Projet Optimisation Ariane

Florent Valbon, Alexandre Rueda Payen, Leonce Fopa Notemi

7 janvier 2020

Résumé

1 Présentation du projet

La mission d'un lanceur spatial est d'amener un satellite en orbite. La mise en orbite est définie en termes d'altitude et de vitesse à atteindre. L'objectif du projet est de trouver le lanceur le plus léger possible permettant d'amener un satellite de masse donné sur une orbite donnée, et de déterminer la vitesse de propulsion nécessaire pour atteindre cette orbite.
Dans un premier temps, nous programmerons l'algorithme SQP (Sequential Quadratic Programming), qui permet de résoudre un problème d'optimisation sous contrainte.
Ensuite nous analyserons le problème d'étagement (déterminer selon une vitesse propulsive fixée, les masses d'ergols (carburant) optimisant la masse de la fusée au décollage.
Nous utiliserons ces masses d'ergols pour simuler les données de vol du lanceur, paramétrées par ses angles de poussée, et essayer de trouver des angles permettant de respecter les conditions de mise en orbite.

2 Algorithme SQP

Nous expliquerons ci-dessous certains choix effectués pour optimiser l'utilisation de l'algorithme SQP : méthodes d'approximation du gradient, de la matrice hessienne, calcul de la fonction mérite, respect de l'aspect physique et processus de globalisation. Enfin nous donnerons une idée de l'utilisation de nos paramètres pour lancer notre algorithme SQP afin de résoudre un problème physique bien défini.

2.1 Approximation du gradient

2.1.1 Méthode des différence finis

Pour calculer $\nabla f(x)$ on utilise la méthode des différences finies :
On notera h_i le$i$-ème élément de la base canonique multiplié par un scalaire suffisamment petit, noté h.

$$\forall i \in I : \frac{\partial f(x)}{\partial x_i} \approx \frac{f(x + h_i) - f(x)}{h}.$$

De plus, nous ferons correspondre la valeur de h à l'ordre de grandeur de x.
Nous choisirons h de la façon suivante : $h_i(x_k) = h_{init}.x_k$ pour un h_{init} paramètre de notre fonction SQP.

Notre fonction gradient prend en argument les valeurs de $f(x)$ et $c(x)$ déjà calculées ainsi que les vecteurs (h_i) et calculera $f(x + h_i)$ et $c(x + h_i)$ pour tous h_i formant la base canonique de l'espace de départ. Enfin nous renverrons les vecteurs calculés par la méthode des différences finies.

2.2 Calcul du Hessien

Nous privilégierons la formule BFGS pour des problèmes de petite dimension et la formule SR1 pour des problèmes de grande dimension, pour des raisons de temps de calcul. Nous avons vérifié que celles-ci donnent exactement les même résultats.

2.3 Respect de l'aspect physique

Principe : toute variable calculée est bornée car correspondant à une réalité physique.
Ainsi à chaque calcul d'une nouvelle solution x_{k+1}, une projection sur un pavé de solutions admissibles sera effectué. Cela sera nécessaire pour le calcul des angles d'incidence optimaux pour la résolution du problème de la mise en orbite.
Cependant il ne faudra pas modifier la direction de descente associée à x_k afin d'éviter de perturber l'algorithme, très sensible à ce genre de problèmes.

2.4 Fonction mérite

L'algorithme SQP est un algorithme de quasi-Josephy-Newton appliqué à la fonction mérite qui sera dynamique tout au long du programme. Le

paramètre ρ appliqué à la fonction mérite nous permet de réguler l'importance du respect des contraintes par rapport au caractère de la valeur de la fonction. Ainsi plus ρ sera grand, plus l'algorithme SQP se focalisera sur le respect des contraintes, avant de se concentrer sur le caractère minimal de la fonction.

2.5 Globalisation

Le coefficient d'Armijo sera augmenté au fur et à mesure : l'objectif étant de s'éloigner le plus de x_k au fur et à mesure que l'on se rapproche de la solution (on évitera ainsi le piège des minimums locaux).

Dans certains cas, la direction de descente associée à la fonction n'est pas une direction de descente pour la fonction mérite. Cela signifie que le poids de la contrainte est trop faible. On augmentera alors la valeur du paramètre ρ.

2.6 Initialisation de la fonction SQP

La fonction SQP (fichier SQP.m) admet le template suivant :

$$[x_{optimal}, f(x_{optimal})] =$$

$$SQP(x_{initial}, callback, choice, \rho, , h_{init}, maxiterations, bounds)$$

où :

— x_{intial} est le point de départ de l'algorithme SQP. Si $x_{initial}$ est situé dans le bassin d'attraction de l'optimum, alors le taux de convergence est quadratique.
— callback sera une fonction qui renverra la valeur de $f(x_k)$ et $c(x_k)$ pour une entrée de x_k.
— choice vaudra 1 si l'on choisira la formule de calcul de Hessien par la méthode SR1 sinon 2.
— ρ sera le poids des contraintes sur la fonction mérite (que l'on cherche à minimiser).
— h_{init} sera le coefficient (à ordre de grandeur près) d'incrément de la fonction gradient.
— max itérations sera le nombre d'itérations maximales de l'algorithme SQP, nous permettant de vérifier si une solution a bien été trouvée.
— bounds sera une matrice représentant les bornes physique associés au problème, dont l'intérieur sera le pavé de projection mentionné ci-dessus.

Nous avons rajouté un compteur pour calculer le nombre d'appels totaux de la fonction.

2.7 Résolution du problème test

Nous donnons ci-dessous un tableau des valeurs prises par les différents paramètres à la fin de chacune des douze itérations de l'algorithme SQP appliquée à la fonction test vue en cours, ainsi que l'initialisation donnée.

Itération	nombre d'appels	f	$\|\|c\|\|_2$	∇L	ρ
1	18	40.6579	2.7378	$1.2027.10^{3}$	1
2	29	23.5247	2.6227	-134.3434	1
3	37	27.7279	1.2596	12.1799	1
4	43	33.7284	0.3573	8.4638	1
5	49	29.6020	0.0208	-2.8638	1
6	55	28.5057	0.0114	$2.8627.10^{-4}$	1
7	61	28.5078	$6.9583.10^{?4}$	-0.0351	1
8	67	28.5078	$1.34410.10^{-4}$	0.0029	1
9	73	28.5078	$4.5445.10^{-4}$	0.0010	1
10	79	28.5078	$2.8033.10^{-5}$	$6.4898.10^{-10}$	1
11	85	28.5078	$3.5389.10^{?12}$	$4.678.10^{-9}$	1
12	91	28.5078	$8.4066.10^{-14}$	$3.2486.10^{-11}$	1

Les codes nous ayant permis de retrouver ces résultats on été mis en commentaire dans la fonction SQP. L'algorithme SQP appliqué à la fonction probleme est appelé à partir de la fonction test sqp.

3 Résolution du problème d'étagement

3.1 Définition et reformulation du problème

On cherche à ré-écrire le problème sous la forme :

$$\min_{x_1,x_2,x_3} f(x_1,x_2,x_3) \text{ sous } c(x_1,x_2,x_3)=0,$$

avec :

$$\begin{cases} f(x_1,x_2,x_3) = -(\frac{1+k_1}{x_1}-k_1)(\frac{1+k_2}{x_2}-k_2)(\frac{1+k_3}{x_3}-k_3) \\ c(x_1,x_2,x_3) = v_{e1}\ln(x_1)+v_{e2}\ln(x_2)+v_{e3}\ln(x_3)-V_p \end{cases} \quad (1)$$

Soit $x_i = \frac{M_{j,i}}{M_{f,i}} \neq 0$. On supposera de plus les v_{ei} non nuls.
En réécrivant

$$J = \frac{m_u}{M_0} = \frac{M_{i,4}M_{i,3}M_{i,2}}{M_{i,3} * M_{i,2}M_{i,1}},$$

et en prenant

$$\frac{M_{i,j+1}}{M_{i,j}} = \frac{(M_{f,j} - m_{s,j})}{M_{i,j}} = \frac{M_{f,j}}{M_{i,j}}.(1 + \left(-\frac{m_{s,j}}{M_{f,j}}\right)),$$

on identifie

$$k_i = \frac{m_{s,j}}{M_{f,j} - M_{i,j}} = \frac{m_{s,j}}{m_{e,j}}.$$

La formule de contrainte vient naturellement.

3.2 Conditions d'optimalité d'ordre 1

Nous devons avoir pour (x_1, x_2, x_3) optimal :

$$\begin{cases} c(x_1, x_2, x_3) = 0 \\ \exists \lambda \in \mathbb{R}\ \nabla f(x_1, x_2, x_3) + \nabla c(x_1, x_2, x_3)^T.\lambda = 0 \end{cases} \tag{2}$$

d'où,

$$\begin{cases} v_{e1}\ln(x_1) + v_{e2}\ln(x_2) + v_{e3}ln(x_3) - V_p = 0 \\ (\frac{1+k_2}{x_2} - k_2)(\frac{1+k_3}{x_3} - k_3)(\frac{1+k_1}{v_{e1}.x_1}) = \lambda \\ (\frac{1+k_2}{x_2} - k_2)(\frac{1+k_1}{x_1} - k_1)(\frac{1+k_3}{v_{e3}.x_3}) = \lambda \\ (\frac{1+k_1}{x_1} - k_1)(\frac{1+k_3}{x_3} - k_3)(\frac{1+k_2}{v_{e2}.x_2}) = \lambda \end{cases} \tag{3}$$

soit,

$$\begin{cases} (\frac{1+k_1}{x_1} - k_1)(\frac{1+k_2}{x_2} - k_2)(\frac{1+k_3}{x_3} - k_3)(\frac{1+k_1}{v_{e1}.x_1}) = \lambda(\frac{1+k_1}{x_1} - k_1) \\ (\frac{1+k_1}{x_1} - k_1)(\frac{1+k_2}{x_2} - k_2)(\frac{1+k_3}{x_3} - k_3)(\frac{1+k_2}{v_{e2}.x_2}) = \lambda(\frac{1+k_2}{x_2} - k_2) \\ (\frac{1+k_1}{x_1} - k_1)(\frac{1+k_2}{x_2} - k_2)(\frac{1+k_3}{x_3} - k_3)(\frac{1+k_3}{v_{e3}.x_3}) = \lambda(\frac{1+k_3}{x_3} - k_3) \end{cases} \tag{4}$$

On pose

$$\psi = \left(\frac{1+k_1}{x_1} - k_1\right)\left(\frac{1+k_2}{x_2} - k_2\right)\left(\frac{1+k_3}{x_3} - k_3\right) \text{ et } \Omega_i = \frac{k_i}{1+k_i}.$$

L'équation ci-dessus se simplifie. Pour i :

$$\psi v_{ei}(1 - \Omega_i x_i)(1 + k_i) = \lambda(1 + k_i)$$

$1 + k_i > 0$
On trouve ainsi :

$$\forall i : v_{ei}(1 - \Omega_i x_i) = \frac{\lambda}{\psi}.$$

3.3 Équation à une inconnue

En reprenant le premier système il vient :

$$\left(\frac{1+k_3}{x_3} - k_3\right)\left(\frac{1+k_2}{v_{e2}.x_2}\right) = \left(\frac{1+k_2}{x_2} - k_2\right)\left(\frac{1+k_3}{v_{e3}.x_3}\right),$$

d'où,

$$x_2 = -\frac{v_{e3}(1 - \Omega_3 x_3)}{\Omega_2 v_{e2}} + \frac{1}{\Omega_2}.$$

On montre de même que :

$$x_1 = -\frac{v_{e2}(1 - \Omega_2 x_2)}{\Omega_1 v_{e1}} + \frac{1}{\Omega_1}.$$

En reprenant la contrainte d'égalité on obtient :

$$v_{e1}\ln\left(-\frac{v_{e3}(1-\Omega_3 x_3)}{\Omega_1 v_{e1}} + \frac{1}{\Omega_1}\right) + v_{e2}\ln\left(-\frac{v_{e3}(1-\Omega_3 x_3)}{\Omega_2 v_{e2}} + \frac{1}{\Omega_2}\right) + v_{e3}\ln(x_3) - V_p = 0.$$

Qui est une équation à une seule inconnue : x_3.

3.4 Résultat analytique et méthode de la sécante

Nous avons utilisé la méthode de la sécante pour résoudre ce problème à une dimension.
Finalement, à partir de la valeur de x_3 trouvée, on déduira les valeurs de x_1 de la façon suivante :

$$\eta = v_{e1}(1 - \omega_3 x_3),$$

$$x_2 = \left(1 - \left(\frac{\eta}{v_{e2}}\right)\right)\omega_2,$$

$$x_1 = \left(1 - \left(\frac{\eta}{v_{e1}}\right)\right)\omega_1,$$

$$M_{i1} = M_0 = f(x_1, x_2, x_3)m_u = -(\frac{1+k_1}{x_1} - k_1)(\frac{1+k_2}{x_2} - k_2)(\frac{1+k_3}{x_3} - k_3)m_u,$$

$$
\begin{aligned}
m_{e1} &= M_{i1}\left(1-\frac{1}{x_1}\right),\\
M_{i2} &= M_{i2} - k_1 m_{e1} - m_{e1},\\
m_{e2} &= M_{i2}\left(1-\frac{1}{x_2}\right),\\
M_{i2} &= M_{i3} - k_1 m_{e2} - m_{e2},\\
m_{e3} &= M_{i3}\left(1-\frac{1}{x_3}\right).
\end{aligned}
$$

On remarquera que la fonction equation.m est strictement croissante : nous utiliserons une méthode de recherche par dichotomie pour trouver des valeurs initiales admissibles (on résoudra l'équation avec une grande tolérance). Cet algorithme est relativement lent, comparé à la méthode de la sécante, qui si les valeurs initiales sont prises dans la bassin d'attraction de l'optimum est de taux de convergence $\phi \simeq 1.618$.

Données de l'algorithme sécante pour $m_u = 1500$:

m_{e1}	m_{e2}	m_{e3}	M_0
11 015	6712.1	4348.6	26 792

3.5 Algorithme SQP

Nous décidons de tester maintenant l'algorithme SQP dessus. Pour cela, connaissant le résultat de l'algorithme de la sécante et les déductions sur les autres masses d'ergols, nous décidons d'appliquer l'algorithme SQP sur la fonction probleme_etagement.m prenant en argument des masses d'ergols et calculant la masse totale de la fusée au décollage. Celles-ci seront initialisées par les masses d'ergols auxquelles on ajoute une perturbation (d'ordre 10^3). Nous obtenons les même résultats.

Nous nous servirons de cette stabilité (grande zone d'attraction) pour résoudre le problème final.

4 Données de vol et mise sur orbite

Nous détaillons ci-dessous les paramètres utilisés dans la résolution du problème de la mise sur orbite.

4.1 Paramètres d'initialisation de la fonction données finales

La fonction finale sera nommée speed_under_constraint.m et se trouve dans le fichier éponyme. Nous utilisons 3 fois la fonction ODE45 avec les paramètres $(\theta_i)_{i\in\{1,..,4\}}$, M_e correspondant à la masse du lanceur initial, puis après que l'on ait détaché le premier étage, et enfin après que le détachement du deuxième étage soit effectué, m_e les masses d'ergols précédemment calculées, V_c vitesse cible à atteindre (on rappelle que vitesse et altitude correspondent dans le cas d'une orbite).

Le paramètre angulaire, nous permettant de calculer la direction de la poussée (angle d'incidence Vitesse/Poussée) : il s'agira d'une variable sur laquelle on appliquera l'algorithme SQP.

4.2 Résolution par la méthode SQP

La difficulté du problème est de trouver un bassin d'attraction pour le maximum.
On supposera dans la suite -par intuition physique- que le première paramètre θ_1 utilisé pour diriger la fusée à son décollage est plus influent sur le résultat (sensibilité de ce paramètre sur la fonction mérite) que θ_2, lui même plus influent que θ_3 et ainsi de suite...

Nous proposons deux l'approche heuristique suivante :
Tout d'abord nous initialisons l'algorithme SQP. Nous effectuerons un balayage en fixant les deux derniers paramètres θ puis nous récupérons, puis un second balayage en fixant le dernier paramètre en utilisant comme valeurs optimales pour θ_1 et θ_2 la valeur renvoyée par le premier balayage et ainsi de suite... Après plusieurs essais expérimentaux, nous constatons que le respect des contraintes est assez difficile. Nous décidons donc de fixer un paramètre ρ pour la fonction de mérite assez grand ($\rho = 1e6$) afin de forcer dans un premier temps la solution obtenue par balayage à respecter les contraintes (avant de maximiser).

Puis nous recommençons le même processus de balayage en trois temps avec une valeur de ρ beaucoup plus faible pour chaque algorithmes SQP, nous permettant ainsi une mise sur orbite.

Face à la difficulté du problème, nous avons tout d'abord fixé une seule contrainte (il s'agira de l'ortho-radialité de la vitesse) et effectué un premier balayage similaire avec cette contrainte seule, puis nous procédons comme ci-dessus.

Nous avons implémenté cette méthode. Voici les résultats pour :

$$V_p = \sqrt{\frac{\mu}{R_c}}$$

θ_1	θ_2	θ_3	$\theta 4$
-1.1119	0.0000	-0.3380	0.0226

Nous joignons également dans le dossier les graphes de la trajectoire optimisée trouvée (voir fichier PLOT).

4.3 Valeur de la tolérance dans SQP

Nous cherchons à trouver une tolérance dont l'ordre de grandeur correspondant avec les données physiques testées.
En effet, la tolérance dans SQP correspond à un pas d'une trop faible amplitude. On pourra vérifier que le gradient du Lagrangien de la fonction speed_under_constraint.m corrobore ce critère d'arrêt en vérifiant que sa norme est presque nulle.

Nous prendrons comme paramètre de tolérance d'ordre de grandeur :
— 10 pour les masses d'ergols
— 0.00001 pour les angles θ

4.4 Problème de la vitesse de propulsion

Nous rappelons que le problème initial à traiter était de trouver une configuration optimale du lanceur. Nous effectuons ainsi une boucle dans laquelle on essaye d'approcher, par la résolution SQP, la vitesse cible de mise en orbite.

Ainsi d'une itération à l'autre nous prendrons comme valeurs initiales les masses d'ergols et les valeurs de θ optimales trouvées pour les précédentes vitesses propulsives.

Voici un tableau récapitulatif des dix dernières itérations de notre logiciel :

m_e1	m_e2	m_e3	δV_c	δV_p	M_0
33457	14201	5900	7451,8	10000	61713
86193	25229	7581	7451,8	10000	134650
206570	37030	7890	7451,8	1	280780
597480	72390	10500	7451,8	1	754580
2097500	0.1631e6	0.0151e6	7451,8	1	$v.a$
1.1078e7	0.0479e7	0.0025e7	7451,8	1	$v.a$
1.9292e8	0.0315e8	0.0006e8	7451,8	1	$v.a$

Toutefois, l'algorithme s'arrête à l'itération 7 et renvoie une erreur. Les valeurs de la masse M_0 sont aberrantes (v.a).