

TP – Informatique Scientifique

Étude du comportement balistique d'un projectile

Université Paul Sabatier

L3 Mécanique et Énergétique

WOLI Sedjro Friedly

Année universitaire 2025–2026

Table des matières

1	Introduction	3
2	Corps du devoir	3
2.1	Modèle 1 : Approche analytique	3
2.1.1	Calcul de la trajectoire	3
2.1.2	Vitesse en fonction du temps	4
2.1.3	Famille de trajectoires et angle optimal	5
2.2	Modèle 2 : Résolution numérique	6
2.2.1	Validation de l'intégrateur	6
2.2.2	Comparaison des trajectoires	6
2.3	Modèle 3 : Effets aérodynamiques complets	6
2.3.1	Trajectoires pour différents coefficients	7
2.3.2	Isocontours de la distance d'impact	7
3	Conclusion	8

1 Introduction

Ce projet porte sur l'étude du mouvement d'un objet lancé dans un champ de pesanteur, un problème classique de mécanique du point. L'objectif est de mettre en œuvre la programmation orientée objet (POO) en Python pour simuler et analyser des trajectoires balistiques de complexité croissante.

Le travail est divisé en trois modules :

1. Le **Modèle 1**, purement analytique, négligeant les forces de frottement.
2. Le **Modèle 2**, utilisant une résolution numérique d'équations différentielles ordinaires (EDO) pour valider l'approche.
3. Le **Modèle 3**, intégrant des forces aérodynamiques (traînée et portance) ainsi qu'une force de propulsion.

2 Corps du devoir

2.1 Modèle 1 : Approche analytique

Le modèle de référence repose sur la seconde loi de Newton, en supposant que seule l'accélération de la pesanteur g agit sur le projectile.

2.1.1 Calcul de la trajectoire

La trajectoire est une parabole définie par

$$z(x) = -\frac{g}{2 v_0^2 \cos^2 \alpha} \frac{x^2}{v_0^2 \cos^2 \alpha} + \tan(\alpha) x + h.$$

Pour un lancement à $h = 20\text{ m}$ et $v_0 = 50\text{ m s}^{-1}$, on obtient une trajectoire régulière (Fig. 1), tracée avec une épaisseur de trait `lw=3` conformément aux consignes.

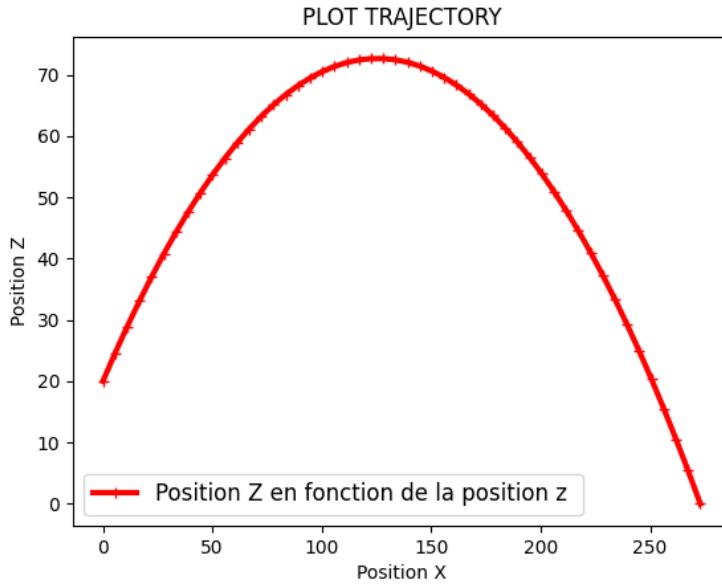


FIGURE 1 – Exemple de trajectoire $z(x)$ pour le Modèle 1.

2.1.2 Vitesse en fonction du temps

L'accélération étant constante ($a_z = -g$), la composante verticale de la vitesse s'écrit

$$v_z(t) = v_0 \sin(\alpha) - gt.$$

La fonction `plot_trajectories` trace $v_z(t)$ ainsi que la norme $\|\mathbf{v}(t)\|$ en fonction du temps (Fig. 2).

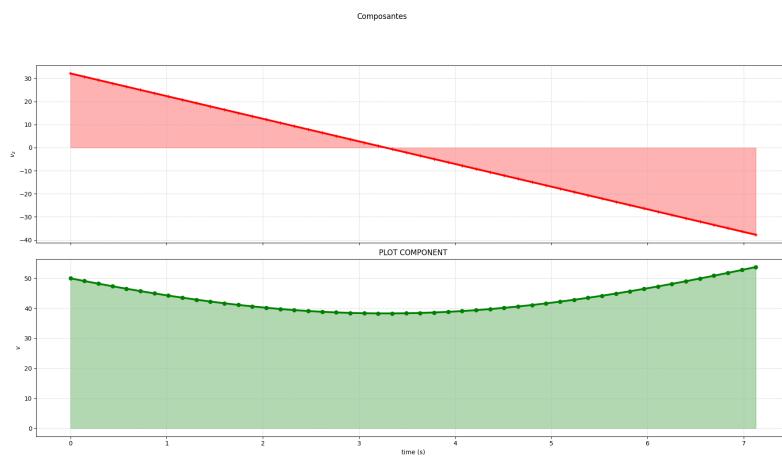


FIGURE 2 – Composantes de la vitesse et norme de $\mathbf{v}(t)$ pour le Modèle 1.

La vitesse v_z diminue linéairement à cause de l'accélération gravitationnelle, tandis que la vitesse totale atteint son minimum à l'apogée, où $v_z = 0$.

2.1.3 Famille de trajectoires et angle optimal

On trace sur un même graphique les trajectoires pour plusieurs angles de lancement α , avec une légende indiquant la valeur de chaque angle (Fig. 3). Cela permet d'analyser l'influence de α sur la portée et l'altitude maximale.

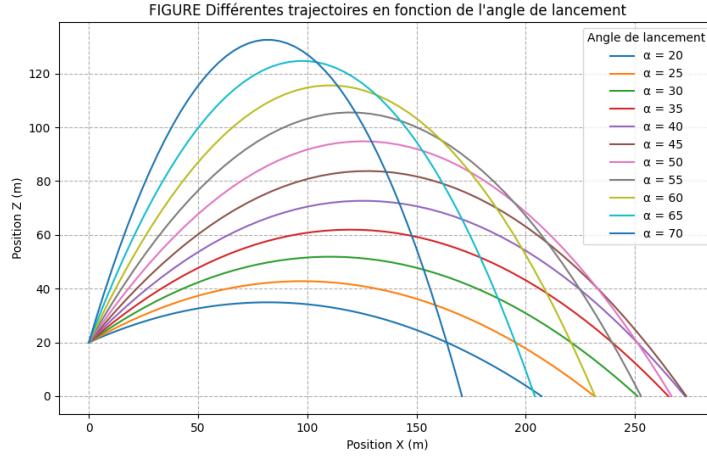


FIGURE 3 – Famille de trajectoires pour différents angles de lancement α .

On observe une grande échelle de portée, car les points d'impact dans le code dépendent de chaque instant d'impact au sol.

Pour un tir à partir du sol ($h = 0$), l'étude montre que l'angle optimisant la portée est $\alpha = 45^\circ$. Dans notre configuration avec $h = 20$ m, l'angle optimal est inférieur à 45° , tandis que l'altitude maximale h_{\max} augmente avec l'angle (Fig. 4).

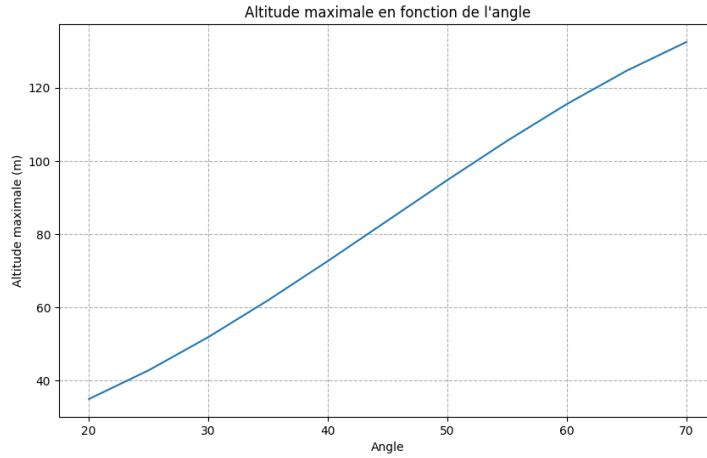


FIGURE 4 – Altitude maximale en fonction de l'angle de lancement α .

2.2 Modèle 2 : Résolution numérique

Le Modèle 2 résout le système différentiel $\frac{dy}{dt} = f(t, y)$, où $y = [x, z, v_x, v_z]^\top$, à l'aide d'un intégrateur de type Runge–Kutta d'ordre 4 (par exemple `odeint` de `scipy.integrate`).

2.2.1 Validation de l'intégrateur

```

117
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL POINTS
(venv) freddyfriendly@DELL-PC16250:~/Documents/General_git_hub_file/UNIV_project_updates$ python3 ./Ballistic/main_model_2.py
9.15780634 8.6702680 8.17381480 7.6685517 7.15442040 6.63119781
6.09974014 5.5591565 5.09974687 4.45151125 3.88444965 3.30856286
2.72384849 2.13638989 1.52794339 0.91675186 0.29673435 -0.33210914
0.59000000 0.00000000 -0.93574256 -3.6887164
analytical solution at t = 3.000000
x, z
numerical solution at the same time
x, z
numerical solution at the same time
x, z
[4]: 0.0000000000000000
l'erreur max est : 9.981526716273947e-11
input values: {'t': 1: np.float64(2.83415194114305), 'p': np.float64(53.26471323527776), 'angle': np.float64(-2756.478782308943), 'v': [np.float64(18.79385241571817), np.float64(-20.9531749749), np.float64(49.07687114077876)]}
input values: {'t': 1: np.float64(2.83415194114305), 'p': np.float64(56.973437822560115), 'angle': np.float64(-57.02229278625156), 'v': [np.float64(15.3208886237956), np.float64(-23.61220795407775), np.float64(28.14722811407091)]}
normal end of execution
(venv) freddyfriendly@DELL-PC16250:~/Documents/General_git_hub_file/UNIV_project_updates$ pkill -9 python
A process terminated with code 134 (SIGABRT) at 10:58:20.000000

```

FIGURE 5 – Comparaison des solutions analytique et numérique pour la trajectoire.

On compare les résultats du Modèle 2 à la solution analytique du Modèle 1 pour valider l'intégrateur numérique. L'erreur absolue maximale obtenue est de l'ordre de $9,98 \times 10^{-11}$, bien inférieure au seuil de 10^{-7} imposé par le cahier des charges, ce qui valide l'implémentation.

2.2.2 Comparaison des trajectoires

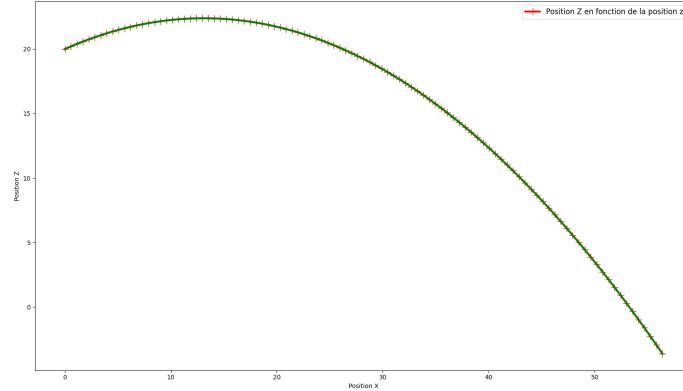


FIGURE 6 – Solution analytique vs solution numérique pour la trajectoire.

La superposition des courbes analytique et numérique montre que les trajectoires coïncident quasiment parfaitement (Fig. 6), confirmant la robustesse du schéma numérique retenu.

2.3 Modèle 3 : Effets aérodynamiques complets

Le Modèle 3 introduit les forces aérodynamiques de traînée (coefficients C_d) et de portance (coefficients C_l), ainsi qu'une force de propulsion T_0 active sur une phase limitée du vol.

2.3.1 Trajectoires pour différents coefficients

On calcule les trajectoires pour différentes combinaisons de C_d et C_l . Les trajectoires obtenues sont représentées sur la Fig. 7.

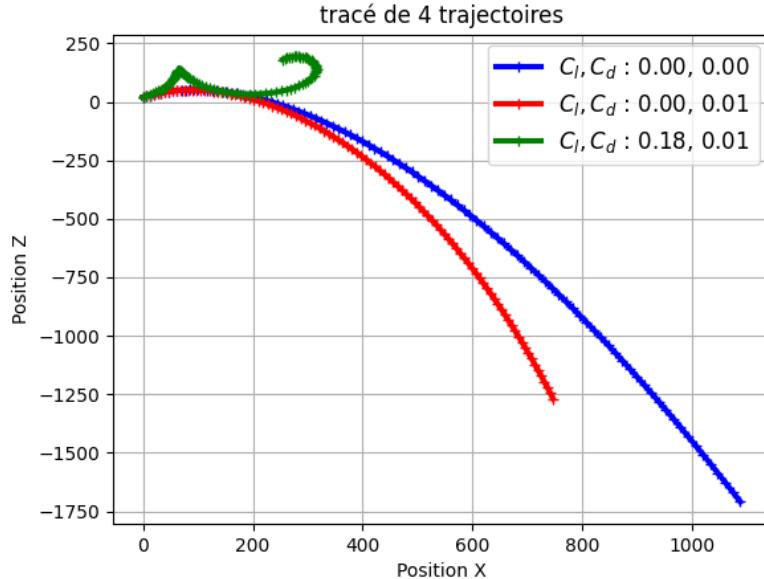


FIGURE 7 – Trajectoires pour différentes valeurs des coefficients C_d et C_l .

La traînée réduit la portée en s'opposant au mouvement et en dissipant l'énergie mécanique, tandis qu'une portance suffisante permet au projectile d'augmenter sa portée, voire générer des trajectoires plus complexes.

On constate que la courbe verte de Cl C_l et C_d oscille après avoir touché le sol.

On remarque également l'ordre de grandeur de z et de x très grande

2.3.2 Isocontours de la distance d'impact

La distance d'impact est évaluée sur une grille de valeurs de l'angle de tir α et du coefficient de traînée C_d . Les isocontours de portée dans le plan (α, C_d) sont représentés sur la Fig. 8.

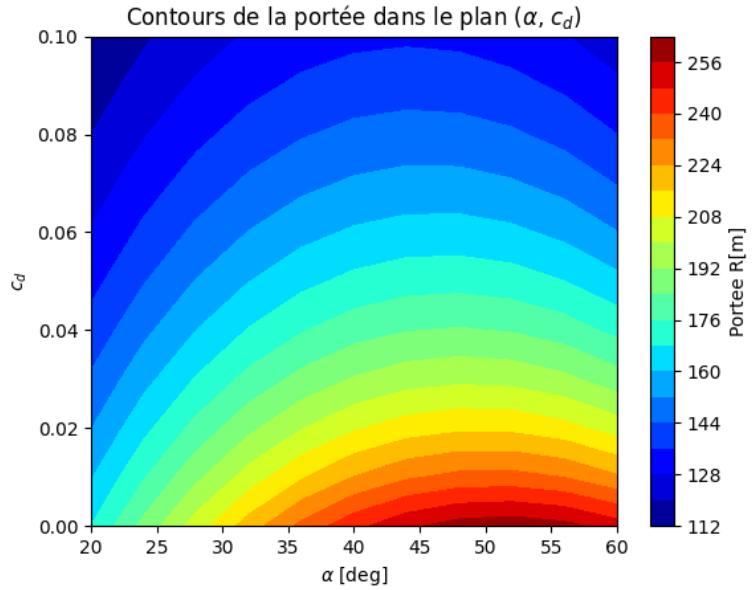


FIGURE 8 – Isocontours de la distance d’impact dans le plan (α, C_d) .

Ces cartes permettent d’identifier visuellement les couples (α, C_d) qui maximisent la portée pour une configuration aérodynamique donnée, et de discuter la sensibilité de la portée à ces paramètres. Mes isocontours ne cadre pas avec la figure proposer dans le malgré plusieurs analyse du code de ma part

3 Conclusion

L’introduction des forces de traînée, de portance et de propulsion a mis en évidence l’impact critique des phénomènes aérodynamiques sur la portée, l’altitude et la forme des trajectoires balistiques, par rapport au cas purement balistique idéal.