

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE MOULOUD MAMMERI, TIZI-OUZOU

FACULTE DES SCIENCES

DEPARTEMENT DE MATHEMATIQUES

THESE DE MASTER

SPECIALITE : RECHERCHE OPERATIONNELLE

OPTION : AIDE A LA DECISION

Présenté par :

M^{elle} **TERKMANI RIMA**

M^{elle} **NAIT ABDESSELAM LOUIZA**

Sujet :

Contrôle optimal non linéaire Application : Décollage d'une fusée

Devant le jury d'examen composé de :

<i>M^{me}</i> Kahina Louadj ;	M.C.B ;	U.M.M.T.O ;	Présidente
<i>M^{me}</i> Nassima Moussouni ;	M.C.B ;	U.M.M.T.O ;	Rapportrice
<i>M^{me}</i> Nacéra Zidelmal ;	M.C.B ;	U.M.M.T.O ;	Examinatrice
<i>M^{elle}</i> Farida Kara ;	M.A.A ;	U.M.M.T.O ;	Examinatrice

Soutenu le : 27/06/2011

✱ Remerciements ✱

Nous tenons à exprimer ici toute notre reconnaissance à M^{me} N.MOUSSOUNI pour l'honneur qu'elle nous a fait en assurant la direction et le suivi scientifique et technique du présent mémoire. On la remercie pour sa grande contribution à l'aboutissement de ce travail, et pour sa disponibilité malgré ses nombreuses activités acceptant de diriger ce travail, ainsi que sa disponibilité et l'intérêt qu'elle lui a porté, malgré ses nombreuses préoccupations. Qu'elle trouve ici notre profonde gratitude.

Nous remercions vivement M^{me} K.LOUADJ maître de conférence pour l'honneur qu'elle nous a fait en acceptant de présider le jury de ce mémoire.

Nos remerciements chaleureux s'adressent également à M^{me} N.ZIDELMAL et M^{elle} F.KARA pour avoir accepté d'examiner ce travail.

On ne peut oublier de remercier tous les membres de nos familles pour leurs soutiens et leurs encouragements particulièrement nos parents. On remercie aussi tout ceux qui nous ont aidé tout au long de notre travail spécialement à M^{me} A.NAIT ABDESSELAM et M^r K.MERAKEB .

~~~~~«««« *Dédicaces* »»»»~~~~~

Je dédie ce modeste travail à :

- * Mes chers parents Rachid et Farida ;
- * Mes frères Sofiane, Essaid et Yacine ;
- * Mes soeurs Sonia et Silya ;
- * Mon grand père et yemma Rosa que j'aime beaucoup ;
- * Mes oncles Mouloud, Farid et leurs familles ;
- * Ma tante Rachida, son époux Arezki et leurs enfants Leatitia, Rayan et Maylis ;
- * Mon cher et aimable ami Brahim ;
- * Mes copines Driffa et Ourida ;
- * Mes amis.

~~~~~«« *TERKMANI RIMA* »»~~~~~

## ~~~~~«««« *Dédicaces* »»»»~~~~~

*Je dédie ce modeste travail à :*

- \* Mes chers parents ;
- \* Mes frères *Arezki, Yazid, Amerane* et *Abdeslam* ;
- \* Ma soeur *Aldjia* et son mari *Rezki* et leurs petit *Beka* ;
- \* Ma soeur *Kahina* et son mari *Rabah* ;
- \* Ma soeur *Ounissa* et son mari *Farhat* ;
- \* Mon fiancé *Brahim* et toute sa famille ;
- \* Mes copines (*Hayet, Katia, Samira*) ;
- \* Mes amies (*Djamila, Zaina, Myriem, Kahina et Kahina*) ;

~~~~~«««*NAIT ABDESSELAM LOUIZA*»»»~~~~~

Table des matières

| | |
|--------------------------------------------------------------------|-----------|
| Table des matières | 1 |
| Introduction générale | 3 |
| 1 Equations différentielles | 5 |
| 1.1 Introduction | 5 |
| 1.2 Equations différentielles ordinaire du premier ordre | 5 |
| 1.3 Equation différentielle et problème de Cauchy | 6 |
| 1.3.1 Théorème de Cauchy-Lipschitz | 6 |
| 1.3.2 Théorème des fonctions implicites | 7 |
| 1.4 Méthodes de résolution des équations différentielles | 7 |
| 1.4.1 Méthode d'Euler | 8 |
| 1.4.2 Méthode de Runge Kutta d'ordre 2 | 9 |
| 1.5 Programmation des méthodes | 10 |
| 1.5.1 Méthode d'Euler fait sur Matlab | 11 |
| 1.5.2 Méthode de Runge Kutta d'ordre 2 | 13 |
| 1.6 Résolution de fonction $f(x)=0$: | 16 |
| 1.6.1 Méthode de Newton | 16 |
| 2 Introduction au contrôle optimal | 20 |
| 2.1 Théorie du contrôle | 20 |
| 2.2 Quelques définitions | 20 |
| 2.3 Position du problème | 21 |
| 2.3.1 Problème de Lagrange | 22 |
| 2.3.2 problème de Mayer | 22 |
| 2.4 Contrôlabilité | 23 |
| 2.4.1 Cas des systèmes linéaires | 23 |
| 2.4.2 Systèmes linéaires autonomes | 24 |

| | | |
|----------|-------------------------------------------------------|-----------|
| 2.4.3 | Systèmes linéaires non autonomes | 24 |
| 2.4.4 | Cas des systèmes non linéaires | 25 |
| 2.4.5 | Application entrée sortie | 25 |
| 2.5 | Méthodes de résolution | 28 |
| 2.5.1 | Méthodes directes | 28 |
| 2.5.2 | Méthodes indirectes | 29 |
| 2.6 | Principe du maximum de Pontryagin | 30 |
| 3 | Exemples d'applications | 33 |
| 3.1 | Introduction | 33 |
| 3.2 | Rappels sur les lois de la physique(Newton) | 34 |
| 3.2.1 | La première loi de Newton | 34 |
| 3.2.2 | La deuxième loi de Newton | 34 |
| 3.2.3 | Contrôlabilité | 36 |
| 3.2.4 | Résolution théorique | 37 |
| 3.2.5 | méthode de tir : tir simple | 41 |
| 3.2.6 | Application de la méthode directe | 46 |
| 3.3 | Position du problème | 49 |
| 3.3.1 | Résolution théorique | 50 |
| 3.3.2 | Résolution numérique | 52 |
| | Conclusion générale | 56 |
| | Bibliographie | 57 |

Introduction générale

Du point de vue mathématique, un système de contrôle est un système dynamique dépendant d'un paramètre appelé le contrôle, pour le modéliser, on peut avoir recours à des équations différentielles, intégrales, fonctionnelles, aux différences finies, aux dérivées partielles, stochastiques, ... Pour cette raison la théorie du contrôle est à l'interconnexion de nombreux domaines des mathématiques, elle s'inscrit dans la continuité du calcul des variations c'est à dire elle s'intéresse à l'analyse des propriétés des systèmes commandés (dynamiques) sur lesquels on peut agir en moyen d'une commande.

Les systèmes abordés sont multiples, système avec bruit, avec retard, ..., leurs origines sont très diverses : mécanique, électricité, électronique, biologie, économie, ... L'objectif peut être de stabiliser le système pour le rendre insensible à certaine perturbation (stabilisation) ou encore de déterminer des solutions optimales pour un certain critère d'optimisation.

Le problème de contrôle optimal se décompose en deux parties : Pour déterminer une trajectoire optimale joignant un ensemble initial à une cible, il faut savoir d'abord si cette cible est atteignable, c'est le problème de contrôlabilité, il existe une caractérisation très simple de la contrôlabilité, apparue dans les années soixante avec les travaux de R.E.Kalman[5]. Pour les systèmes non linéaires, le problème de contrôlabilité est beaucoup plus difficile.

Une fois le problème de contrôlabilité est résolu, il faut chercher parmi toutes les trajectoires possibles celle qui donne le coût minimum (maximum) et pour ce faire, on dispose de deux grandes classes des méthodes à savoir la méthode directe et la méthode indirecte.

Le présent travail est donc devisé en trois parties organisées de la manière suivante :

Partie1 : Equations différentielles.

Partie2 : Introduction au contrôle optimal.

Partie3 : Contrôle optimal d'une fusée.

Dans la première partie intitulé équations différentielles, nous nous sommes procédés à la définition des équations différentielles du premier ordre, on a défini quelques méthodes de résolution des équations différentielles du premier ordre les plus utiles pour notre thème par exemple la méthode d'Euler et la méthode de Runge Kutta et on termine le chapitre avec un exemple résolu avec les deux méthodes cités auparavant ainsi la comparaison entre ces deux méthodes avec la solution exacte.

Dans la deuxième partie intitulé introduction au contrôle optimal, nous nous intéressons au problème de contrôle optimal et nous présentons ici quelques notions de base d'un problème de contrôle optimal, la contrôlabilité des systèmes linéaires et non linéaires ainsi les méthodes de résolution.

Le dernière partie est consacré à l'application des méthodes de résolution comme la méthode directe et la méthode de tir simple, on s'est intéresser dans cette partie à étudier le décollage d'une fusée qui se déplace d'une position initiale à une position finale arbitraire sans tenir compte des forces extérieures en minimisant le temps. Par la suite, nous avons modélisé le même problème en tenant compte de la variation de la masse de la fusée en fonction du temps. Notre but dans cette partie est de calculer le contrôle optimal (poussée) permettant de maximiser l'altitude finale. Cette modélisation nous a conduit à un problème de contrôle optimal avec contrainte sur l'état.

Chapitre 1

Equations différentielles

Ce chapitre a pour objectif de présenter quelques rappels essentielles de résolution d'équations différentielles en se rappelant de la version générale du théorème de Cauchy-Lipschitz, adaptée à la théorie de contrôle, qui établit sous certaines conditions l'existence et l'unicité d'une solution d'une équation différentielle.

1.1 Introduction

Les équations différentielles jouent un rôle important dans différents domaines, en physique, en chimie, en biologie, Ces équations ont rarement des solutions exactes, autrement dit, la résolution de ces équations différentielles n'est pas toujours possible. Ces équations ne s'intègrent pas d'une manière exacte. On fait appel dans ce cas à des méthodes numériques afin de trouver des solutions approchées.

1.2 Equations différentielles ordinaire du premier ordre

Définition 1.1. On appelle équation différentielle ordinaire du premier ordre, une équation de type :

$$a(x)y' + b(x)y = c(x)$$

ou a, b, c des fonctions avec $a(x) \neq 0 \forall x \in I \subset \mathbb{R}$, I étant un intervalle sur lequel a, b, c sont définies. Une fonction f est solution de cette équation sur I , lorsque f est définie et dérivable sur I et :

$$\forall x \in I : a(x)f'(x) + b(x)f(x) = c(x).$$

1.3 Equation différentielle et problème de Cauchy

On s'intéresse aux équations différentielles du premier ordre de la forme :

$$y'(t) = f(t, y(t))$$

Avec $f : I \times \mathbb{R}^p \rightarrow \mathbb{R}^p$ (I intervalle de \mathbb{R}) une fonction continue si $p > 1$, il s'agit en pratique d'un système différentiel.

Le problème avec condition initiale est appelé problème de Cauchy défini comme suit :

$$\begin{cases} y'(t) = f(t, y(t)) \\ y(t_0) = y_0, \quad t_0 \in I, \quad y_0 \in \mathbb{R}^p \end{cases} .$$

1.3.1 Théorème de Cauchy-Lipschitz

Théorème 1.1. [4] *Considérons le problème de Cauchy suivant :*

$$\begin{cases} y'(t) = f(t, y(t)) \\ y(t_0) = y_0, \quad t_0 \in I, \quad y_0 \in \mathbb{R}^p \end{cases} \quad (1.1)$$

avec $f : (t, y) \in I \times \mathbb{R}^p \rightarrow f(t, y) \in \mathbb{R}^p$.

– f est continue sur $I \times \mathbb{R}^p$.

– f est lipschitzienne en y , localement en y c'est à dire :

$\exists L > 0$ telle que :

$$\forall t \in I, \forall y_1, y_2 \in v_{\mathbb{R}^p}(y_0) : \|f(t, y_1) - f(t, y_2)\| \leq L\|y_1 - y_2\|.$$

Alors le problème de Cauchy (1.1) possède une unique solution. Cette solution est définie sur un intervalle contenant t .

Définition 1.2. On définit l'exponentielle de la matrice A (dans le cas où A est nilpotante) par :

$$\exp(A) = e^A = \sum_{k=0}^n \frac{A^k}{k!}$$

avec $\exp(0) = I_n$.

I_n : Désigne la matrice identité d'ordre n .

Définition 1.3. Considérons le problème de Cauchy dans \mathbb{R}^n

$$\begin{cases} \dot{x}(t) = A(t)x(t) + B(t), \\ x(t_0) = 0, \quad t_0 \in [0, T] \end{cases} \quad (1.2)$$

où les applications $t \mapsto A(t) \in M_n(\mathbb{R})$ et $t \mapsto B(t) \in \mathbb{R}^n$ sont localement intégrables sur l'intervalle I considéré.

1.3.2 Théorème des fonctions implicites

Nous nous intéresserons dans ce travail au théorème de la submersion :

Théorème 1.2. [3] Soient E et F deux espaces de Banach, $f : E \rightarrow F$ une application de classe C^1 , soit $x_0 \in E$. Supposons que $df(x_0) : E \rightarrow F$ est linéaire continue, surjective ; alors f est localement surjective au voisinage de x_0 . Autrement dit

$$\exists v \in v(f(x_0)); \exists w \in v(x_0) / \forall y \in v; \exists x \in w : y = f(x)$$

Définition 1.4. (Différentiabilité au sens de Fréchet) Soit V et W deux espaces vectoriels normés. Une application $f(x)$ de V dans W est différentiable (au sens de Fréchet) en $x \in V$ si il existe une application linéaire continue de V dans W , notée $Df(x)$ telle que

$$\forall h \in V; f(x+h) = f(x) + Df(x).h + \epsilon(h)||h||$$

Avec $\lim_{h \rightarrow 0} ||\epsilon(h)|| = 0$.

Définition 1.5. (ensemble compact dans \mathbb{R}) L'ensemble compact dans \mathbb{R} est un ensemble fermé borné.

1.4 Méthodes de résolution des équations différentielles

On s'intéresse dans notre travail à la résolution du problème de Cauchy définit précédemment :

1.4.1 Méthode d'Euler

Considérons le problème :

$$\begin{cases} y'(t) = f(t, y(t)) \\ y(0) = y_0, \quad t \in [0, T] \end{cases} \quad (1.3)$$

Nous donnerons une subdivision comme suit :

$$0 = t_0 < t_1 < \dots < t_n < t_{n+1} < \dots < t_N = T$$

et posons :

$$h_n = t_{n+1} - t_n; \quad h = \max h_n; \quad 0 \leq n \leq N$$

La solution de (1.3) vérifie pour $0 \leq n \leq N$:

$$y(t_{n+1}) = y(t_n) + \int_{t_n}^{t_{n+1}} f(s, y(s)) ds$$

On construit par récurrence une approximation y_n et $y(t_n)$ en remplaçant la relation précédente par :

$$y_{n+1} = y_n + hf(t_n, y_n); \quad n = 0, 1, \dots, N-1 \quad (1.4)$$

Ce qui revient à approcher, pour $s \in]t_n, t_{n+1}[$, $f(s, y(s))$ par $f(t_n, y_n)$.

Le schéma défini par (1.4) s'appelle schéma d'Euler.

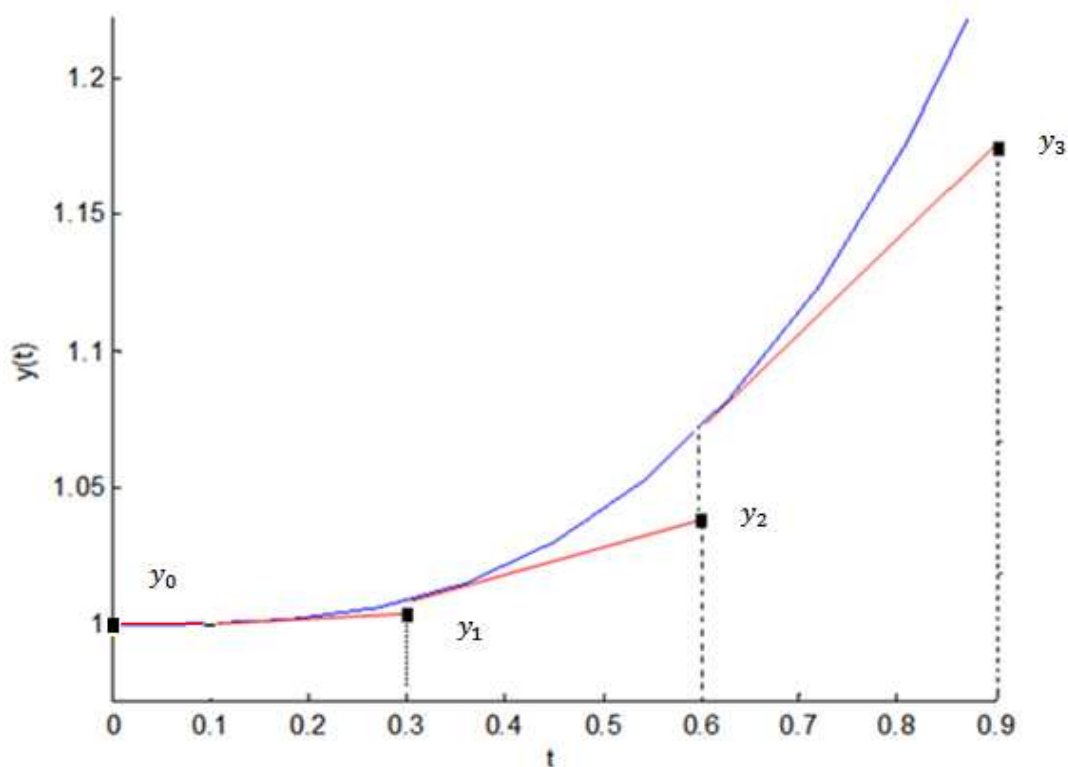


FIG. 1.1 – Méthode d'Euler

1.4.2 Méthode de Runge Kutta d'ordre 2

On considère le problème de Cauchy :

$$\begin{cases} y'(t) = f(t, y(t)) \\ y(0) = y_0, \quad t \in I = [0, T] \end{cases}$$

On cherche à discrétiser ce problème par rapport à une subdivision :

$$0 = t_0 < t_1 < \dots < t_n < t_{n+1} < \dots < t_N = T$$

L'idée est de calculer par récurrence les points (t_n, y_n)

posons :

$$h_n = t_{n+1} - t_n; \quad h = \max h_n; \quad 0 \leq n \leq N$$

La méthode de Runge Kutta correspondante est définie par l'algorithme suivant :

$$\begin{cases} K_1 = h \times f(t_{n-1}, y_{n-1}) \\ K_2 = h \times f(t_{n-1} + h/2, y_{n-1} + K_1/2) \\ y_n = y_{n-1} + K_2 \end{cases}$$

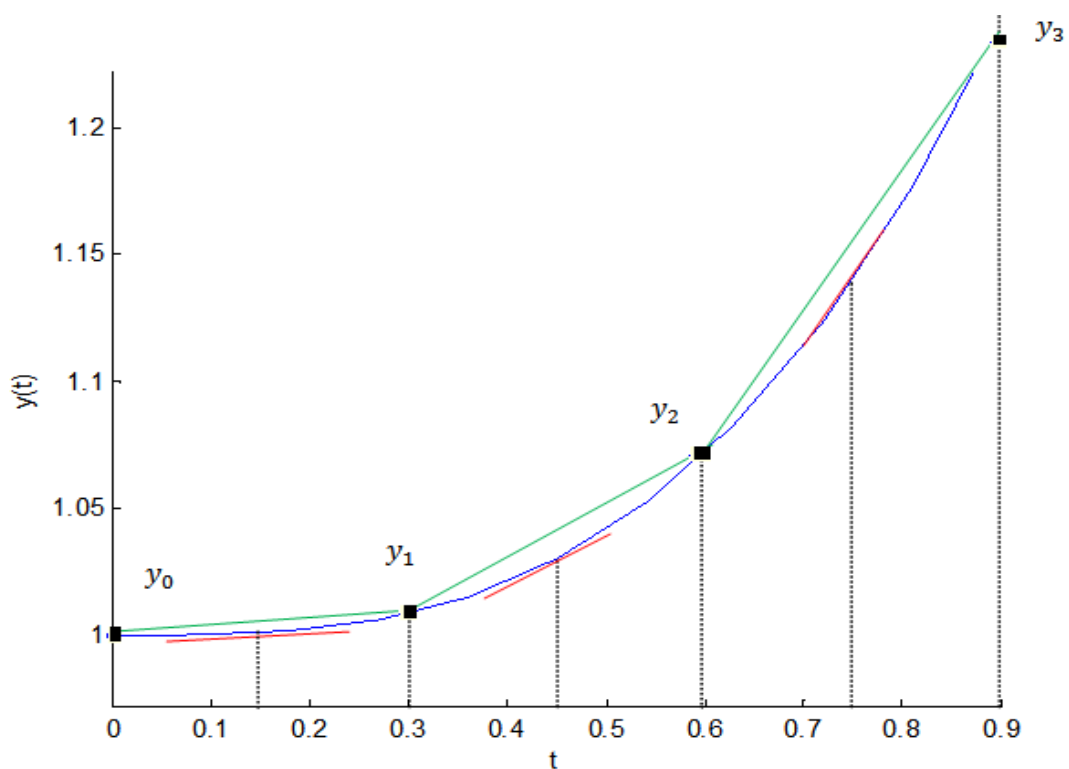


FIG. 1.2 – Méthode de Runge Kutta d'ordre 2

1.5 Programmation des méthodes

Soit l'exemple d'équation différentielle suivant :

$$\begin{cases} y'(t) = t^2 \\ y(0) = 1 \end{cases} .$$

On résoud cet exemple avec deux méthodes différentes, Euler et Runge Kutta d'ordre 2.

1.5.1 Méthode d'Euler fait sur Matlab

En Matlab, on peut facilement programmer la méthode d'Euler avec la fonction suivante :

fonction $[t, y] = \text{Euler}(f, t_{min}, t_{max}, Nint, y_0)$: Méthode d'Euler.

$Nint$: nombre de sous intervalles N .

t_{min} : temps t_0 .

t_{max} : temps $t_0 + T$.

f est une fonction avec comme arguments t et $y(t) : f(t, y(t))$.

y_0 à l'entrée est composé des valeurs des conditions limites $y_0 = y(t_0)$.

$h = (t_{max} - t_{min})/Nint$: valeur de pas

$t = \text{linspace}(t_{min}, t_{max}, Nint + 1)$: vecteur de t discrétisé $t = [t_{min}, t_{max}]$.

$y(1) = y(t_0) = y_0$: initialisation

for $n = 2 : Nint + 1$

$y(n) = y(n - 1) + h * \text{feval}(f, t(n - 1), y(n - 1))$.

calcul d'Euler

end

end

Les résultats d'exécution sur Matlab sont donnés comme suit :

valeur de pas :

$h = 0.3000$

temps t_0 :

$t_{\min} = 0$

temps final :

$t_{\max} = 0.9000$

conditions initiales :

$y = 1$

$t = 0$

nombre de sous intervalles n

$n = 3$

l'equation differentielle $dy=f=t^2$

$t = 0 \quad 0.3000$

$y = 1 \quad 1$

$t = 0 \quad 0.3000 \quad 0.6000$

$y = 1.0000 \quad 1.0000 \quad 1.0270$

$t = 0 \quad 0.3000 \quad 0.6000 \quad 0.9000$

$y = 1.0000 \quad 1.0000 \quad 1.0270 \quad 1.1350$

$ans = 0 \quad 0.3000 \quad 0.6000 \quad 0.9000$

FIG. 1.3 – Résultats de la méthode d'Euler sur Matlab

Ces résultats sont représentés dans la figure suivante :

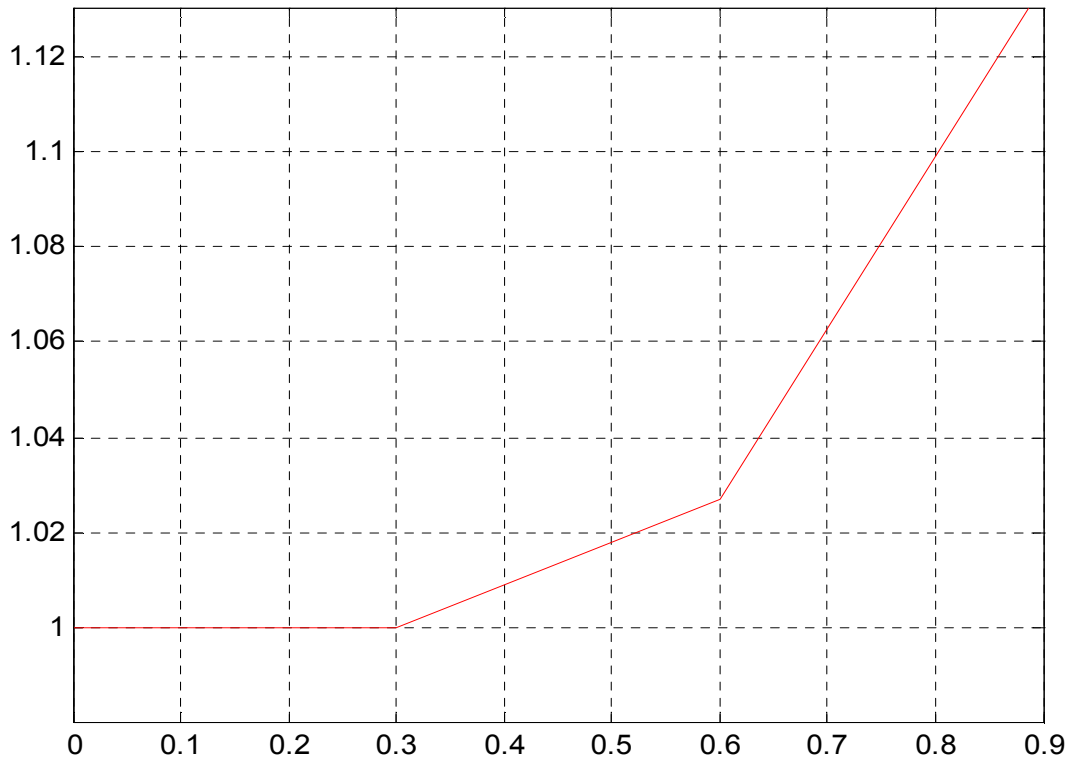


FIG. 1.4 – Solution par la méthode d'Euler

1.5.2 Méthode de Runge Kutta d'ordre 2

La méthode de Runge Kutta d'ordre 2 est programmée sur Matlab comme suit :
 fonction $[t, y] = RK2(f, t_{min}, t_{max}, Nint, y_0)$: méthode de Runge Kutta d'ordre 2.

$Nint$: nombre de sous intervalles N .

t_{min} : temps t_0 .

t_{max} : temps $t_0 + T$.

f est une fonction avec comme arguments t et $y(t)$: $f(t, y(t))$.

y_0 contient les valeurs des conditions limites

$h = (t_{max} - t_{min})/Nint$: valeur de pas

$t = linspace(t_{min}, t_{max}, Nint + 1)$: vecteur de t discrétisé $t = [t_{min}, t_{max}]$.

y contient les solutions de $y(t_n)$, $n = 1, 2, \dots, Nint + 1$, $y(1) = y_0$

for $n = 2 : Nint + 1$

```

 $K_1 = h * feval(f, t(n-1), y(n-1)).$ 
 $K_2 = h * feval(f, t(n-1) + h/2, y(n-1) + K_1/2)$ 
 $y(n) = y(n-1) + K_2;$ 
end
end.

```

Les résultats d'exécution sur Matlab de cette méthode sont comme suit :

```

Nint: nombre de sous intervalles N
Nint = 3
tmin: temps t0
t = 0
tmax: temps t0 + T.
tmax = 0.9000
y(0)= y0 : initialisation
y = 1
h : valeur de pas
h = 0.3000
n = 3
t = 0  0.3000
K1 = 0
K2 = 0.0067
y = 1.0000  1.0068
t = 0  0.3000  0.6000
K1 = 0.0270
K2 = 0.0607
y = 1.0000  1.0068  1.0675
t = 0  0.3000  0.6000  0.9000
K1 = 0.1080
K2 = 0.1687
y = 1.0000  1.0068  1.0675  1.2363
ans = 0  0.3000  0.6000  0.9000

```

FIG. 1.5 – Résultats de la méthode Runge Kutta d'ordre 2 sur Matlab

Ces résultats sont représentés par la figure suivante :

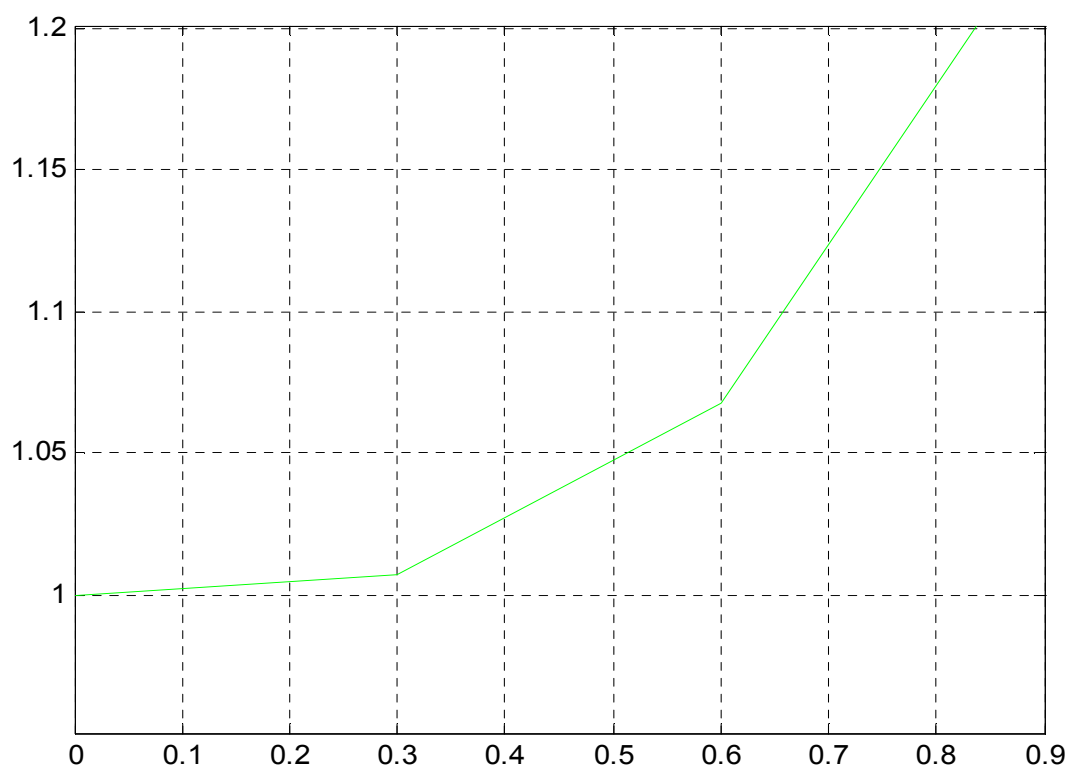


FIG. 1.6 – Solution par la méthode de Runge Kutta d'ordre 2

La comparaison entre la solution exacte et les solutions des deux méthodes de Runge Kutta et d'Euler est donnée par la figure suivante :

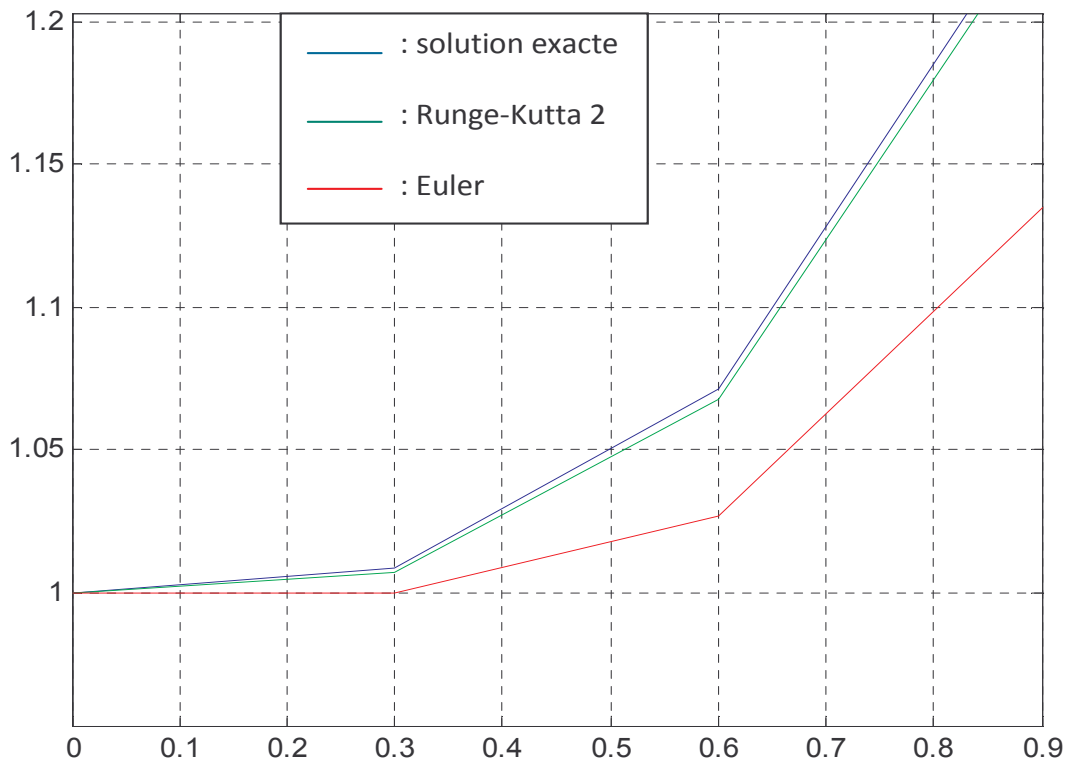


FIG. 1.7 – Comparaison des solutions

Remarque 1.1. On remarque que la méthode de Runge-Kutta d'ordre 2 est plus précise (proche) que celle d'Euler.

1.6 Résolution de fonction $f(x)=0$:

On veut résoudre l'équation $f(x) = 0$, où f est une fonction satisfaisant les deux hypothèses suivantes :

(H_0) : la fonction $f : [a, b] \rightarrow \mathbb{R}$ est continue.

(H_1) : l'équation $f(x) = 0$ admet une seule racine notée \bar{x} qui est dans l'intervalle $]a, b[$.

Autrement dit : $f(a) \cdot f(b) < 0$.

1.6.1 Méthode de Newton

Soit I un intervalle de \mathbb{R} et $f : I \rightarrow \mathbb{R}$ une application dérivable .

Pour déterminer une approximation numérique des solutions de l'équation $f(x) = 0$, la méthode de Newton part d'une solution approchée x_0 et remplace l'équation $f(x) = 0$ par

l'équation approchée $f(x_0) + (x - x_0)f'(x_0) = 0$, d'où la solution : $x = x_0 - f(x_0)/f'(x_0)$. Bien entendu, cette formule n'a un sens que si $f'(x_0) \neq 0$. Par récurrence, on obtient :

$$x_{n+1} = x_n - f(x_n)/f'(x_n), \quad x_0 \in I$$

d'où le schéma de Newton suivant :

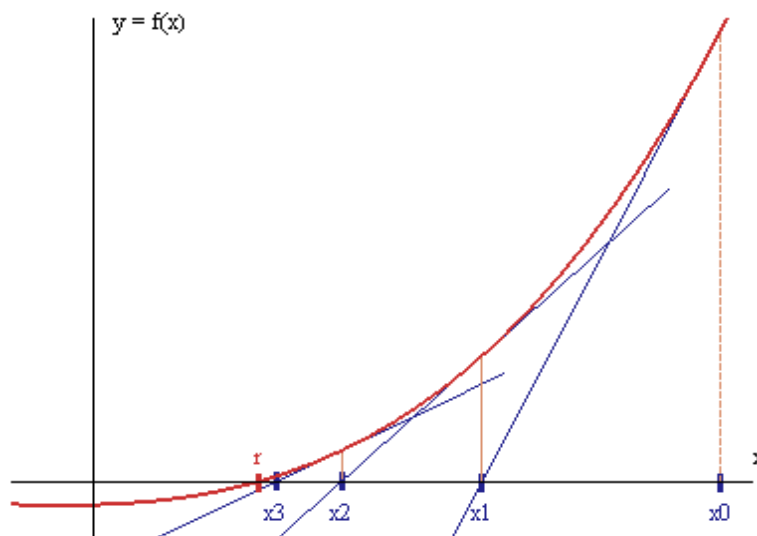


FIG. 1.8 – Méthode de Newton

Exemple 1.1. Soit la fonction suivante :

$$f(x) = \exp(x) - 2\cos(x).$$

On cherche le zéro de cette fonction en appliquant la méthode de Newton .

Les résultats d'exécution sur Matlab sont donnés par :

Donner la valeur initiale $x(1)$:

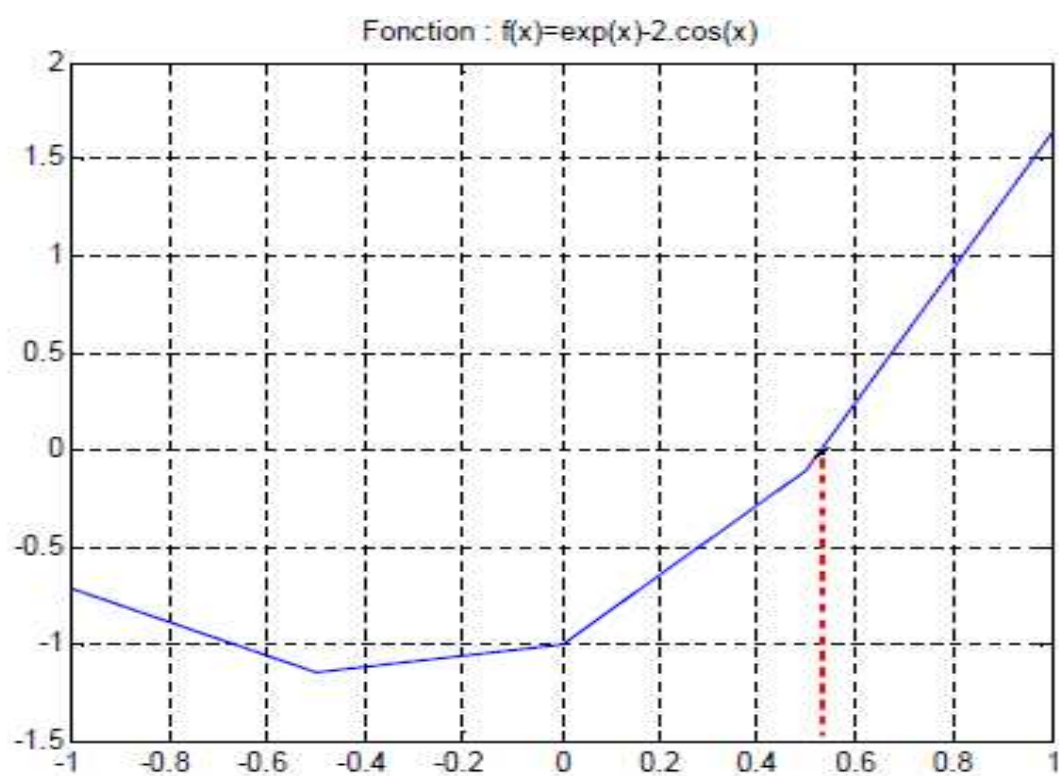
1

Le zéro de cette fonction est :

$x_p=0.539785$

FIG. 1.9 – Résultats d'exécution de la méthode Newton sur Matlab.

La figure de cette fonction par la méthode de Newton est donnée par :

FIG. 1.10 – Le zéro de la fonction f

Chapitre 2

Introduction au contrôle optimal

Ce chapitre est consacré aux notions de base de la théorie de contrôle optimal et la contrôlabilité des systèmes linéaires et non linéaires en utilisant les différentes méthodes de résolution des problèmes de contrôle optimal.

2.1 Théorie du contrôle

En mathématique, la théorie du contrôle optimal s'inscrit dans la continuité du calcul des variations. Cette méthode est apparue après la seconde guerre mondiale, répondant à des besoins pratiques de guidage, notamment dans le domaine de l'aéronautique et de la dynamique. La formalisation de cette théorie a posé des questions nouvelles ; par exemple dans la théorie des équations différentielles ordinaires, elle a motivé un concept de solutions généralisées et a engendré de nouveaux résultats d'existence de trajectoires optimales. La théorie du contrôle optimal est très liée à la mécanique classique, en particulier aux principes de Fermat, de Huygens, équations d'Euler-Lagrange

2.2 Quelques définitions

Définition 2.1.

- Les variables nommées variables d'état seront notées x_i , $i = 1, \dots, n$. On a des systèmes qui évoluent dans le temps, donc les x_i sont des fonctions de temps notées : $x_i(t)$, $t \in [0, T]$.
- Les composantes du contrôle seront notées $u_j(t)$; $j = 1, \dots, m$, elles doivent être intégrables par rapport à t .

- Les n variables d'états vont être gouvernées par n équations différentielles du premier ordre nommées équations d'états de la forme

$$\dot{x} = f(t, x, u), \quad t \in [0, T]$$

où \dot{x} est le vecteur dérivé par rapport au temps t de toutes les composantes de x .

- L'ensemble U est l'ensemble des contrôles admissibles qui peut être non borné, borné ou du type bang-bang.

Définition 2.2. (Commande bornée) la commande $u_j(t)$ est dite commande bornée si elle peut être minorer et majorer par des constantes a_j et b_j , par la forme suivante : $a_j \leq u_j(t) \leq b_j$, $j = 1, \dots, m$, $t \in [0, T]$.

Définition 2.3. Le contrôle $u(t)$, $t \in [0, T]$ est dit admissible s'il satisfait les contraintes :

$$\begin{cases} \dot{x} = Ax + Bu; \quad x(0) = x_0 \\ f_{\star} \leq u(t) \leq f^{\star} \end{cases} .$$

Définition 2.4. (Commande bang-bang). Un contrôle $u \in U$ est appelé contrôle bang-bang, si pour chaque instant t et chaque indice $j = 1, \dots, m$, on a $|u_j(t)| = 1$.

2.3 Position du problème

La formulation du problème de contrôle optimal est le suivant :

$$\left\{ \begin{array}{ll} J(T, u) = g(T, x(T)) + \int_0^T f^o(t, x(t), u(t)) dt \rightarrow \min_u, & (1) \\ \dot{x}(t) = f(t, x(t), u(t)), & (2) \\ x(0) = x_0 \in M_0, & (3) \\ x(T) = x_1 \in M_1, & (4) \\ u \in U, \quad t \in I = [0, T], & (5) \end{array} \right. \quad (2.1)$$

où M_0 (ensemble de départ) et M_1 (Ensemble d'arrivée) sont des sous ensembles de \mathbb{R}^n , I un intervalle de \mathbb{R} , $x_0 = x(0)$ est la position initiale du système (2), $x(T)$ est sa position terminale. En pratique, la position du système peut représenter la vitesse, la position, la température,... etc. $u(\cdot)$ est la commande du système (2.1). U est l'ensemble des applications mesurables, localement bornées sur I à valeurs dans $\Omega \subset \mathbb{R}^n$.

$$J(T, u) = g(T, x(T)) + \int_0^T f^0(t, x(t), u(t)) dt$$

est appelé coût, critère de qualité ou but du problème (2.1).

On distingue deux problèmes importants :

2.3.1 Problème de Lagrange

C'est le problème dont le critère à minimiser est égale à :

$$J(T, u) = \int_0^T f^0(t, x(t), u(t)) dt$$

C'est à dire $g \equiv 0$.

2.3.2 problème de Mayer

Ici c'est le problème dont le critère est le suivant :

$$J(T, u) = g(T, x(T))$$

C'est à dire $f^0 = 0$, $J(T, u)$ est le coût terminal.

L'unicité du système (2)-(4) est assurée par le théorème d'existence et d'unicité des équations différentielles. Soit $x(\cdot)$ la solution de l'équation (2) du système (2.1). $x(\cdot)$ varie en fonction du contrôle u .

Avant de résoudre un problème de contrôle optimal, on se pose les questions suivantes :

Question 1 : Existe-t-il un contrôle $u \in U$ tel que la trajectoire associée à $u \in U$ joigne l'état initial $x_0 \in \mathbb{R}^n$ à un état final $x_1 \in \mathbb{R}^n$ en un temps fini ? C'est le problème de contrôlabilité.

Question 2 : Si le système est contrôlable c'est à dire une fois la contrôlabilité est vérifiée, on peut vouloir déterminer un contrôle $u \in U$ tel que la trajectoire associée à ce contrôle joigne l'état initiale $x_0 \in \mathbb{R}^n$ à un état terminal $x_1 = x(T) \in \mathbb{R}^n$, en minimisant un certain critère de performance.

2.4 Contrôlabilité

Définition 2.5. le système $\dot{x} = f(t, x(t), u(t))$, $x(0) = x_0$, $t \in [0, T]$, est dit contrôlable si pour tous points $x_0 \in M_0$ et $x_1 \in M_1$, il existe un contrôle $u(\cdot)$ telle que la trajectoire associée relie x_0 à x_1 en temps fini.

Pour ce problème de contrôlabilité, Kalman [5] a donné en 1949 une caractérisation des systèmes linéaires autonomes, contrôlables en dimension finie. Pour les systèmes non linéaires, le problème mathématique de contrôlabilité est beaucoup plus compliqué.

2.4.1 Cas des systèmes linéaires

Ensemble accessible

Considérons le système dynamique suivant :

$$\dot{x}(t) = f(t, x(t), u(t)), \quad x(0) = x_0, \quad t \in I = [0, T] \quad (\star)$$

Définition 2.6. L'ensemble des points accessibles à partir de x_0 en temps final T est :

$$Acc(x_0, T) = \{x_u(T), u \in U\}$$

où $x_u(\cdot)$ est la solution du système (\star) associée au contrôle u .

Autrement dit $Acc(x_0, T)$ est l'ensemble des extrémités des solutions du système (\star) , en temps T lorsque le contrôle u varie.

Définition 2.7. Le système (\star) est dit Contrôlable en temps T si

$$Acc(x_0, T) = \mathbb{R}^n$$

Définition 2.8. Le contrôle u est dit extrémal sur $[0, T]$ si la trajectoire du système (2) du problème de contrôle (2.1) associée à u vérifie :

$$x(t) \in \partial Acc(x_0, t), \quad t \in I = [0, T].$$

2.4.2 Systèmes linéaires autonomes

Définition 2.9. Le système linéaire autonome décrit par :

$$\dot{x} = f(x, u)$$

est dit contrôlable, si pour tous $x_0 \in M_0$ et $x_1 \in M_1$, il existe un contrôle $u(\cdot)$ dans l'ensemble U , tel que la trajectoire correspondante $x(\cdot)$ transfère le système de l'état initial donné x_0 à un état final x_1 en temps fini.

Théorème 2.1. [14] *Le système autonome*

$$\dot{x}(t) = Ax(t) + Bu(t), \quad x(0) = x_0, \quad t \in I = [0, T]$$

est dit Contrôlable si la matrice

$$K = [B, AB, A^2B, \dots, A^{n-1}B]$$

est de rang n .

Remarque 2.1. K est une matrice à n lignes et $n \times m$ colonnes, elle est appelée matrice de Kalman. La condition $rgK = n$ est appelée condition de Kalman.

2.4.3 Systèmes linéaires non autonomes

Considérons le système

$$\dot{x}(t) = A(t)x(t) + B(t)u(t), \quad x(0) = x_0, \quad t \in I = [0, T] \quad (2.2)$$

la solution du système (2.2) en temps t est

$$x(t) = R(t)x_0 + \int_0^t R(t)R^{-1}(s)B(s)u(s)ds, \quad t \in [0, T]$$

où $R(\cdot)$ est la résolvante, elle est solution du système :

$$\begin{cases} \dot{R}(t) = A(t)R(t), \\ R(0) = Id, \end{cases}$$

où Id est la matrice identité.

Théorème 2.2. [14] *Le système (2.2) est contrôlable en temps T si et seulement si la matrice*

$$D = \int_0^T R(t)^{-1}B(t)B(t)'R(t)^{-1'}dt$$

est inversible.

où $B(t)'$ est la transposée de $B(t)$ et $R^{-1}(t)$ est l'inverse de $R(t)$.

D est appelée matrice de contrôlabilité.

2.4.4 Cas des systèmes non linéaires

Considérons le système

$$\begin{cases} \dot{x}(t) = Ax(t) + B(u(t)), \\ x(0) = x_0, \quad u \in U, \quad t \in I = [0, T] \end{cases} \quad (2.3)$$

où x est vecteur de \mathbb{R}^n , A une matrice dans $M_n(\mathbb{R})$, $B : \mathbb{R} \rightarrow \mathbb{R}^n$, fonction non linéaire du contrôle $u(\cdot)$, $x_0 \in \mathbb{R}^n$ est l'état initial du système (2.3).

Pour les systèmes de contrôle non linéaires, il est impossible d'étudier la contrôlabilité globale, le problème est beaucoup plus compliqué du fait qu'on ne peut pas utiliser la caractérisation de Kalman. Dans ce qui suit, on s'intéressera à l'étude de la contrôlabilité locale du système (2.3).

Définition 2.10. Soit $x_1 \in \mathbb{R}^n$. On dit que le système (2.3) est localement contrôlable au voisinage de x_1 en temps T depuis x_0 , si $x_1 \in \text{Acc}(x_0, T)^\circ$.

Autrement dit, il existe un voisinage V dans $V(x_1)$ tel que $V \subset \text{Acc}(x_0, T)$.

Considérons pour le système (2.3) le problème de contrôle suivant : étant donné un point $x_1 \in \mathbb{R}^n$, trouver un temps T et un contrôle u sur $[0, T]$ tel que la trajectoire x_u associée à u , solution de (2.3), vérifie :

$$x_u(0) = x_0, \quad x_u(T) = x_1.$$

Ceci conduit à la définition suivante :

2.4.5 Application entrée sortie

Définition 2.11. Soit $T > 0$. L'application entrée sortie en temps T du système (2.3) initialisée à x_0 est l'application

$$\begin{aligned} E_T : U &\longrightarrow \mathbb{R}^n \\ u &\longrightarrow x_u(T), \end{aligned}$$

où U est l'ensemble des contrôles u tel que la trajectoire associée est bien définie sur $[0, T]$. Autrement dit (l'application entrée sortie en temps T associe à un contrôle u , le point final de la trajectoire associée à u).

On donne une autre caractérisation de la contrôlabilité locale.

Définition 2.12. Soit $x_1 \in \mathbb{R}^n$. On dit que le système (2.3) est localement contrôlable au voisinage de x_1 en temps T depuis x_0 si :

$$\forall V \in \mathcal{V}(x_1) \not\subset \forall y \in V, \exists u \in U \text{ tel que } y = E_T(u).$$

Corollaire 2.1. Soit u un contrôle définie sur $[0, T]$ tel que sa trajectoire associée à x_u issue de $x(0) = y_0$ est définie sur $[0, T]$ est vérifie au temps T

$$x(T) \in \partial \text{Acc}(x_0, T)$$

Alors le contrôle u est singulier sur $[0, T]$. Où $\partial \text{Acc}(x_0, T)$ est la frontière de l'ensemble $\text{Acc}(x_0, T)$

Définition 2.13.

Si u est un contrôle singulier sur $[0, T]$. Alors le système est localement contrôlable le long de la trajectoire associée à ce contrôle.

Proposition 2.1. Si u n'est pas singulier sur $[0, T]$, alors il est régulier c'est à dire

$$x(T) \notin \partial \text{Acc}(x_0, T)$$

Remarque 2.2. L'application entrée sortie E_T est différentiable au sens de Fréchet. Calculons sa différentielle à l'ordre un.

$$E_T(u + \delta u)(t) = E_T(u(t)) + dE_T(u(t))\delta u + o(\|\delta\|_{L^\infty}), \quad t \in [0, T] \quad (2.4)$$

Soit

$$\dot{x}_{u+\delta u}(t) = Ax_{u+\delta u}(t) + B((u + \delta u)(t)), \quad x_{u+\delta u}(0) = x_0, \quad t \in [0, T],$$

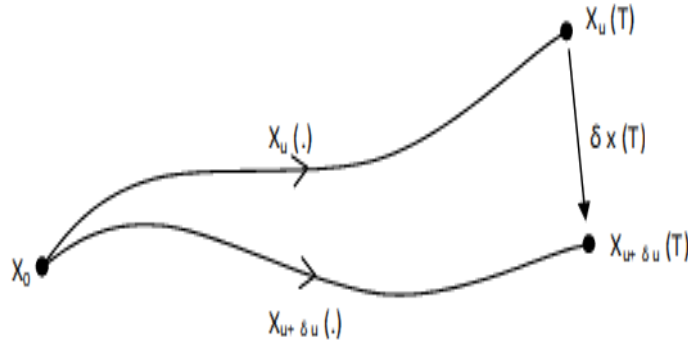
Notons par

$$\delta x(t) = x_{u+\delta u}(t) - x_u(t)$$

la variation de la trajectoire $x_u(\cdot)$. (voir figure 2.1).

$$\delta \dot{x}(t) = \dot{x}_{u+\delta u}(t) - \dot{x}_u(t)$$

$$\begin{aligned}
&= Ax_{u+\delta u}(t) + B((u + \delta u)(t)) - Ax_u(t) - B(u(t)), \quad t \in [0, T] \\
&= A(x_{u+\delta u}(t) - x_u(t)) + B((u + \delta u)(t)) - B(u(t)), \quad t \in [0, T] \\
&= A\delta x(t) + dB(u(t))\delta u(t), \quad t \in [0, T]
\end{aligned}$$

FIG. 2.1 – Variation de la trajectoire $x_u(\cdot)$

De la condition d'admissibilité $\delta x(0) = 0$, on obtient le système suivant :

$$\begin{cases} \delta \dot{x}(t) = A\delta x(t) + dB(u(t))\delta u(t), \\ \delta x(0) = 0, \quad u \in U, \quad t \in [0, T]. \end{cases} \quad (2.5)$$

Définition 2.14. Le système (2.5) est appelé système linéarisé du système (2.3) le long de la commande $u(\cdot)$.

La solution du système linéarisé (2.5) en temps T est donnée par :

$$\begin{aligned}
\delta x(T) &= \int_0^T e^{(T-s)} A dB(u(s)) \delta u(s) ds, \\
dE_T(u) \delta u &= x_{u+\delta u}(T) - x_u(T) = \delta x(T)
\end{aligned}$$

Où $dE_T(u)$ est l'application entrée sortie du système (2.5).

Proposition 2.2. [14] $dE_T(u)$ est surjective si et seulement si le système linéarisé est contrôlable.

Proposition 2.3. [14] Le système (2.3) est localement contrôlable au voisinage de $x_1 \in \mathbb{R}^n$ si et seulement si E_T est localement surjective au voisinage de x_1 .

Remarque 2.3. Si le système linéarisé (2.5) est contrôlable alors $dE_T(u)$ est surjective en utilisant le théorème des fonctions implicites, on déduit que E_T est localement surjective, de la proposition 2.1, on déduit que le système de départ (2.3) est localement contrôlable au voisinage de x_1 .

Théorème 2.3. [14] Le système linéarisé (2.5) est contrôlable en temps T si et seulement si la matrice

$$D = \int_0^T e^{-tA} dB(u(t)) dB(u(t))' e^{-tA'} dt$$

est inversible.

2.5 Méthodes de résolution

Dans la littérature, on trouve deux approches de résolution des problèmes de contrôle optimal :

- Les méthodes directes.
- Les méthodes indirectes.

2.5.1 Méthodes directes

Les méthodes directes consistent à transformer le problème de contrôle optimal en un problème d'optimisation non linéaire en dimension fini, parmi ces méthodes, on trouve la méthode de résolution par l'approche de la programmation linéaire, qui est la méthode adaptée appelée aussi méthode du support [1, 6, 10] seulement cette méthode résout les problèmes linéaires. Elle permet d'avoir une solution approchée ou une solution exacte.

Une autre méthode directe est la méthode de discrétisation du problème initial. Pour un problème de départ linéaire facile à résoudre. L'inconvénient de cette dernière approche est l'obtention d'une solution approchée.

La mise en œuvre des méthodes directes est simple, car elles ne nécessitent pas une étude théorique préalable, on n'a pas à étudier les variables adjointes ou bien à connaître à l'avance la structure des commutations, néanmoins ces méthodes sont moins précises.

2.5.2 Méthodes indirectes

Les méthodes indirectes sont basées sur le principe du maximum de pontryagin [11, 12] qui donne une condition nécessaire d'optimalité, il faut vérifier à priori l'optimalité de la trajectoire calculée. Ces méthodes ont l'extrême précision numérique, mais elles sont très sensibles au choix de la condition initiale. Contrairement aux méthodes directes, les méthodes indirectes nécessitent une étude théorique préalable et l'étude des variables adjointes. Pour ces méthodes la structure des commutations doit être connue à l'avance, elles sont efficaces en toute dimension.

Méthode de tir simple

Considérons le problème de contrôle optimal suivant :

$$\left\{ \begin{array}{l} \min J(t, u) = g(T, x(T)) + \int_0^T f^0(t, x(t), u(t)) dt \\ \dot{x}(t) = f(t, x(t), u(t)), \\ x(0) = x_0 \in M_0, \quad x(T) = x_1 \in M_1 \\ p(0) = p_0, \quad p(T) = p_1 \\ t \in [0, T] \end{array} \right. \quad (2.6)$$

Et supposons que le temps final T est fixé.

Le principe du maximum donne une condition nécessaire d'optimalité et affirme que toute trajectoire optimale est la projection d'une extrémale. Si l'on est capable, à partir de la condition de maximum, d'exprimer le contrôle extrémal en fonction de $(x(t), p(t))$, alors le système extrémal est un système différentiel de la forme $\dot{z} = F(t, z(t))$, où $z(t) = (x(t), p(t))$ et les conditions initiale, finale sous la forme $R(z(t_0), z(T)) = 0$. Finalement on obtient le problème aux valeurs limites suivant :

$$\left\{ \begin{array}{l} \dot{z} = F(t, z(t)), \\ R(z(t_0), z(T)) = 0 \end{array} \right. \quad (2.7)$$

Notons par $z(t, z_0)$ la solution du problème de cauchy suivant :

$$\dot{z} = F(t, z(t)), \quad z(t_0) = z_0$$

La fonction de tir est alors définie par :

$$G(T, z_0) = R(z_0, z(T, z_0)).$$

Le problème (2.7) aux valeurs limites est alors équivalent à la résolution du système suivant :

$$G(T, z_0) = 0.$$

Il s'agit de déterminer un zéro de la fonction G . Ceci peut être résolu par la méthode de Newton.

2.6 Principe du maximum de Pontryagin

Avant d'énoncer le principe du maximum, introduisons certaines définitions et propriétés essentielles.

Définition 2.15. Un contrôle $u^0(t)$, $t \in [0, T]$ est dit optimal si $u^0(\cdot)$ est extrémal et $J(u^0(t)) < J(u(t))$ pour tout contrôle extrémal $u(t)$, $t \in [0, T]$.

Théorème 2.4. [2] *Considérons le système*

$$\forall t \in I, \dot{x}(t) = A(t)x(t) + B(t)u(t), \quad x(0) = x_0$$

Supposons que le domaine de contraintes noté Ω est compact. Soit $T > 0$. Le contrôle u est extrémal sur $I=[0, T]$ si et seulement si il existe une solution non triviale $p(t)$, $t \in I$, de l'équation $\dot{p}(t) = -p(t)A(t)$ telle que :

$$p(t)B(t)u(t) = \max_{v \in \Omega} p(t)B(t)v. \quad (2.8)$$

pour presque tout $t \in [0, T]$.

Définition 2.16. Le vecteur $p(t) \in \mathbb{R}^n$ est appelé vecteur adjoint.

Définition 2.17. Le temps t_c auquel le contrôle extémal $u(t)$, $t \in [0, T]$ change de signe est appelé temps de commutation.

Le théorème qui suit est l'énoncé général du principe du maximum de Pontryaguin.

Théorème 2.5. [14] *Considérons le système de contrôle dans \mathbb{R}^n :*

$$\dot{x}(t) = f(t, x(t), u(t)), \quad (2.9)$$

où $f : \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^m \longrightarrow \mathbb{R}^n$ de classe C^1 , les contrôles sont des applications mesurables bornées à valeurs dans $\Omega \subset \mathbb{R}^m$. Soit M_0 et M_1 deux ensembles de \mathbb{R}^n . Notons par U l'ensemble des contrôles admissibles u dont les trajectoires associées relient un point initial de M_0 à un point final M_1 en temps t .

On définit le coût

$$J(T, u) = \int_0^T f^o(t, x(t), u(t)) dt + g(T, x(T))$$

où $f : \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^m \longrightarrow \mathbb{R}^n$ et $g : \mathbb{R} \times \mathbb{R}^n \longrightarrow \mathbb{R}$ sont de classe C^1 , $x(\cdot)$ est la solution de (2.9) associée au contrôle u .

On considère le problème de contrôle optimal suivant : Déterminer une trajectoire reliant M_0 à M_1 en minimisant le coût J . Le temps final peut être fixé ou non.

Si le contrôle $u \in U$ associé à la trajectoire $x(\cdot)$ est optimal sur $[0, T]$, alors il existe une application $p(\cdot) : [0, T] \longrightarrow \mathbb{R}^n$ absolument continue, appelé vecteur adjoint, et un réel $p^0 \leq 0$ tel que le couple $(p(\cdot), p^0)$ est non trivial et tels que pour presque tout $t \in [0, T]$,

$$\dot{x}(t) = \frac{\partial H}{\partial p}(t, x(t), p(t), p^0, u(t)), \quad (2.10)$$

$$\dot{p}(t) = -\frac{\partial H}{\partial x}(t, x(t), p(t), p^0, u(t)), \quad (2.11)$$

où $H(t, x, p, p^0, u) = p'(t)f(t, x, u) + p^0 f^0(t, x, u)$ est le Hamiltonien du système, et on a la condition de maximisation presque partout sur $[0, T]$

$$H(t, x, p, p^0, u(t)) = \max_{v \in U} H(t, x(t), p(t), p^0, v). \quad (2.12)$$

Si de plus M_0 et M_1 (ou juste l'un des deux ensembles) sont des variétés de \mathbb{R}^n ayant des espaces tangent en $x(0) = x_0 \in M_0$ et $x(T) = x_1 \in M_1$, alors le vecteur adjoint peut être construit de manière à vérifier les conditions de transversalités aux deux extrémités (ou juste l'une des deux)

$$p(0) \perp T_{x(0)}M_0 \quad (2.13)$$

$$\left(p(T) - p^0 \frac{\partial g}{\partial x}(T, x(T)) \right) \perp T_{x(T)}M_1 \quad (2.14)$$

Remarque 2.4. Si f et f^0 ne dépendent pas du temps t c'est à dire si le système considéré est autonome, alors l'Hamiltonien H ne dépend pas de t et on a :

$$\forall t \in [0, T], \quad \max_{v \in U} H(t, x(t), p(t), p^0, v) = cst$$

Remarque 2.5. La convention $p^0 \leq 0$ conduit au principe du maximum. La condition $p^0 \geq 0$ conduira au principe du minimum.

Remarque 2.6. Lorsqu'il n'y a pas de contraintes sur le contrôle, la condition de maximum (2.12) devient $\frac{\partial H}{\partial u} = 0$.

Définition 2.18. Les conditions (2.13) et (2.14) sont appelées conditions de transversalité sur le vecteur adjoint.

Chapitre 3

Exemples d'applications

3.1 Introduction

Dans ce chapitre, nous allons résoudre dans un premier temps, théoriquement puis numériquement, un problème d'un mouvement rectiligne d'une fusée, dans ce premier cas, on a commencé par un exemple simple, on a négligé toutes les forces externes. Par la suite on a considéré le problème de décollage d'une fusée en négligeant uniquement la résistance de l'air. Dans ce dernier cas, on a obtenu un problème avec contraintes sur l'état.

Le vol d'une fusée se décompose en deux phases :

- La phase propulsée.
- La phase balistique.

La période s'écoulant de l'instant de la mise à feu à la fin de combustion du propulseur est appelée la phase propulsée. Elle comprend une partie où la fusée est guidée par rampe de lancement et une partie où la fusée est livrée à elle-même. Après l'extinction du propulseur commence la phase balistique pendant laquelle la fusée, uniquement soumise à son poids et à la résistance de l'air, exploite la vitesse acquise pendant la propulsion pour atteindre son altitude maximale.

Notre objectif dans cette partie est la résolution du problème du temps minimal pour une fusée à mouvement rectiligne. Au cours de son vol, la fusée est soumise à trois forces : le poids de la fusée, la poussée du moteur et la résistance de l'air. En négligeant la résistance de l'air, notons $m(t)$ la masse de l'engin au temps t , $u(t)$ la poussée du moteur, au temps t , γ l'accélération. Supposons dans le premier cas que la pesanteur g est négligé. Notons aussi $x(t)$ la vitesse de l'engin au temps t .

Remarque 3.1. La phase propulsée d'une fusée est marquée par la diminution du poids

du à la combustion (flambée) du moteur.

D'où le choix de la masse en fonction du temps t .

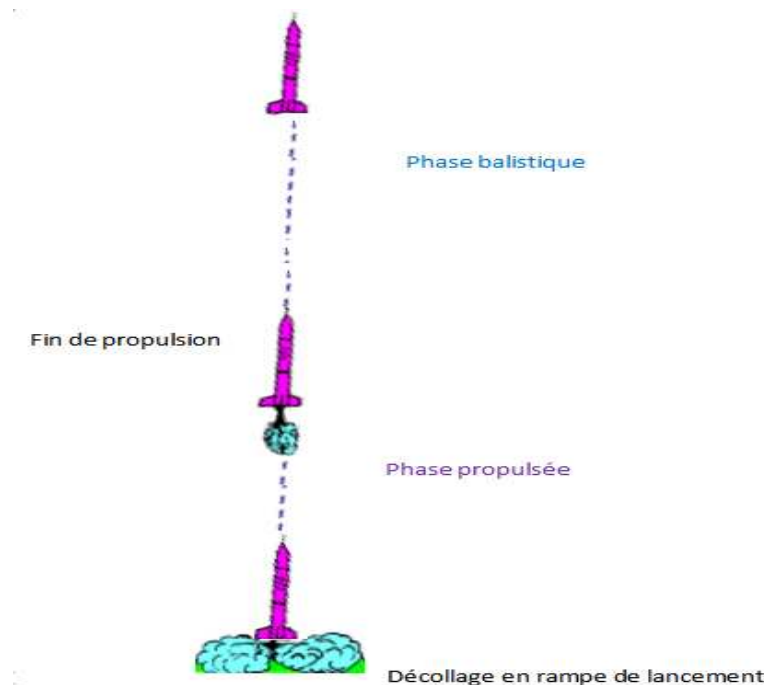


FIG. 3.1 – Les phases de la fusée

3.2 Rappels sur les lois de la physique(Newton)

Les lois du mouvement de Newton sont les trois principes fondamentaux concernant le mouvement des corps, on cite dans ce chapitre :

3.2.1 La première loi de Newton

Dans un référentiel Galiléen ; si un corps est au repos ou s'il est en mouvement rectiligne uniforme, les forces qu'il subit se compensent (la somme des forces égale à zéro)

$$\sum \vec{F} = 0.$$

3.2.2 La deuxième loi de Newton

Cette loi est appelée aussi principe fondamental de la dynamique.
Quand une force résultante s'exerce sur un objet, celui-ci est soumis à une accélération qui

a la même direction que la force :

$$\sum \vec{F} = m \vec{\gamma}.$$

où m : masse d'inertie de l'objet.

Définition 3.1. En physique, un référentiel galiléen, ou inertiel, est un référentiel dans lequel un objet isolé (sur lequel ne s'exerce aucune force ou sur lequel la résultante des forces est nulle) est soit immobile, soit en mouvement de translation rectiligne uniforme. Cela signifie que le principe d'inertie, qui est énoncé dans la première loi de Newton, s'y applique.

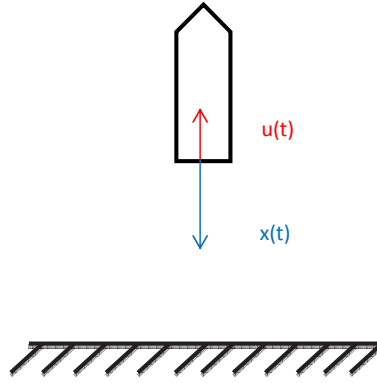


FIG. 3.2 – Application de la loi de Newton sur la fusée

La modélisation mathématique de notre exemple est obtenu en appliquant la deuxième loi de Newton et en négligeant toutes les forces extérieures, on obtient le système ci-dessous :

$$\left\{ \begin{array}{l} J(u(t)) = \int_0^T dt \longrightarrow \min_u, \\ \dot{x}(t) = u(t) \\ \dot{y}(t) = u^2(t) \\ |u(t)| \leq 1, \quad t \in [0, T] \\ x(0) = 0, \quad y(0) = 0 \\ x(T) = 2, \quad y(T) = 2 \end{array} \right. \quad (3.1)$$

où :

$x(t)$ représente la vitesse, $y(t)$ est inversement proportionnelle à la masse de l'engin. Le contrôle $u(t)$ est la poussée et vérifie la contrainte $|u(t)| \leq 1$.

$$\text{où } A = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}, \quad B(u(t)) = \begin{pmatrix} u(t) \\ u^2(t) \end{pmatrix}, \quad X_0 = \begin{pmatrix} x(0) \\ y(0) \end{pmatrix}.$$

On remarque que B est un vecteur fonction non linéaire de la commande $u(t)$, $t \in [0, T]$, le problème (3.1) est un problème de contrôle optimal avec une entrée non linéaire.

3.2.3 Contrôlabilité

Pour vérifier la contrôlabilité du système du problème (3.1), considérons la matrice

$$D = \int_0^T e^{-tA} dB(u(t)) dB(u(t))' e^{-tA'} dt$$

où

$$A = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix},$$

$$B(u(t)) = \begin{pmatrix} u(t) \\ u^2(t) \end{pmatrix},$$

et de là, on a la dérivé de $B(u(t))$:

$$dB(u(t)) = \begin{pmatrix} 1 \\ 2u(t) \end{pmatrix}$$

la résolvante est donnée par :

$$e^{-tA} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

la matrice D sera alors égale à :

$$D = \int_0^T \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 2u(t) \end{pmatrix} \begin{pmatrix} 1 \\ 2u(t) \end{pmatrix}' \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} dt$$

Le déterminant de D est égal à : $4T \int_0^T u^2(t)dt - 4(\int_0^T u(t)dt)^2$, et d'après le théorème de Cauchy Lipschitz on a :

$$(\int_0^T u(t)dt)^2 < T \int_0^T u^2(t)dt \implies T \int_0^T u^2(t)dt - (\int_0^T u(t)dt)^2 > 0 \text{ alors le problème (3.1)}$$

est contrôlable en temps T.

3.2.4 Résolution théorique

Application du Principe du maximum de Pontryagin le Hamiltonien du système (3.1) précédent s'écrit :

$$\begin{aligned} H(t, x(t), p(t), p^0, u(t)) &= p_1 \dot{x} + p_2 \dot{y} - 1 \\ &= p_1 u(t) + p_2 u^2(t) - 1 \end{aligned}$$

où p_1 et p_2 sont les composantes du vecteur adjoint, elles vérifient :

$$\begin{cases} \dot{p}_1(t) = -\frac{\partial H}{\partial x(t)} = 0 \\ \dot{p}_2(t) = -\frac{\partial H}{\partial y(t)} = 0 \end{cases} \quad (3.2)$$

les conditions de transversalité

$$p_1(T) = \frac{\partial g(u(t))}{\partial x(t)} = k_1$$

$$p_2(T) = \frac{\partial g(u(t))}{\partial y(t)} = k_2$$

De l'équation (3.2) on obtient :

$$\dot{p}_1(t) = 0 \implies p_1(t) = c_1 = cst$$

$$\dot{p}_2(t) = 0 \implies p_2(t) = c_2 = cst$$

Cherchons le contrôle optimal :

$$\frac{\partial H}{\partial u(t)} = 0 \iff p_1(t) + 2p_2(t)u(t) = 0$$

$$\iff 2p_2(t)u(t) = -p_1(t)$$

$$\iff u(t) = \frac{-p_1(t)}{2p_2(t)}$$

$$\frac{\partial^2 H}{\partial^2 u(t)} = 2p_2(t).$$

Dans ce cas on doit étudier les deux cas : $p_2(t) < 0$ et $p_2(t) > 0$.

Si $p_2(t) < 0$:

$$u(t) = \begin{cases} -\frac{p_1(t)}{2p_2(t)} , & si -\frac{p_1(t)}{2p_2(t)} \in]-1, +1[\\ +1 , & si -\frac{p_1(t)}{2p_2(t)} \geq 1 \\ -1 , & si -\frac{p_1(t)}{2p_2(t)} \leq -1 \end{cases} .$$

Si $p_2(t) > 0$:

$$u(t) = \begin{cases} +1 , & si -\frac{p_1(t)}{2p_2(t)} \leq -1 \\ -1 , & si -\frac{p_1(t)}{2p_2(t)} \geq 1 \\ \pm 1 , & si -\frac{p_1(t)}{2p_2(t)} \in]-1, +1[\end{cases} \quad (*)$$

De (*) on obtient :

$$p_2(t) > 0, si -\frac{p_1(t)}{2p_2(t)} \in]-1, +1[:$$

$$u(t) = \begin{cases} +1, & si p_1(t) > 0 \\ -1, & si p_1(t) < 0 \end{cases}$$

De là le contrôle optimal est :

$$u(t) = \begin{cases} +1, & \text{si } (p_1(t) \geq -2p_2(t)) \text{ ou } (p_1(t) \geq 2p_2(t)) \text{ ou } (p_1(t) > 0 \text{ et } p_2(t) > 0) \\ -1, & \text{si } (p_1(t) \leq 2p_2(t)) \text{ ou } (p_1(t) \leq -2p_2(t)) \text{ ou } (p_1(t) < 0 \text{ et } p_2(t) > 0) \\ -\frac{p_1(t)}{2p_2(t)}, & \text{sinon} \end{cases}$$

Autrement dit :

$$u(t) \in \left\{ -1, +1, \frac{-p_1(t)}{2p_2(t)} \right\}$$

Soit (x_1, y_1) le point final à atteindre, on distingue les cas suivants :

- $0 < y_1 < x_1$, il existe un seul contrôle optimal constant.
- $y_1 = |x_1|$, il existe un seul contrôle optimal égal à 1 ou -1.
- si $y_1 > x_1$, il existe une infinité de contrôles optimaux qui sont des successions d'arcs ± 1 .

En effet :

1^{er} cas : Supposons que $0 < y_1 < x_1$:

$$x_1 = x(T) = \int_0^T u(s) ds = \begin{cases} \int_0^T ds = T, & \text{si } u(t) = +1. \\ \int_0^T -ds = -T, & \text{si } u(t) = -1. \\ \int_0^T \frac{-p_1(t)}{2p_2(t)} ds = \frac{-p_1(t)}{2p_2(t)} T, & \text{si } u(t) = \frac{-p_1(t)}{2p_2(t)}. \end{cases}$$

$$y_1 = y(T) = \int_0^T u(s)^2 ds = \begin{cases} \int_0^T ds = T, & \text{si } u(t) = +1 \\ \int_0^T ds = T, & \text{si } u(t) = -1 \\ \int_0^T \left(\frac{-p_1(t)}{2p_2(t)}\right)^2 ds = \frac{p_1^2(t)}{4p_2^2(t)} T, & \text{si } u(t) = \frac{-p_1(t)}{2p_2(t)}. \end{cases}$$

C'est à dire :

$y_1 < x_1 \implies T < T$: cas impossible (pour $u(t) = +1$).

$y_1 < x_1 \implies T < -T$: cas impossible (pour $u(t) = -1$).

$y_1 < x_1 \implies \frac{p_1^2(t)}{4p_2^2(t)} T < \frac{-p_1(t)}{2p_2(t)} T$: cas possible (pour $u(t) = \frac{-p_1(t)}{2p_2(t)}$).

On a donc un seul contrôle optimal constant (singulier car il n'atteint pas les bornes : $u \in]-1, 1[$).

2^{eme} cas : Supposons que $y_1 = |x_1|$

$$x_1 = x(T) = \int_0^T u(s) ds = \begin{cases} \int_0^T ds = T, & \text{si } u(t) = +1 \\ \int_0^T -ds = -T, & \text{si } u(t) = -1 \\ \int_0^T \frac{-p_1(t)}{2p_2(t)} ds = \frac{-p_1(t)}{2p_2(t)} T, & \text{si } u(t) = \frac{-p_1(t)}{2p_2(t)} \end{cases}$$

$$y_1 = y(T) = \int_0^T u(s)^2 ds = \begin{cases} \int_0^T ds = T, & \text{si } u(t) = -1 \\ \int_0^T ds = T, & \text{si } u(t) = +1 \\ \int_0^T \left(\frac{-p_1(t)}{2p_2(t)}\right)^2 ds = \left(\frac{p_1(t)}{2p_2(t)}\right)^2 T, & \text{si } u(t) = \frac{-p_1(t)}{2p_2(t)} \end{cases}$$

C'est à dire :

$y_1 = |x_1| \implies T = |T|$: cas possible (pour $u(t) = +1$).

$y_1 = |x_1| \implies T = |-T|$: cas possible (pour $u(t) = -1$).

$y_1 = |x_1| \implies \frac{p_1^2(t)}{4p_2^2(t)} T = \left|\frac{-p_1(t)}{2p_2(t)} T\right|$: cas impossible (pour $u(t) = \frac{-p_1(t)}{2p_2(t)}$).

On a un seul contrôle optimal égal à 1 ou -1.

3^{eme} cas : Supposons que $y_1 > x_1$:

$$\begin{aligned}
x_1 = x(T) = \int_0^T u(s) ds &= \begin{cases} \int_0^T ds = T, & \text{si } u(t) = +1 \\ \int_0^T -ds = -T, & \text{si } u(t) = -1 \\ \int_0^T \frac{-p_1(t)}{2p_2(t)} ds = \frac{-p_1(t)}{2p_2(t)} T, & \text{si } u(t) = \frac{-p_1(t)}{2p_2(t)} \end{cases} \\
y_1 = y(T) = \int_0^T u(s)^2 ds &= \begin{cases} \int_0^T ds = T, & \text{si } u(t) = -1 \\ \int_0^T ds = T, & \text{si } u(t) = +1 \\ \int_0^T \left(\frac{-p_1(t)}{2p_2(t)}\right)^2 ds = \frac{p_1^2(t)}{4p_2^2(t)} T, & \text{si } u(t) = \frac{-p_1(t)}{2p_2(t)} \end{cases}
\end{aligned}$$

C'est à dire :

$$y_1 > x_1 \implies T > T : \text{ cas impossible (pour } u(t) = +1).$$

$$y_1 > x_1 \implies T > -T : \text{ cas possible (pour } u(t) = -1).$$

$$y_1 > x_1 \implies \frac{p_1^2(t)}{4p_2^2(t)} T > \frac{-p_1(t)}{2p_2(t)} T : \text{ cas possible (pour } u(t) = \frac{-p_1(t)}{2p_2(t)}).$$

On a une infinité de solutions.

3.2.5 méthode de tir : tir simple

La méthode de tir simple permet de trouver la commande et la trajectoire associée. Il existe une fonction $z(t) = (x(t), p(t))$ définie dans U , et les conditions initiales, finales et les conditions de transversalités, se mettent sous la forme

$$h(z(t), p(t))$$

définit toutes les trajectoires optimales, et d'après le principe de maximum on a

$$u(t) = \text{signe} \left(-\frac{p_1(t)}{2p_2(t)} \right)$$

La connaissance de la fonction de tir permet de considérer que ce problème optimal est entièrement résolu mathématiquement.

Considérons le cas $y_1 = |x_1|$, nous avons le problème suivant :

$$\left\{ \begin{array}{l} \dot{x} = u(t) \\ \dot{y} = u^2(t) \\ \dot{p}_1(t) = 0 \\ \dot{p}_2(t) = 0 \\ x(0) = 0, \quad y(0) = 0 \\ x(T) = 2, \quad y(T) = 2 \end{array} \right. \quad (3.3)$$

En posant $z(t) = (x(t), y(t), p_1(t), p_2(t)) = (z_1(t), z_2(t), z_3(t), z_4(t))$ Résoudre le problème (3.3) est équivalent à chercher un zéro de l'équation $G(z)=0$. Où G est la fonction de tir associé à notre problème :

$$G : \mathbb{R}^4 \longrightarrow \mathbb{R}^4$$

$$z \longrightarrow G(z) = z(t, 0)$$

Où $z(t, 0, h)$ est la solution du système suivant :

$$\left\{ \begin{array}{l} \dot{z}_1(t) = u(t) \\ \dot{z}_2(t) = u^2(t) \\ \dot{z}_3(t) = 0 \\ \dot{z}_4(t) = 0 \\ z_1(0) = 0, \quad z_2(0) = 0 \end{array} \right.$$

On définit la fonction de tir suivante :

$$G(z) = \begin{pmatrix} z_1(t_f) - x_f \\ z_2(t_f) - y_f \end{pmatrix}$$

Voici le programme de la méthode de tir fait sous Matlab :

```

function tirsimple
% Méthode de tir simple, en utilisant fsolve, pour le système
de contrôle
%      xdot=u, ydot=u^2, |u|<=1.
% On veut aller de (0,0) à (2,2) en temps minimal.

clear all ; clf ; clc ; format long ;

global x0 ; x0=[0;0] ;

P0=[1;-1] ; tf=3;

% Calcul de P0,tf
options=optimset('Display','iter','LargeScale','off');

[P0tf,FVAL,EXITFLAG]=fsolve(@F,[P0;tf],options);

EXITFLAG % 1 si la méthode converge, -1 sinon

% Trace de la trajectoire optimale
options = odeset('AbsTol',1e-9,'RelTol',1e-9) ;

[t,z] = ode45(@sys,[0;P0tf(3)],[x0;P0tf(1);P0tf(2)],options) ;

subplot(321) ; plot(t,z(:,3)) ; title('Trajectoire P1') ;
subplot(322) ; plot(t,z(:,4)) ; title('Trajectoire P2') ;
subplot(323) ; plot(z(:,1),z(:,2)) ; title('Trajectoire ') ;

if ((z(3)>=-2*z(4)) || (z(3)>=2*z(4)) || (z(3)>0 && z(4)>0))
    z(:,5)=+1 ;
else
    if((z(3)<=2*z(4)) || (z(3)<=-2*z(4)) || (z(3)<0 && z(4)>0))

        z(:,5)=-1;
    else
        z(:,5)=(-z(3)/(2*z(4)));
    end
end

subplot(324) ; plot(t,z(:,5)) ; title('Contrôle u') ;

t(end)
z(1,3)
z(1,4)

```

```

%-----
function Xzero=F(X)      % Définition de la fonction dont on
                           cherche un zéro.

global x0 ;
options = odeset('AbsTol',1e-9,'RelTol',1e-9) ;

[ t,z] = ode113(@sys,[0;X(3)],[x0;X(1);X(2)],options) ;

if ((z(3)>=-2*z(4)) || (z(3)>=2*z(4)) || (z(3)>0 && z(4)>0))

    v=+1 ;
else
    if((z(3)<=2*z(4)) || (z(3)<=-2*z(4)) || (z(3)<0 && z(4)>0))

        v=-1;
    else
        v=-z(3)/(2*z(4));
    end
end
HamEnd = z(end,3)*v + z(end,4)*(v^2)-1 ;

Xzero = [ z(end,1)- 2          % on impose xf=2
          z(end,2)- 2          % on impose yf=2
          HamEnd           ] ; % tf libre donc H(tf)=0

%-----
function zdot=sys(t,z)

if ((z(3)>=-2*z(4)) || (z(3)>=2*z(4)) || (z(3)>0 && z(4)>0))

    u=+1 ;
else
    if((z(3)<=2*z(4)) || (z(3)<=-2*z(4)) || (z(3)<0 && z(4)>0))

        u=-1;
    else
        u=z(3)/(-2*z(4));
    end
end

zdot = [ u
         u^2
         0
         0 ] ; % système extrémal

```

FIG. 3.3 – La simulation de la méthode de tir sous Matlab

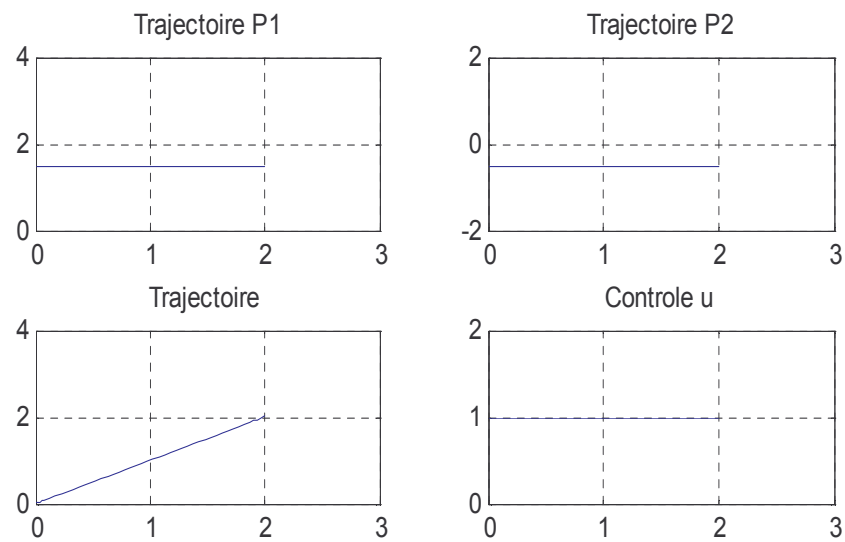


FIG. 3.4 – Les trajectoires optimales par la méthode de tir

On remarque dans ce cas que le contrôle optimal $u(t)$ est constant. Les deux vecteurs adjoints sont constants en fonction du temps t .

EXITFLAG =

1

La valeur du temps minimal :

ans =

2.000000019869594

La valeur des vecteurs adjoints :

ans =

1.499999997453296

ans =

-0.500000000502919

Remarque 3.2.

- Si $y_1 = |x_1|$, la méthode de tir converge (exitflag=1), mais pour $0 < y_1 < x_1$ cette méthode ne converge pas (exitflag=-2) car la trajectoire singulière a une propriété de rigidité c'est-à-dire c'est la seule trajectoire joignant ses extrémités et donc elle est optimale indépendamment du critère d'optimisation choisit. De même pour le troisième cas ou $y_1 > x_1$ on a la non convergence de la méthode de tir.
- On déduit que la méthode de tir est très sensible au choix de conditions initiales.

3.2.6 Application de la méthode directe

Voici le programme de la méthode directe fait sous Matlab :

```

function directe
%% Discrétisation directe (en utilisant fmincon) du problème de temps minimal
%% xdot=u, ydot=u^2, |u|<=1,
%% le problème étant de joindre (0,0) a (2,2) en temps minimal.

clear all ; close all ; clc ;

N = 20; % nombre de pas de discrétisation
uinit = 2*rand(N,1)-1 ; % initialisation aléatoire du contrôle
tfinit = 1 ; xinit = [uinit ; tfinit] ; % point de départ pour fmincon
lb = -ones(N+1,1) ; lb(N+1) = 0 ; ub = ones(N+1,1) ; ub(N+1) = 20 ;
% contrainte sur le contrôle |u| <= 1, et 0 <= tf <= 20

[rep,Fval,exitflag] = fmincon(@tempsfinal,xinit,[],[],[],[],lb,ub,@cond) ;

exitflag
tf = rep(end) ; x(1)=0 ; y(1) = 0 ;
for i=1:N
    x(i+1) = x(i) + (tf/N)*rep(i);
    y(i+1) = y(i) + (tf/N)*(rep(i)^2);
end
% calcul de la trajectoire optimale
subplot(321) ; plot(linspace(0,tf,N),x(1:N)) ; title('Trajectoire X') ;
subplot(322) ; plot(linspace(0,tf,N),y(1:N)) ; title('Trajectoire Y') ;
subplot(323) ; plot(linspace(0,tf,N),x(1:N),'b',linspace(0,tf,N),y(1:N),'b')
; title('Trajectoire') ;
subplot(324) ; plot(linspace(0,tf,N),rep(1:N)) ; title('Contrôle u') ;

%-----

function [c,ceq] = cond(x)
N = length(x)-1 ;
c = [] ; % ou bien c=0
tf = x(end) ; xf = 0 ; yf = 0 ;
for i=1:N
    xf = xf +(tf/N)*x(i); % calcul du point final au temps tf
    yf = yf + (tf/N)*(x(i)^2) ; % avec la méthode d'Euler explicite
end
ceq = [ xf-2 ; yf-2] ; % on impose la condition finale xf=2, yf=2
%-----

function val = tempsfinal(x)
val = x(end) % x=[u;tf], ou u est le discrétise du contrôle,
% et tf est le temps final

```

FIG. 3.5 – La simulation de la méthode directe sous Matlab

les résultats d'exécution sur matlab nous donne les figures suivantes :

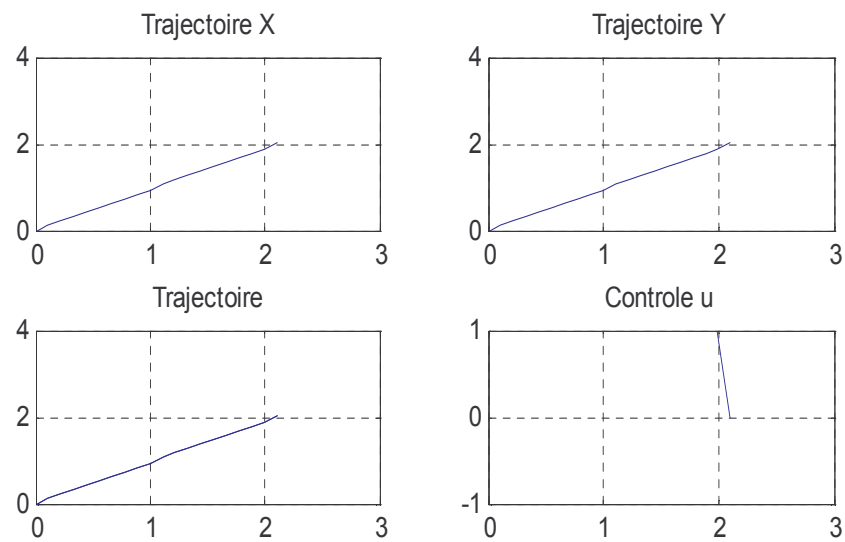


FIG. 3.6 – Les trajectoires optimales par la méthode directes

La valeur du temps final :

val =

2.105263189265602

exitflag =

1

Remarque 3.3.

- On comparant les résultats de la méthode directe avec la méthode indirecte, on déduit que cette dernière est plus précise.
- On a $\varphi(t) = -\frac{p_1(t)}{2p_2}$, alors il ya commutation si : $\varphi(t) = 0 \implies p_1(t) = 0$ et cela pour $t=T$.

3.3 Position du problème

Dans cette section, nous allons modéliser et résoudre théoriquement et numériquement le décollage d'une fusée, en tenant compte des forces externes, sans la résistance de l'air. Notons par $h(t)$ l'altitude au temps t , $v(t)$, $t \in [0, T]$, le module de la vitesse, $m(t)$ la masse de la fusée au temps t , g l'accélération de la pesanteur.

D'après la loi de Newton :

$$\dot{h}(t) = v(t) \text{ et } h(0) = 0. \quad (3.4)$$

Supposons qu'on a négligé la résistance de l'air.

La somme des forces est égale à la masse multipliée par l'accélération notée γ .

Notons $u(t)$ la poussée de la fusée au temps t .

$$\begin{aligned} \sum \vec{F} &= m(t)\vec{\gamma} \\ \Downarrow \\ u(t) - m(t)g &= m(t)\gamma = m\dot{v}(t) \\ \Downarrow \\ \dot{v}(t) &= \frac{u(t)}{m(t)} - g, \quad v(0) = 0 \end{aligned} \quad (3.5)$$

A noter que la masse de la fusée au temps t est égale à la masse de la fusée sans carburant plus la masse du carburant.

A chaque fois que le temps augmente, la masse $m(t)$ diminue et la poussée augmente, donc :

$$\dot{m}(t) = -u(t). \quad (3.6)$$

Le problème considéré dans ce cas est de maximiser l'altitude finale. La fonction objectif sera donc :

$$J(u(t)) = h(T) \longrightarrow \max$$

Ce qui est équivalent à :

$$J(u(t)) = -h(T) \longrightarrow \min.$$

Le problème de contrôle optimal obtenu de la modélisation est :

$$\left\{ \begin{array}{ll} J(u(t)) = -h(T) \longrightarrow \min & \\ \dot{h}(t) = v(t), \quad h(0) = 0 & (1) \\ \dot{v}(t) = \frac{u(t)}{m(t)} - g, \quad v(0) = 0 & (2) \\ \dot{m}(t) = -u(t), \quad m(0) = m_0 & (3) \\ 0 \leq u(t) \leq M = u_{max}, \quad t \in [0, T] & (4) \end{array} \right. \quad (3.7)$$

u_{max} = La poussée maximale.

Notons par m_0 la masse de la fusée en absence du carburant et m_1 la masse de la fusée au temps $t = 0$.

La troisième contrainte du problème (3.7) est sous conditions :

$$m_0 \leq m(t) \leq m_1, \quad t \in [0, T]$$

On a donc un problème de contrôle optimal avec contrainte sur l'état.

Quelle est donc la nature du contrôle optimal et la nature des états adjoints ?

3.3.1 Résolution théorique

Soit H l'hamiltonien du système (3.7)

$$H(t, v(t), p(t), u(t)) = p_h(t)v(t) + p_v(t)\left(\frac{u(t)}{m(t)} - g\right) + p_m(t)(-u(t))$$

$$\frac{\partial H}{\partial u(t)} = \frac{p_v(t)}{m(t)} - p_m(t)$$

Le contrôle est donc bang-bang c'est à dire :

$$u(t) = \begin{cases} 0 & \text{si } \frac{p_v(t)}{m(t)} - p_m(t) < 0 \\ U_{max} & \text{si } \frac{p_v(t)}{m(t)} - p_m(t) > 0 \end{cases} \quad (3.8)$$

Lemme 3.1. $\frac{p_v(t)}{m(t)} - p_m(t)$ ne s'annule pas dans un sous intervalle de $[0, T]$

Démonstration

$\varphi(t) = \frac{p_v(t)}{m(t)} - p_m(t)$, sa dérivée par rapport à t , avec $t \in [0, T]$ est :

$$\begin{aligned} \dot{\varphi}(t) &= \frac{\dot{p}_v(t)m(t) - \dot{m}(t)p_v(t)}{m^2(t)} - \dot{p}_m(t) \\ &= -\frac{p_h(t)}{m(t)} + \frac{u(t)p_v(t)}{m^2(t)} - \frac{p_v(t)}{m^2(t)}u(t) \\ &= -\frac{p_h(t)}{m(t)} \neq 0. \end{aligned}$$

$p_h(t)$, $p_v(t)$ et $p_m(t)$, $t \in [0, T]$ sont les composantes du vecteur adjoint elles sont telle que :

$$\begin{cases} \dot{p}_h(t) = -\frac{\partial H}{\partial h(t)} = 0 \implies p_h(t) = c_1, \forall t \in [0, T] \\ \dot{p}_v(t) = -\frac{\partial H}{\partial v(t)} = -p_h(t) \implies p_v(t) = -c_1 t + c_2, \forall t \in [0, T] \\ \dot{p}_m(t) = -\frac{\partial H}{\partial m(t)} = \frac{p_v(t)}{m^2(t)}u(t) \end{cases}$$

On remarque que $\dot{p}_m(t)$ n'est pas continu en fonction de t (Il dépend de la commande bang-bang).

On a ce cas, car le modèle considéré est un problème de contrôle optimal avec contrainte sur l'état (le dernier état).

Les conditions de transversalité

$$g(T, X(T)) = -h(T)$$

$$\text{où } X^t = (h, v, m).$$

On a $p^0 = -1$.

$$\begin{cases} p_h(T) = -\frac{\partial g(T, X(t))}{\partial h(t)} = 1 \\ p_v(T) = -\frac{\partial g(T, X(t))}{\partial v(t)} = 0 \\ p_m(T) = 0 \end{cases}$$

De (3.8), on déduit que la fonction de commutation est :

$$\varphi(t) = \frac{p_v(t)}{m(t)} - p_m(t), \text{ sa dérivée par rapport à } t, \text{ avec } t \in [0, T] \text{ est :}$$

$$\dot{\varphi}(t) = -\frac{p_h(t)}{m(t)} \neq 0.$$

Au début la poussée $u(t) \geq 0$

donc :

$$\frac{p_v(t)}{m(t)} - p_m(t) = \varphi(t) \geq 0, \quad t \in [0, T].$$

On a au temps $t=T$:

$$\varphi(t) = \frac{p_v(t)}{m(t)} - p_m(t) = 0.$$

Il y a au plus une commutation au temps $t=T$, dans ce cas $u(t) = u_{max}$.

3.3.2 Résolution numérique

On a un problème avec contrainte sur l'état, pour cela on le résoud avec la méthode directe. A noter que le pesantier est négligé.

D'après les résultats obtenus (ci dessous), on remarque que la masse diminue en fonction du temps de 1500 Kg à 800 Kg au bout de $t=7,14$ s. La vitesse augmente en fonction du temps t pour atteindre une vitesse de 0,62 Km/s et le contrôle est de type bang-bang.

Voici la simulation du programme de la méthode directe fait sous Matlab :

```
function directe
%% Discrétisation directe (en utilisant fmincon) du problème d'altitude
maximale
clearall ; close all ; clc ;
M=100;
N = 50; % nombre de pas de discrétisation
uinit = 2*rand(N,1)-1 ; % initialisation aléatoire du contrôle
tfinit = 1 ; xinit = [uinit ; tfinit] ; % point de départ pour fmincon
lb = zeros(N+1,1) ; lb(N+1) = 0 ; ub = M*ones(N+1,1) ; ub(N+1) = 50 ;
% contrainte sur le contrôle  $0 \leq u \leq M$ , et  $0 \leq tf \leq 50$ 

[rep,Fval,exitflag] = fmincon(@hfinal,xinit,[],[],[],[],lb,ub,@cond) ;
exitflag
tf = rep(end) ; x(1)=0 ; y(1)=0 ; z(1)=1500 ;
for i=1:N-1
x(i+1) = x(i) + (tf/N)*y(i) ;
y(i+1) = y(i) + (tf/N)*((rep(i)/z(i)));
z(i+1) = z(i) + (tf/N)*(-rep(i));
end % calcul de la trajectoire optimale
subplot(321) ; plot(linspace(0,tf,N),x(1:N)) ; title('Hauteur');grid ;

subplot(322) ; plot(linspace(0,tf,N),y(1:N)) ; title('Vitesse');grid ;

subplot(323) ; plot(linspace(0,tf,N),z(1:N)) ; title('Masse');grid ;

subplot(324) ; plot(linspace(0,tf,N),rep(1:N)) ; title('Contrôle u')
;grid;
```

```

function [c,ceq] = cond(x)

N = length(x)-1 ;

c = [];          % ou bien c=0

tf =x(end); yf = 0; zf = 1500; xf = 0;

for i=1:N-1

    xf = xf +(tf/N)*yf;          % calcul du point final au temps tf

    yf = yf + (tf/N)*((x(i)/zf)) ;    % avec la méthode d'Euler

    zf = zf +(tf/N)*(-x(i));

end

ceq = [zf-800 ] ;                %on impose la condition finale zf=800

xf(end)
yf(end)

%-----

function val = hfinal(x)

val = x(end) ;

```

FIG. 3.7 – La simulation de la méthode directe sous Matlab

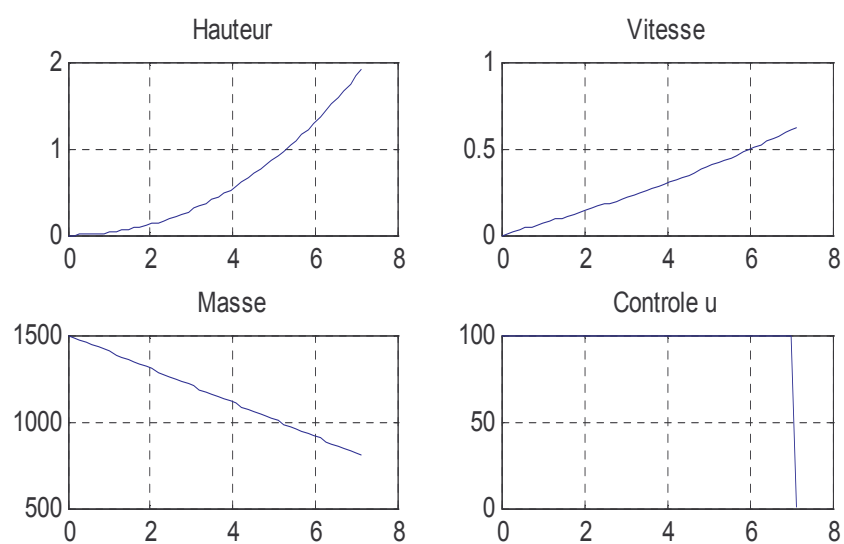


FIG. 3.8 – Les trajectoires optimales par la méthode directe

Conclusion générale

Dans ce travail on a considéré un problème de contrôle optimal avec une entrée non linéaire. On a modélisé un problème de décollage d'une fusée, en négligeant en premier lieu toutes les forces externes y compris le pesanteur g et on a considéré la masse de la fusée égale 1.

Numériquement on a programmé deux méthodes de résolution à savoir la méthode directe appelée aussi (tir directe), puis la méthode indirecte appelée méthode de tir. On a remarqué que la méthode indirecte est très sensible au choix du point de départ. On a résolu théoriquement le problème considéré, on a supposé que le point final est arbitraire (x_1, y_1) , on a considéré tous les cas possibles de x_1, y_1 .

Par la suite, on a modélisé le même problème en tenant compte de la pesanteur et surtout de la variation de la masse, ceci nous a conduit à un problème de contrôle optimal avec contrainte sur l'état final.

Théoriquement le vecteur adjoint résultant est discontinu. Pour cette raison la mise en œuvre de la méthode indirecte est compliquée.

Bibliographie

- [1] M. Aidene., I.L. Vorbev., and B. Oukacha. Algorithm for solving a linear optimal control problem with minimaux. *Performance Index Computational Mathematics and mathematical Physics.*, 45 vol 10 :1691–1700, 2005.
- [2] F. Boyer. A propos de contrôle et de l’optimisation de trajectoires. *Laboratoire d’Analyse Topologie et Probabilités, Aix Marseille University. Aix en Provence.*, Decembre 2010.
- [3] H. Brezis. Analyse fonctionnelle. théorie et application. *Masson Paris*, 1983.
- [4] P. Demailly. Analyse numérique et équations différentielles. *3e édition. EDP SCIENCES*, 2006.
- [5] R.E. Kalman. Mathematical discription of linear dynamical system siam. *journal on control.*, 1963.
- [6] K. Louadj. Résolution des problèmes paramétrés en contrôle optimal. *Thèse de doctorat*, page 116, 2012.
- [7] F. Mauger. Méthodes numériques pour la physique1.résolution des équations différentielles ordinaires. *University de Caen Basse-Normandie.*, 3 Septembre 2012.
- [8] N. Moussouni. Contrôle optimal : Optimisation d’une production cérélière. *Thèse de doctorat Université Mouloud Mammeri de Tizi-Ouzou*, page 102, 2012.
- [9] E. Neuman. Tutorial 1.getting started with matlab. *Departement of Mathematics. Southern Illinois university at Carbondale.*
- [10] B. Oukacha. Résolution de problème de contrôle optimal. *Thèse de doctorat Université Mouloud Mammeri de Tizi-Ouzou*, 2005.
- [11] L.S. Pontryagin., V.G. Boltyanskiand., and R.V. Gamkrelidze.and E.F. Mischenko. The mathematical theory of optimal processes. *Interscience Pubishers New York*, 1962.
- [12] L. Pontyagin. and al. Mathematical theory of optimal processes. *Eds Mir Moscou*, 43 :504, 1974.

- [13] B. Stout. Méthodes numériques de résolution d'équations différentielles. *University de Provence. Institut Fresnel. case 161 Faculté de St Jérôme. Marseille France.*, Février 2007.
- [14] E. Trélat. Contrôle optimal : Théorie et application. *Université Paris-Sud Laboratoire EN-EDP Mathématique UMR 8628 Bat 425 ; 91405 Orsay Cedex*, page 238., 2005.