

**École de Contrôle Optimal Numérique,
Événement CIMI, Toulouse, France, 3-7 septembre 2018.**

Méthodes indirectes (TP)

Olivier Cots & Joseph Gergaud

30 août 2018

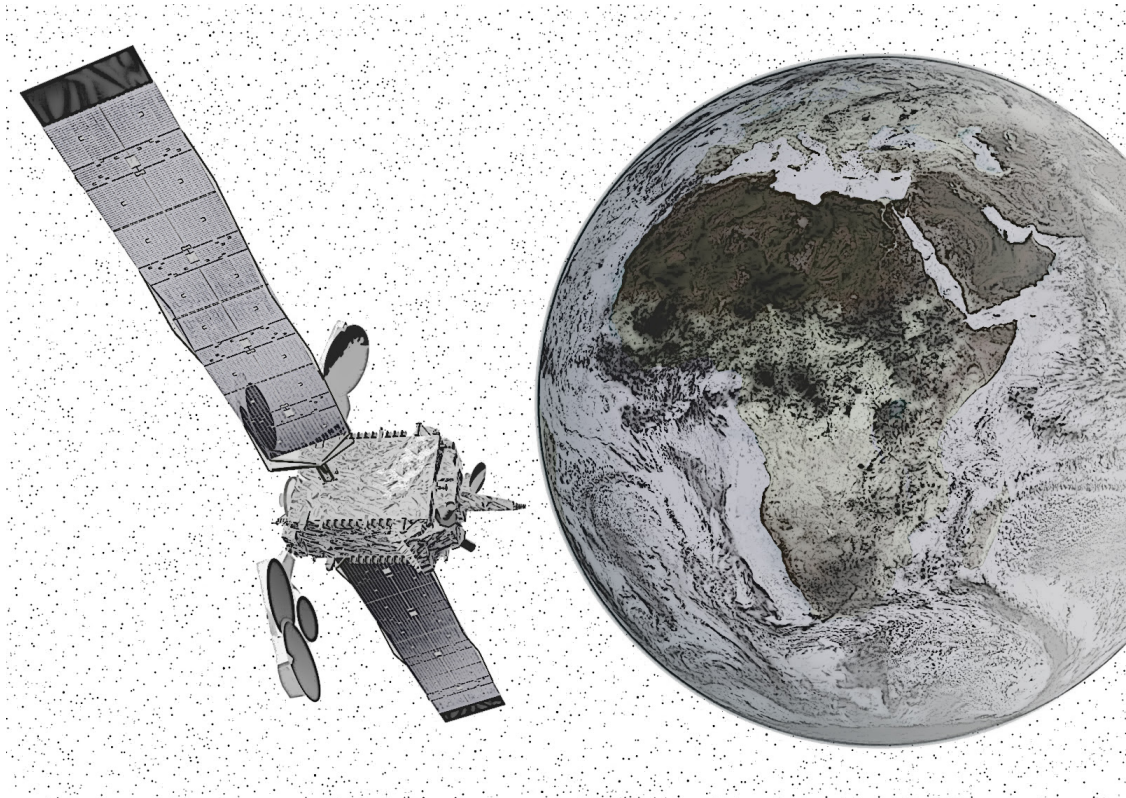


Table des matières

Sujet 1. Méthode de tir simple	1
1.1 Introduction	1
1.2 Présentation de la méthode de tir simple sur un exemple	1
1.2.1 Le problème de contrôle optimal	1
1.2.2 Application du principe du maximum de Pontryagin	1
1.2.3 Problème aux deux bouts	2
1.2.4 Fonction de tir et méthode de tir (simple)	2
1.2.5 Calcul du zéro de la fonction de tir	2
1.3 Présentation rapide de HamPath	3
1.3.1 Introduction	3
1.3.2 Schéma général de HamPath	3
1.4 Résolution numérique	4
1.4.1 Problème (P1)	4
1.4.2 Ajout de contraintes sur le contrôle	4
Sujet 2. Tir simple versus tir multiple	5
2.1 Double intégrateur : énergie min	5
2.2 Double intégrateur : temps min	6
2.2.1 Tir simple	6
2.2.2 Tir multiple	7
Sujet 3. Méthode homotopique, régularisation et tir multiple	8
3.1 Résolution du problème régularisé	8
3.2 Résolution du problème de minimisation L^1	9
Sujet 4. Transfert orbital	10
4.1 Introduction	10
4.2 Résolution du problème en temps minimal	10
Sujet A. (Annexe) Principe du Maximum de Pontryagin (PMP)	12
A.1 Formulation du problème de contrôle optimal	12
A.2 Principe du Maximum de Pontryagin	13
A.3 Définitions et propriétés importantes	14

Méthode de tir simple

1.1 Introduction

Nous allons dans cette partie voir une méthode de résolution numérique d'un problème simple de contrôle optimal utilisant les conditions nécessaires d'optimalité, c'est-à-dire le Principe du Maximum de Pontryagin (PMP), cf. annexe A. Cette méthode fait partie des méthodes dites indirectes et s'appelle la méthode de tir simple.

1.2 Présentation de la méthode de tir simple sur un exemple

1.2.1 Le problème de contrôle optimal

Considérons le problème de contrôle optimal suivant.

$$(P1) \quad \begin{cases} J(u(\cdot)) := \frac{1}{2} \int_0^{t_f} u(t)^2 dt \longrightarrow \min \\ \dot{x}(t) = -x(t) + u(t), \quad u(t) \in \mathbb{R}, \quad t \in [0, t_f] \text{ p.p.}, \quad x(0) = x_0, \\ x(t_f) = x_f, \end{cases}$$

avec $t_f := 1$, $x_0 := -1$, $x_f := 0$ et $\forall t \in [0, t_f]$, $x(t) \in \mathbb{R}$. Notons

$$H(x, p, p^0, u) := p(-x + u) + p^0 \frac{1}{2} u^2,$$

le pseudo-hamiltonien associé au problème (P1).

1.2.2 Application du principe du maximum de Pontryagin

D'après le PMP, si $u(\cdot)$ est une solution optimal de (P1) (avec $x(\cdot)$ la trajectoire associée) alors il existe un vecteur adjoint $p(\cdot) \in AC([0, t_f], \mathbb{R})$, un scalaire $p^0 \leq 0$, tels que $(p(\cdot), p^0) \neq (0, 0)$, et tels que les équations suivantes sont vérifiées pour $t \in [0, t_f]$ p.p. :

$$\begin{cases} \dot{x}(t) &= \partial_p H[t] = -x(t) + u(t), \\ \dot{p}(t) &= -\partial_x H[t] = p(t), \\ 0 &= \partial_u H[t] = p(t) + p^0 u(t), \end{cases}$$

où $[t] := (x(t), p(t), p^0, u(t))$. Il n'y a pas d'anormale car si $p^0 = 0$ alors $p(\cdot) \equiv 0$ ce qui est impossible. Ainsi, on a $p^0 < 0$. Notons $u_s(x, p, p^0) := -p/p^0$, la solution de l'équation $0 = \partial_u H(x, p, p^0, u)$ pour (x, p, p^0) fixé. On peut alors fixer arbitrairement $p^0 \neq 0$ car pour tout $\alpha \in \mathbb{R}_+^*$, $u_s(x, \alpha p, \alpha p^0) = u_s(x, p, p^0)$ et la trajectoire associée reste inchangée. Fixons $p^0 = -1$ et notons maintenant

$$u_s(x, p) = p$$

le contrôle optimal. De même, on notera $H(x, p, u)$ le hamiltonien tel que $p^0 = -1$.

1.2.3 Problème aux deux bouts

L'application du PMP nous mène à résoudre le *problème aux deux bouts* (Two Points Boundary Value Problem) suivant :

$$(P2) \quad \begin{cases} \dot{x}(t) &= -x(t) + u_s(x(t), p(t)) = -x(t) + p(t), \\ \dot{p}(t) &= p(t), \\ x(0) &= x_0, \quad x(t_f) = x_f. \end{cases}$$

L'inconnue de ce problème aux deux bouts est le vecteur adjoint initial $p(0)$. En effet si l'on fixe $p_0 := p(0)$ alors d'après le théorème de Cauchy–Lipschitz, il existe une unique solution maximale $z(\cdot, x_0, p_0) := (x(\cdot, x_0, p_0), p(\cdot, x_0, p_0))$ vérifiant la dynamique (sur x et p) et la condition initiale $z(0, x_0, p_0) = (x_0, p_0)$. Le problème est donc de trouver p_0 tel que $x(t_f, x_0, p_0) = x_f$.

1.2.4 Fonction de tir et méthode de tir (simple)


On va transformer le problème aux deux bouts (P2) en un système d'équations non linéaires, que l'on appelle équations de tir. Pour cela, on définit tout d'abord le système hamiltonien


$$\vec{H}(x, p, u) := \left(\frac{\partial H}{\partial p}(x, p, u), -\frac{\partial H}{\partial x}(x, p, u) \right).$$

On note $z := (x, p)$, puis $z(\cdot, x_0, p_0)$ la solution de l'équation différentielle $\dot{z}(t) = \vec{H}(z(t), u_s(z(t)))$ vérifiant $z(0, x_0, p_0) = (x_0, p_0)$. On définit enfin la *fonction de tir* suivante :

$$\begin{aligned} S: \mathbb{R} &\longrightarrow \mathbb{R} \\ y &\longmapsto S(y) := \Pi_x(z(t_f, x_0, y)) - x_f, \end{aligned} \quad (1.1)$$

où Π_x est simplement la projection canonique $\Pi_x(x, p) = x$. Résoudre le problème aux deux bouts (P2) revient à trouver un zéro de la fonction de tir, *i.e.* consiste à résoudre $S(y) = 0$. C'est ce que l'on appelle la *méthode de tir simple*. On utilise alors une méthode de type Newton pour la résolution, à laquelle nous pouvons fournir la jacobienne de S .

Remarque 1.1.  Si \bar{p}_0 vérifie $S(\bar{p}_0) = 0$, alors la courbe intégrale $\bar{z}(\cdot) := z(\cdot, x_0, \bar{p}_0)$, avec le contrôle $\bar{u}(\cdot) := u_s(\bar{z}(\cdot))$, est une BC-extrémale du problème (P1), *i.e.* cette extrémale satisfait les conditions nécessaires d'optimalité données par le PMP.

Remarque 1.2.  La courbe intégrale $z(\cdot, z_0)$ est aussi solution du système hamiltonien $\dot{z}(t) = \vec{h}(z(t))$, $z(0) = z_0$, où $h(z) := H(z, u_s(z))$ et $\vec{h} := (\partial_p h, -\partial_x h)$.

1.2.5 Calcul du zéro de la fonction de tir

Calculons la solution à la main sur cet exemple simple.

$$\dot{p}(t) = p(t) \implies p(t) = e^t p_0, \quad p_0 := p(0) \implies x(t) = (0.5 p_0 e^{2t} + C) e^{-t}, \quad C \in \mathbb{R}.$$

Or $x(0) = x_0$ donc $x(t) = (0.5 p_0 (e^{2t} - 1) + x_0) e^{-t}$ et finalement, puisque $x_0 = -1$, $x(t_f) = x_f = 0$ et $t_f = 1$, on a

$$\bar{p}_0 = \frac{2(x_f e^{t_f} - x_0)}{e^{2t_f} - 1} = \frac{2}{e^2 - 1} \approx 0.313.$$

La fonction de tir est ici : $p_0 \mapsto (0.5 p_0 (e^{2t_f} - 1) + x_0) e^{-t_f} - x_f$.

1.3 Présentation rapide de HamPath

1.3.1 Introduction

Le code `HamPath` est développé par J.-B. Caillaud, O. Cots et J. Gergaud. Ce code permet :

- ⊛ de résoudre les problèmes de contrôle optimal à commandes continues via la méthode du tir simple indirect ;
- ⊛ de résoudre ceux à commandes non continues via la méthode du tir multiple indirect ;
- ⊛ de résoudre les familles de problèmes de contrôle optimal à un paramètre via la méthode homotopique de suivi de chemin différentiel ;
- ⊛ de vérifier les conditions du deuxième ordre via le calcul de points conjugués.

1.3.2 Schéma général de HamPath

L'utilisateur fournit à `HamPath` en FORTRAN la fonction de tir $S(y)$ et le hamiltonien maximisé $h(z) := \max_u H(z, p^0, u)$. Après compilation, `HamPath` fournit des fonctions MATLAB (ou PYTHON) : `hfun`, `hvfun`, `exphvfun`, `dhvfun`, `expdhvfun`, `sfun`, `sjac`, `ssolve`, `hampath`...

Remarque 1.3. La notation `exphvfun` vient de la notation mathématique $\exp(t_f \vec{h})(z_0) := z(t_f, z_0)$, où $z(\cdot, z_0)$ est la solution de l'edo $\dot{z}(t) = \vec{h}(z(t))$, $z(0) = z_0$.

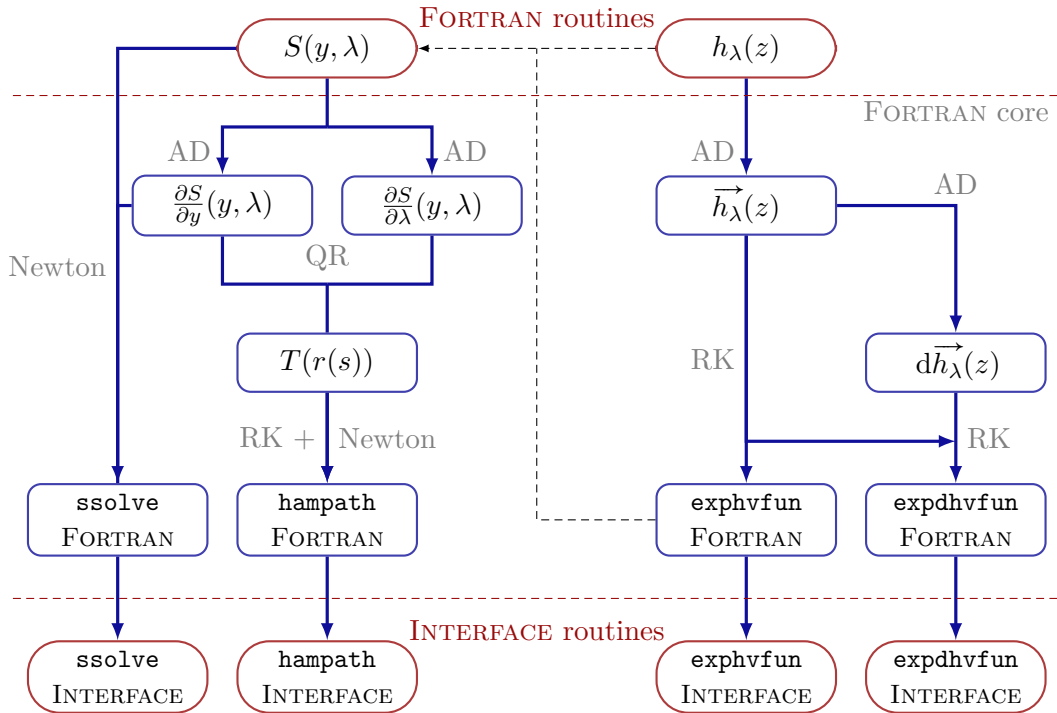


FIGURE 1.1 – Schéma général de `HamPath`. AD signifie Différentiation Automatique, RK signifie schémas d'intégration numérique de Runge-Kutta utilisés pour la résolution d'équations différentielles ordinaires. L'appellation Newton est employée dès qu'un solveur d'équations non linéaires de type Newton est appelé et enfin QR veut dire factorisation QR.

1.4 Résolution numérique

Remarque 1.4. Se rendre dans le répertoire TP_MI et exécuter la commande (dans un terminal) :

```
source hampath_on
```

Cette commande doit être exécutée à chaque ouverture d'un nouveau terminal.

1.4.1 Problème (P1)

- ▷ **Exercice 1.1.** Se rendre dans le répertoire TP_1_tir_simple_pb_scalaire/exo_1.
1. Jeter un œil à la routine `control` du fichier `afun.f90` codant le contrôle $u_s(x, p, p^0) = -p/p^0$, $p^0 = -1$.
 2. Jeter un œil à la routine `hfun` de `hfun.f90` codant le hamiltonien maximisé $h(z) := H(z, p^0, u_s(z, p^0))$, $p^0 = -1$, $z := (x, p)$ et $H(z, p^0, u) = p(-x + u) + p^0 u^2/2$.
 3. Jeter un œil à la routine `sfun` de `sfun.f90` codant la fonction de tir (1.1). On remarquera l'utilisation de la routine FORTRAN `exphv` pour l'intégration numérique du système hamiltonien.
 4. Compiler le problème : exécuter la commande `hampathCIMI` dans un terminal dans le répertoire de définition du problème.
 5. Jeter un œil au script MATLAB `main11.m` puis l'exécuter. Vérifier que l'on retrouve bien $\bar{p}_0 = \frac{2}{e^2 - 1}$.

Remarque 1.5. 📌 Les paramètres du problème sont transmis via le vecteur `par`, des fonctions MATLAB vers les routines FORTRAN.

1.4.2 Ajout de contraintes sur le contrôle

On considère le problème de contrôle (P1) où l'on ajoute la contrainte sur le contrôle par $u(t) \in [-u_{\max}, u_{\max}]$, $u_{\max} := 1$. La condition de maximisation nous donne comme loi de contrôle :

$$u(t) = \begin{cases} u_s(x(t), p(t), p^0) = -p(t)/p^0 & \text{si } |p(t)| \leq u_{\max}, \\ +u_{\max} & \text{si } p(t) > u_{\max}, \\ -u_{\max} & \text{si } p(t) < -u_{\max}. \end{cases}$$

- ▷ **Exercice 1.2.** Se rendre dans le répertoire TP_1_tir_simple_pb_scalaire/exo_2_contraintes_sur_u.
1. Décommenter la partie codant pour l'ajout de la contrainte sur le contrôle dans le fichier `afun.f90`. Attention, u_{\max} est donné dans le vecteur `par`.
 2. Compiler le problème.
 3. Jeter un œil au script MATLAB `main12.m` puis l'exécuter.

Tir simple versus tir multiple

2.1 Double intégrateur : énergie min

Considérons le problème de contrôle optimal suivant :

$$(P3) \quad \begin{cases} J(u(\cdot)) := \frac{1}{2} \int_0^{t_f} u(t)^2 dt \longrightarrow \min \\ \dot{x}_1(t) = x_2(t) \\ \dot{x}_2(t) = u(t), \quad |u(t)| \leq u_{\max}, \quad t \in [0, t_f] \text{ p.p.}, \\ x(0) = x_0, \quad x(t_f) = x_f, \end{cases}$$

avec $t_f := 1$, $x_0 := (-1, 0)$, $x_f := (0, 0)$ et $\forall t \in [0, t_f]$, $x(t) = (x_1(t), x_2(t)) \in \mathbb{R}^2$. Notons

$$H(x, p, p^0, u) := p_1 x_2 + p_2 u + p^0 \frac{1}{2} u^2, \quad x = (x_1, x_2), \quad p = (p_1, p_2),$$

le pseudo-hamiltonien. On considère le cas normal et on fixe $p^0 = -1$. La condition de maximisation nous donne comme loi de contrôle :

$$u(t) = \begin{cases} u_s(x(t), p(t)) := p_2(t) & \text{si } |p_2(t)| \leq u_{\max}, \\ +u_{\max} & \text{si } p_2(t) > u_{\max}, \\ -u_{\max} & \text{si } p_2(t) < -u_{\max}. \end{cases}$$

La fonction de tir est donnée par :

$$\begin{aligned} S: \mathbb{R}^2 &\longrightarrow \mathbb{R}^2 \\ y &\longmapsto S(y) := x(t_f, x_0, y) - x_f, \end{aligned}$$

où y joue ici le rôle du vecteur adjoint initial $p(0)$.

► **Exercice 2.1.** Se rendre dans le répertoire

TP_2_tir_simple_et_multiple_double_integrateur/exo_1_energie_min.

1. Compléter `control`, `hfun` et `sfun`, respectivement dans les fichiers `afun.f90`, `hfun.f90` et `sfun.f90`. Attention pour `sfun`, même si $x_f = (0, 0)$, il est passé en paramètre à l'aide du vecteur `par` : `x_f = par(5 : 6)`.
2. Compiler le problème.
3. Jeter un œil au script MATLAB `main21.m` puis l'exécuter. Vous pouvez tester la résolution du problème pour deux valeurs de u_{\max} .

Remarque 2.1. 🏹 La loi de commande $\bar{u}(\cdot)$ associée à la solution des équations de tir est lisse lorsque $u_{\max} \geq 6$. Pour $u_{\max} < 6$, la loi de commande perd en régularité. Pour $u_{\max} = 4.5$, elle est continue mais pas C^1 , seulement C^1 par morceaux (elle est même lisse par morceaux).

2.2 Double intégrateur : temps min

Considérons le problème (P3) où l'on remplace le critère par le suivant : $J(u(\cdot), t_f) := t_f$. On remarque alors que dans ce cas, le temps final t_f est libre. Fixons de plus la borne sur le contrôle : $u_{\max} = 1$. Notons le pseudo-hamiltonien

$$H(x, p, u) := p_1 x_2 + p_2 u.$$

La condition de maximisation nous donne comme loi de contrôle :

$$u(t) = \begin{cases} +u_{\max} & \text{si } p_2(t) > 0, \\ -u_{\max} & \text{si } p_2(t) < 0, \\ ? & \text{si } p_2(t) = 0. \end{cases}$$

Remarque 2.2.  Il n'est pas possible d'avoir $p_2(t) = 0$ sur un intervalle de temps non réduit à un point. Ainsi, nous pouvons ignorer ce cas.

Puisque t_f est libre, le PMP nous donne la condition de transversalité

$$H(x(t_f), p(t_f), u(t_f)) = -p^0,$$

où l'on fixe toujours $p^0 = -1$ car nous ne considérons que le cas normal.


2.2.1 Tir simple

La fonction de tir simple est donnée par :

$$\begin{aligned} S: \mathbb{R}^3 &\longrightarrow \mathbb{R}^3 \\ y := (p_0, t_f) &\longmapsto S(y) := \begin{bmatrix} x(t_f, x_0, p_0) - x_f \\ H(x(t_f, x_0, p_0), p(t_f, x_0, p_0), u(t_f)) + p^0 \end{bmatrix}. \end{aligned}$$

▷ **Exercice 2.2.** Se rendre dans le répertoire
TP_2_tir_simple_et_multiple_double_integrateur/
exo_2_temps_min_tir_simple.

1. Compléter `control`, `hfun` et `sfun`, respectivement dans les fichiers `afun.f90`, `hfun.f90` et `sfun.f90`. Attention pour `sfun`, même si $x_f = (0, 0)$, il est passé en paramètre à l'aide du vecteur `par` : $x_f = \text{par}(3 : 4)$.
2. Compiler le problème.
3. Jeter un œil au script MATLAB `main22.m` puis l'exécuter.

Remarque 2.3.  Remarquer que le solveur `fsolve` de MATLAB ne fait qu'une itération mais ne progresse pas.

2.2.2 Tir multiple

Nous introduisons quelques notations pour clarifier l'expression de la fonction de tir multiple que l'on définit ci-après. On note $H_+(z) := H(z, u_{\max})$, $H_-(z) := H(z, -u_{\max})$ et \overrightarrow{H}_+ , \overrightarrow{H}_- les systèmes hamiltoniens associés. Pour définir la fonction de tir multiple, nous devons nécessairement connaître la structure du contrôle. Ici, nous savons que la structure contient deux arcs : $+u_{\max}$ suivi de $-u_{\max}$. Le PMP nous donne qu'à l'instant de la commutation (noté t_1), on a $p_2(t_1) = 0$. Nous pouvons maintenant définir la fonction de tir multiple (de structure), qui est donnée par :

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

$$y := (p_0, t_f, t_1) \longmapsto S(y) := \begin{bmatrix} \Pi_x(z_f) - x_f \\ H_-(z_f) + p^0 \\ \Pi_{p_2}(z_1) \end{bmatrix},$$

où

$$z_1 := \exp(t_1 \overrightarrow{H}_+)(x_0, p_0), \quad z_f := \exp((t_f - t_1) \overrightarrow{H}_-)(z_1).$$

On rappelle que $\Pi_x(z) = x$ avec $z = (x, p)$, $p = (p_1, p_2)$, et on définit $\Pi_{p_2}(z) = p_2$.

▷ **Exercice 2.3.** Se rendre dans le répertoire
 TP_2_tir_simple_et_multiple_double_integrateur/
 exo_3_temps_min_tir_multiple_de_structure.

1. Jeter un œil à la routine `control` de `afun.f90`. Noter l'utilisation de la variable `iarc` indiquant si l'on se trouve sur le premier ou le second arc.
2. Jeter un œil à la routine `hfun` de `hfun.f90`. Elle reste inchangée.
3. Compléter `sfun` de `sfun.f90`. Attention à bien utiliser la variable `iarc`.
4. Compiler le problème.
5. Jeter un œil au script MATLAB `main23.m` puis l'exécuter.

Remarque 2.4. 🦋 Comparer la précision de la solution numérique entre le tir multiple et le tir simple. On notera que pour le tir multiple, nous avons besoin d'une information supplémentaire : la structure du contrôle optimal.

Sujet 3

Méthode homotopique, régularisation et tir multiple

Considérons le problème de contrôle optimal suivant (double intégrateur, minimisation L^1) :

$$(P4) \quad \begin{cases} J(u(\cdot)) := \int_0^{t_f} |u(t)| \, dt \longrightarrow \min \\ \dot{x}_1(t) = x_2(t) \\ \dot{x}_2(t) = u(t), \quad |u(t)| \leq u_{\max}, \quad t \in [0, t_f] \text{ p.p.}, \\ x(0) = x_0, \quad x(t_f) = x_f, \end{cases}$$

avec $t_f := 3$, $x_0 := (-1, 0)$, $x_f := (0, 0)$ et $\forall t \in [0, t_f]$, $x(t) = (x_1(t), x_2(t)) \in \mathbb{R}^2$. Notons

$$H(x, p, p^0, u) := p_1 x_2 + p_2 u + p^0 |u|, \quad x = (x_1, x_2), \quad p = (p_1, p_2),$$

le pseudo-hamiltonien. On considère le cas normal et on fixe $p^0 = -1$. La condition de maximisation nous donne comme loi de contrôle :

$$u(t) = \begin{cases} \text{sign}(p_2(t)) u_{\max} & \text{si } |p_2(t)| + p^0 > 0, \\ 0 & \text{si } |p_2(t)| + p^0 < 0, \\ ? & \text{si } |p_2(t)| + p^0 = 0. \end{cases}$$

Nous voulons ici résoudre le problème par une méthode de tir multiple. Nous avons donc besoin dans un premier temps, de déterminer la structure du contrôle. Pour ce faire, nous pourrions utiliser une méthode de tir simple en prenant certaines précautions, notamment sur le calcul de la jacobienne de la fonction de tir. Une autre possibilité est d'utiliser une méthode homotopique différentielle de suivi de chemin combinée à une méthode de régularisation. Nous introduisons donc le critère suivant afin de régulariser le contrôle optimal :

$$J_\lambda(u(\cdot)) := \int_0^{t_f} |u(t)|^{2-\lambda} \, dt, \quad \lambda \in [0, 1[. \quad (3.1)$$

3.1 Résolution du problème régularisé

Nous considérons ici le problème (P4) où l'on remplace le critère J par J_λ défini en (3.1). Le pseudo-hamiltonien est donné par :

$$H_\lambda(x, p, p^0, u) := p_1 x_2 + p_2 u + p^0 |u|^{2-\lambda}$$

et le contrôle maximisant s'écrit

$$u(t) = \text{sign}(p_2(t)) \left(-\frac{|p_2(t)|}{p^0 (2-\lambda)} \right)^{\frac{1}{1-\lambda}}.$$

La fonction de tir est simplement donnée par :

$$\begin{aligned} S: \mathbb{R}^2 &\longrightarrow \mathbb{R}^2 \\ y &\longmapsto S(y) := x(t_f, x_0, y) - x_f. \end{aligned}$$

▷ **Exercice 3.1.** Se rendre dans le répertoire
`TP3_homotopie_regularisation_double_integrateur/exo_1_regularisation.`

1. Compléter seulement `hfun` dans `hfun.f90`.
2. Compiler le problème.
3. Jeter un œil au script MATLAB `main31.m` puis l'exécuter.

3.2 Résolution du problème de minimisation L^1

D'après la régularisation (pour λ proche de 1), le contrôle optimal solution du problème (P4) possède trois arcs : $u = u_{\max}$ suivi de $u = 0$ puis $u = -u_{\max}$. On note $H_+(z) := H(z, p^0, u_{\max})$, $H_-(z) := H(z, p^0, -u_{\max})$, $H_0(z) := H(z, p^0, 0)$ et \overrightarrow{H}_+ , \overrightarrow{H}_- , \overrightarrow{H}_0 les systèmes hamiltoniens associés. La fonction de tir correspondant à cette structure s'écrit :

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

$$y := (p_0, t_1, t_2) \longmapsto S(y) := \begin{bmatrix} \Pi_{p_2}(z_1) + p^0 \\ \Pi_{p_2}(z_2) - p^0 \\ \Pi_x(z_f) - x_f \end{bmatrix},$$

où

$$z_1 := \exp(t_1 \overrightarrow{H}_+)(x_0, p_0), \quad z_2 := \exp((t_2 - t_1) \overrightarrow{H}_0)(z_1), \quad z_f := \exp((t_f - t_2) \overrightarrow{H}_-)(z_2).$$

▷ **Exercice 3.2.** Se rendre dans le répertoire
`TP3_homotopie_regularisation_double_integrateur/exo_2_min_L1.`

1. Compléter `control`, `hfun` et `sfun`, respectivement dans les fichiers `afun.f90`, `hfun.f90` et `sfun.f90`.
2. Compiler le problème.
3. Compléter le script MATLAB `main32.m` puis l'exécuter. Il faut fournir au solveur des équations de tir, la prédiction initiale du vecteur adjoint initial p_0 et des instants de commutations t_1 et t_2 . Utiliser pour cela, la solution du problème régularisé avec λ proche de 1.

Sujet 4

Transfert orbital

4.1 Introduction

Le système considéré est un satellite de masse fixé m libéré par une fusée dans le plan de l'équateur ; l'orbite initiale du satellite est une ellipse de forte excentricité, voir figure 4.1. L'objectif de ce travail est de réaliser le transfert en temps minimal de cette orbite elliptique à une orbite circulaire géostationnaire.

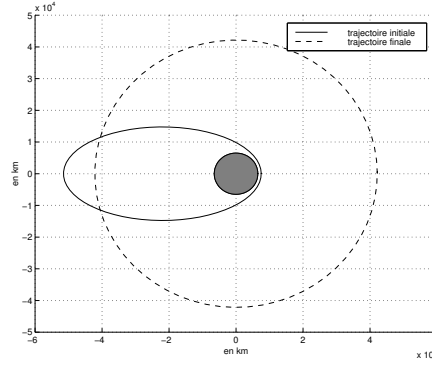


FIGURE 4.1 – Transfert orbital 2D.

4.2 Résolution du problème en temps minimal

Considérons le problème de transfert orbital à temps minimal suivant :

$$(P5) \quad \begin{cases} J(u(\cdot), t_f) = t_f \longrightarrow \min \\ \dot{x}_1(t) = x_3(t), \\ \dot{x}_2(t) = x_4(t), \\ \dot{x}_3(t) = -\frac{\mu x_1(t)}{r(t)^3} + u_1(t), \\ \dot{x}_4(t) = -\frac{\mu x_2(t)}{r(t)^3} + u_2(t), \quad \|u(t)\| \leq \gamma_{\max}, \quad t \in [0, t_f] \text{ p.p.}, \quad u(t) := (u_1(t), u_2(t)), \\ x_1(0) = x_{0,1}, \quad x_2(0) = x_{0,2}, \quad x_3(0) = x_{0,3}, \quad x_4(0) = x_{0,4}, \\ r(t_f)^2 = r_f^2, \quad x_3(t_f) = -\sqrt{\frac{\mu}{r_f^3}} x_2(t_f), \quad x_4(t_f) = \sqrt{\frac{\mu}{r_f^3}} x_1(t_f), \end{cases}$$

avec $r(t) := \sqrt{x_1(t)^2 + x_2(t)^2}$. Les unités choisies sont le kilomètre pour les distances et l'heure pour les temps, cf. table 4.1.

Paramètre	Valeur	Unité
μ	$5.1658620912 \times 10^{12}$	$\text{km}^3 \text{ h}^{-2}$
r_f	42165	km
γ_{\max}	388.8	km h^{-2}

TABLE 4.1 – Unité et valeurs des paramètres constants. Cette valeur de γ_{\max} correspond à une accélération de 60N et à une masse de 2000kg, cf. $\gamma_{\max} = \frac{F_{\max}}{m} = \frac{60 \times 3600^2}{2000 \times 10^3} = 388.8$.

Le pseudo-hamiltonien est donné par :

$$H(x, p, u) = x_3 p_1 + x_4 p_2 + \left(u_1 - \frac{\mu x_1}{\|r\|^3} \right) p_3 + \left(u_2 - \frac{\mu x_2}{\|r\|^3} \right) p_4.$$

On considère le cas normal et on fixe $p^0 = -1$. La condition de maximisation nous donne comme loi de commande :

$$u(t) = \bar{u}(z(t)) := \frac{\gamma_{\max}}{\sqrt{p_3(t)^2 + p_4(t)^2}} (p_3(t), p_4(t)).$$

Le temps final étant libre, on a la condition au temps final $H(x(t_f), p(t_f), u(t_f)) = -p^0$ et puisque le point terminal sur l'orbite final n'est pas entièrement fixé, le PMP nous donne la condition de transversalité en t_f suivante :

$$x_2(t_f) \left(p_1(t_f) + \sqrt{\frac{\mu}{r_f^3}} p_4(t_f) \right) = x_1(t_f) \left(p_2(t_f) - \sqrt{\frac{\mu}{r_f^3}} p_3(t_f) \right).$$

En introduisant $\alpha := \sqrt{\frac{\mu}{r_f^3}}$, la fonction de tir multiple s'écrit :

$$S: \mathbb{R}^5 \longrightarrow \mathbb{R}^5$$

$$(p_0, t_f) \longmapsto \begin{bmatrix} \sqrt{x_1(t_f, z_0)^2 + x_2(t_f, z_0)^2} - r_f \\ x_3(t_f, z_0) + \alpha x_2(t_f, z_0) \\ x_4(t_f, z_0) - \alpha x_1(t_f, z_0) \\ x_2(t_f, z_0) (p_1(t_f, z_0) + \alpha p_4(t_f, z_0)) - x_1(t_f, z_0) (p_2(t_f, z_0) - \alpha p_3(t_f, z_0)) \\ H(z(t_f, z_0), \bar{u}(z(t_f, z_0))) + p^0 \end{bmatrix},$$

où $z_0 := (x_0, p_0)$, $x_0 := (x_{0,1}, x_{0,2}, x_{0,3}, x_{0,4})$, $z(\cdot, z_0) := (x(\cdot, z_0), p(\cdot, z_0))$ et $x := (x_1, x_2, x_3, x_4)$, $p := (p_1, p_2, p_3, p_4)$.

Remarque 4.1. Pour la résolution numérique, il est préférable de remplacer l'équation $r(t_f)^2 = r_f^2$ par $r(t_f) = r_f$.

▷ **Exercice 4.1.** Se rendre dans le répertoire TP4_transfert_orbital_temps_min.

1. Compléter seulement `sfun` dans `sfun.f90`.
2. Compiler le problème.
3. Jeter un œil au script MATLAB `main41.m` puis l'exécuter.

ANNEXE A

(Annexe) Principe du Maximum de Pontryagin (PMP)

Dans ce chapitre nous présentons le *principe du maximum de Pontryagin* :

L. S. Pontryagin, V. G. Boltyanskiï, R. V. Gamkrelidze & E. F. Mishchenko,
The Mathematical Theory of Optimal Processes, Translated from the Russian by K.
 N. Tirogoff, edited by L. W. Neustadt, Interscience Publishers John Wiley & Sons,
 Inc., New York-London, 1962.

Pour un problème de contrôle optimal donné, ce principe nous fournit des conditions nécessaires d'optimalité que doit vérifier toute solution de ce problème.

A.1 Formulation du problème de contrôle optimal

Considérons le problème de contrôle optimal suivant :

$$(OCP) \quad \begin{cases} J(x_0, u(\cdot), t_0, t_f) := g(t_0, x_0, t_f, x(t_f)) + \int_{t_0}^{t_f} f^0(t, x(t), u(t)) \, dt \longrightarrow \min \\ \dot{x}(t) = f(t, x(t), u(t)), \quad u(t) \in U, \quad t \in [t_0, t_f] \text{ p.p.}, \quad x(t_0) = x_0, \\ c(t_0, x_0, t_f, x(t_f)) = 0, \end{cases}$$

C'est un problème très général sous la forme de Bolza avec une dynamique non autonome, dont les instants initial et final sont libres, où l'on considère des conditions aux limites mélangées et où le contrôle peut éventuellement être contraint. On définit les hypothèses suivantes.

Hypothèse HA.1. Considérons une EDO contrôlée non autonome

$\dot{x}(t) = f(t, x(t), u(t))$, où f est une application de classe C^1 de $\mathcal{I} \times \Omega \times U$ dans \mathbb{R}^n , \mathcal{I} un intervalle ouvert de \mathbb{R} , Ω un ouvert de \mathbb{R}^n et U un ensemble **quelconque** de \mathbb{R}^m .

Considérons de plus une fonction f^0 de classe C^1 sur $\mathcal{I} \times \Omega \times U$ et deux applications¹ g et c de classe C^1 de $\mathcal{I} \times \Omega \times \mathcal{I} \times \Omega$ respectivement sur \mathbb{R} et \mathbb{R}^p , $p \leq 2(n+1)$. Notons

$$X_c := \{(t_0, x_0, t_f, x_f) \in \mathcal{I} \times \Omega \times \mathcal{I} \times \Omega \mid c(t_0, x_0, t_f, x_f) = 0\}$$

et supposons que $c'(t_0, x_0, t_f, x_f)$ est surjective pour tout $(t_0, x_0, t_f, x_f) \in X_c$.

Remarque A.1. On peut supposer seulement que f et f^0 sont C^0 par rapport à t et u .

On cherche alors une solution $(x_0, u(\cdot), t_0, t_f)$, $x_0 \in \Omega$, $u(\cdot) \in L^\infty([t_0, t_f], U)$, $0 \leq t_0 \leq t_f$ dans \mathbb{R} , telle que la trajectoire associée $\mathbf{t} \mapsto \mathbf{x}(\mathbf{t}) := \mathbf{x}(\mathbf{t}, \mathbf{t}_0, \mathbf{x}_0, \mathbf{u}(\cdot))$ soit définie sur $[t_0, t_f]$ et telle que la contrainte différentielle, la contrainte sur le contrôle et les conditions aux limites soient vérifiées, et qui de plus minimise le critère.

1. On notera (t_0, x_0, t_f, x_f) l'argument des applications g et c . On écrira donc $\frac{\partial g}{\partial x_f}$ la dérivée partielle de g par rapport à la quatrième variable.

A.2 Principe du Maximum de Pontryagin

Énonçons maintenant le Principe du Maximum de Pontryagin (PMP).

Théorème A.1 (Principe du Maximum de Pontryagin). Si $(x_0, u(\cdot), t_0, t_f)$, avec $x(\cdot)$ la trajectoire associée, est solution du problème (OCP) sous les hypothèses HA.1, alors il existe un vecteur adjoint $p(\cdot) \in AC([t_0, t_f], (\mathbb{R}^n)^*)$, un réel $p^0 \leq 0$, tels que $(p(\cdot), p^0) \neq (0, 0)$, et un multiplicateur $\lambda \in (\mathbb{R}^p)^*$, tels que les équations suivantes sont vérifiées pour $t \in [t_0, t_f]$ p.p. :

$$\begin{aligned}\dot{x}(t) &= \frac{\partial H}{\partial p}(t, x(t), p(t), p^0, u(t)), \\ \dot{p}(t) &= -\frac{\partial H}{\partial x}(t, x(t), p(t), p^0, u(t)),\end{aligned}\tag{A.1}$$

où $H(t, x, p, p^0, u) := p f(t, x, u) + p^0 f^0(t, x, u)$ est le pseudo-hamiltonien associé au problème (OCP), et on a la condition de maximisation du hamiltonien suivante pour $t \in [t_0, t_f]$ p.p. :

$$H(t, x(t), p(t), p^0, u(t)) = \max_{w \in U} H(t, x(t), p(t), p^0, w).\tag{A.2}$$

Les conditions aux limites $c(t_0, x_0, t_f, x(t_f)) = 0$ et $x(t_0) = x_0$ sont vérifiées et on a en plus les conditions de transversalité suivantes :

$$p(t_0) = -\left(\lambda \frac{\partial c}{\partial x_0} + p^0 \frac{\partial g}{\partial x_0}\right), \quad p(t_f) = \left(\lambda \frac{\partial c}{\partial x_f} + p^0 \frac{\partial g}{\partial x_f}\right),\tag{A.3}$$

appliqué en $(t_0, x_0, t_f, x(t_f))$. Enfin, puisque les temps initial et final sont libres, si $u(\cdot)$ est continu aux temps t_0 , respectivement t_f , alors on a les conditions sur le hamiltonien suivantes :

$$H[t_0] = \left(\lambda \frac{\partial c}{\partial t_0} + p^0 \frac{\partial g}{\partial t_0}\right), \quad H[t_f] = -\left(\lambda \frac{\partial c}{\partial t_f} + p^0 \frac{\partial g}{\partial t_f}\right),\tag{A.4}$$

toujours appliqué en $(t_0, x_0, t_f, x(t_f))$ et où $[t] := (t, x(t), p(t), p^0, u(t))$.

Remarque A.2. On rappelle : $\lambda \frac{\partial c}{\partial x_0} = \sum_{i=1}^p \lambda_i \frac{\partial c_i}{\partial x_0}$, avec $\lambda := (\lambda_1, \dots, \lambda_p)$.

Remarque A.3. La convention $p^0 \leq 0$ conduit au principe du *maximum* tandis que $p^0 \geq 0$ conduit au principe du *minimum*, i.e. la condition (A.2) serait une minimisation.

Remarque A.4. Dans le cas où U est un ouvert de \mathbb{R}^m , i.e. lorsqu'il n'y a pas de contraintes sur le contrôle, la condition de maximisation (A.2) implique $\partial_u H[t] = 0$.

A.3 Définitions et propriétés importantes

Définition A.1. On introduit les définitions suivantes.

- ⊛ Une *extrémale* du problème (OCP) est un quadruplet $(x(\cdot), p(\cdot), p^0, u(\cdot))$ solution des *équations hamiltoniennes* (A.1) et de la *condition de maximisation* (A.2).
- ⊛ On parle de *BC-extrémale* (BC vient de “Boundary Conditions”) si l’extrémale vérifie en plus les *conditions aux limites* $c(t_0, x_0, t_f, x(t_f)) = 0$, $x(t_0) = x_0$, les *conditions de transversalité* (A.3) et les *conditions sur le hamiltonien* (A.4).
- ⊛ Une extrémale $(x(\cdot), p(\cdot), p^0, u(\cdot))$ est dite *anormale* si $p^0 = 0$ et *normale* dans le cas contraire. Dans le cas normal on peut fixer p^0 à -1 par exemple, par homogénéité.
- ⊛ Une extrémale définie sur un intervalle $I \subset [t_0, t_f]$, $t_0 < t_f$, est dite *singulière* si $\partial_u H[t] = 0$ pour tout $t \in I$, elle est dite *régulière* sinon. On appelle *arc bang*, une portion d’extrémale régulière sur laquelle le contrôle $u(t)$ est de norme constante.

Remarque A.5. Le problème important du *temps minimal* correspond à $f^0 \equiv 1$ et $g \equiv 0$ ou bien à $f^0 \equiv 0$ et $g(t_0, x_0, t_f, x_f) = t_f$. Dans tous les cas, les conditions de transversalité obtenues sont bien les mêmes.

On peut de plus définir un véritable hamiltonien et son système hamiltonien sous certaines conditions, d’après la proposition suivante.

Proposition A.2. Soit $(\bar{x}(\cdot), \bar{p}(\cdot), p^0, \bar{u}(\cdot))$ une extrémale du problème (OCP). On note $z(\cdot) := (x(\cdot), p(\cdot))$. Si pour presque tout $t \in [t_0, t_f]$, le **hamiltonien maximisé**

$$z := (x, p) \mapsto h(t, z) := \max_{u \in U} H(t, x, p, p^0, u)$$

est défini et lisse sur un voisinage de l’extrémale, alors pour presque tout $t \in [t_0, t_f]$

$$\dot{\bar{z}}(t) = \overrightarrow{h}(t, \bar{z}(t)) := \left(\frac{\partial h}{\partial p}(t, \bar{z}(t)), -\frac{\partial h}{\partial x}(t, \bar{z}(t)) \right),$$

et $h(t, z)$ définit un véritable hamiltonien (il ne dépend pas de u).