

# Projet d'Optimisation d'un Lanceur Spatial

NAYET AYMERIC

MISRACHI LAURA

*Master 2 Ingénierie Mathématique, MPE, Université Pierre et Marie Curie*



## LIST OF FIGURES

1	: Vue d'ensemble du solveur . . . . .	4
2	: Vue d'ensemble de l'algorithme SQP . . . . .	10
3	: Convergence cas test MHW4D . . . . .	15
4	: Convergence cas test Ariane 1. . . . .	18
5	: Convergence cas d'Ariane 6. . . . .	21
6	: Caractéristiques du lanceur Ariane 6 au cours du temps et trajectoire dans l'espace des phases. . . . .	22

## CONTENTS

<b>I</b>	<b>Introduction</b>	<b>4</b>
<b>II</b>	<b>Outils mathématiques</b>	<b>5</b>
I	Algorithme SQP avec restriction supplémentaire du domaine . . . . .	5
I.1	Méthode de projection . . . . .	5
I.2	Problème de Newton modifié . . . . .	6
I.3	Recherche linéaire . . . . .	8
I.4	Mise en oeuvre globale de l'algorithme . . . . .	9
II	Simulation de trajectoire . . . . .	11
II.1	Forces extérieures au lanceur . . . . .	11
II.2	Implémentation du simulateur . . . . .	12
<b>III</b>	<b>Validation du solveur</b>	<b>14</b>
I	Le cas test MHW4D . . . . .	14
II	Le cas test Ariane 1 . . . . .	15
II.1	Solution théorique . . . . .	15
II.2	Solution approchée par l'algorithme SQP . . . . .	17
<b>IV</b>	<b>Optimisation du lanceur Ariane 6</b>	<b>19</b>
I	Introduction . . . . .	19
II	Résultats . . . . .	19
II.1	Résultat théorique pour le problème d'étagement . . . . .	20
II.2	Résultat de l'optimiseur pour le problème couplé . . . . .	20
<b>A</b>	<b>Programmes MATLAB</b>	<b>23</b>

## I. INTRODUCTION

Ce projet porte sur l'optimisation du lanceur spatial *Ariane 6* à 3 étages, dont on souhaite déterminer les masses minimales d'ergols ( $me_1, me_2, me_3$ ) nécessaires au lancement de son satellite de masse  $mu = 1500$  kg sur une orbite à 250 km de la Terre. Dans une optique de diminution des coûts et de la consommation en carburant, l'optimisation du poids de ces véhicules spatiaux s'avère effectivement primordiale.

Pour ce faire, le problème sera posé sous la forme d'un premier *problème d'étagement* lié à la détermination des masses d'ergols minimisant la masse totale du lanceur  $M_{i,1}$ , sous contrainte d'atteindre la vitesse orbitale souhaitée  $V_c$ . Il sera couplé à un *problème de trajectoire* lié à la détermination des angles d'orientation des étages dans leurs différentes configurations afin de maximiser la vitesse fournie au satellite, sous contrainte d'atteindre l'altitude orbitale souhaitée  $R_c$ .

Le premier sous-problème fait porter une contrainte sur la vitesse propulsive tandis que le second en fait le coeur de son optimisation (fonction objectif). La résolution du problème de trajectoire permettra ainsi de connaître la vitesse maximale  $V_r$  atteignable par le lanceur et de vérifier si sa configuration est adaptée à la mission. Une heuristique sur la vitesse propulsive  $V_r$  sera ensuite mise en place afin de coupler les deux problèmes et de déterminer la configuration et la trajectoire optimale du lanceur (masses d'ergols ( $me_1, me_2, me_3$ ) et orientation du lanceur dans ses différentes configurations ( $\theta_0, \theta_1, \theta_2, \theta_3$ )).

Le graphe ci-dessous résume les différentes étapes du solveur que nous allons implémenter sous MATLAB.

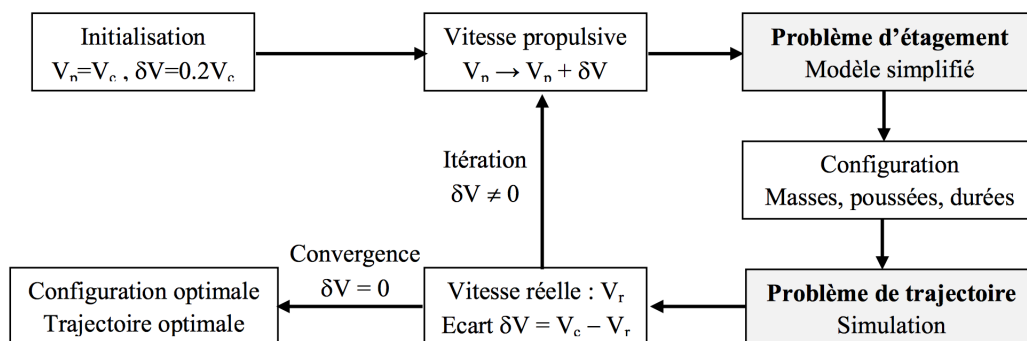


Figure 1: : Vue d'ensemble du solveur

## II. OUTILS MATHÉMATIQUES

### I. Algorithme SQP avec restriction supplémentaire du domaine

Le but de l'algorithme est de résoudre numériquement des problèmes d'optimisation du type :

$$\begin{cases} \min_{x \in \mathbb{R}^n} f(x) \\ \text{tel que : } c(x) = 0 \end{cases} \quad (1)$$

Avec  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ , et  $c : \mathbb{R}^n \rightarrow \mathbb{R}^m$

Pour la suite de notre projet, il est nécessaire de pouvoir restreindre le domaine des points admissibles en plus des contraintes d'égalités, avec des contraintes d'inégalité du type :

$$m_j < x_j < M_j, \forall j = \{1 \dots n\} \quad (2)$$

Nous verrons que prendre en compte ce type de contrainte revient à projeter des points dans un hypercube  $[m_j, M_j]^n$ .

Dans un premier temps, un manière de pouvoir traiter le problème (1) est de le reformuler en introduisant des multiplicateurs de Lagrange  $\lambda \in \mathbb{R}^m$  :

$$\begin{cases} \min_{x \in \mathbb{R}^n, \lambda \in \mathbb{R}^m} f(x) + \lambda^T c(x) = \min_{x \in \mathbb{R}^n, \lambda \in \mathbb{R}^m} L(x, \lambda) \\ \text{tel que : } c(x) = 0 \end{cases} \quad (3)$$

Le théorème de K.K.T. nous dit alors que toutes solutions  $(\bar{x}, \bar{\lambda})$  de (3) respectent :

$$\begin{cases} \nabla f(\bar{x}) + \bar{\lambda}^T \nabla c(\bar{x}) = 0 \\ c(\bar{x}) = 0 \end{cases} \quad (4)$$

(Le théorème assure l'unicité de la solution uniquement dans le cas où  $f$  est convexe ainsi que son domaine d'admissibilité).

On pourrait alors résoudre le système (4) par la méthode de Quasi-Newton, associée à une méthode de globalisation (recherche linéaire d'un pas optimal). Pour autant, l'algorithme SQP propose d'agir légèrement différemment.

Avant d'entrer dans le détail de l'algorithme, nous explicitons la méthode de projection que nous utiliserons dans l'algorithme pour respecter le domaine d'admissibilité  $[m_j, M_j]^n$ .

#### I.1 Méthode de projection

Comme indiqué précédemment, en plus de respecter la contrainte  $c$ , on souhaite pouvoir restreindre notre domaine par des contraintes du type (2). On définit alors l'opérateur suivant :

$$\begin{cases} \Pi : \mathbb{R}^n \longrightarrow [m_j, M_j]^n \\ \Pi(x)_j = \begin{cases} x_j & \text{si } x_j \in [m_j, M_j] \\ m_j & \text{si } x_j < m_j \\ M_j & \text{si } x_j > M_j \end{cases} \end{cases} \quad \text{où } j \text{ désigne la } j\text{-ème application partielle de } \Pi \quad (5)$$

On utilisera cet opérateur tout au long de l'algorithme pour projeter les pas intermédiaires obtenus.

## I.2 Problème de Newton modifié

Pour débiter notre algorithme, on écrit tout d'abord une étape de l'algorithme de Newton pour (4), en posant :

$$\begin{aligned} F : \mathbb{R}^n \times \mathbb{R}^m &\longrightarrow \mathbb{R}^2 \\ F_1(x, \lambda) &= \nabla f(x) + \lambda^T \nabla c(x) = L(x, \lambda) \\ F_2(x, \lambda) &= c(x) \end{aligned}$$

Une itération de l'algorithme est alors :

$$\begin{cases} \text{Soit } z_k = (x_k, \lambda_k) \in \mathbb{R}^n \times \mathbb{R}^m \\ \nabla F(z_k)(z_{k+1} - z_k) = -F(z_k) \end{cases} \quad (6)$$

Or, de manière générale, résoudre le système linéaire  $Ax = b$  peut se voir comme résoudre  $\min \frac{1}{2}x^T Ax - b^T x + c$ .

*Proof.* Pour résoudre le problème sans contrainte  $\min \frac{1}{2}x^T Ax - b^T x + c$ , il suffit (uniquement dans le cas où  $A$  est symétrique définie positive) d'annuler le gradient de la fonctionnelle :

$$\nabla \left( \frac{1}{2}x^T Ax - b^T x + c \right) = 0$$

Soit en simplifiant :

$$Ax - b = 0$$

De plus, s'agissant d'une fonctionnelle convexe, la solution est unique.  $\square$

Ainsi, à chaque étape  $k$ , on peut localement en  $z_k$  résoudre le problème

$$\begin{cases} \text{Soit } z_k = (x_k, \lambda_k) \in \mathbb{R}^n \times \mathbb{R}^m \\ \min_{z \in \mathbb{R}^n \times \mathbb{R}^m} \frac{1}{2}(z - z_k)^T \nabla F(z_k)(z - z_k) - F(z_k)^T(z - z_k) \end{cases} \quad (7)$$

Le minimum du problème précédent est alors le nouveau pas  $z_{k+1}$ . On a que :

$$\nabla F(x, \lambda) = \begin{bmatrix} \nabla^2 L(x, \lambda) & \nabla c(x) \\ \nabla c(x) & 0 \end{bmatrix} \quad (8)$$

Le problème (7) se réécrit donc, en posant  $d_x = x - x_k$  et  $d_\lambda = \lambda - \lambda_k$ :

$$\begin{cases} \text{Soit } z_k = (x_k, \lambda_k) \in \mathbb{R}^n \times \mathbb{R}^m \\ \min_{(d_x, d_\lambda) \in \mathbb{R}^n \times \mathbb{R}^m} \frac{1}{2} (d_x^T \nabla^2 L(x_k, \lambda_k) d_x) + d_x^T \nabla c(x_k) d_\lambda - d_x^T \nabla L(x_k, \lambda_k) - d_\lambda^T c(x_k) \end{cases} \quad (9)$$

Pour des raisons pratiques (coût de calcul), il est préférable de remplacer le terme  $d_x^T \nabla L(x_k, \lambda_k)$  par  $\nabla f(x_k)$ . L'erreur commise est donc logiquement égale à  $d_\lambda^T \nabla c(x_k)$ , soit  $(d_\lambda + \lambda_k)^T \nabla c(x_k)$  à chaque étape. Cela implique que si  $x_k$  est dans le domaine admissible alors  $c(x_k)$  est nulle et donc l'erreur également.

On peut interpréter le problème (9) sous une forme Lagrangienne à l'image de (3). En d'autres termes, on peut réécrire ce problème comme un problème de minimisation sous contrainte, en considérant  $d_\lambda$  comme une variable de Lagrange :

$$\begin{cases} \text{Soit } z_k = (x_k, \lambda_k) \in \mathbb{R}^n \times \mathbb{R}^m \\ \min_{d_x \in \mathbb{R}^n \times \mathbb{R}^m} \frac{1}{2} (d_x^T \nabla^2 L(x_k, \lambda_k) d_x) - d_x^T \nabla f(x_k) \\ \text{s.c. } d_x^T \nabla c(x_k) - c(x_k)^T = 0 \end{cases} \quad (10)$$

Ce problème admet une solution analytique unique si  $\nabla^2 L(x, \lambda)$  est symétrique définie positive car la fonction objectif est convexe ainsi que l'espace admissible (c'est un sous-espace affine si  $\nabla c(x_k)$  est de rang maximal, ce qui est le cas si il n'y a pas de contraintes redondantes).

$$\begin{cases} d_\lambda = -(\nabla c(x_k)^T \nabla^2 L(x_k, \lambda_k)^{-1} \nabla c(x_k))^{-1} (\nabla c(x_k)^T \nabla^2 L(x_k, \lambda_k)^{-1} \nabla f(x_k) - c(x_k)) \\ d_x = -\nabla^2 L(x_k, \lambda_k)^{-1} (\nabla c(x_k) d_\lambda + \nabla f(x_k)) \end{cases} \quad (11)$$

*Proof.* Pour obtenir ce résultat, il suffit d'écrire (10) sous forme Lagrangienne :

$$\min_{d_x \in \mathbb{R}^n \times \mathbb{R}^m} \frac{1}{2} (d_x^T \nabla^2 L(x_k, \lambda_k) d_x) - d_x^T \nabla f(x_k) + d_\lambda (d_x^T \nabla c(x_k) - c(x_k)^T)$$

Puis d'annuler le gradient de ce Lagrangien (d'où les hypothèses d'existence) : on obtient donc le système :

$$\begin{cases} \nabla^2 L(x_k, \lambda_k) d_x - \nabla f(x_k) + d_\lambda \nabla c(x_k) = 0 \\ d_x^T \nabla c(x_k) - c(x_k)^T = 0 \end{cases}$$

On agence la première équation pour avoir une expression pour  $d_x$  :

$$\begin{cases} d_x = \nabla^2 L(x_k, \lambda_k)^{-1} (\nabla f(x_k) - d_\lambda \nabla c(x_k)) \\ d_x^T \nabla c(x_k) - c(x_k)^T = 0 \end{cases}$$

Et on l'injecte dans la deuxième équation, ce qui donne :

$$\begin{cases} d_x = -\nabla^2 L(x_k, \lambda_k)^{-1} (\nabla c(x_k) d_\lambda + \nabla f(x_k)) \\ d_\lambda = -(\nabla c(x_k)^T \nabla^2 L(x_k, \lambda_k)^{-1} \nabla c(x_k))^{-1} (\nabla c(x_k)^T \nabla^2 L(x_k, \lambda_k)^{-1} \nabla f(x_k) - c(x_k)) \end{cases} \quad (12)$$

□

Le calcul du hessien  $\nabla^2 L(x_k, \lambda_k)$  est particulièrement coûteux en terme de temps de calcul. Pour remédier à cela, on utilise une approximation de ce hessien à chaque étape à l'aide de l'algorithme BFGS ou SR1 (que l'on ne rappelle pas dans ce document). Ces algorithmes ne donnent pas nécessairement une matrice symétrique définie positive (or on en a nécessairement besoin cf. preuve de la minimisation d'une fonctionnelle convexe faite plus haut) : on modifie donc ce Hessien à chaque étape. Soit  $Q \approx \nabla^2 L(x_k, \lambda_k)$ , alors dans notre cas, pour symétriser  $Q$ , nous effectuons :  $\hat{Q} = \frac{1}{2}(Q^T + Q)$ . Puis, on note  $\gamma_{min}$  la plus petite valeur propre de  $\hat{Q}$ . Si  $\gamma_{min} < 0$  (i.e. la matrice n'est pas définie positive) alors on effectue l'opération  $\tilde{Q} = \hat{Q} + (0.1 + \gamma_{min})\mathbb{I}$ .

Ajoutons également que  $\nabla f(x_k)$  et  $\nabla c(x_k)$  sont approchés par différences finies.

Il est maintenant impératif de projeter la direction  $d_x$  avec l'opérateur  $\Pi$ , pour respecter nos contraintes d'inégalités :

$$\begin{aligned} \hat{x}_{k+1} &= x_k + d_x \\ x_{k+1} &= \Pi(\hat{x}_{k+1}) \\ \text{nouveau } d_x \text{ projeté : } d_x &= x_{k+1} - x_k \end{aligned}$$

**Note :** Il semble logique de projeter  $d_x$  de cette manière. Pour autant, nous avons remarqué que cette méthode ne fonctionne pas toujours. Il est parfois nécessaire de projeter uniquement  $x_{k+1}$  et pas  $d_x$ . A la date d'écriture de ce document, nous n'avons pas d'explication à ce problème.

Ainsi, à chaque étape  $k$ , on obtient une direction de recherche  $d_x$  pour faire progresser l'algorithme vers un minimum : il s'agit de la phase QP de l'algorithme. Pour autant, il est nécessaire de vérifier à chaque étape que cette direction est pertinente en un certain sens, et dans ce cas optimiser le déplacement à effectuer.

### I.3 Recherche linéaire

Supposons connu la direction de Newton à l'étape  $k$  :  $d_x$ . Il est clair que rien n'indique que cette direction minimise  $f$  sous les contraintes  $c$  puisque le nouveau point ne fait que faire décroître le gradient du Lagrangien du problème initial. Nous introduisons alors une fonction mérite qui représente à quel point cette direction (et donc le nouveau point  $x_{k+1}$ ) est pertinent-e :

$$\begin{cases} h_\rho : \mathbb{R}^n \longrightarrow \mathbb{R} \\ h_\rho(x) = f(x) + \rho c(x) \end{cases} \quad (13)$$



Avec  $\rho > 0$ . Ainsi, plus  $x$  est loin du domaine admissible, plus  $h_\rho$  augmente.  $\rho$  mesure donc la sensibilité par rapport aux contraintes.

On teste donc à l'étape  $k$  si  $d_x$  fait bien diminuer  $h_\rho$ , c'est à dire si  $h_\rho(x_{k+1}) < h_\rho(x_k)$ . Si c'est le cas, alors on accepte la direction  $d_x$  et donc le nouveau point  $x_{k+1}$ . Sinon, on va affiner le déplacement.

Pour cela, on souhaite trouver un certain  $\alpha \in \mathbb{R}$  pour lequel  $h_\rho(x_k + \alpha d_x)$  est optimal. On doit donc d'abord vérifier si  $d_x$  est une direction de descente pour  $h_\rho$  i.e si  $\nabla h_\rho(x_k) \cdot d_x < 0$ .

Supposons que ce soit le cas : on effectue alors une recherche linéaire avec l'algorithme d'Armijo (dont on ne rappelle pas le fonctionnement dans ce document) pour déterminer  $\alpha$ . On précise qu'il faut absolument projeter les points intermédiaires d'Armijo à l'aide de  $\Pi$ , autrement, on risque de s'éloigner du domaine  $[m_j, M_j]^n$ . A la suite de cela, on obtient un nouveau point  $x_{k+1}$  qui fait bien décroître  $h_\rho$ , qu'il faut bien sûr projeter à l'aide de  $\Pi$  :

$$\widehat{x}_{k+1} = x_k + \alpha d_x \quad (14)$$

$$x_{k+1} = \Pi(\widehat{x}_{k+1}) \quad (15)$$

Dans le cas contraire, il faut trouver une autre direction  $d_x$ . On a dit précédemment qu'en pratique on utilise une approximation du hessien  $\nabla^2 L(x_k, \lambda_k)$ . On va réinitialiser cette approximation avec une matrice identité et recalculer le déplacement  $d_x$ , ce qui donne dans ce cas:

$$\begin{cases} d_x = \nabla f(x_k) - \nabla c(x_k) d_\lambda \\ d_\lambda = -(\nabla c(x_k)^T \nabla c(x_k))^{-1} (\nabla c(x_k)^T \nabla f(x_k) - \nabla c(x_k)) \end{cases}$$

Puis on recommence : on teste la décroissance de  $h_\rho$  et on calcule  $\alpha$  dans le cas où ce n'est pas le cas. Si une nouvelle fois la direction ne convient pas, on va augmenter  $\rho$  de la fonction  $h_\rho$  d'un facteur 10 pour mieux prendre en compte les contraintes  $c$ , puis recommencer à nouveau.

#### I.4 Mise en oeuvre globale de l'algorithme

Voici une vue d'ensemble de l'algorithme que nous avons décrit précédemment :

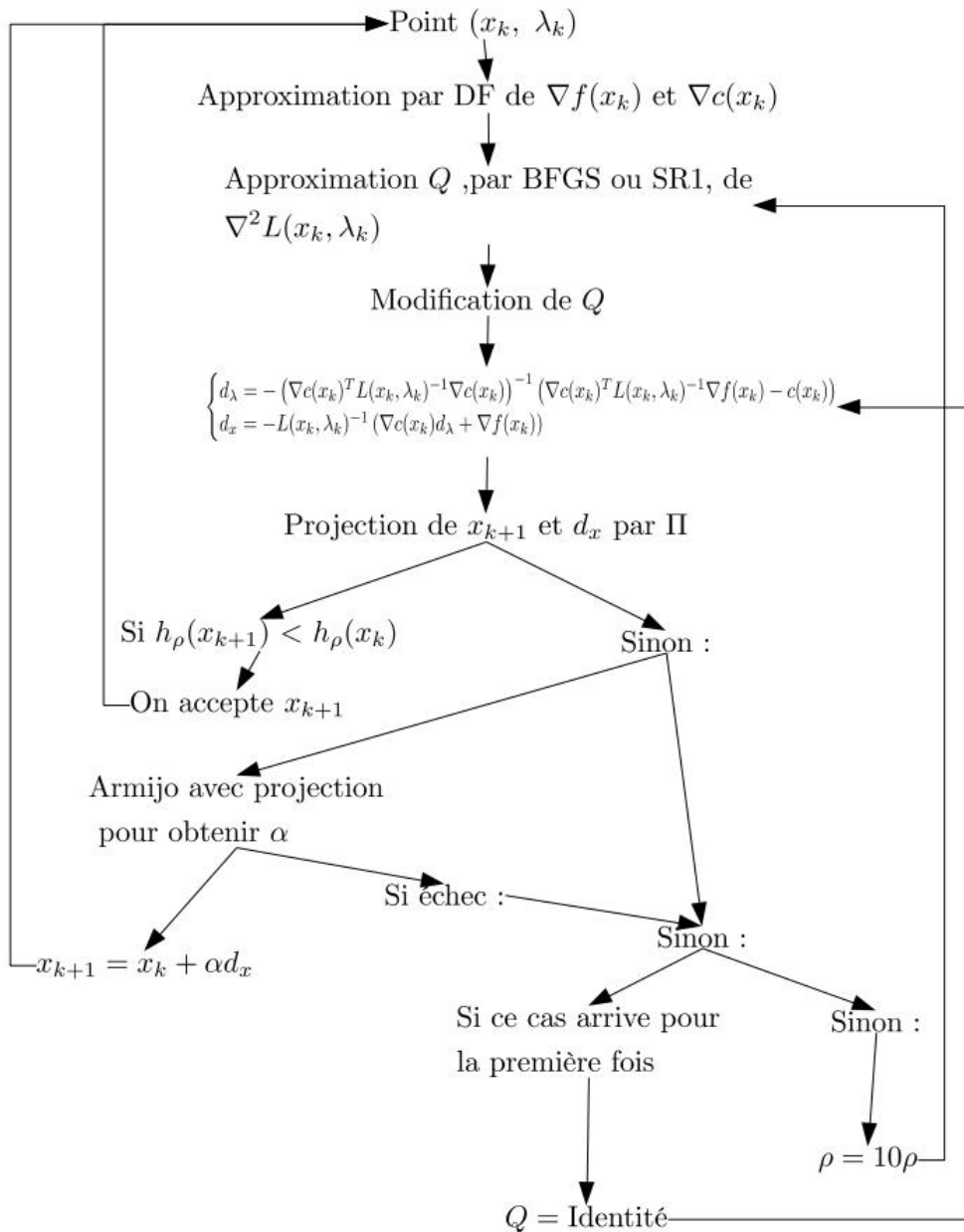


Figure 2: : Vue d'ensemble de l'algorithme SQP

## II. Simulation de trajectoire

Dans cette partie, nous allons construire le simulateur de trajectoire d'une fusée à 3 étages. Chaque étage de la fusée est défini par une masse sèche (soit sans ergols) que l'on notera  $m_{s,j}$  où  $j$  est le numéro de l'étage, et sa quantité d'ergols initial  $m_{e,j}$ . De plus, on dimensionne un étage connaissant la masse d'ergol que l'on souhaite lui injecter ; autrement dit il existe une constante constructeur qui relie la masse d'ergol de chaque étage à sa masse sèche :

$$m_{s,j} = k_j * m_{e,j} \text{ avec } k_j \text{ la constante constructeur.} \quad (16)$$

Ainsi, on note  $M_{i,j}$  la masse initiale de l'étage  $j$  et on a donc :

$$\forall j \in \{1, 2, 3\} : M_{i,j} = k_j m_{s,j} + m_{e,j} = (1 + k_j) m_{e,j} \quad (17)$$

Le but du lanceur étant de placer un satellite en orbite, on doit ajouter à cela le poids de celui-ci :  $m_u$ .

Chaque étage a une vitesse propulsive propre notée  $v_{e,j}$  et une accélération initiale propre  $a_{e,j}$ .

Nous nous plaçons dans un repère géocentrique, dans un plan orbital. Ce plan orbital correspond au plan défini par les directions:

1. Le centre de la Terre - Point du décollage
2. Le centre de la Terre - Le pôle nord

Ce repère forme un repère cartésien et on note ces axes normalisés et orthogonaux  $e_1$  et  $e_2$ .

Dans ce repère, nous rappelons la loi de la dynamique pour un point matériel :

$$\begin{cases} \frac{d}{dt}(m(t)v(t)) = F_{\text{ext}} \\ \frac{d}{dt}x(t) = v(t) \\ \frac{d}{dt}v(t) = a(t) \end{cases} \quad (18)$$

où  $m(t)$  est la masse du lanceur au temps  $t$ ,  $v(t) \in \mathbb{R}^2$  sa vitesse,  $x(t)$  sa position et  $F_{\text{ext}}$  les forces extérieures qui s'appliquent au point.

### II.1 Forces extérieures au lanceur

Nous commençons par citer en premier la gravité. Cette force volumique a pour expression :

$$W(t) = -G \frac{m_T m(t)}{(\|x(t)\| - R_T)^2} \frac{x(t)}{\|x(t)\|} \quad (19)$$

où  $G \approx 6.674.10^{11} \text{ m}^3 \text{ kg}^{-1} \text{ s}^{-2}$  est la constante gravitationnelle,  $m_T = 5,972.10^{24} \text{ kg}$  la masse de la Terre,  $R_T = 6371.10^3 \text{ m}$  le rayon de la Terre,  $x(t)$  la position du lanceur au temps  $t$ ,  $\theta$  l'angle formé par  $x(t)$  et  $e_1$ .

Une force de traînée est également présente : elle est due au frottement de l'air. Cette force est donc naturellement dirigée dans la direction opposée au mouvement, soit à l'opposé de  $v(t)$ . Elle dépend de trois choses : la densité de l'air, la vitesse du lanceur et la forme de la fusée. Concernant la densité de l'air, on la modélise selon la relation :

$$\rho(x) = \rho_0 e^{\frac{-||x(t)|| - R_T}{H}} \text{ où } H = 7000m \text{ et } \rho_0 \text{ est la densité de l'air au sol soit } 1.225kg.m^{-3}$$

Pour tenir compte de la forme de la fusée, on définit un nombre représentant la résistance de celle-ci au déplacement dans l'air : ce coefficient est  $c_x = 0.1$ . On définit alors la traînée par la relation :

$$D(t) = -c_x \rho(||x(t)|| - R_T) ||v(t)|| v(t) \quad (20)$$

Avant de définir la force de poussée, il faut définir ce qu'est l'axe de poussée. Cette direction correspond à l'orientation des tuyères par rapport à la vitesse  $v(t)$  du lanceur, que l'on note  $u$  (unitaire). Il s'agit maintenant de définir cette poussée dans notre repère initial. On définit donc un angle  $\theta$  de telle manière à d'abord pouvoir projeter la poussée sur  $v(t)/||v(t)||$  et son orthogonal.

$$u(t) = \cos(\theta) \frac{v(t)}{||v(t)||} + \sin(\theta) w(t) \text{ où } w(t) \text{ est le vecteur orthogonal à } v(t), \text{ unitaire} \quad (21)$$

On définit alors le vecteur  $w$  orthogonal à  $v$ . On résout donc le système :

$$\begin{cases} \langle w, \frac{v}{||v||} \rangle = \cos(\gamma) = 0 \\ ||w||^2 = 1 \end{cases} \quad (22)$$

Avec  $\gamma = \widehat{w, \frac{v}{||v||}}$

Il suffit alors de poser  $\gamma = \arcsin(x(t)/||x(t)||, v(t)/||v(t)||)$  et on obtient la direction  $u$  dans le repère cartésien  $(e_1, e_2)$  :

$$u = \begin{bmatrix} \cos(\gamma + \theta) + \sin(\gamma + \theta) \\ \sin(\gamma + \theta) - \cos(\gamma + \theta) \end{bmatrix} \quad (23)$$

On définit alors la poussée (qui est différente pour chaque étage  $j$ , mais constante) par :

$$T_j = a_j M_{i,j} \quad (24)$$

où  $a_j$  est l'accélération initiale de l'étage  $j$  et  $M_{i,j}$  la masse initiale de l'étage  $j$ .

Ainsi,  $F_{\text{ext}}$  du problème (18) est obtenu par la somme des 3 contributions précédentes  $W, T$  et  $D$ .

## II.2 Implémentation du simulateur

Voici le scénario de vol pour le lanceur :

1. Le lanceur décolle, avec une poussée orientée  $\theta_1$
2. Il consomme une quantité  $m_{e_1}$  d'ergols

3. Il éjecte l'étage 1 vide, de masse sèche  $k_1 m_{e,1}$
4. Il consomme une quantité  $m_{e,2}$  d'ergols selon un angle de poussée  $\theta_2$
5. Il éjecte l'étage 2 vide, de masse sèche  $k_2 m_{e,2}$
6. Il consomme une quantité  $m_{e,3}$  d'ergols
7. Il éjecte le dernier étage, de masse sèche  $k_3 m_{e,3}$
8. Il ne reste que le satellite, qui est maintenant en orbite si la mission est réussie

Etant donné la non continuité de  $m(t)$ ,  $a(t)$  et  $v(t)$ , il est informatiquement impossible d'intégrer le système (18) en une seule fois. On décompose donc l'intégration numérique en 3 plages temporelles :  $[0, t_{c,1}]$ ,  $[t_{c,1}, t_{c,1} + t_{c,2}]$ ,  $[t_{c,1} + t_{c,2}, t_{c,1} + t_{c,2} + t_{c,3}]$ .  $t_{c,j}$  désigne la durée de combustion des ergols de l'étage  $j$ . Avant de définir cette quantité, il est nécessaire de décrire la perte de masse sur chaque plage d'intégration :

$$\begin{cases} \text{Pour chaque étage } j \\ \frac{d}{dt}m(t) = T_j/v_{e,j} = q_j \end{cases} \quad (25)$$

Et alors la durée de combustion de l'étage  $j$ , obtenue par analyse dimensionnelle, est :

$$t_{c,j} = \frac{m_{e,j}}{q_j} \quad (26)$$

Pour conclure, à la fin de chaque intégration, il faut soustraire à la masse obtenue la quantité  $k_j m_{e,j}$  correspondant au largage de l'étage (d'où la non continuité de  $m(t)$ ). On utilisera l'intégrateur ODE45 de MATLAB 3 fois pour résoudre le problème (18), avec pour valeur initiale à l'étage  $j$  : la masse  $m_j$  calculé à l'étape précédente,  $a_j$  l'accélération initiale de l'étage  $j$  et  $v_j$  la vitesse propulsive de l'étage  $j$ .

### III. VALIDATION DU SOLVEUR

Afin de valider notre implémentation de l'algorithme décrit précédemment, nous allons tester le solveur SQP sur différents cas tests:

- La minimisation d'une fonction polynomiale assez complexe  $f$  de  $\mathbb{R}^5 \rightarrow \mathbb{R}$ , sous une contrainte  $c$  de  $\mathbb{R}^5 \rightarrow \mathbb{R}^3$ .
- Le problème d'étagement du lanceur *Ariane 1*.

#### I. Le cas test MHW4D

Le problème se formule comme indiqué par l'équation (1), avec pour fonction objectif et contrainte:

$$f(x) = (x_1 - 1)^2 + (x_1 - x_2)^2 + (x_2 - x_3)^3 + (x_3 - x_4)^4 + (x_4 - x_5)^4$$

$$c(x) = \begin{cases} x_1 + x_2^2 + x_3^2 - 3\sqrt{2} - 2 \\ x_2 - x_3^2 + x_4 - 2\sqrt{2} + 2 \\ x_1 \cdot x_5 - 2 \end{cases}$$

Il existe une solution analytique  $X^*$  à ce problème qui nous a été fournie et nous pouvons donc tester notre algorithme sur ce cas test. Afin de contrôler nos résultats, le tableau suivant résume les trois premières et trois dernières itérations fournies par notre algorithme SQP, en partant du point initial  $X_0 = [-1, 2, 1, -2, -2]$ . Il est important de noter que les résultats ont été tronqués au centième près et peuvent donc parfois donner l'impression qu'il n'y a pas d'évolution d'une itération à l'autre. Ce n'est généralement pas le cas.

$k$	$X$	$\lambda$	$f(X)$	$c(X)$	$\rho$	$  \nabla_x L(X)  $
0	$[-1, 2, 1, -2, -2]$	$[0, 0, 0]$	95	$[-2.24, -1.83, 0]$	1.0	[-]
1	$[-0.86, 1.64, 1.68, -0.22, -2.28]$	$[-12.19, 51.19, -8.88]$	40.66	$[-1.58, -2.23, -0.04]$	1.0	151.2
2	$[-1.72, 1.70, 1.93, 0.85, -0.01]$	$[-5.52, -12.63, -9.78]$	21.1	$[-1.32, -2.02, -1.98]$	1.0	196.8
3	$[0.11, 2.4, 1.19, -0.71, -1.17]$	$[-3.52, 5.55, -13.14]$	20.95	$[1.04, -0.56, -2.13]$	1.0	69.8
17	$[-1.24, 2.46, 1.19, -0.22, -1.62]$	$[-2.51, 0.12, -8.9]$	28.51	$[6e^{-5}, -5e^{-5}, 0]$	1.0	0.33
18	$[-1.24, 2.46, 1.19, -0.22, -1.62]$	$[-2.51, 0.12, -8.9]$	28.51	$[5e^{-5}, -4e^{-5}, -6e^{-5}]$	1.0	0.04
19	$[-1.24, 2.46, 1.19, -0.22, -1.62]$	$[-2.51, 0.12, -8.9]$	28.51	$[1e^{-5}, -1e^{-5}, 3e^{-5}]$	1.0	0.04

Table 1: Tableau des itérations pour le cas test MHW4D.

Certaines informations supplémentaires (nombres d'appel aux différentes fonctions  $f$  et  $c$ , norme de l'erreur...) concernant la convergence de l'algorithme sont présentées dans le tableau ci-dessous:

$n_{iter}$	$n_{appel_f}$	$n_{appel_c}$	$  X_{SQP} - X^*  $	$n_{armijo}$
20	189	190	0.001542	50

Table 2: Information sur la convergence du cas test MHW4D.

Les deux tableaux précédents permettent de démontrer la bonne convergence de notre algorithme vers la solution attendue  $X^*$ . Pour s'en convaincre définitivement, on trace la norme du gradient par rapport à  $x$  du Lagrangien en fonction du nombre d'itérations. C'est effectivement cette quantité que l'on cherche à annuler avec l'algorithme SQP afin de respecter une des conditions KKT. L'autre condition d'annulation des contraintes (assurée par l'algorithme également) se lit bien dans le tableau 1.

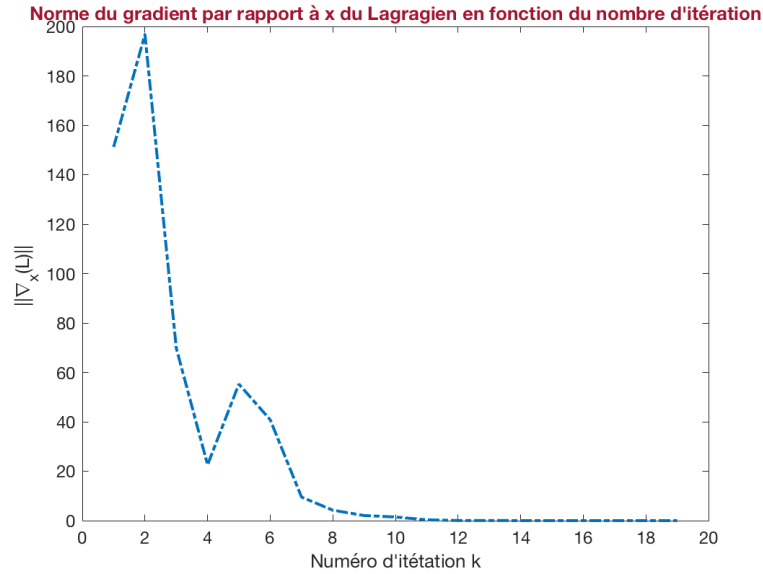


Figure 3: : Convergence cas test MHW4D

## II. Le cas test Ariane 1

Dans le cas d'Ariane 1 et d'Ariane 6, il est effectivement possible de déterminer une solution théorique du problème d'étagement.

### II.1 Solution théorique

Dans cette sous-section, nous allons brièvement caractériser cette solution théorique. Le problème d'étagement consiste à minimiser la masse totale de la fusée, sous contrainte d'atteindre la vitesse souhaitée  $V_c$ . Dans un premier temps, on remarque que l'on souhaite minimiser la fonction suivante:

$$-J = -\frac{M_{i,4}}{M_{i,1}} = -\frac{M_{i,4}}{M_{i,3}} \frac{M_{i,3}}{M_{i,2}} \frac{M_{i,2}}{M_{i,1}} \quad (27)$$

En introduisant les nouvelles inconnues  $\forall j \in \llbracket 1, 3 \rrbracket$ ,  $x_j = \frac{M_{i,j}}{M_{f,j}}$ , on obtient :

$$\frac{M_{i,j+1}}{M_{i,j}} = \frac{k_j + 1}{x_j} - k_j \quad (28)$$

Notre problème d'optimisation sous contrainte peut donc se réécrire en fonction de ces nouvelles variables:

$$\begin{cases} \min_{x \in \mathbb{R}^3} f(x_1, x_2, x_3) = - \left( \frac{1+k_1}{x_1} - k_1 \right) \left( \frac{1+k_2}{x_2} - k_2 \right) \left( \frac{1+k_3}{x_3} - k_3 \right) \\ \text{s.c } c(x_1, x_2, x_3) = v_{e_1} \ln(x_1) + v_{e_2} \ln(x_2) + v_{e_3} \ln(x_3) - V_c \end{cases} \quad (29)$$

On introduit ensuite une nouvelle variable définie par:

$$\forall j \in \llbracket 1, 3 \rrbracket, \quad y_j = \frac{k_j + 1}{x_j} - k_j \quad (30)$$

et en écrivant les conditions KKT (4), moyennant quelques calculs que l'on ne présente que partiellement par soucis de clarté, on obtient le système d'équation suivant:

$$\begin{cases} \forall j \in \llbracket 1, 3 \rrbracket, \quad v_{e_j}(1 - \Omega_j x_j) = cte \\ \forall j \in \llbracket 1, 3 \rrbracket, \quad \Omega_j = \frac{k_j}{1+k_j} \\ v_{e_1} \ln(x_1) + v_{e_2} \ln(x_2) + v_{e_3} \ln(x_3) - V_p = 0 \end{cases} \quad (31)$$

La première équation du système (31) a été obtenue en remarquant tout d'abord que l'on a:

$$\frac{\partial f}{\partial x_j} = - \frac{f^*}{\Omega_j x_j \left( 1 - x_j + \frac{1}{k_j} \right)} \quad (32)$$

où  $f^*$  est tel que  $f^* = -y_1 y_2 y_3$ . Ensuite en écrivant la condition suivante

$$\nabla_x L(x) = 0 \Leftrightarrow \frac{\partial f}{\partial x_j} + \lambda \frac{v_{e_j}}{x_j} = 0 \quad (33)$$

on obtient l'équation souhaitée avec  $cte = \frac{f^*}{\lambda}$ .

A partir de (31), on remarque que l'on peut exprimer  $x_1$  et  $x_2$  en fonction de  $x_3$  et réinjecter leurs expressions dans l'équation de la contrainte. On obtient ainsi une équation à une seule inconnue:

$$v_{e_1} \ln \left( \frac{1}{\Omega_1} \left( 1 - \frac{v_{e_3}}{v_{e_1}} (1 - \Omega_3 x_3) \right) \right) + v_{e_2} \ln \left( \frac{1}{\Omega_2} \left( 1 - \frac{v_{e_3}}{v_{e_2}} (1 - \Omega_3 x_3) \right) \right) + v_{e_3} \ln(x_3) - V_p = 0 \quad (34)$$

L'étape suivante consiste à implémenter une méthode de Newton pour résoudre l'équation ci-dessus:  $g(x) = 0$ . Nous pourrions calculer la dérivée de  $g$ , mais il y a de grandes chances pour que celle-ci s'annule en certains points. Nous allons donc privilégier une méthode de Newton approchée, la **méthode de la sécante** qui approche la dérivée de  $g$  par une différence finie d'ordre 1. Au préalable, il est important d'estimer grossièrement un intervalle sur  $x$  pour lequel l'équation ci-dessus est bien définie (les termes internes au logarithme doivent être strictement positifs). Ensuite, lors de la descente de gradient, il sera primordial de projeter les points en dehors du domaine de définition de  $g$  sur cet intervalle (sur la borne *min* par exemple) afin d'obtenir des résultats cohérents.

Une fois la solution  $x_3$  déterminée, on peut remonter à  $x_1$  et  $x_2$  et finalement à  $me_1, me_2, me_3$  en utilisant les relations présentées plus haut. On présente les résultats théoriques du problème d'étagement pour Ariane 1 et de la méthode de la sécante :



$x_3$	$g(x_3)$	$m_{e_{theo}}$
3.3276	$9 \cdot 10^{-6}$	$[14.5 \cdot 10^4, 31.2 \cdot 10^3, 7.935 \cdot 10^3]$

Table 3: Résultat théorique Ariane 1

La convergence de la méthode de la sécante semble assurée au vu de la valeur que prend la fonction  $g$  (cf (34)) en  $x_3$ , qui est censé l'annuler.

## II.2 Solution approchée par l'algorithme SQP

Pour le problème d'étagement d'Ariane 1, on lance l'algorithme SQP avec pour point de départ  $X_0 = [10^5, 5 \cdot 10^4, 10^4]$ . Par ailleurs, le domaine d'admissibilité des trois masses d'ergols a été décidé comme suit:

$$(m_{e_1}, m_{e_2}, m_{e_3}) \in ([10^4; 10^5], [10^3; 10^5], [10^2; 10^4])$$

Le tableau des itérations est donné ci-après. Il est important de noter que les valeurs 0 dans les dernières lignes du tableau peuvent être interprétées comme un zéro numérique (pour une précision jusqu'à  $10^{-7}$  ici).

$k$	$X$	$\lambda$	$f(X)$	$c(X)$	$\rho$	$\ \nabla_x L(X)\ $
1	$[12.3 \cdot 10^4, 61.0 \cdot 10^4, 12.7 \cdot 10^4]$	$-18.1 \cdot 10^5$	$22.4 \cdot 10^4$	-58.6	1.0	2.01
2	$[12.7 \cdot 10^4, 62.9 \cdot 10^4, 12.2 \cdot 10^4]$	$-19.9 \cdot 10^3$	$23.0 \cdot 10^4$	-2.3	10.0	$3.3 \cdot 10^4$
3	$[12.7 \cdot 10^4, 63.0 \cdot 10^4, 12.2 \cdot 10^4]$	$-18.7 \cdot 10^1$	$23.0 \cdot 10^4$	$-4 \cdot 10^{-3}$	100.0	$2.9 \cdot 10^2$
38	$[14.5 \cdot 10^4, 31.2 \cdot 10^3, 7.9 \cdot 10^3]$	$-11.7 \cdot 10^1$	$20.8 \cdot 10^4$	0	1000.0	$10^{-6}$
39	$[14.5 \cdot 10^4, 31.2 \cdot 10^3, 7.9 \cdot 10^3]$	$-11.7 \cdot 10^1$	$20.8 \cdot 10^4$	0	1000.0	0
40	$[14.5 \cdot 10^4, 31.2 \cdot 10^3, 7.9 \cdot 10^3]$	$-11.7 \cdot 10^1$	$20.8 \cdot 10^4$	0	1000.0	0

Table 4: Tableau des itérations pour le cas test Ariane 1.

$n_{iter}$	$n_{appel_f}$	$n_{appel_c}$	$\ X_{SQP} - X^*\ $	$\ X_{SQP} - X_{theo}^*\ $	$n_{armijo}$
44	251	252	57.01	0.0230	32

Table 5: Information sur la convergence du cas test Ariane 1.

Dans le tableau précédent,  $X^*$  fait référence à la solution qui nous a été donnée en cours, tandis que  $X_{theo}^*$  fait référence à la solution théorique du problème d'étagement obtenue en section II.1. Les résultats obtenus semblent tout à fait satisfaisant et en guise de dernière vérification on trace à nouveau la norme du gradient du lagrangien en fonction des itérations:

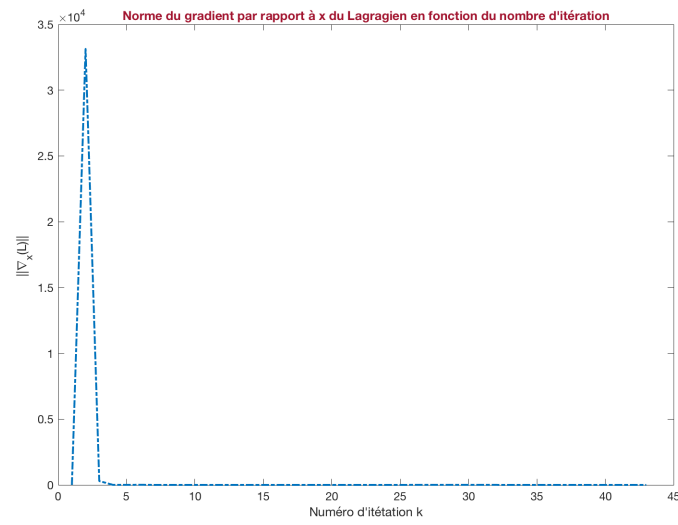


Figure 4: : Convergence cas test Ariane 1.

## IV. OPTIMISATION DU LANCEUR ARIANE 6

### I. Introduction

Dans cette partie, nous allons optimiser les masses d'ergols  $me_1, me_2, me_3$  et les angles de poussée  $\theta_0, \theta_1, \theta_2, \theta_3$  du lanceur Ariane 6 pour mettre en orbite un satellite de masse  $m_u = 1500\text{kg}$ , sur une orbite circulaire à  $250\text{km}$  du sol et à une vitesse  $V_c = \sqrt{\frac{3.986 \cdot 10^{14}}{250 \cdot 10^3}} \approx 7754 \text{ m.s}^{-1}$ .

Pour cela, nous allons utiliser l'optimiseur SQP que nous avons développé ainsi que le simulateur de vol. Le branchement est résumé par le graphique de l'introduction du document (Figure 1). Ainsi on se donne une certaine vitesse propulsive  $V_r$  de départ. On utilise cette vitesse pour résoudre notre problème d'étagement, c'est à dire pour trouver les masses d'ergols nécessaires pour atteindre cette vitesse propulsive. Puis, on optimise la trajectoire du lanceur en ajustant les angles  $\theta_j$ , en minimisant la fonction suivante (avec SQP) :

$$\begin{cases} \min_{\theta_0, \theta_1, \theta_2, \theta_3} ||v(T_{max})|| \\ \text{s.c. } \langle x(T_{max}), v(T_{max}) \rangle = 0 \\ x(T_{max}) = R_c \end{cases} \quad (35)$$

Avec :

- $T_{max}$  : le temps final de la simulation de la trajectoire.
- $x(t), v(t)$  : la solution numérique du problème (18) avec les masses d'ergols calculées par l'étagement précédent.
- $R_c$  : la hauteur à atteindre soit dans notre cas  $R_T + 250 \cdot 10^3 \text{ m}$ .

Ainsi, en sortie de cette optimisation, si c'est possible par rapport aux masses d'ergols  $me_1, me_2, me_3$ , on a des angles de poussée qui permettent de placer en orbite le satellite à la vitesse voulue, cette vitesse est orthogonale au déplacement (i.e. tangente à l'orbite) avec la contrainte 1, et l'orbite est à la bonne distance par la contrainte 2.

Cette optimisation fonctionne uniquement si les masses d'ergols sont correctes. Autrement dit, il faut boucler ensuite l'algorithme en posant  $V_r = V_r + dv$ , où  $dv$  est la distance entre la vitesse  $V_c$  à atteindre et  $v(T_{max})$ , et recommencer tant que  $dv \neq 0$ .

On fixe donc un critère d'arrêt qui stoppe l'algorithme lorsque la précision voulue est atteinte. Dans notre cas, ce critère est  $dv < 1$  (on souhaite une précision à  $1\text{m.s}^{-1}$  près pour la vitesse cible du satellite).

### II. Résultats

Compte tenu de tout ce qui a été présenté plus haut, le couplage des deux problèmes d'étagement et de trajectoire nous a permis d'optimiser le lanceur Ariane 6. Ici, nous présentons les résultats principaux de cette optimisation.

## II.1 Résultat théorique pour le problème d'étagement

Avant de nous lancer dans la détermination des masses d'ergols et des orientations  $(\theta_i, \forall i \in \llbracket 0, 3 \rrbracket)$  optimales de façon pratique, il est intéressant d'examiner les résultats théoriques que nous détenons sur le problème d'étagement seul. Cela nous permet d'obtenir un ordre de grandeur de la solution du problème couplé, et d'affiner ainsi notre initialisation et nos domaines d'admissibilité pour les masses d'ergols. Les résultats théoriques du problème d'étagement d'Ariane 6 sont présentés ci-dessous:

$x_3$	$g(x_3)$	$m_{e_{theo}}$
2.9368	$3 \cdot 10^{-3}$	[8402.6, 7734.2, 4742.4]

Table 6: Résultat théorique Ariane 6.

Lors de la résolution du problème couplé avec notre solveur, il est clair que nous ne retrouverons pas ces valeurs exactement, puisqu'on résout simplement le problème d'étagement ici, ie on trouve  $(me_{1_{theo}}, me_{2_{theo}}, me_{3_{theo}})$  pour atteindre une certaine vitesse et non pas une certaine altitude. Cependant, nous utiliserons ces valeurs pour initialiser nos masses d'ergols en conséquence.

## II.2 Résultat de l'optimiseur pour le problème couplé

Suite aux commentaires de la section précédente, on choisit d'initialiser les masses d'ergols à  $X_0 = [10000, 5000, 3000]$  et les orientations des différentes configurations à  $\theta_0 = [5, 5, 0, 0]$ . On définit nos domaines d'admissibilité pour les masses d'ergols et pour les orientations comme suit:

$$(me_1, me_2, me_3) \in ([5000; 100000], [3000; 50000], [100; 10000])$$

$$(\theta_0, \theta_1, \theta_2, \theta_3) \in ([-45; 45], [-45; 45], [-90; 90], [-180; 180])$$

De cette manière, on reste autour des ordres de grandeurs du résultat théorique pour les masses d'ergols, tout en introduisant une certaine hiérarchie naturelle de valeurs entre les étages. Le choix du domaine d'admissibilité pour les  $\theta$  a été motivé par le souhait d'atteindre une trajectoire à courbure lisse et peu abrupte en début de trajectoire. Davantage de degrés de libertés ont été alloués aux  $\theta$  sur la fin de trajectoire.

On présente désormais le tableau des itérations qui donne la dernière itération locale de l'algorithme SQP pour les  $me$  et les  $\theta$ , et ce pour les trois premières et trois dernières itérations globales.

$k_{global}$	$q$	$n_{iter}$	$X$	$\lambda$	$f(X)$	$c(X)$	$\rho$	$  \nabla_x L(X)  $
1	$m_e$	28	$[26.3 \cdot 10^3, 12.6 \cdot 10^3, 5.5 \cdot 10^3]$	$[-24.3]$	$51.5 \cdot 10^3$	0	100	$10^{-6}$
	$\theta$	39	$[0.76, 0.79, 1.57, 2.82]$	$[-22.3, 5.6]$	-11.4	$[4 \cdot 10^{-3}, -2 \cdot 10^{-2}]$	$10^{10}$	$3.2 \cdot 10^{-1}$
2	$m_e$	28	$[20.6 \cdot 10^3, 11.2 \cdot 10^3, 5.3 \cdot 10^3]$	$[-20.2]$	$43.4 \cdot 10^3$	$10^{-6}$	100	$1.5 \cdot 10^{-4}$
	$\theta$	42	$[0.70, 0.79, 1.50, 2.75]$	$[-24.6, 5.6]$	-11.0	$[7 \cdot 10^{-4}, -9 \cdot 10^{-3}]$	$10^{10}$	$2.5 \cdot 10^{-1}$
3	$m_e$	33	$[16.4 \cdot 10^3, 10.1 \cdot 10^3, 5.1 \cdot 10^3]$	$[-17.2]$	$37.3 \cdot 10^3$	$10^{-6}$	100	$5 \cdot 10^{-5}$
	$\theta$	34	$[0.66, 0.79, 1.37, 2.68]$	$[-28.3, 5.92]$	-10.5	$[5 \cdot 10^{-4}, -3 \cdot 10^{-3}]$	$10^{10}$	$1.2 \cdot 10^{-1}$
18	$m_e$	7	$[5.6 \cdot 10^3, 6.8 \cdot 10^3, 4.6 \cdot 10^3]$	$[-9.2]$	$20.9 \cdot 10^3$	0	1	$4 \cdot 10^{-4}$
	$\theta$	6	$[0.33, 0.01, 0.29, 2.36]$	$[-113.3, 16.9]$	-7.8	$[0, 0]$	100	$1.5 \cdot 10^{-4}$
19	$m_e$	5	$[5.6 \cdot 10^3, 6.8 \cdot 10^3, 4.6 \cdot 10^3]$	$[-9.2]$	$20.9 \cdot 10^3$	$-1.5 \cdot 10^{-4}$	1	$8.9 \cdot 10^{-4}$
	$\theta$	6	$[0.33, 0.01, 0.29, 2.36]$	$[-114.0, 17.1]$	-7.8	$[0, 0]$	100	$4 \cdot 10^{-5}$
20	$m_e$	5	$[5.6 \cdot 10^3, 6.8 \cdot 10^3, 4.6 \cdot 10^3]$	$[-9.2]$	$20.9 \cdot 10^3$	$-1.5 \cdot 10^{-4}$	1	$5.5 \cdot 10^{-4}$
	$\theta$	6	$[0.33, 0.01, 0.29, 2.36]$	$[-114.9, 17.1]$	-7.8	$[0, 0]$	100	$5 \cdot 10^{-5}$

Table 7: Tableau des itérations pour Ariane 6

La convergence semble bien assurée, au vu des valeurs de la norme du gradient par rapport à  $x$  du Lagrangien pour les dernières itérations. On trace ensuite l'évolution de l'erreur de vitesse et de hauteur en fonction du nombre d'itérations. Les résultats sont présentés ci-dessous. Les deux premières valeurs de  $dv$  peuvent sembler un peu étrange sur le graphe de gauche et laisser entendre que la première itération n'est pas concluante, mais cela résulte simplement de notre initialisation de  $dv$  à  $dv = 0.2 \cdot V_c$ , où  $V_c$  est la vitesse de propulsion à atteindre pour notre problème.

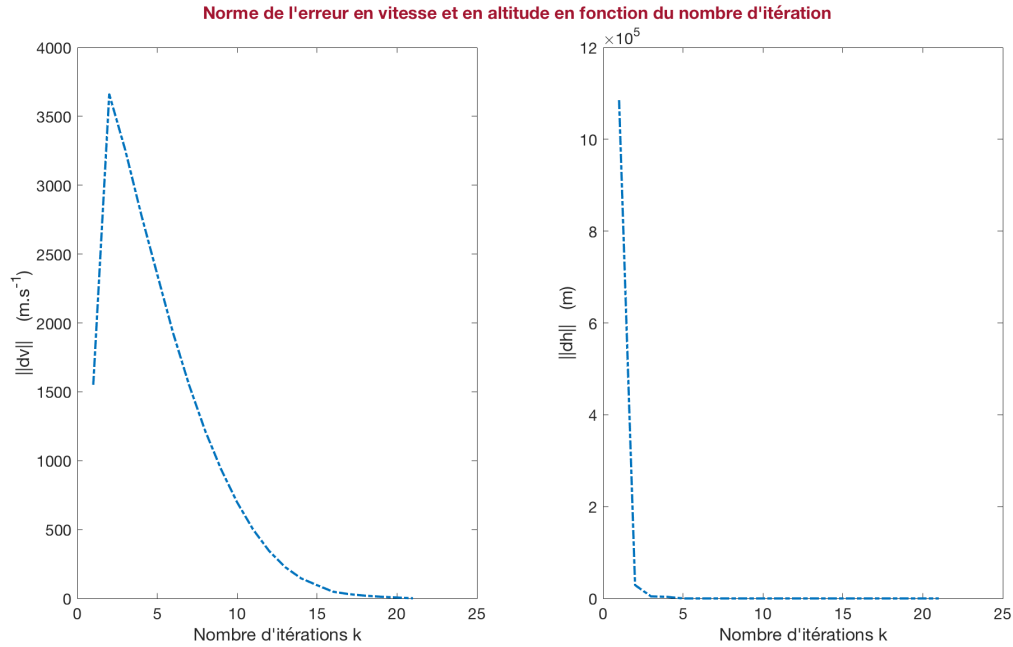


Figure 5: : Convergence cas d'Ariane 6.

En investiguant davantage nos itérations globales du problème *Ariane 6*, il est intéressant de remarquer qu'à l'itération globale 7, on retrouve un résultat pour le problème d'étagement très proche du résultat théorique, à savoir respectivement  $me_{k=7} = [8384.5, 7730.5, 4745.0]$ , contre  $me_{theo} = [8402.6, 7734.2, 4742.4]$ . Cette observation est réconfortante. Cependant, à l'itération globale numéro 7, l'erreur sur la vitesse  $dv$  est toujours importante, ce qui est logique, puisqu'on couple à présent le problème d'étagement avec celui de trajectoire.

Par ailleurs, rappelons que la solution du problème d'étagement et de trajectoire est  $m_e = [5607.7, 6766.0, 4575.2]$ , donc notre lanceur aura une masse totale d'ergol d'environ 16.9 tonnes, tandis que le problème d'étagement théorique seul prévoit une masse totale d'ergol d'environ 20.9 tonnes. Ceci est réconfortant puisqu'il est naturel que le problème de trajectoire (pour lequel on tente de déterminer les configurations angulaires des étages afin de maximiser la vitesse du lanceur) ait un effet positif sur la consommation totale en carburant.

Les quatre graphiques ci-dessous montrent l'évolution des quantités caractérisant le lanceur au cours du temps, à savoir sa masse, sa vitesse et sa hauteur. On peut également observer sa trajectoire en haut à gauche; celle-ci semble appréciable.

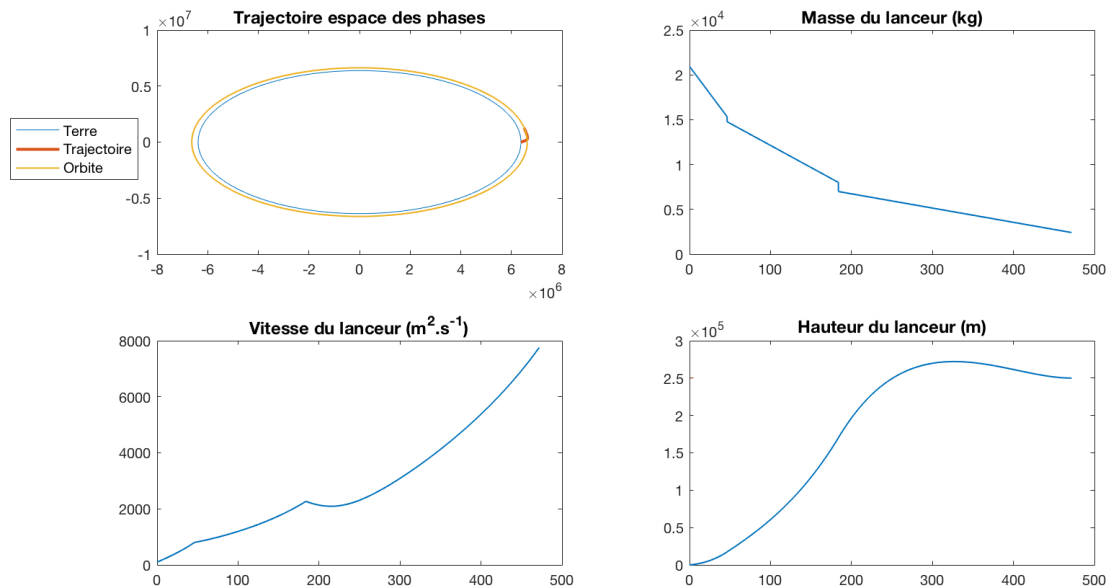


Figure 6: : Caractéristiques du lanceur Ariane 6 au cours du temps et trajectoire dans l'espace des phases.

## A. PROGRAMMES MATLAB

Le dossier **./Programmes** fourni avec ce document contient 5 programmes principaux :

- **M\_etagement\_theorique.m** : Ce programme résout le problème d'étagement théorique de la partie III - II.1
- **M\_main.m** : Ce programme contient l'optimiseur complet pour Ariane 6 (Section IV)
- **M\_test1.m** : Ce programme contient le test MHW4D pour SQP
- **M\_test2.m** : Ce programme contient la solution approchée du problème d'étagement d'Ariane 1 (III-II.2)
- **M\_TestSimulateur.m** : Ce programme contient un test du simulateur de vol