

Auteurs :

Geoffrey Pouliquen

Apolline El Baz

03/01/2018

Projet Optimisation

Optimisation de la trajectoire d’un lanceur spatial

**Table des matières**

**Introduction3**

**Optimiseur SQP4**

Description de l’optimiseur4

Validation de l’optimiseur5

Cas test : MH4WD5

Cas test : Ariane 16

Résolution du problème d’étagement7

Résolution analytique7

Résolution numérique9

**Simulateur de trajectoire10**

Description du simulateur10

Modélisation d’une première trajectoire11

Validation du simulateur11

Optimisation des angles vitesse-poussée11

**Réglage de la vitesse de propulsion12**

**Conclusion14**

**Introduction**

Le but de ce projet est de trouver le lanceur le plus léger permettant d’envoyer un satellite de masse sur une orbite se trouvant à de distance de la Terre. Notre modèle de lanceur (cf. schéma ci-après) est divisé en quatre étages, où les trois premiers étages, de masse initiale contiennent une certaine quantité de combustible, des masses d’ergol . Le dernier « étage », fictif, correspond au satellite de masse .

Les masses de propergol sont éjectées à des vitesses , ce qui permet de propulser le lanceur. On suppose que les sont brulées successivement, et . L’étage est détaché après la combustion de la correspondante. A chaque étape , le lanceur est donc allégé.

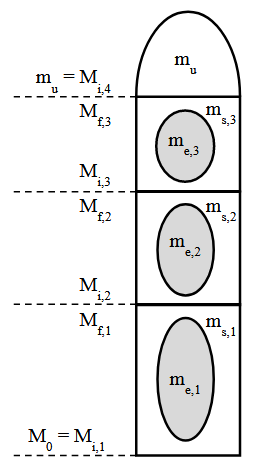
Notre problème se subdivise donc en deux : un problème d’étagement et un problème de trajectoire.

Le premier consiste à trouver la meilleure configuration pour le lanceur, c’est-à-dire à chercher les masses d’ergol telles que pour un satellite de masse le lanceur soit le plus léger possible.

Le second problème consiste à optimiser, à chaque étape, les angles vitesse-poussée qui détermine la trajectoire du lanceur.

Enfin, Il faut simuler des trajectoires pour différentes vitesses de propulsion , jusqu’à ce que la vitesse de propulsion initiale du lanceur lui permette d’atteindre l’orbite visée à la vitesse cible , vitesse de l’orbite en question.

Pour répondre à notre problème, nous avons eu besoin de programmer un optimiseur basé sur l’algorithme *SQP* (Sequential Quadratic Programming) qui permet de traiter des problèmes d’optimisation non linéaires. Les résultats de *SQP* ont été validés à partir d’une résolution analytique du problème d’étagement. Nous avons ensuite fait appel à un intégrateur, *ODE45,* de *Matlab*, afin de résoudre les équations du mouvement du lanceur, et ainsi simuler sa trajectoire.

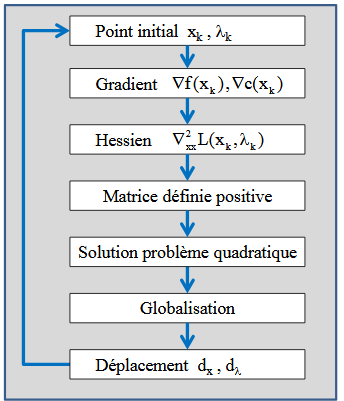


**Optimiseur *SQP***

**Description de l’optimiseur**

L’algorithme *SQP* est basé sur l’algorithme de Newton appliqué au lagrangien du problème d’optimisation, c’est-à-dire que l’on cherche tel que .

Nous avons alors besoin d’une approximation de et de son hessien Si n’est pas définie positive, on la modifie de telle sorte à ce qu’elle le soit. On peut alors calculer une solution de manière explicite au problème quadratique équivalent. La convergence est rapide, mais elle n’est pas nécessairement assurée, on force alors la convergence si les itérés sont mauvais par une technique de globalisation : la recherche linéaire (RL).



*SQP.m* est appelé en un point sur un problème qui correspond à une fonction à minimiser sous certaines contraintes . La fonction *Gradient.m* calcule par une méthode de différences finies - - une approximation de et renvoie également les valeurs de . On calcule ensuite

On initialise H à l’identité avant de faire appel au programme *Hessien.m* qui calcule par la formule *BFGS* (*v*=0) ou par *SR1* (*v*=1), à partir de la 2e itération. Si n’est pas définie positive, on le modifie en lui ajoutant la valeur absolue de sa plus grande valeur propre.

On peut alors inverser la matrice et utiliser une formule directe qui nous donne un déplacement et un multiplicateur . Le couple est accepté si la fonction (*Merite.m)* décroît. En effet, doit être une direction de descente pour . Elle l’est si

Si ce n’est pas le cas, on cherche tel vérifie la condition d’Armijo i.e. , avec On obtient alors .

Il faut de plus imposer que

Les tests de *SQP* se trouvent dans le script *Test\_SQP.m*, on peut les lancer en choisissant la variable *test* = 1 pour *MH4WD* et *test* = 2 pour *Ariane 1*.

**Validation de l’optimiseur**

Cas Test MH4WD

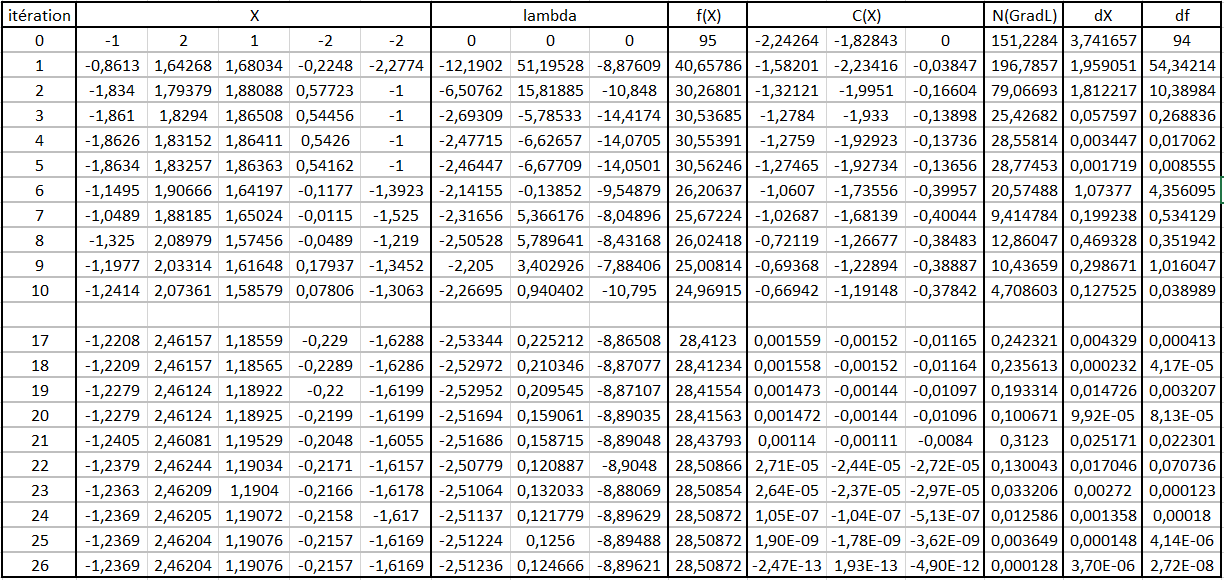
On cherchait à minimiser la fonction

sous les contraintes

(cf. *MH4WD.m*)

Le point d’initialisation devait être et nous devions obtenir et .

*SQP* renvoie :



Cas Test Ariane 1

*SQP* nous a permis d’optimiser les masses d’ergols pour un problème-type dont les données sont les indices constructifs des étages, les vitesses d’éjections , la vitesse de propulsion à atteindre et la masse du satellite. Les données constantes d’*Ariane 1* nous étaient fournies :

 ;

 ;

 ;

;

en plus des masses d’ergol à obtenir : et

Le problème-type nous était donné sous la forme :

où

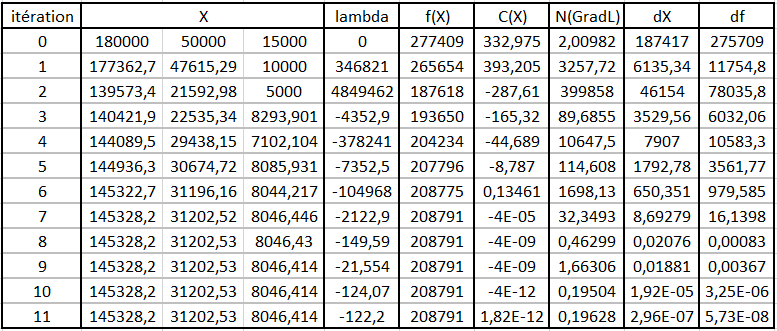
On peut alors expliciter les termes en fonction des  à partir des relations Ce qui donne :

.

avec .

Dans le fichier *Ariane1.m,* notre fonction ne prend donc que les masses d’ergol en argument et revoie les valeurs de la fonction et de la contrainte pour ces masses d’ergol.

Les sorties *SQP*sont les suivantes :



Donc .

**Résolution du problème d’étagement**

On cherche le lanceur de masse, la plus légère possible, pour qu’un satellite de masse atteigne une certaine vitesse propulsive.

Or dépend des masses d’ergol :

Le problème peut donc s’écrire :

Résolution analytique

On réécrit alors ce problème comme un problème de minimisation sous contraintes.

Sachant que et en posant , on obtient pour la contrainte .

= = =

Or

Et

Alors

Or

Si on pose , alors

Montrons que si vérifient les conditions KKT alors où .

Le lagrangien s’écrit

Déroulons les calculs sur la première composante :

En posant, on obtient et .

En faisant de même sur les autres composantes :

⇔⇔

Or

Or

. Ainsi :

On cherche donc les tels que .

Ainsi revient à résoudre avec pour inconnue.

Nous avons résolu cette équation en implémentant la fonction correspondante dans *fC\_Etagement.m* et une méthode de Newton (*Newton.m*).

avec .

Les fonctions sont appelées et les résultats sont obtenus à partir de *Test\_Etagement.m*.

On récupère alors la valeur de . Par, on a .

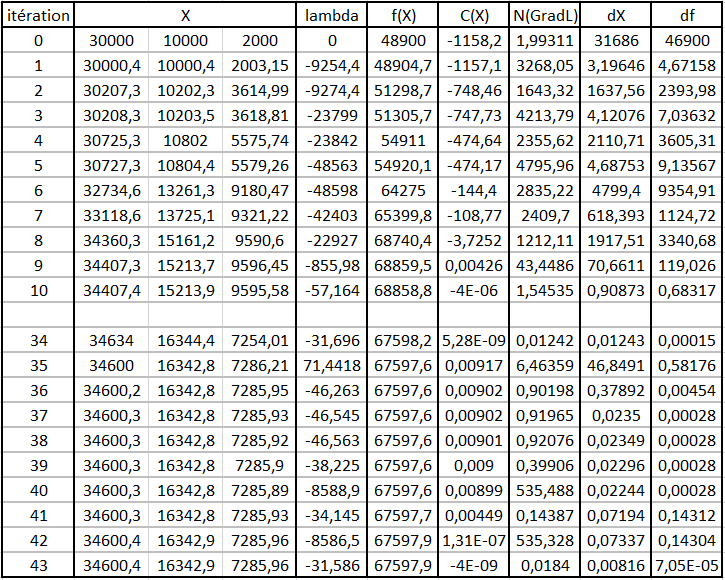
Le multiplicateur est donné par

Les conditions KKT sont alors bien vérifiées, et à partir des formules, on a Alors

Résolution numérique

On a vérifié que nos résultats étaient bien cohérents avec ceux obtenu en lançant notre optimiseur. On fournit *Etagement* -dans *Etagement.m* – à *SQP.* *Etagement* prend en argument . Les données sont :

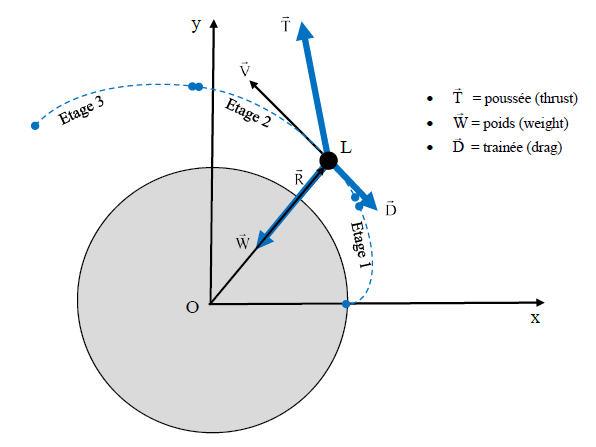
et les sorties :



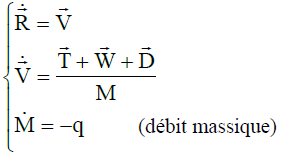
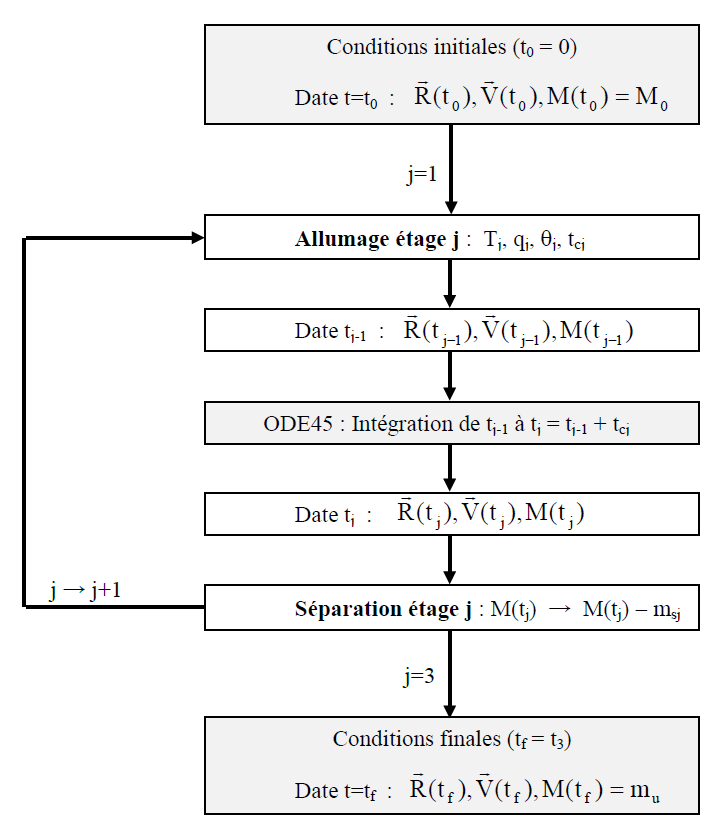
Donc

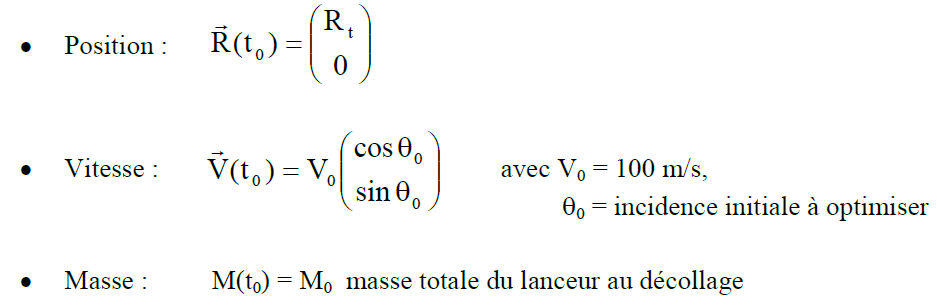
**Simulateur de trajectoire**

**Description du simulateur**

La deuxième partie du projet consiste en la réalisation d’un simulateur de trajectoire pour le lanceur (*Simulateur.m*). Ce simulateur doit permettre, à partir des angles vitesse-poussée et des masses d’ergol, solutions du problème d’étagement, de déterminer la trajectoire du lanceur de la surface de la Terre jusqu’à une orbite située à l’altitude Hc = 300 km.

On considère que le lanceur est soumis à trois forces : sa poussée T, son poids W et sa traînée D. On obtient alors, selon la deuxième loi de Newton, les équations du mouvement suivantes :



Pour résoudre ces équations (transcrites dans *Mouvement.m*), on utilise l’intégrateur numérique ODE45 de Matlab. Le trajet est, de plus, découpé en trois étapes correspondant chacune à la consommation du carburant d’un étage du lanceur. Ces trois étapes sont précédées, pour simplifier le problème, d’une « étape 0 » qui représente les premiers instants suivants le décollage. Le problème est initialisé avec les données suivantes :

**Modélisation d’une première trajectoire**

Validation du simulateur

L’efficacité du simulateur a été testée sur un groupe d’angles [θ0, θ1, θ2, θ3] déterminé manuellement, par essais successifs, de telle sorte que la trajectoire du lanceur atteigne l’altitude souhaitée.

*Données initiales de la simulation*

Masses d’ergols (en kg) Angles (en degrés)

|  |  |  |
| --- | --- | --- |
| me1 | me2 | me3 |
| 34 353 | 16 562 | 7 300 |

|  |  |  |  |
| --- | --- | --- | --- |
| θ0 | θ1 | θ2 | θ3 |
| 1 | -1 | 5 | 10 |

*Résultats attendus*

Altitude à atteindre Rc = 6678 km

Vitesse finale à atteindre Vc = 7725 m/s

Masse du satellite mu = 2000 kg

*Résultats finals*

Données finales

|  |  |  |
| --- | --- | --- |
| Altitude | Vitesse | Masse |
| 6686 km | 7297 m/s | 2000 kg |

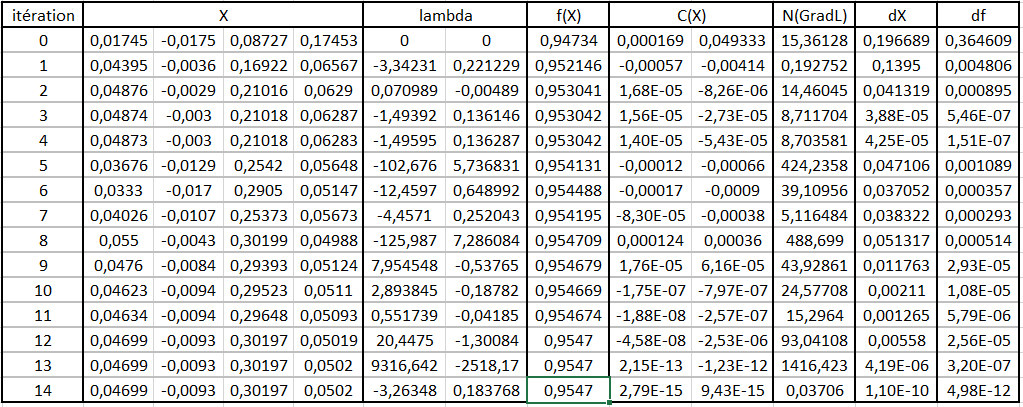
Les données finales sont cohérentes avec les résultats attendus ce qui valident l’efficacité simulateur.

Optimisation des angles vitesse-poussée

L’objectif de cette étape est de trouver les angles qui maximisent la vitesse du lanceur au temps final, i.e. en respectant les contraintes suivantes :

Pour ce faire on utilise de nouveau l’optimiseur *SQP*, initialisé avec le groupe d’angles [θ0, θ1, θ2, θ3] déterminé à l’étape précédente, et appliqué au problème décrit ci-dessus (écrit dans *Trajectoire.m* avec des fonctions normalisées pour des raisons pratiques).

*Sorties :*



*Résultat final (en degrés)*

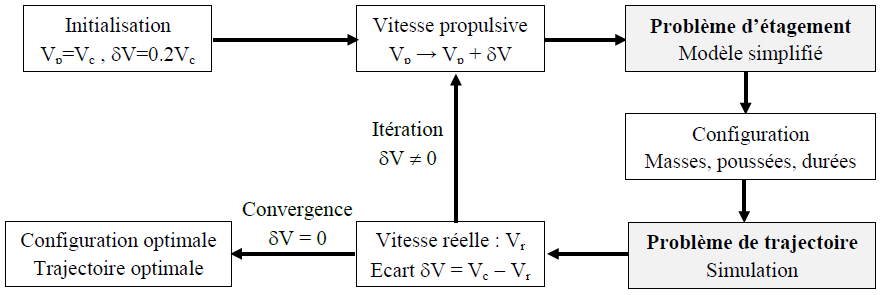
|  |  |  |  |
| --- | --- | --- | --- |
| θ0 | θ1 | θ2 | θ3 |
| 2,6926° | -0,5313° | 17,3018° | 2,8760° |

**Réglage de la vitesse de propulsion**

Les deux composantes principales du projet étant fonctionnelles, il s’agit désormais de les rassembler (*Projet.m*) afin de déterminer la vitesse de propulsion initiale nécessaire pour atteindre l’orbite souhaitée à la vitesse-cible 7725 m/s.

On initialise le programme avec une valeur initiale puis on résout successivement les problèmes d’étagement et de trajectoire. Enfin, à chaque étape on incrémente de , l’écart entre (la vitesse finale réelle) et .

Pour la première itération, 1.2\* = 9271 m/s.



*Résultats finals*

Vitesse propulsive initiale Vp = 9 545 m/s

Masse d'ergol des différents étages (en kg) :

|  |  |  |
| --- | --- | --- |
| me1 | me2 | me3 |
| 38 995 | 19 939 | 7 694 |

Angle pour chaque étape :

|  |  |  |  |
| --- | --- | --- | --- |
| θ0 | θ1 | θ2 | θ3 |
| 12,396° | 5,498° | 14,157° | 2,565° |



Altitude finale : 300 km Vitesse finale : 7 716 m/s

Ecart avec l'altitude cible : 0,000% Ecart avec la vitesse cible : 0,128%

**Conclusion**

En segmentant les différentes étapes de la résolution du problème, nous avons vérifié nos résultats au fur et à mesure du projet. Les tests *MH4D*, *Ariane 1* et la résolution analytique du problème d’étagement ont permis de vérifier le bon fonctionnement de l’optimisateur. On a donc pu optimiser les masses d’ergol.

La réalisation d’une trajectoire convenable à partir du simulateur nous a permis d’établir que le simulateur était acceptable. Nous en avons alors tiré les angles de vitesse-poussée (*p.12*) permettant d’initialiser la minimisation de ces mêmes angles par le programme SQP.

Enfin à partir des problèmes d’étagement et de trajectoire, nous avons abouti au résultat escompté : notre lanceur Ariane est arrivé en orbite, à 300 km d’altitude, avec une vitesse finale de 7 716 m/s très proche de la vitesse-cible attendue.