TP1: Résolution des problèmes de contrôle optimal Méthode de tir simple

Olivier Cots & Joseph Gergaud

1 Introduction

Nous allons dans ce TP voir une méthode de résolution numérique d'un problème simple de contrôle optimal utilisant la condition nécessaire de solution, c'est-à-dire le principe du maximum de Pontriaguine.

Considérons le problème simple de contrôle optimal suivant :

$$(\mathcal{P}_0)\left\{\begin{array}{l} Min \int_0^2 u^2(t)dt \\ \dot{x}(t) = -x(t) + u(t) \\ |u(t)| \leq 1 \\ x(0) = x_0 = 0; \\ x(2) = x_f = 0.5; \end{array}\right.$$

2 Problème aux deux bouts

La condition nécessaire de solution (principe du maximum de Pontriaguine) nous conduit à un système différentiel à deux équations et à une condition initiale et une condition terminale ; en d'autre terme à un problème aux deux bouts (Two Points Boundary Value Problem)

$$(TPBVP_0)\left\{\begin{array}{l} \dot{x}(t) = -x(t) + u(t) \\ \dot{p}(t) = p(t) \\ u(t) = h(p(t)) \\ x(0) = x_0 = 0; \\ x(2) = x_f = 0.5; \end{array}\right.$$

où $u(t) = h(p(t))$ est donné par la maximisation de l'Hamiltonien :

$$\left\{\begin{array}{l} Max \quad H(t, x(t), p(t), w) = -w^2 + (-x(t) + w)p(t) \\ \mid w \mid \leq 1 \end{array}\right.$$

Nous obtenons immédiatement ici

- $u(t) = \dfrac{p(t)}{2}$ si $\mid p(t) \mid \leq 2$
- $u(t) = sign(p(t))$ sinon

3 Méthode de tir

Posons $z(t) = (x(t), p(t))$. Résoudre le problème $(TPBVP_0)$ est alors équivalent à rechercher un zéro de l'équation $S(y) = 0$ où la fonction S, qui sera appelée la fonction de tir associée à notre problème, est définie par

$$\begin{array}{rcl} S : \mathbf{R} & \longrightarrow & \mathbf{R} \\ y & \longmapsto & S(y) = z_1(2, x_0, y) - 0.5, \end{array}$$

avec $z(., x_0, y)$ solution du système à valeur initiale (Initial Value Problem) (cf. la figure 1).

$$(IVP_0) \left\{ \begin{array}{l} \dot{z}_1(t) = -z_1(t) + u(t) \\ \dot{z}_2(t) = z_2(t) \\ u(t) = h(z_2(t)) \\ z_1(0) = x_0 = 0; \\ z_2(0) = y. \end{array} \right.$$

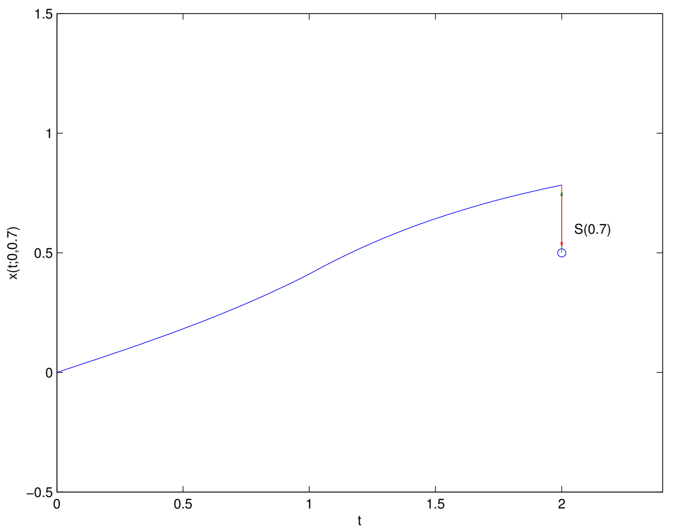

FIGURE 1 – *Valeur de la fonction de tir pour $y = 0.7$.*

L'algorithme de résolution numérique de ce problème sera alors complètement défini si on se donne :

1. l'algorithme de résolution de $S(y) = 0$;
2. l'algorithme d'intégration d'un système différentiel à valeur initial pour calculer la fonction de tir S.

4 Travail demandé

4.1 Interfaces

On respectera les interfaces suivantes :

— Deuxième membre du problème aux deux bouts

$$\dot{z}(t) = \vec{H}(z(t))$$

```
%    Description
%
%        Computes the Hamiltonian vector field associated to H.
%
%-------------------------------------------------------------------------------------------
%
%    Matlab Usage
%        hv = hvfun(t, z, par)
%
%    Inputs
%
%        t    -  real = time
%        z    -  real vector = state and costate
%        par  -  real vector, parameters, par=[] if no parameters
%
%    Outputs
%
%        hv   -  real vector, hamiltonian vector field at time t
%
```

— Intégration numérique du système hamiltonien

$$\dot{z}(t) = \vec{H}(z(t))$$

```
%    Description
%
%        Computes the chronological exponential of the Hamiltonian vector field hv
%        defined by h.
%-------------------------------------------------------------------------------------------
%
%    Matlab Usage
%        [tout,exphv,flag ] = exphvfun(tspan, z0, options, par)
%
%    Inputs
%
%        tspan   - real row vector of dimension 2, tspan = [t0 tf]
%        z0      - real vector, initial flow
%        options - options for the numerical integration
%        par     - real vector, parameters, par=[] if no parameters
%
%    Outputs
%
%        tout    - real row vector, time at each integration step
%        exphv   - real matrix, exphv(:,i) : flow at tout(i)
%        flag    - integer, flag should be 1 (ODE integrator output)
```

— Fonction de tir :

```
%------------------------------------------------------------------------------------------
%
%    Description
%
%        Computes the shooting function
%
%
%------------------------------------------------------------------------------------------
%
%    Matlab Usage
%
%        s = sfun(y,options,par)
%
%    Inputs
%
%        y       - real vector, shooting variable
%        options - options for numerical integration
%        par     - real vector, par=[] if no parameters
%
%    Outputs
%
%        s       - real vector, shooting value
%
%------------------------------------------------------------------------------------------
```

4.2 Travail demandé

En utilisant MATLAB et les fonctions `ode45` et `fsolve`

1. (a) Visualiser $z(.,0,y)$ pour $y = 0$ et 0.7.
   (b) Quelle est la valeur de $S(0)$?
   (c) Vérifier que $S(0.7) = 0.2833$.
2. Visualiser la fonction de tir S pour $y \in [-3; 3]$ (*cf.* la figure 2).
3. Résoudre l'équation $S(y) = 0$. On prendra comme point de départ $y^{(0)} = -0.5$ (la solution est $y^* = 0.2759$) et $y^{(0)} = 2$ pour `fsolve`.
4. Résoudre le problème suivant :

$$(P_1)\begin{cases} Min \int_0^2 u^2(t)dt \\ \dot{x}_1(t) = x_2(t) \\ \dot{x_2}(t) = u(t) \\ |u(t)| \leq 1 \\ x_1(0) = 0; x_2(0) = 0 \\ x_1(2) = 0.5; x_2(2) = 0. \end{cases}$$

   La solution est $y^* = \begin{pmatrix} 3/2 & 3/2 \end{pmatrix}$ et les états, états adjoints et contrôle sont donnés à la figure 3.

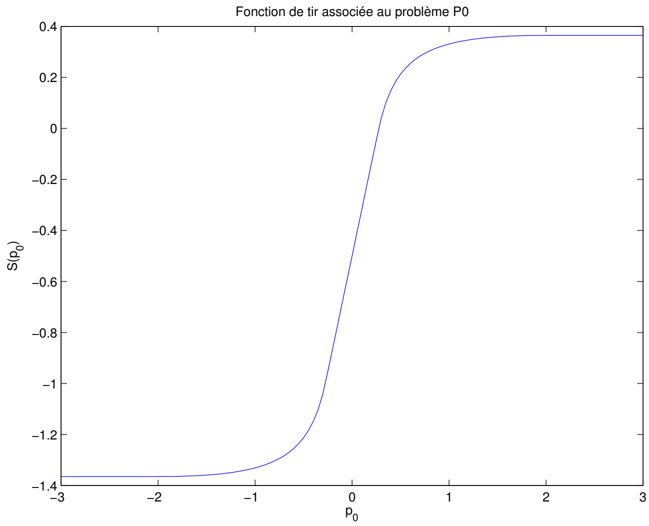

FIGURE 2 – *Fonction de tir $S(y)$.*

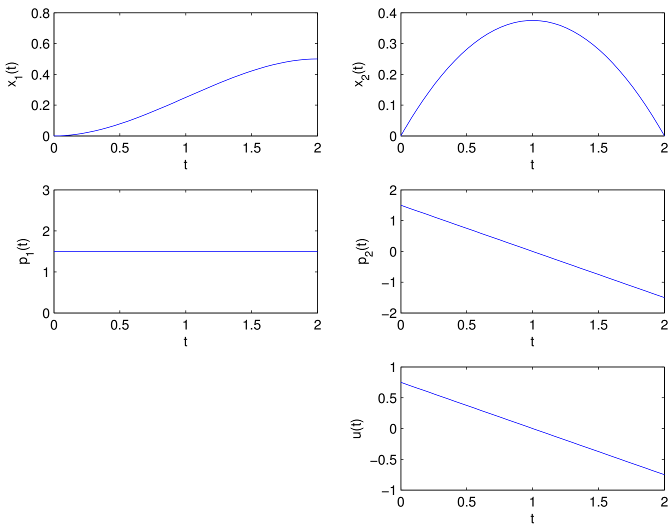

FIGURE 3 – *Solution du problème (P_1).*