissage grâce au logiciel Latispro en veillant à sélectionner l'étalon (ici le lanceur) et l'origine.

APPEL n°1		
	Appeler le professeur pour lui présenter le protocole expérimental, ou en cas de difficulté	

Mettre en œuvre le protocole pour obtenir ces données d'un point du lanceur entre les images n° 60 et n° 77. Le point à considérer sera pris à la limite entre la zone blanche et la zone sombre inférieure du lanceur.



Point à considérer

1.3. Quelle est la durée séparant deux positions successives ?

Une seconde se découle pour 3 images.

$$\Delta t = 0,33 \text{ s}$$

Suivre les instructions mentionnées dans la notice permettant la création du fichier **donnees.csv** contenant les trois variables t , x et y .

Ouvrir le programme Python fourni « **Programme_A.py** ». Compléter les lignes 15, 16 et 17 et lancer le programme.

1.4. L'hypothèse précédemment formulée est-elle confirmée ? Justifier en analysant la représentation graphique issue du programme.

En accord avec l'allure des points, valider ou reconsidérer l'hypothèse.

2. Étude du mouvement du lanceur (30 minutes conseillées)

Le programme **B.py** exploite également le fichier **donnees.csv** pour estimer vitesse et accélération du lanceur à différents instants.

2.1. Programme B – Partie 1 - Estimation de la vitesse juste avant l'atterrissage

Compléter la **partie 1** du programme **B** aux lignes 14 et 28 afin de calculer les coordonnées du vecteur vitesse pour tous les points et d'estimer la valeur de la composante verticale de la vitesse du lanceur à l'instant de l'image 75 (point n° 15).

Exécuter le programme et noter la valeur de la composante verticale de la vitesse obtenue.



$$v_{15} = 0,87 \text{ m.s}^{-1}$$

2.2. Programme B – Partie 2 - Représentation du vecteur accélération

Supprimer les trois guillemets autour de la **partie 2** du programme (lignes 37 et 77) puis compléter la **partie 2** du programme **B** aux endroits indiqués afin :

- d'estimer la valeur des composantes de l'accélération aux points 6, 10 et 13 ;
- de représenter le vecteur accélération en ces points.

APPEL n°2

	Appeler le professeur pour lui présenter les résultats expérimentaux ou en cas de difficulté	
-----------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------

3. Exploitation des résultats expérimentaux (10 minutes conseillées)

3.1. La vitesse du lanceur est assimilée à la valeur absolue de la coordonnée verticale de la vitesse.

La vitesse du lanceur obtenue à l'instant de l'image 75 grâce à la **partie 1** du programme **B** est-elle compatible avec une réutilisation ultérieure du lanceur ? Justifier la réponse.

A la question précédente on a trouvé pour le point 15, un vitesse égale à $0,87\text{m}\cdot\text{s}^{-1}$.

Soit $v_{15} = 0,87 \times 3,6 = 3,13 \text{ km/h} < 20\text{km}\cdot\text{h}^{-1}$

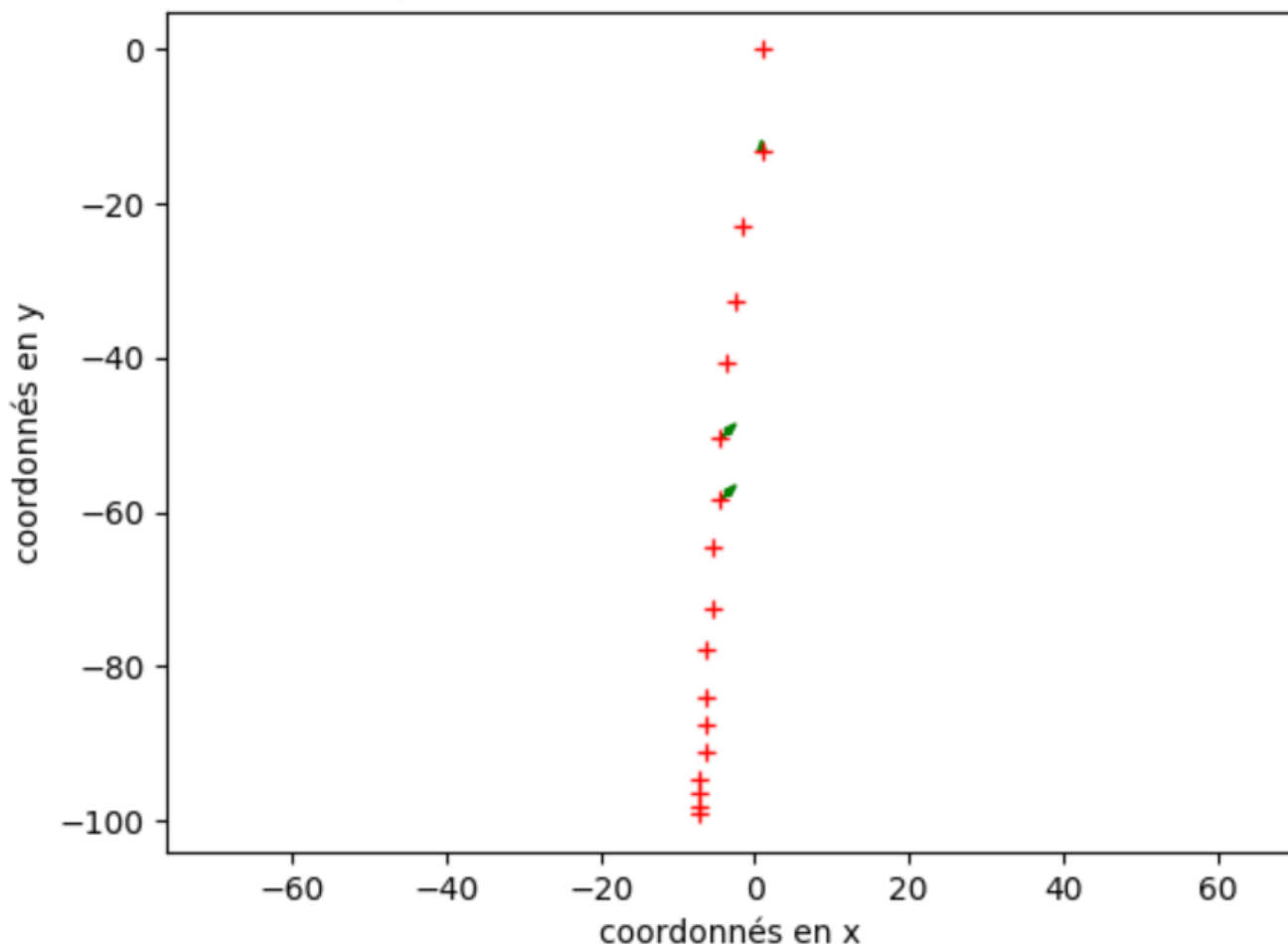
Cette vitesse est donc compatible avec une réutilisation ultérieure du lanceur.

3.2. La représentation des vecteurs accélération donnée par la **partie 2** du programme **B** permet-elle de confirmer l'hypothèse initiale ? Justifier la réponse.

A mesure que le lanceur atterrit, on remarque que le vecteur accélération est de plus en plus dirigé vers le haut. On peut d'ailleurs s'appuyer sur des valeurs numériques. En effet, si au point 1 l'accélération est nulle, elle est de $0,89\text{m}\cdot\text{s}^{-2}$ au point 5 et 6. Autrement dit, le vecteur accélération s'oppose plus au moins au vecteur vitesse (dans le même sens que la trajectoire de notre système) au cours du temps. Il s'agit donc d'un mouvement rectiligne ralenti

Défaire le montage et ranger la pailasse avant de quitter la salle.

évolution de la position du sommet du lanceur au cours du temps



Programme:

```

# PROGRAMME B - PARTIE 1
#####

# import des bibliothèques
import pandas # lecture de fichiers de données

# création de deux Listes contenant les coordonnées x et y des points au cours du mouvement
donnees = pandas.read_csv("donnees9.csv", sep=";", decimal=",")
x= donnees.x
y = donnees.y

# intervalle de temps entre deux points consécutifs
Deltat = 0.33# ***commande à compléter***

point = 15

vx = []
for i in range(1, len(x)-1):
    vx.append(abs(x[i+1]-x[i])) # ***commande à compléter (calcul de la vitesse selon l'axe des abscisses et ajout à la liste vx)***

# selon l'axe des y
vy = []
for i in range(1, len(x)-1):
    vy.append(abs(y[i+1]-y[i])) # ***commande à compléter (calcul de la vitesse selon l'axe des ordonnées et ajout à la liste vy)***

# affichage de la vitesse verticale au point i
print("La coordonnée de la vitesse selon l'axe y au point", point, "vaut : vy =", vy[point-1], "m/s.")

import des bibliothèques
import matplotlib.pyplot as plt # tracés de graphiques

# calcul des coordonnées du vecteur accélération pour tous les points
# selon l'axe des x
ax = []
for i in range(1, len(vx)-1):
    ax.append(abs(vx[i+1]-vx[i])) # ***commande à compléter (calcul de l'accélération selon l'axe des abscisses et ajout à la liste ax)***

# selon l'axe des y
ay = []
for i in range(1, len(vy)-1):
    ay.append(abs(vx[i+1]-vx[i])) # ***commande à compléter (calcul de l'accélération selon l'axe des ordonnées et ajout à la liste ay)***

point = 6

# affichage de l'accélération verticale au point choisi
print("L'accélération verticale au point", point, "vaut : ay =", ay[point-2], "m/s².")

# construction du vecteur accélération au point choisi
plt.arrow(x[point],y[point],ax[point-2],ay[point-2], head_width=1, color = "green")

point = 1

# affichage de l'accélération verticale au point choisi
print("L'accélération verticale au point", point, "vaut : ay =", ay[point-2], "m/s².")

# construction du vecteur accélération au point choisi
plt.arrow(x[point],y[point],ax[point-2],ay[point-2], head_width=1, color = "green")
point = 10

# affichage de l'accélération verticale au point choisi
print("L'accélération verticale au point", point, "vaut : ay =", ay[point-2], "m/s².")

```

```
# construction du vecteur accélération au point choisi
plt.arrow(x[point],y[point],ax[point-2],ay[point-2], head_width=1, color = "green")

# représentation graphique
plt.plot(x,y,"r+") # ***commentaire à compléter***
plt.axis("equal") # définition d'un repère orthonormé
plt.xlabel("coordonnés en x") # ***commande à compléter (titre de l'axe des abscisses)***
plt.ylabel("coordonnés en y") # ***commande à compléter (titre de l'axe des ordonnées)***
plt.title("évolution de la position du sommet du lanceur au cours du temps") # ***commande à compléter (titre du
graphique)***
plt.show() # affichage du graphique
```