

1st week report

Reda YAHOU

6 août 2018

Introduction

Ce document a pour but de résumer le travail qui a été réalisé durant la première semaine de stage. L'objectif de ce papier est principalement de synthétiser les concepts qui interviennent dans la trajectographie de fusée, ainsi, certaines notions physiques et mécaniques seront passées sous silence et seul l'essentiel sera mis en lumière.

Dans un premier temps vont être abordés et présentés les éléments de base nécessaires à la compréhension de la trajectographie de fusée. Par la suite sera établie un parallèle entre ces notions théoriques et le code de calcul Pégase : on verra comment le vol de fusée y est intégré ainsi que les étapes et les méthodes utilisées par cette interface pour la simulation. un dernier paragraphe viendra conclure ce rapport en introduisant au cas multi-étagé.



FIGURE 1 – Kourou - Décollage d'Ariane 5.

Première partie

Trajectographie de fusée

1 Le vol de fusée

En trajectographie, on décompose le vol de fusée en 3 grandes phases (voir schéma ci-dessous) :

1. La phase propulsée
2. La phase balistique
3. La descente sous parachute

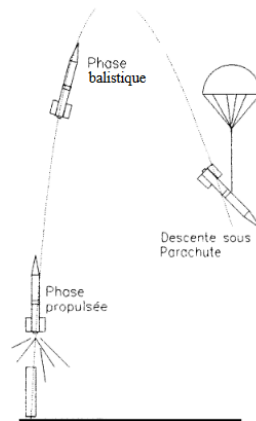


FIGURE 2 – Les phases du vol.

1. Cette première phase s'écoulant de l'instant de la mise à feu à la fin de combustion du propulseur, comprend une partie où la fusée est guidée par la rampe de lancement et une partie où la fusée est livrée à elle-même.
2. Après l'extinction du propulseur commence la phase balistique pendant laquelle la fusée, uniquement soumise à son poids et à la résistance de l'air, exploite la vitesse acquise pendant la propulsion pour atteindre son altitude maximale.
3. Une fois la culmination atteinte, lorsque l'engin commence à retomber, la phase balistique se poursuit jusqu'à l'ouverture du parachute, ce qui lui permet de revenir au sol tout en demeurant la plus intact possible.

Pendant le vol d'une fusée, deux paramètres vont être étudié :

- la simulation de sa trajectoire, c'est à dire l'allure de sa courbe, son altitude maximale, sa portée, le temps de vol. Ceci est entre autre modélisable à partir d'un système d'EDO.

— la stabilité de la fusée.

La seconde notion soulève deux aspect : un aspect "esthétique", on désire avoir une belle courbe et un aspect sécuritaire.

En effet, au delà du fait d'obtenir la trajectoire d'une fusée, on souhaite aussi que cette dernière soit stable, c'est à dire qu'elle retrouve vite sa position initiale dès qu'elle se met en incidence. Le paragraphe suivant aura pour objectif de détailler cette notion de stabilité.

2 Stabilité de la fusée

Les forces Pendant le vol, la fusée est soumise à trois forces : le Poids \vec{P} , la Poussée du moteur, \vec{F} et la résistance à l'air \vec{R} .

C'est la portance \vec{R}_N , composante normale de \vec{R} , qui va créer un moment et être responsable du mouvement de rotation. Elle va s'appliquer sur le centre de poussée dynamique **Cpa** et va dépendre de $C_N\alpha$, le coefficient de portance et du carré de la vitesse v^2 .

Stabilité statique La distance entre le **Cpa** et le **Cdm** (Centre de masse) est appelé la **marge statique**.

La position du **Cpa** et $C_N\alpha$ peuvent être déterminés grâce aux méthodes de Barrowmann, basées sur la géométrie des éléments de la fusée.

L'expérience montre que pour certains type de fusée, la stabilité dépend de la valeur de la marge statique i.e que la fusée est stable si la marge statique est comprise dans un certain intervalle de valeur.

De manière similaire, le terme $C_N\alpha$ va aussi influencer sur la stabilité dans la mesure ou il intervient dans le moment de rotation provoquer par la portance : la fusée sera stable si la rotation est bien contrôlée.

D'un point de vie pratique, c'est les dimensions des ailes qui vont influencer sur le facteur $C_N\alpha$.

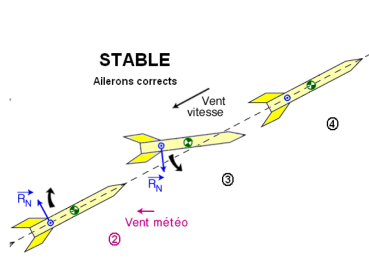


FIGURE 3 – Vol stable

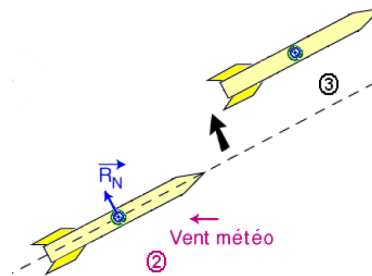


FIGURE 4 – Vol instable

En pratique, la fusée est stable lorsque :

1. la marge statique est entre 1.5 et 7.

2. $C_N\alpha$ est comprise entre 15 et 30.
3. le produit de ces deux paramètres est entre 30 et 100.

Stabilité dynamique Pendant le vol, la fusée va être soumise à des perturbation extérieure telles que le vent. Elles vont provoquer des oscillations chez cette dernière autour de sa position d'incidence nulle sources d'instabilités.

En pratique, ces oscillations vont décroître : c'est le phénomène d'amortissement. Dans le cas stable, le ratio entre deux extremums dans les oscillations diminue jusqu'à que l'incidence soit nulle.

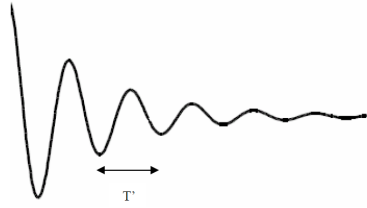


FIGURE 5 – Évolution de l'incidence d'une fusée stable.

L'équation donnée par le Principe fondamentale de la dynamique en rotation donne :

$$\frac{d^2\alpha}{dt^2} + 2\omega_0\xi\frac{d\alpha}{dt} + \omega_0^2\alpha = 0 \quad (1)$$

Le terme ξ représente le taux d'amortissement. D'un point de vue pratique, l'amortissement des oscillations d'une fusée ne doit être ni trop petit, ni trop grand : Si ξ est trop grand, l'amortissement serait trop faible et source d'instabilité. Trop petit, cela entraînerait une sur-stabilité, déviant totalement la fusée de sa trajectoire initiale.

Pour des mini-fusées, l'expérience suggère que le facteur doit se situer entre 0.05 à 0.2.

Conclusion : On peut obtenir la stabilité en respectant les conditions suivantes :

1. la marge statique est entre 1.5 et 7.
2. $C_N\alpha$ est comprise entre 15 et 30.
3. le produit de ces deux paramètres est entre 30 et 100.
4. ξ est entre 0.05 et 0.2.

3 Trajectoire

La trajectoire s'obtient grâce à la mise en équation du PFD ainsi qu'à un jeu d'intégrations successives, permettant de retrouver la position de la fusée. Durant

toute la phase de vol, les paramètres et les grandeurs varient et il conviendrait donc d'utiliser ces intégrations continues. Néanmoins, pour de courts instant (infinitésimaux), ces paramètres peuvent être considérés comme constant, et en particulier l'accélération.

Dans ce cas, et sur une durée dt limitée et assez petite, on a un mouvement uniformément accéléré. Ces équations deviennent alors facile à écrire et la base d'un schéma numérique permettant de modéliser la trajectoire. Il va être question de discrétiser le temps et l'espace le plus finement afin d'obtenir une bonne approximation de la trajectoire réelle. Les modèles sont simple et l'erreur d'approximation est faible (1^{er} ordre).

Ci-dessous les enchainements de calculs pour une trajectoire de fusée 2D.

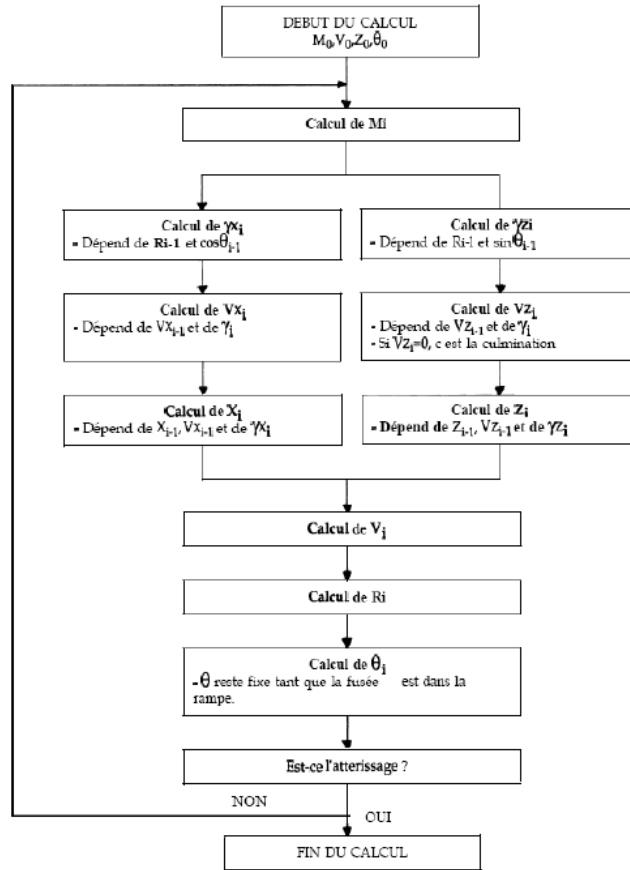


FIGURE 6 – Exemple pour le cas 2D.

Deuxième partie

Description du module Pégase

Maintenant que l'essentiel de la théorie sur le vol de fusée a été présenté, cette partie aura pour vocation de décrire l'interface de calcul de et de simulation JAVA Pégase qui fait appelle a tout le bagages vu ci-dessus.

4 Interface

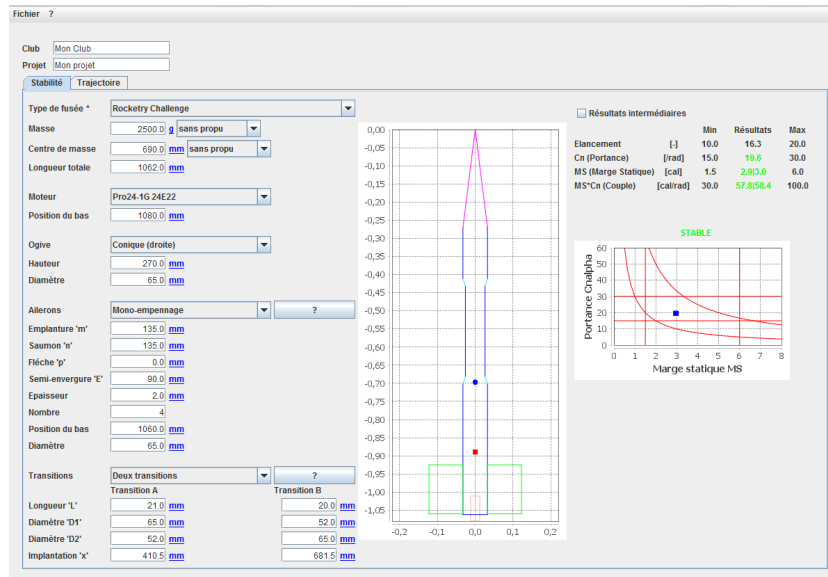


FIGURE 7 – Interface Pégase - calcul de la stabilité.

L'interface de Pégase se divise en deux sections : un premier module pour le calcul de stabilité et un module pour le calcul de trajectoire 2D.

1. **Stabilité** : Dans cette section, l'utilisateur est invité à modéliser une fusée en entrant ses caractéristiques dans des cellules (??). Une fois l'en-gin modélisé, Pégase stocke ces informations inputs et réalise les calculs afin de vérifier si la fusée vérifie les conditions de stabilité évoquées plus haut (marge statique, coefficient $C_N\alpha$ etc.) Au milieu de la fenêtre peut être observé un graphe représentant la fusée créée. À droite, un graphe de $C_N\alpha$ en fonction de la marge statique, indiquant si les caractéristiques sont dans le bon intervalle de stabilité ??.
2. **Trajectoire** : L'utilisateur doit entrer les inputs pour modéliser la phase de lancement via la rampe (hauteur, angle d'inclinaison, longueur etc.),

le vitesse du vent pendant le vol et les paramètre de la phase de parachute et le code de calcul s'occupe de tracer la trajectoire de la fusée.

À droite est représenté la trajectoire en fonction du temps et en fonction de l'axe des x ?? . Juste en dessous sont affichés les informations calculés à des moments clés du vol (sortie de rampe, apogée, vitesse max etc.) ??.

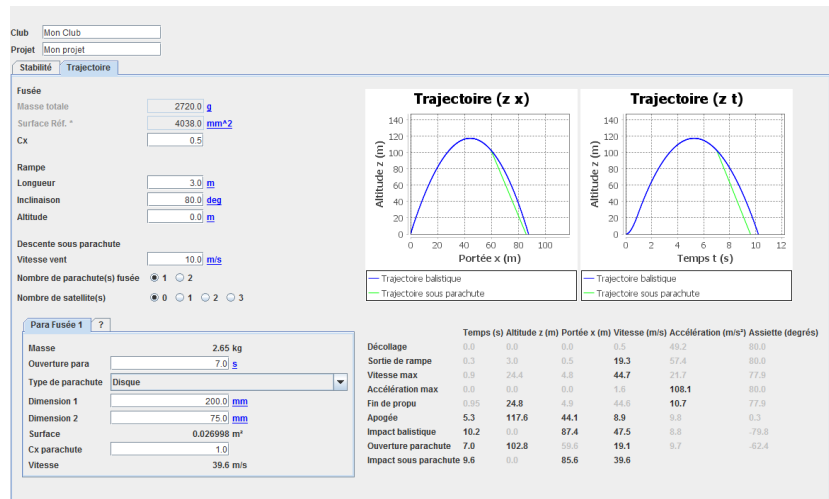


FIGURE 8 – Interface Pégase - calcul de la trajectoire.

L'algorithme classique pour le calcul de la trajectoire est inspiré de ?? et est de la forme suivante :

```

\\Calcul des forces et accélérations
M_i = M0 - dm.ti \\nouvelle masse avec débit supposée constant dm
R_i = (1/2)rho_i.C_Ai.S(V_i-1)^2 \\ Résistance de l'air
\\ P_i = poussée à t_i calculée par la méthode getloidePoussée()

Gamma_xi = (P_i - R_i)/M_i . cos(theta_i-1)
Gamma_zi = (P_i - R_i)/M_i .sin(theta_i-1) - g

\\Calcul des vitesses
V_xi = (V_xi-1) + (Gamma_xi).dt_i
V_zi = (V_zi-1) + (Gamma_zi).dt_i
V_i = ((V_xi)^2 + (V_zi)^2)^(1/2) \\norme vitesse à t_i

\\Calcul de la position et trajectoire
X_i = X_i-1 + (V_xi-1).dt_i +(1/2)(Gamma_xi).(dt_i)^2 \\selon x
Z_i = Z_i-1 + (V_zi-1).dt_i +(1/2)(Gamma_zi).(dt_i)^2 \\selon z

\\Màj de theta_i
a=tan(theta_i-1) = (V_zi)/(V_xi)
theta_i = arctan(a)

```

Lorsque la composante verticale de la vitesse V_{X_i} s'annule, la fusée est en phase de culmination.

Dans le cas où la fusée est sur une rampe de lancement, il faut prendre en considération la hauteur à laquelle la fusée commence sa trajectoire, et l'angle d'inclinaison plus une force supplémentaire qui est la réaction \vec{R} . En adaptant le PFD, l'accélération à cette étape se calcul comme tel :

$$\begin{aligned} \Gamma_{xi} &= (P_i - R_i)/M_i \cdot \cos(\theta_i) - g \cdot \cos(\theta_i) \sin(\theta_i) \\ \Gamma_{zi} &= (P_i - R_i)/M_i \cdot \sin(\theta_i) - g + g \cdot \cos(\theta_i)^2 \end{aligned}$$

avec $\theta_i = \theta_0$ l'angle formé par la rampe.

On ne tient plus compte de la rampe de lancement lorsque l'altitude $Z_i - Z_0$ devient supérieure à $L \cdot \sin(\theta_0)$ (L est la longueur de la rampe).

Quand à la phase du parachute, sous l'action du poids et de la résistance de l'air seulement, la vitesse augmente ou diminue jusqu'à ce que la traînée égale le poids. L'accélération du mouvement étant alors nulle, la vitesse reste constante (vitesse limite V_l).

$$\begin{aligned} P &= R \quad V = \text{Cste} = V_l \\ M \cdot g &= (1/2) \cdot \rho \cdot S_{\text{para}} \cdot C_{x_{\text{para}}} \cdot V_l^2 \\ V_l &= (2 \cdot M \cdot g / (\rho \cdot S_{\text{par}} \cdot C_{x_{\text{par}}}))^{1/2} \end{aligned}$$

5 Code

Le code de Pégase est écrit en JAVA. Le paragraphe suivant va résumer l'essentiel des éléments de code générant l'interface.

Le code est basé autour de 3 classes principales :

```
class Fusée
class Stabilité
class Trajectoire
```

La première classe va permettre de modéliser la fusée et ses caractéristiques sont : son type, sa masse, la forme de son ogive etc.

La stabilité va dépendre du type auquel appartient la fusée. un calcul est fait pour vérifier si les inputs respectent bien les conditions de stabilités caractéristiques du type de fusée sélectionné.


```

public class Traj {

    // ~~~~ Phénomènes physiques ~~~~
    private static double g0 = 9.80665; //intensité de la pesanteur (m/s²)//atmosphère
    private static double rhoMoyen = 1.225; //masse volumique de l'air (en kg/m3)
    private static double T = 298; //température de l'atmosphère, 25°C (en K)
    // ~~~~ Caractéristiques moteurs ~~~~
    private static double iInst[] = new double [1000]; //impulsion totale instantanée (en N.s)
    private static double iTotale = 0; //impulsion totale (en N.s)
    private static double isp = 0; //impulsion spécifique (en s)
    // ~~~~ Descente parachutée ~~~~
    private static double tParaMin = 4; //instant de déclenchement du premier parachute
    // ~~~~ Variables utilitaires ~~~~
    private static double pasDeCalcul = 0.01; //pas de calcul de la trajecto (en s)
    private static double dtLongFeu = 100; //durée à partir de laquelle on considère que la fusée ne décollera pas (10s)
    // ~~~~ tableaux de calcul ~~~~
    private static double mach[] = new double [100000]; //nombre de Mach
    private static double trajectoire[][] = new double [100000][20]; //tableaux pour stocker les valeurs du calcul pas à pas

    //~~~~~ Méthode principale ~~~~~//
}

```

FIGURE 9 – Extrait de code.

La trajectoire va être calculée à l'aide d'une classe. Les positions, vitesses et autres paramètres relatifs au vol de la fusée vont être stockés dans un tableau et les coordonnées de la position vont servir à tracer la trajectoire. Le schéma numérique utilisé est le même que celui décrit dans la partie précédente.

6 Fusée multi-étagée

L'interface Pégase ne réalise les calculs que pour les fusées mono étagées. Le cas des fusées multi-étagée est un peu différent dans la mesure où il fait entrer d'autres paramètres en jeu :

- La fusée se sépare de ses étages à des moments données, entraînant un changement de masse et de géométrie.
- Il faut par la suite considérer la stabilité et la trajectoire de chaque élément (étage) de la fusée.

Pour une fusée bi-étagée par exemple il faut considérer les configurations suivantes :

- Vol en configuration bi-étage,
- Vol du second étage seul
- Vol du premier étage seul

Les changements de masse et de géométrie vont avoir un impact direct sur les forces mise en jeu et donc sur le schéma numérique modélisant la trajectoire.

7 Conclusions

Durant les prochains jours, on va tenter de formaliser la modélisation des fusée bi-étagée et l'étude de leur stabilité et trajectoire et enfin, d'inclure cela à l'interface de calcul Pégase.