



COMPTE RENDU PROJET OPTIMISATION

Optimisation d'un lanceur spatial avec un algorithme SQP



02 JANVIER 2019

MAHAMADOU KLANAN DIARRA/LAURENT SEKONIAN

Sommaire

Introduction	2
I/Algorithme SQP	3
1/Principe de l'algorithme	4
II/Cas Tests	5
1/Cas test n°1	5
2/Cas test MHW4D	6
3/Cas Ariane 1	7
III/Projet Lanceur	9
1/Objectif du projet	9
2/Problème d'étagement	9
a. Résolution par l'algorithme SQP	9
b. Résolution analytique	10
3/Trajectoire du lanceur	16
a. Modélisation de la trajectoire	16
b. Formulation du problème	16
4/Résolution du problème globale	17
Bonus	19
Conclusion	20

Introduction :

Dans ce cours d'optimisation, il nous a été confié le projet d'implémenter un algorithme d'optimisation SQP dans un environnement Matlab puis de l'appliquer dans le cas d'un lanceur spatiale afin de déterminer les masses d'ergols à embarquer fonction de l'altitude et de la vitesse nécessaire à la mise en orbite d'un satellite.

Dans une première partie nous allons aborder le principe de fonctionnement de l'algorithme SQP ainsi que la globalisation afin d'améliorer les solutions données par celui-ci.

Dans la deuxième partie nous présenterons les tests effectués par l'intermédiaire des cas tests afin de vérifier le comportement du programme.

Enfin nous procéderons à la résolution du problème du lanceur en le décomposant en deux problèmes distincts pouvant être résolu à l'aide de notre algorithme SQP. Puis nous effectuerons la résolution globale par une méthode itérative.

I / Algorithme SQP :

Les algorithmes newtoniens sont basés sur la linéarisation d'équations caractérisant les solutions que l'on cherche, fournies par les conditions d'optimalité d'ordre 1. Ces algorithmes sont primaux-duaux dans le sens où ils génèrent à la fois une suite primale x_k convergeant vers une solution x^* du problème considéré, et une suite duale λ_k de multiplicateurs convergeant vers un multiplicateur optimal λ^* associé à x^* .

Dans notre implémentation de l'algorithme SQP nous auront seulement comme exemple le cas de contraintes d'égalités.

Le problème d'optimisation avec contraintes d'égalité s'écrit :

$$\begin{aligned} \min f(x) \text{ avec } x \in \mathbb{R}^n \\ \text{sous } h_i(x) = 0; i = 1, \dots, m \end{aligned}$$

Où $f: \mathbb{R}^n \rightarrow \mathbb{R}$ et $h_i: \mathbb{R}^n \rightarrow \mathbb{R}$, $i = 1, \dots, m$

Les conditions d'optimalité de Lagrange s'écrivent :

$$\begin{aligned} \nabla L(x; \lambda) = 0 \quad \text{soit} \quad \nabla f(x) + \sum_{i=1}^m \lambda_i \nabla h_i(x) = 0 \\ h(x) = 0 \end{aligned}$$

Pour résoudre le système d'équation on utilise la méthode de newton :

$$H[L](x^k; \lambda^k)(x^{(k+1)} - x^k, \lambda^{(k+1)} - \lambda^k)^T = -\nabla L(x^k; \lambda^k),$$

$$\text{Soit} \quad \begin{pmatrix} H_x[L](x^k; \lambda^k) & Dh(x^k)^T \\ Dh(x^k) & 0 \end{pmatrix} (x^{(k+1)} - x^k, \lambda^{(k+1)} - \lambda^k)^T = -(\nabla_x L(x^k; \lambda^k), h(x^k))^T$$

Où $Dh(x)$ désigne la matrice Jacobienne définie par

$$Dh(x) = [\nabla h_1(x) \mid \dots \mid \nabla h_m(x)]$$

On pose $H_k = H_x[L](x^k; \lambda^k)$, $d = x^{(k+1)} - x^k$, et $\mu = \lambda^{(k+1)}$

On peut réécrire l'itération de newton sous la forme :

$$\begin{aligned} H_k d + \sum_{i=1}^m \mu_i \nabla h_i(x^k) = -\nabla f(x^k) \\ \nabla h_i(x^k)^T d + h_i(x^k) = 0, i = 1, \dots, m \end{aligned}$$

On a alors le problème quadratique suivant :(QP)

$$\begin{aligned} \min \quad \nabla f(x^k)^T d + \frac{1}{2} d^T H_k d \\ \text{sous } h_i(x^k) + \nabla h_i(x^k)^T d = 0 \end{aligned}$$

On remplace donc le problème initial par une suite de problèmes quadratiques sous contraintes linéaires plus faciles à résoudre.

Il nous reste alors à itérer tant que la norme deux du gradient du lagrangien évalué en x^k et λ^k est inférieur à un certain epsilon (précision demandée). On obtient alors la solution primale d_k et le multiplicateur λ associé.

Dans notre cas, l'algorithme est destiné à un utilisateur qui a juste à rentrer la fonction et la contrainte. C'est-à-dire que pour le gradient de la fonction et de la contrainte nous n'avons pas d'expression algébrique. Nous utiliserons donc la méthode des différences finies pour déterminer le gradient de la fonction et de la contrainte.

Afin que le sous-programme quadratique (QP) admette une unique solution, la plupart des implémentations actuelles de SQP utilisent une approximation du hessien H_k du Lagrangien qui soit définie positive. En particulier celle fournie par les techniques quasi-newtonienne (BFGS), (SR1) par exemple.

Etant une méthode newtonienne, l'algorithme SQP converge localement quadratiquement pourvu que les points initiaux (x_0, λ_0) soient dans un voisinage d'un point stationnaire à x^* et de ses multiplicateurs associés λ^* .

En effet cet algorithme ne peut fournir que des extrema locaux, on globalise l'algorithme en ajoutant une étape de recherche linéaire. Cette étape permet d'améliorer la robustesse en identifiant si la solution QP peut être utilisé comme direction de recherche pour construire un déplacement acceptable.

Dans la globalisation, on utilise une fonction mérite à la place du gradient du lagrangien car l'évaluation est trop coûteuse. Cette fonction mérite est :

$$F(x) = f(x) + \rho \|h(x)\|_1$$

ρ est le coefficient de pénalisation qui est adaptée pour respecter les contraintes à la précision voulue.

si $F(x^{(k+1)}) < F(x^k)$; Alors la solution QP est acceptée.

Dans le cas contraire on part de x^k et on cherche un pas dans la direction donnée par la direction d qui vérifie la condition d'Armijo :

$$F(x_k + sd) < F(x_k) + \omega s F'_d(x_k)$$

Si $F'_d(x_k) > 0$ alors on réinitialise le Hessien et on augmente le facteur de pénalisation.

II / Cas tests :

Une fois l'algorithme implémenter conformément à la description vue précédemment. Nous testons notre algorithme grâce à des cas tests. Le but de ces tests est de vérifier le comportement du programme.

Pour cela nous avons des cas dont nous connaissons les données d'entrée ainsi que les résultats. On utilise alors notre programme SQP pour résoudre ces problèmes d'optimisation tests.

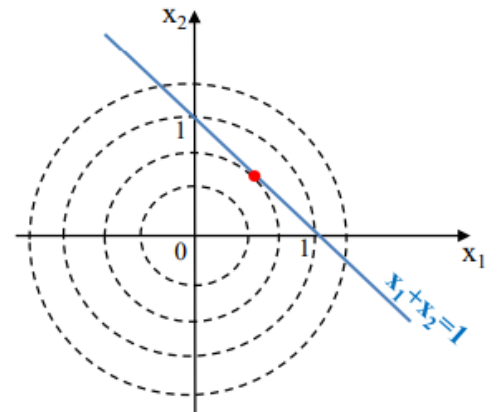
1/Cas test n°1 :

On considère le problème d'optimisation suivant :

$$\begin{aligned} \min \quad & x_1^2 + x_2^2 \\ \text{sous} \quad & x_1 + x_2 = 1 \end{aligned}$$

Données en entrée :

```
h=1e-5;
eps = 1e-7;
choix = "bfgs";
rho=10;
x_0=[1;1];
lamb=-2;
born_min = [-10,-10];
born_max = [10,10];
```



Résultats :

```
----- premier cas test -----
pts min =[ 0.500000 0.500000 ]
f_value = 0.500000
cont =[ -0.000000 ]
```

cas test 1							
Iterations	X		Contrainte	fonction	λ	coeff_pen(rho)	nb_appel de la fonction
	x1	x2					
1	1	1	1	2	-2	-2	1
2	0,5	0,5	6,55E-12	0,5	-1,50001	-1,50001	8
3	0,5	0,5	-2,22E-16	0,5	-1,00001	-1,00001	19

2/Cas test MHW4D :

On considère le problème d'optimisation suivant :

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

$$\text{Sous } c_1(x) = x_1 + x_2^2 + x_3^2 - 3\sqrt{2} - 2 = 0$$

$$c_2(x) = x_2 - x_3 + x_4 - 2\sqrt{2} + 2 = 0$$

$$c_3(x) = x_1 x_5 - 2 = 0$$

Données en entrée :

```
h=1e-5;
eps = 1e-7;
choix = "bfgs";
x_0=[-1;2;1;-2;-2];
lamb=[100;100;100];
rho=1000;
born_min = [-1000,-1000,-1000,-1000,-1000];
born_max = [1000,1000,1000,1000,1000];
```

Résultats :

```
-----Problème MHW4D-----
pts min =[ -1.273043 2.410357 1.194854 -0.154255 -1.571039 ]
f_value = 27.871905
cont =[ 0.000000 0.000000 0.000000 ]
```

cas test MHW4D								
Iterations	X					Contrainte		
	x1	x2	x3	x4	x5	C1	C2	C3
1	-1,00000	2,00000	1,00000	-2,00000	-2,00000	-2,24264	-1,82843	0,00000
2	-0,87410	1,49249	1,65807	-0,11920	-2,25179	-0,33085	-2,20434	-0,03170
3	-0,99577	1,46739	1,68253	0,02146	-1,93893	-0,32206	-2,17050	-0,06927
4	-0,97608	2,19617	1,44734	-0,41353	-2,01205	0,63631	-1,14057	-0,03608
5	-1,22347	2,17161	1,49122	-0,11938	-1,50672	0,56582	-0,99992	-0,15658
6	-1,40171	2,53177	1,19864	-0,35222	-1,41519	0,48764	-0,08561	-0,01631
7	-1,38535	2,47495	1,14846	-0,33008	-1,44335	0,01216	-0,00252	-0,00046
8	-1,34324	2,45706	1,15943	-0,28637	-1,48730	0,00985	-0,00201	-0,00220
9	-1,29521	2,43051	1,17864	-0,21426	-1,54130	0,00692	-0,00137	-0,00369
10	-1,26716	2,40868	1,19572	-0,15081	-1,57753	0,00151	-0,00029	-0,00102
11	-1,27296	2,41049	1,19468	-0,15479	-1,57112	0,00001	0,00000	-0,00004
12	-1,27301	2,41037	1,19484	-0,15431	-1,57108	0,00000	0,00000	0,00000
13	-1,27304	2,41036	1,19485	-0,15426	-1,57104	0,00000	0,00000	0,00000
14	-1,27304	2,41036	1,19485	-0,15426	-1,57104	0,00000	0,00000	0,00000
15	-1,27304	2,41036	1,19485	-0,15426	-1,57104	0,00000	0,00000	0,00000

cas test MHW4D					
fonction	lambda			coeff_pen (rho)	nb_appel de la fonction
	lamb_1	lamb_2	lamb_3		
95,00000	100,00000	100,00000	100,00000	1000	1
39,76962	-10,14339	47,81395	-8,05735	1000	21
32,42288	-4,42534	9,95893	-9,73208	1000	52
32,90859	-1,65343	-2,35338	-9,26080	1000	78
27,21874	3,04271	-20,21770	-5,66352	1000	106
30,67126	-1,34522	-3,99804	-8,78859	1000	129
29,24084	-1,64504	-4,83258	-9,39182	1000	152
28,56761	-2,04383	-1,32795	-8,14003	1000	179
27,97647	-2,08624	-1,62326	-8,70653	1000	205
27,86730	-2,11425	-1,57619	-8,86502	1000	228
27,87159	-2,12701	-1,55267	-8,93049	1000	251
27,87191	-2,12546	-1,55363	-8,93470	1000	274
27,87191	-2,12531	-1,55371	-8,93551	1000	297
27,87191	-2,12530	-1,55372	-8,93559	1000	320
27,87191	-2,12530	-1,55372	-8,93559	1000	343

3/Cas test Ariane 1 :

On considère le problème d'optimisation suivant :

$$\min f(x) = m_u + (k_1 + 1)m_{e1} + (k_2 + 1)m_{e2} + (k_3 + 1)m_{e3}$$

$$\text{sous } c(x) = v_1 \ln(M_{i,1}/M_{f,1}) + v_2 \ln(M_{i,2}/M_{f,2}) + v_3 \ln(M_{i,3}/M_{f,3}) - V_p$$

Données en entrée :

```

h=1e-5;
eps = 1e-7;
choix = "bfgs";
lamb = 1000;
x_0 = [4000;3641;1263];
rho=1000;
mu = 1700;
vit_p = 11527;
vit_ejec = [2647.2 ; 2922.4 ; 4344.3];
k = [0.1101 ; 0.1532 ; 0.2154];
born_min = [10000,1000,1000];
born_max = [1000000,900000,100000];

```


Résultats :

```
-----Probleme etagement Ariane-----
pts min =[ 145400.800391 31238.587363 7935.512066 ]
f_value = 208778.588827
cont =[ -0.000000 ]
```

Cas test Ariane 1								
Iterations	Masses ergols			Contrainte	fonction	lambda	coeff_pen(rho)	nb_appel de la fonction
	me1	me2	me3					
1	4000	3641	1263	-6322,738276	11874,2514	1000	1000	1
2	10000	6865,89648	10107,7595	-3899,155909	33003,72267	-15759,27691	1000	10
3	32170,2296	31048,3817	19082,4497	-2037,792042	96409,97509	-242614,7313	1000	25
4	64063,3085	62146,9365	2505,44954	-1852,968778	147529,6494	-700711,5556	1000	40
5	77614,2854	74732,2498	6016,43728	-770,9777657	181353,2265	-120861,2755	1000	55
6	102237,452	95487,299	9419,69872	-201,1705679	236758,4506	-228296,3168	1000	70
7	114945,423	105917,499	10780,7146	-18,59771879	264547,8552	-79868,51205	1000	85
8	116382,924	107090,651	10929,3369	-0,186141897	267677,1386	-2061,171146	1000	100
9	116397,522	107101,259	10930,9566	-1,82E-05	267707,5451	-172,1221966	1000	115
10	116395,795	107073,104	10933,0633	-6,08E-05	267675,7202	-163,4144613	1000	130
11	115751,582	96518,3205	11723,2807	-9,272687087	255749,2341	-202,4250173	1000	149
12	117039,645	87046,2681	12755,0821	-15,40438514	247509,9929	-201,0145114	1000	167
13	121322,178	78839,8776	14196,3812	-13,13788922	244552,1783	-196,109738	1000	182
14	123392,799	78421,1652	14589,857	-0,531963229	246846,1459	-192,9063083	1000	197
15	123669,097	78162,8799	14659,2241	-0,019999049	246939,3185	-192,9516196	1000	212
16	123723,242	78096,8283	14673,6465	-0,000948157	246940,7834	-192,8079936	1000	227
17	123869,216	77905,4313	14712,103	-0,007634368	246928,8497	-192,7474839	1000	245
18	124118,504	77574,3005	14774,9271	-0,023612662	246900,0807	-192,7015038	1000	263
19	124487,454	77078,4177	14862,4352	-0,052420324	246844,1582	-192,6307671	1000	281
20	125012,881	76362,5092	14977,1868	-0,101325311	246741,3181	-192,5756029	1000	299
21	125771,465	75311,0654	15125,3908	-0,190625642	246551,024	-192,2909106	1000	317
22	126872,955	73753,6389	15310,0969	-0,354098452	246202,255	-191,9492255	1000	335
23	128477,844	71432,6879	15527,021	-0,6551992	245570,9718	-191,1460096	1000	353
24	130786,877	68008,4775	15751,7611	-1,197615001	244458,5787	-189,7982842	1000	371
25	134004,915	63094,7386	15922,1919	-2,16375986	242571,5408	-187,2253813	1000	389
26	138169,499	56491,3479	15914,0599	-3,809901228	239569,7314	-183,0047998	1000	407
27	142865,5	48560,2189	15537,3775	-6,333311622	235178,7641	-176,6046404	1000	425
28	147131,884	40173,5096	14554,651	-9,965894638	229048,9183	-168,0652439	1000	443
29	149427,616	32476,0844	12670,7016	-15,78928225	220430,9873	-157,0986994	1000	461
30	148598,804	30135,2613	11273,71	-16,05439555	215113,5822	-142,9761177	1000	480
31	146154,122	29481,2775	9795,85627	-18,74602357	209849,384	-123,3567285	1000	499
32	145399,395	29504,1299	8979,08586	-17,26515585	208045,2122	-119,8775068	1000	518
33	144958,333	30538,4092	7467,60269	-17,69726343	206911,2627	-119,3201322	1000	533
34	145299,084	31016,1656	8006,51114	-2,493605494	208495,4686	-118,6462004	1000	548
35	145461,913	31164,2286	7945,94348	-0,048439143	208773,3581	-117,6397393	1000	563
36	145398,952	31239,9681	7935,51641	-0,003880454	208778,1344	-117,1478195	1000	578
37	145401,065	31238,3455	7935,4995	-1,26E-06	208778,5887	-117,1379954	1000	593
38	145400,8	31238,5874	7935,51207	-4,24E-08	208778,5888	-117,1391239	1000	608
39	145400,8	31238,5874	7935,51207	-4,24E-08	208778,5888	-117,1410333	1000	649

IV/Projet Lanceur :

1/Objectif 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 donnée sur une orbite donnée.

Pour notre étude on prendra $m_u = 2500 \text{ kg}$ qui correspond à la masse du satellite et l'altitude cible $H_c = 300 \text{ km}$.

Le satellite doit tourner autour de la terre à une vitesse spécifique permettant de rester en orbite autour de la terre à une altitude constante. Dans le cas contraire notre satellite s'écraserait à terme sur la terre conséquence du champ gravitationnel terrestre ou serait perdu dans l'espace.

On calcul la vitesse cible du lanceur grâce à l'expression

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

Avec $R_c = R_t + H_c$ et $\mu = 3.986 \cdot 10^{14} \text{ m}^3/\text{s}^2$ la constante gravitationnelle terrestre.

On obtient $V_c = 7726 \text{ m/s}$.

L'optimisation simultanée de la configuration du lanceur et de sa trajectoire étant difficile, on adopte une approche itérative en découplant les deux problèmes. Dans un premier temps nous allons déterminer les masses d'ergols à embarquer pour que le lanceur atteigne une vitesse propulsive donnée. Dans un second temps nous déterminerons la vitesse maximale que peut réellement atteindre le lanceur en prenant en compte les forces extérieures auquel il est soumis tel que : le poids, la traînée.

Nous itérerons alors sur la vitesse propulsive jusqu'à obtenir une vitesse réelle correspondant à la vitesse cible.

2/Problème étagement :

L'objectif du problème d'étagement est de déterminer les masses d'ergols rendant le lanceur le plus léger possible tout en respectant la vitesse propulsive donnée.

a/Résolution par l'algorithme SQP :

On cherche à minimiser la masse du lanceur, on peut formuler le problème comme suit :

$$\begin{aligned} \min \quad & f(x) = m_u + (k_1 + 1)m_{e1} + (k_2 + 1)m_{e2} + (k_3 + 1)m_{e3} \\ \text{sous} \quad & c(x) = v_1 \ln(M_{i,1}/M_{f,1}) + v_2 \ln(M_{i,2}/M_{f,2}) + v_3 \ln(M_{i,3}/M_{f,3}) - V_p \end{aligned}$$

On cherche alors à exprimer la fonction et la contrainte fonction des masses d'ergols

$m = (m_{e1}, m_{e2}, m_{e3})$ et des données m_u , $k = (k_1, k_2, k_3)$.

$$M_{i,3}/M_{f,3} = \frac{m_u + (1 + k_3)m_{e3}}{m_u + k_3 m_{e3}}$$

$$M_{i,2}/M_{f,2} = \frac{(m_u + (1 + k_3)m_{e3} + (1 + k_2)m_{e2})}{(m_u + (1 + k_3)m_{e3} + k_2 m_{e2})}$$

$$M_{i,1}/M_{f,1} = \frac{m_u + (1 + k_3)m_{e3} + (1 + k_2)m_{e2} + (1 + k_1)m_{e1}}{m_u + (1 + k_3)m_{e3} + (1 + k_2)m_{e2} + k_1 m_{e1}}$$

On réinjecte alors les masses avant combustion et après combustion des étages dans l'expression de la contrainte.

On implémente alors la fonction et la contrainte dans Matlab exprimée selon les variables et données ci-dessus.

On notera que l'initialisation de la vitesse propulsive correspond à $V_p = 1.2V_c$.

Données d'entrée :

```
h=1e-5;
eps = 1e-7;
choix = "bfgs";
lamb = 1000;
rho=1000;
x_0 = [4000;3641;1263];
rt = 6378137;
mu = 2500;
vit_p = 9271;
vit_cible = 7726;
vit_ejec = [2647.2 ; 2922.4 ; 4344.3];
k = [0.1101 ; 0.1532 ; 0.2154];
```

Résultats :

```
-----Probleme etagement-----
pts min =[ 53030.080049 19923.737841 8616.085786 ]
masse_fusee = 94816.737006
cont =[ 0.000000 ]
```

b/Résolution analytique :

Pour résoudre le problème analytiquement nous allons utiliser 3 relations :

$$m_{s,j} = k_j m_{e,j}$$

$$M_{f,j} = M_{i,j} - m_{e,j}$$

$$M_{i,j+1} = M_{f,j} - m_{s,j}$$

Avec $M_0 = M_{i,1}$ = masse du lanceur au decollage

Et $m_u = M_{i,4}$ = masse du satellite

Dans un premier temps on cherche à reformuler le problème d'optimisation (PE) de la façon suivante :

$$\min f(x_1, x_2, x_3) \quad \text{Sous} \quad c(x_1, x_2, x_3) = 0$$

$$\text{Avec} \quad f(x_1, x_2, x_3) = -\left(\frac{1+k_1}{x_1} - k_1\right)\left(\frac{1+k_2}{x_2} - k_2\right)\left(\frac{1+k_3}{x_3} - k_3\right)$$

$$C(x_{\{1\}}, x_{\{2\}}, x_{\{3\}}) = v_{\{e1\}} \ln(x_{\{1\}}) + v_{\{e2\}} \ln(x_{\{2\}}) + v_{\{e3\}} \ln(x_{\{3\}}) - V_p$$

Minimiser la masse du lanceur revient à maximiser le critère de performance suivant :

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

C'est-à-dire minimiser $-J$. On a donc $f(x_1, x_2, x_3) = -J$

On pose $x_j = \frac{M_{i,j}}{M_{f,j}}$ et on exprime $\frac{M_{i,j+1}}{M_{i,j}}$ fonction de x_j .

$$\begin{aligned} \frac{M_{i,j+1}}{M_{i,j}} &= \frac{M_{f,j} - k_j m_{ej}}{M_{i,j}} = \frac{1}{x_j} - \frac{k_j m_{ej}}{M_{i,j}} = \frac{1}{x_j} - \frac{k_j (M_{i,j} - M_{f,j})}{M_{i,j}} \\ \frac{1}{x_j} - \frac{k_j (M_{i,j} - M_{f,j})}{M_{i,j}} &= \frac{1}{x_j} - k_j \frac{M_{f,j}}{M_{i,j}} - k_j = \frac{1+k_j}{x_j} - k_j \end{aligned}$$

On a alors $\frac{M_{i,2}}{M_{i,1}} = \frac{1+k_1}{x_1} - k_1$; $\frac{M_{i,3}}{M_{i,2}} = \frac{1+k_2}{x_2} - k_2$; $\frac{M_{i,4}}{M_{i,3}} = \frac{1+k_3}{x_3} - k_3$

On a donc

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

Maintenant nous écrivons les conditions nécessaires d'optimalité d'ordre 1 (conditions KKT) :

$$\nabla f(x) + \lambda \nabla c(x) = 0$$

$$c(x) = 0$$

La fonction étant une fonction de \mathbb{R}^3 dans \mathbb{R} nous aurons 3 équations liées au gradient du lagrangien.

Et étant donné que nous avons une seule contrainte, nous aurons un seul multiplicateur de Lagrange.

On écrit alors le gradient du lagrangien :

$$\frac{\partial f(x)}{\partial x_1}(x) + \lambda \frac{\partial c(x)}{\partial x_1}(x) = \left(\frac{1+k_1}{x_1^2}\right)\left(\frac{1+k_2}{x_2} - k_2\right)\left(\frac{1+k_3}{x_3} - k_3\right) + \lambda \frac{v_{e1}}{x_1} = 0$$

$$\frac{\partial f(x)}{\partial x_2}(x) + \lambda \frac{\partial c(x)}{\partial x_2}(x) = \left(\frac{1+k_1}{x_1} - k_1\right)\left(\frac{1+k_2}{x_2^2} - k_2\right)\left(\frac{1+k_3}{x_3} - k_3\right) + \lambda \frac{v_{e2}}{x_2} = 0$$

$$\frac{\partial f(x)}{\partial x_3}(x) + \lambda \frac{\partial c(x)}{\partial x_3}(x) = \left(\frac{1+k_1}{x_1} - k_1\right)\left(\frac{1+k_2}{x_2} - k_2\right)\left(\frac{1+k_3}{x_3^2} - k_3\right) + \lambda \frac{v_{e3}}{x_3} = 0$$

On pose $\Omega_j = \frac{k_j}{1+k_j}$

$$\frac{1}{k_1 k_2 k_3} \left(\frac{\partial f(x)}{\partial x_1}(x) + \lambda \frac{\partial c(x)}{\partial x_1}(x) \right) = \left(\frac{1+k_1}{x_1^2 k_1} \right) \left(\frac{1+k_2}{x_2 k_2} - 1 \right) \left(\frac{1+k_3}{x_3 k_3} - 1 \right) + \lambda \frac{v_{e1}}{x_1 k_1 k_2 k_3} = 0$$

$$\frac{1}{k_1 k_2 k_3} \left(\frac{\partial f(x)}{\partial x_2}(x) + \lambda \frac{\partial c(x)}{\partial x_2}(x) \right) = \left(\frac{1+k_1}{x_1 k_1} - 1 \right) \left(\frac{1+k_2}{x_2^2 k_2} - 1 \right) \left(\frac{1+k_3}{x_3 k_3} - 1 \right) + \lambda \frac{v_{e2}}{x_2 k_1 k_2 k_3} = 0$$

$$\frac{1}{k_1 k_2 k_3} \left(\frac{\partial f(x)}{\partial x_3}(x) + \lambda \frac{\partial c(x)}{\partial x_3}(x) \right) = \left(\frac{1+k_1}{x_1 k_1} - 1 \right) \left(\frac{1+k_2}{x_2 k_2} - 1 \right) \left(\frac{1+k_3}{x_3^2 k_3} - 1 \right) + \lambda \frac{v_{e3}}{x_3 k_1 k_2 k_3} = 0$$

On identifie $\Omega_1 \Omega_2 \Omega_3$ et on obtient les relations suivantes.

$$\left(\frac{1}{\Omega_1 x_1^2} \right) \left(\frac{1}{\Omega_2 x_2} - 1 \right) \left(\frac{1}{\Omega_3 x_3} - 1 \right) + \lambda \frac{v_{e1}}{x_1 k_1 k_2 k_3} = 0$$

$$\left(\frac{1}{\Omega_1 x_1} - 1 \right) \left(\frac{1}{\Omega_2 x_2^2} \right) \left(\frac{1}{\Omega_3 x_3} - 1 \right) + \lambda \frac{v_{e2}}{x_2 k_1 k_2 k_3} = 0$$

$$\left(\frac{1}{\Omega_1 x_1} - 1 \right) \left(\frac{1}{\Omega_2 x_2} - 1 \right) \left(\frac{1}{\Omega_3 x_3^2} \right) + \lambda \frac{v_{e3}}{x_3 k_1 k_2 k_3} = 0$$

$$\frac{1}{v_{e1}} \left(\frac{1}{\Omega_1 x_1} \right) \left(\frac{1}{\Omega_2 x_2} - 1 \right) \left(\frac{1}{\Omega_3 x_3} - 1 \right) = -\lambda \frac{1}{k_1 k_2 k_3}$$

$$\frac{1}{v_{e2}} \left(\frac{1}{\Omega_1 x_1} - 1 \right) \left(\frac{1}{\Omega_2 x_2} \right) \left(\frac{1}{\Omega_3 x_3} - 1 \right) = -\lambda \frac{1}{k_1 k_2 k_3}$$

$$\frac{1}{v_{e3}} \left(\frac{1}{\Omega_1 x_1} - 1 \right) \left(\frac{1}{\Omega_2 x_2} - 1 \right) \left(\frac{1}{\Omega_3 x_3} \right) = -\lambda \frac{1}{k_1 k_2 k_3}$$

Et $-\lambda \frac{1}{k_{e1} k_{e2} k_{e3}}$ est une constante car $k = (k_1, k_2, k_3)$ sont des constantes et le multiplicateur de Lagrange est identique pour les 3 équations.

On a alors :

$$\frac{1}{v_{e1}} \left(\frac{1}{\Omega_1 x_1} \right) \left(\frac{1}{\Omega_2 x_2} - 1 \right) \left(\frac{1}{\Omega_3 x_3} - 1 \right) = \frac{1}{v_{e2}} \left(\frac{1}{\Omega_1 x_1} - 1 \right) \left(\frac{1}{\Omega_2 x_2} \right) \left(\frac{1}{\Omega_3 x_3} - 1 \right)$$

$$\frac{1}{v_{e2}} \left(\frac{1}{\Omega_1 x_1} - 1 \right) \left(\frac{1}{\Omega_2 x_2} \right) \left(\frac{1}{\Omega_3 x_3} - 1 \right) = \frac{1}{v_{e3}} \left(\frac{1}{\Omega_1 x_1} - 1 \right) \left(\frac{1}{\Omega_2 x_2} - 1 \right) \left(\frac{1}{\Omega_3 x_3} \right)$$

$$\frac{1}{v_{e3}} \left(\frac{1}{\Omega_1 x_1} - 1 \right) \left(\frac{1}{\Omega_2 x_2} - 1 \right) \left(\frac{1}{\Omega_3 x_3} \right) = \frac{1}{v_{e1}} \left(\frac{1}{\Omega_1 x_1} \right) \left(\frac{1}{\Omega_2 x_2} - 1 \right) \left(\frac{1}{\Omega_3 x_3} - 1 \right)$$

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

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

$$\frac{1}{v_{e3}} \left(\frac{1}{\Omega_1 x_1} - 1 \right) \left(\frac{1}{\Omega_3 x_3} \right) = \frac{1}{v_{e1}} \left(\frac{1}{\Omega_1 x_1} \right) \left(\frac{1}{\Omega_3 x_3} - 1 \right)$$

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

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

$$v_{e3}(1 + \Omega_3 x_3) = v_{e1}(1 - \Omega_1 x_1)$$

On a alors $v_{e1}(1 - \Omega_1 x_1) = v_{e2}(1 - \Omega_2 x_2) = v_{e3}(1 + \Omega_3 x_3) = -\lambda \frac{1}{k_1 k_2 k_3} = Cste$

Et donc $v_{ej}(1 + \Omega_j x_j) = Cste$

On cherche à se ramener à un problème à une équation à une inconnue, on choisit comme inconnue x_3 et on exprime x_1 et x_2 en fonction de x_3 .

On déduit des relations $v_{e2}(1 - \Omega_2 x_2) = v_{e3}(1 + \Omega_3 x_3)$ et $v_{e3}(1 + \Omega_3 x_3) = v_{e1}(1 - \Omega_1 x_1)$

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

$$x_1 = \frac{v_{e1} - v_{e3}(1 - \Omega_3 x_3)}{\Omega_1 v_{e1}}$$

On exprime alors la contrainte en fonction de x_3

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

On résout alors l'équation $c(x_3) = 0$ à l'aide de l'algorithme de la sécante :

Algorithme 2.3 : L'algorithme de la sécante
<p>Données : La fonction $g(x)$, tolérance ε</p> <p>Résultat : x^* tel que $g(x^*) = 0$</p> <p>Initialisation : $k = 0$, x_0 approximation initiale de $g(x) = 0$. a_0 approximation initiale de $g'(x_0)$ ($=1$ par défaut)</p> <p>tant que $g(x_k) > \varepsilon$ faire</p> <div style="margin-left: 20px;"> $x_{k+1} = x_k - \frac{g(x_k)}{a_k}$ $a_{k+1} = \frac{g(x_k) - g(x_{k+1})}{x_k - x_{k+1}}$ $k \leftarrow k + 1$ </div> <p>fin</p> <p>$x^* \leftarrow x_k$</p>

Nous avons pris soin lors de la programmation de l'algorithme de borner la solution à chaque itération car la présence de logarithmes dans l'équation peut amener à itérer sur des x complexes.

Nous avons créé une fonction « Newton.m » dans Matlab et nous avons écrit un fichier « main_newton .m » pour différencier la partie résolution SQP et la partie analytique.

En sortie de l'algorithme on obtient x_3 :

```
-----Probleme etagement-----
x_3 =[ 2.978024 ]
f_value = 0.000000
iteration = 8.000000
```

On peut alors en déduire x_1 et x_2 grâce à leurs expression fonction de x_3 .

```
-----Probleme etagement-----
x_1 =[ 2.269067 ]
x_2 =[ 2.243346 ]
x_3 =[ 2.978024 ]
f_value = 0.000000
iteration = 8.000000
```

Nous avons automatiquement calculé x_1 et x_2 lorsque le x_3 optimale est trouvé.

A partir de $x = (x_1, x_2, x_3)$ on cherche à déduire les masses d'ergols et la masse totale du lanceur.

On obtient alors :

$$m_3 = \frac{m_u}{1 - k_3(x_3 - 1)}(x_3 - 1)$$

$$m_2 = \frac{m_u + (k_3 + 1)m_3}{1 - k_2(x_2 - 1)}(x_2 - 1)$$

$$m_1 = \frac{m_u + (k_3 + 1)m_3 + (k_2 + 1)m_2}{1 - k_1(x_1 - 1)}(x_1 - 1)$$

Données d'entrée :

```
h=1e-5;
eps = 1e-7;
x_0 = [4];
mu = 2500;
vit_p = 9271;
vit_cible = 7726;
vit_ejec = [2647.2 ; 2922.4 ; 4344.3];
k = [0.1101 ; 0.1532 ; 0.2154];
Omega=(k./([1;1;1]+k));
```

On obtient alors les résultats suivants pour l'algorithme de la sécante.

```
-----Probleme etagement-----
x_1 =[ 2.269067 ]
x_2 =[ 2.243346 ]
x_3 =[ 2.978024 ]
f_value = 0.000000
iteration = 8.000000
masses ergol =[ 53030.076960 19923.746400 8616.080487 ]
masse_fusee = 94816.737006
```

On les compare à ceux obtenus ci-dessous par le SQP.

```
-----Probleme etagement-----
pts min =[ 53030.080049 19923.737841 8616.085786 ]
masse_fusee = 94816.737006
cont =[ 0.000000 ]
```

On observe que les résultats sont très proches. Compte tenu de l'ordre de grandeur des masses d'ergol et du lanceur, on peut conclure que les résultats sont équivalents.

3/Trajectoire du lanceur :

a/Modélisation de la trajectoire du lanceur :

Pour étudier la trajectoire on a considéré le lanceur comme étant un point matériel avec des coordonnées de position et vitesses.

Afin de modéliser la trajectoire il nous est donné les équations du mouvement :

$$\begin{aligned}\dot{\vec{R}} &= \vec{V} \\ \dot{\vec{V}} &= \frac{\vec{T} + \vec{W} + \vec{D}}{M} \\ \dot{M} &= -q\end{aligned}$$

Avec \vec{T} la poussée, \vec{W} le poids et \vec{D} la traînée.

En exprimant la poussée, le poids, la traînée ainsi que le débit massique « -q » avec expressions fournies pages 8 et 9 de l'énoncé, nous trouvons l'équation différentielle du mouvement.

Cette équation différentielle est spécifiée dans la fonction Matlab « phi.m ».

Cette équation est une équation différentielle ordinaire que nous intégrons numériquement pour chaque étage de la fusée à l'aide de l'intégrateur de Matlab Ode45.

Pour cela avons spécifier la fonction à intégrer (phi) , le domaine d'intégration $[t_0 \ t_f]$, les conditions initiales y_0 et enfin en option le pas de temps .

Cet intégrateur numérique nous renvoie alors un vecteur t et un vecteur Y. Le vecteur Y étant composé des coordonnées de la position, vitesse ainsi que de la masse : $Y = [x, y, \dot{x}, \dot{y}, M]$

b/Formulation du problème de trajectoire :

On cherche à atteindre l'altitude de l'orbite en maximisant la vitesse fournie au satellite

On peut alors formuler le problème de la manière suivante (Problème PT) :

$$\begin{aligned}\max \quad & V(t_f) \\ \text{sous} \quad & R(t_f) = R_c \text{ et } \vec{R}(t_f)\vec{V}(t_f) = 0\end{aligned}$$

Les variables sont les angles $\theta_0, \theta_1, \theta_2, \theta_3$ qui orientent la vitesse initiale et la poussée du lanceur pour chaque étage.

La résolution de ce problème permet de connaître la vitesse maximale V_r que nous pouvons atteindre avec les masses d'ergols embarquées et les angles $\theta_0, \theta_1, \theta_2, \theta_3$ optimaux.

Si cette vitesse est inférieure ou supérieure à la vitesse cible, alors nous revenons au problème d'étagement en augmentant ou diminuant respectivement la vitesse propulsive à atteindre.

On incrémente la vitesse propulsive en faisant :

```
delta_v = vit_cible-vitesse_finale; (avec vitesse final = vitesse maximale)
```

```
vit_p = vit_p + delta_v; (vit_p = vitesse propulsive)
```

Le delta v pouvant être ou positif ou négatif.

Pour une seule itération avec $V_c = 7726$ m/s et $V_p = 1.2V_c$

Données en entrée :

Résultats :

```
theta_0 = [10;5;7;-5];
born_min = [0,-10,-50,-50];
born_max = [20,30,50,50];
```

```
-----Probleme trajectoire-----
pts min =[ 10.311764 4.403301 7.597861 3.518411 ]
f_value = 7513.403770
cont =[ -0.000031 -2.044323 ]
```

Avec f_value qui correspond à la vitesse maximale et pts min correspondant aux angles.

On observe que la vitesse maximale obtenue lors de la première itération est inférieure à la vitesse cible.

4/Résolution finale du problème :

Récapitulatif des données d'entrée :

```
h=1e-5;
eps = 1e-7;
choix = "bfgs";
lamb = 1000;
rho=1000;
x_0 = [4000;3641;1263];
rt = 6378137;
H_c=300000;
mu = 2500;
vit_p = 9271;
vit_cible = 7726;
vit_ejec = [2647.2 ; 2922.4 ; 4344.3];
k = [0.1101 ; 0.1532 ; 0.2154];
theta_0 = [10;5;7;-5];
born_min_masses = [45000,10000,3000];
born_max_masses = [750000,90000,25000];
born_min_theta = [0,-10,-50,-50];
born_max_theta = [20,30,50,50];
```

Nous avons défini pour ce problème une précision de 0.01m/s. lorsque nous itérons sur la vitesse propulsive nous obtenons alors les résultats suivants.

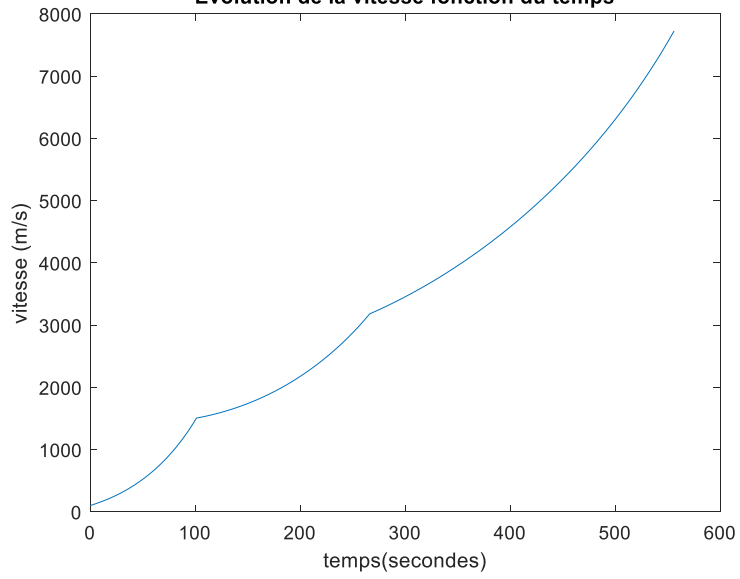
```
-----Probleme etagement-----
pts min =[ 59049.740914 21207.110924 8796.285089 ]
masse_fusee = 103198.162604
cont =[ -0.000000 ]
-----Probleme trajectoire-----
pts min =[ 10.341581 4.338822 7.515230 3.097091 ]
f_value = 7726.009172
cont =[ -0.000017 -0.024853 ]
iteration = 6.000000
```

Pour le problème de trajectoire et d'étagement nous observons bien que les contraintes sont proches de 0.

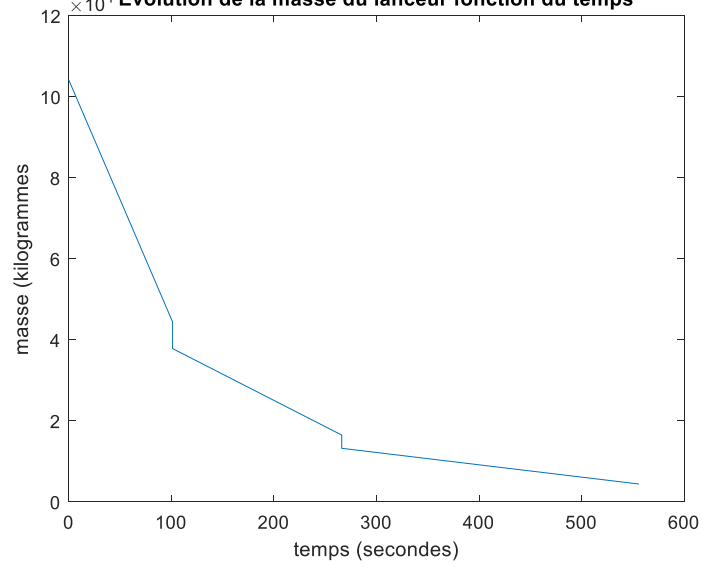
Afin de consulter l'historique des valeurs correspondant aux variables, aux fonctions et aux contraintes, nous avons noté « hist_ » suivie du nom, leur historique respectif. Ces valeurs sont accessibles dans le Workspace.

Graphiques :

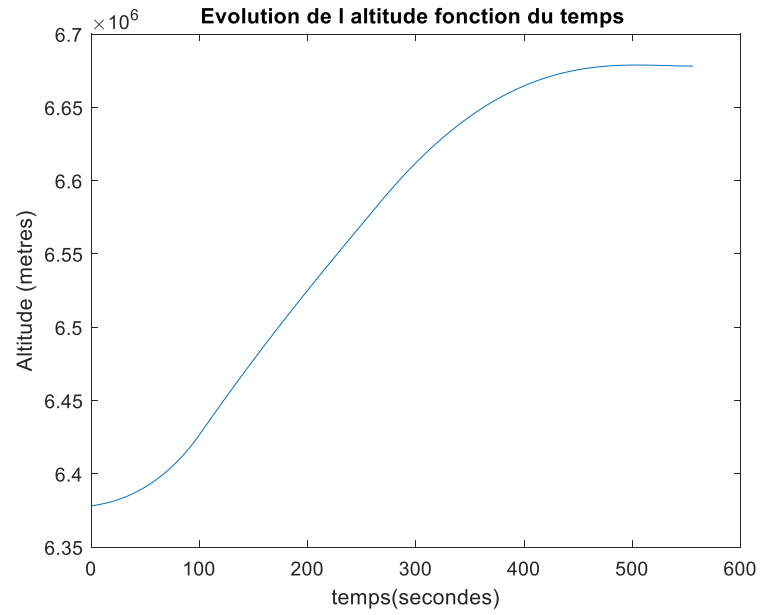
Evolution de la vitesse fonction du temps



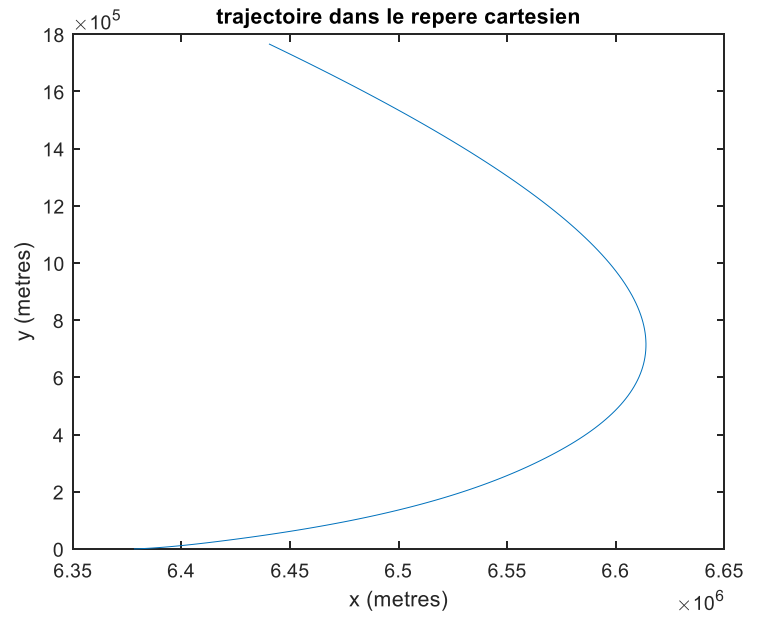
Evolution de la masse du lanceur fonction du temps

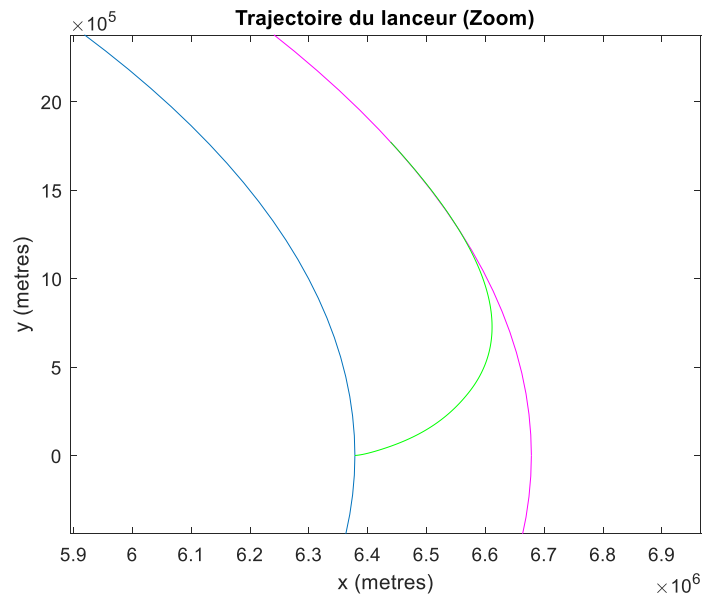
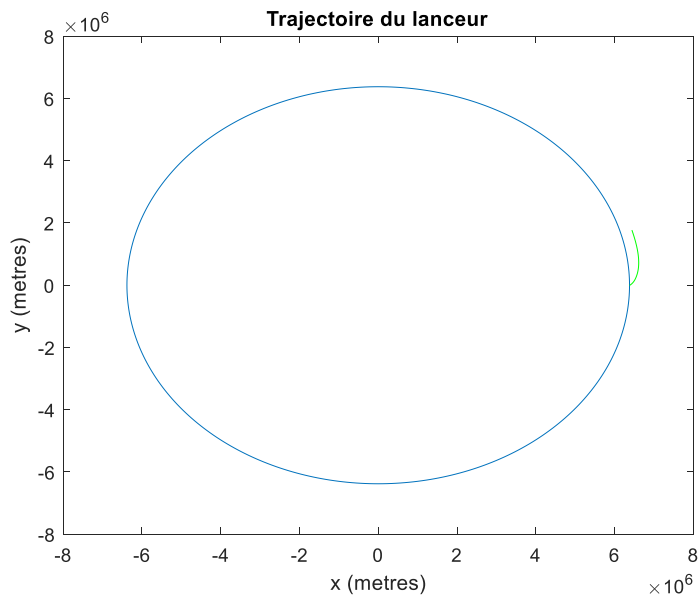


Evolution de l'altitude fonction du temps

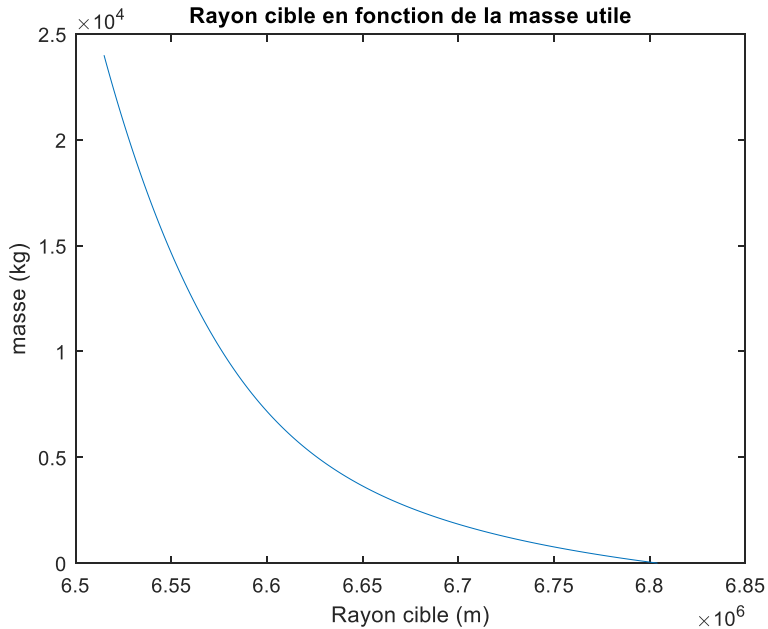


trajectoire dans le repere cartésien





Question Bonus :



On fixe les masses d'ergols et les angles.
On a donc comme variable la masse utile pour la fonction calculant la masse du lanceur et la masse de la fusée comme argument d'entrée qui change à chaque itération sur la masse utile.

En se servant de la première contrainte du problème de trajectoire on détermine le rayon cible.

On observe que la masse utile décroît exponentiellement avec le rayon cible.

Le programme est implémenté sous forme d'une fonction « bonus.m » appelé dans le fichier principal « M_main.m ».

Conclusion :

Grace à l'algorithme SQP nous avons pu mettre en application un algorithme d'optimisation sur un cas concret. Nous avons ainsi pu modéliser le lanceur à partir de l'équation de la dynamique et aborder la résolution d'une équation différentielle à partir d'un intégrateur numérique (ode45 dans notre cas).

D'autre part l'utilisation d'une méthode itérative et la décomposition d'un problème globale en plusieurs problèmes nous a montré une approche permettant de résoudre des problèmes complexes.

Cependant la décomposition fait apparaître la nécessité de choisir des précisions élevées et de bien prendre en compte les approximations de calcul. Dans le cas d'un calcul réel de trajectoire, l'addition des erreurs de calculs et la simplification d'équations par des hypothèses peut conduire à un résultat éloigné de la réalité.