

République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Mouloud Mammeri de Tizi-Ouzou

Faculté des Sciences
Département de Mathématiques

MEMOIRE DE Master II
En Mathématiques Appliquées
Option : Recherche Opérationnelle

Thème :

Résolution d'un problème de contrôle optimal avec
une entrée libre par la méthode de relaxation

Présenté par : M^{elle} HAKIMI Sarah

Dirigé par :
M^{me} LOUADJ Kahina

Soutenu le : 18/09/2011 devant le jury d'examen composé de :

Président : Mohamed AIDENE	Professeur	UMMTO
Examineur : A.K.MERAKEB	Maître de conférence classe B	UMMTO
Rapporteur : Kahina LOUADJ	Maître assistant classe A	UMMTO
Examinatrice : Nassima MOUSSOUNI	Maître assistant classe A	UMMTO
Examinatrice : Faiza HADOUCHE	Maître assistant classe A	UMMTO

Promotion 2011

Remerciements

Je tiens à exprimer toute ma reconnaissance à *M^{me}* LOUADJ KAHINA doctorante et enseignante à l'université de Mouloud Mammeri de T.O, qui a proposé et accepté de diriger ce travail. Elle a apporté une grande contribution à son aboutissement. Je tiens particulièrement à la remercier pour sa grande disponibilité et son sens de la responsabilité au-delà de son rôle d'encadreur.

Je remercie vivement l'ensemble des membres du jury pour l'honneur qu'ils me font en acceptant de juger ce travail.

Je suis également reconnaissante à tous les enseignants qui nous ont formé et particulièrement pour la chance qu'ils nous ont offert en nous ouvrant l'inscription au Master 2.

Je dédie ce modeste travail à :

- Mes chers parents, votre affection, vos encouragements et vos conseils n'ont pas d'égal. Puisse Dieu vous accorder longue vie ;
- Ma soeur : Nina ;
- Mes cousines Sihem, Samia, Hakima et Célia ;
- Mes très chers amis : Lydia, Maya, Ghania, Nadia, Ouahab et Lyes ;

Table des matières

Introduction générale	3
1 Méthodes numériques d'intégration des équations différentielles	5
1.1 Introduction	5
1.2 Théorèmes d'existence et d'unicité	5
1.3 La méthode d'EULER	6
1.3.1 Majoration de l'erreur dans la méthode d'Euler	7
1.4 Méthodes numériques à un pas	8
1.4.1 Définition	8
1.4.2 Méthode de Taylor d'ordre p	9
1.4.3 Méthode du point milieu	10
1.4.4 Méthodes consistantes, stables et convergentes	11
1.4.5 Méthode de Runge-Kutta	11
1.5 Méthodes à pas multiples	17
2 Contrôle Optimale	18
Introduction	18
2.1 Formulation du problème	19
2.1.1 Formulation générale (Problème de Bolza)	19
2.2 Contrôlabilité	19
2.2.1 Cas avec contraintes sur le contrôle	20
2.2.2 Contrôlabilité dans le cas non linéaire	21
2.3 Définitions	22
2.4 Temps-Optimalité	22
2.4.1 Existence de trajectoires temps-optimales	22
2.5 Condition nécessaire d'optimalité : principe du maximum dans le cas linéaire	22
2.6 Théorie linéaire-quadratique	23
2.6.1 Problème LQ	24
2.6.2 Existence de trajectoires optimales	24
2.6.3 Condition nécessaire et suffisante d'optimalité : principe du maximum dans le cas LQ	25

2.7	Théorie du contrôle optimal non linéaire	26
2.7.1	Problème de Lagrange	26
2.8	Méthodes numériques en contrôle optimale	29
2.8.1	Méthodes Indirectes	29
2.8.2	Méthodes Directes	32
3	Résolution d'un problème de contrôle optimale par la méthode de relaxation	34
3.1	Optimisation dynamique : problème de commande optimale associé à des EDO	34
3.1.1	Cas des systèmes linéaires avec coût quadratique : régulation d'un processus thermique	34
3.1.2	Algorithme de relaxation associé	35
3.2	Etude d'un système en anneau	36
3.2.1	Algorithmes de résolution	37
3.2.2	Analyse de la convergence	39
3.3	Application numérique : Résolution d'un problème de contrôle optimale à entrée libre	40
3.3.1	Position du problème	40
3.3.2	La solution exacte	43
3.3.3	Résultats numériques	50
3.4	Conclusion	52
	Conclusion générale	53
	Bibliographie	55

Introduction générale

La théorie du contrôle optimal est une extension du calcul des variations, c'est un ensemble de méthodes utilisé pour calculer les politiques de contrôle d'un modèle mathématique d'optimisation. La théorie du contrôle optimal est largement dû au travail de *Lev Pontryagin* et ses collaborateurs de l'Union soviétique ainsi qu'à celui de *Richard Bellman*, Etats-Unis.

Le contrôle optimal sert à trouver une loi de commande pour un système donné de telle sorte qu'un certain critère d'optimalité soit atteint. Un problème de commande comprend donc un coût à optimiser, une fonction d'état et une variable de contrôle. Résoudre un problème de contrôle optimal revient à trouver un ensemble d'équations différentielles décrivant les trajectoires des variables d'état ainsi que le contrôle qui minimisent le coût fonctionnel. Ce problème peut se résoudre en utilisant le principe du maximum de Pontryagin, ou encore en résolvant l'équation d'hamilton-Jacobi-Bellman.

Dans ce mémoire nous avons étudié la **méthode de décomposition et coordination**, qui est basée sur un principe de décomposition des équations différentielles. Depuis les années 70, les automaticiens se sont intéressés à la thématique de la commande et de l'analyse des "grands systèmes", beaucoup d'études ont porté sur les techniques de décomposition et de résolution de problèmes numériques complexes développées en mathématiques appliquées. Ce principe est à l'origine utilisé pour traiter des problèmes d'optimisation de grande taille, en effet, leur résolution conduit à un programme mathématique complexe, une alternative consiste à décomposer le problème d'origine en un nombre de sous-systèmes plus petits.

Plusieurs études ont montré l'efficacité des méthodes de décomposition et coordination par relaxation, leur principe est de diviser un programme en multiples niveaux et en multiples systèmes à chaque niveau, on obtient ce qu'on appelle un programme mathématique hiérarchique ou multi-niveaux.

Dans le but d'obtenir une solution optimale du problème d'origine, il est nécessaire de coordonner ses sous-problèmes. Différentes méthodologies de décomposition et coordination ont été développées et étudiées.

Nous nous sommes intéressé à la **coordination par relaxation**, il s'agit d'algorithmes itératifs synchrones et asynchrones qui ont été développés pour résoudre de grands systèmes d'équations linéaires ou non linéaires issus de discrétisations d'équations ou d'inéquations aux dérivées partielles. La modélisation des algorithmes asynchrones a été développée successivement par D.Chazan et Miranker [5] dans le cadre linéaire, J.C Miellou ([6],[12]), G.Baudet [2], D.Bertsekas [3] dans le cadre non linéaire, citons également les travaux de F.Robert [17] dans le cas synchrone. La modélisation de ces algorithmes est effectuée en introduisant une stratégie de choix des composantes pour rendre compte du parallélisme ainsi qu'une notion de retards pour rendre compte de l'asynchronisation entre les processeurs.

Notre objectif est de résoudre **un problème de contrôle optimale à entrée libre** c'est à dire la condition initial n'est pas fixe, et on cherche à ramener un système contrôlé à un état final vérifiant une certaine contrainte à partir d'un état initial x_0 appartenant à un domaine prédéfini X_0 , en minimisant un critère sur un intervalle de temps fixé. Pour ce faire, nous avons utilisé la méthode de relaxation [8, 9, 10] que nous avons simulé sur Matlab.

Ce mémoire est organisé en trois chapitres :

Chapitre 1 : est une introduction aux équations différentielles, on y présente des notions essentielles, suivies des méthodes numériques de résolution.

Dans le second chapitre, on s'intéresse à la théorie du contrôle optimal et aux méthodes numériques de résolution.

Le chapitre trois, expose la méthode de relaxation, avec trois exemples de son application : la régulation d'un processus thermique, l'étude d'un système en anneau, et enfin notre exemple d'application numérique : un problème de contrôle à entrée libre. Pour ce dernier cas, nous présenterons les résultats numériques obtenus sur Matlab.

Chapitre 1

Méthodes numériques d'intégration des équations différentielles

1.1 Introduction

Dans tout ce chapitre I_0 désigne un intervalle de \mathbb{R} non réduit à un point et t_0 un instant fixé dans I_0 ; on se donne une fonction f définie et continue sur $I_0 \times \mathbb{R}^m$ à valeurs dans \mathbb{R}^m , un élément y_0 de \mathbb{R}^m , et on cherche à trouver une fonction y continue et dérivable sur l'intervalle I_0 , à valeurs dans \mathbb{R}^m , telle que

$$\forall t \in I_0, y'(t) = f(t, y(t)) \quad (1.1)$$

et

$$y(t_0) = y_0 \quad (1.2)$$

Ce problème s'appelle un *problème de Cauchy* pour le système différentiel (1.1); la condition (1.2) s'appelle une *condition de Cauchy*. Une fonction y qui vérifie les équations (1.1) est appelée une intégrale du système différentiel (1.1). Dans de nombreux exemples physiques, la variable t représente le temps; l'instant t_0 est alors appelé *instant initial* et la condition (1.2) est appelée *condition initiale*.

1.2 Théorèmes d'existence et d'unicité

Théorème 1 (Cauchy-Péano) [6] *On suppose que la fonction f est continue dans un voisinage du point (t_0, y_0) dans I_0 de \mathbb{R}^m , alors il existe un intervalle J_0 , voisinage de t_0 dans I_0 , et une fonction $y \in C^1(J_0)$ tels que*

$$\forall t \in J_0, y'(t) = f(t, y(t)), y(t_0) = y_0$$

Définition 1.1 On appelle solution locale du problème (1.1),(1.2) la donnée d'un couple (I, y) où I est un intervalle de \mathbb{R} qui est un voisinage de t_0 dans I_0 et où y est une fonction appartenant à $C^1(I)$ telle que

$$y(t_0) = y_0 \quad \text{et} \quad \forall t \in I, y'(t) = f(t, y(t))$$

Définition 1.2 On dit que la solution locale (J, l) prolonge la solution locale (I, y) si on a $I \subset J$, et $\forall t \in I, y(t) = l(t)$; si de plus $I \neq J$, on dit que (J, l) prolonge strictement (I, y) .

Définition 1.3 On dit que la solution locale (I, y) est une solution maximale du problème (1.1),(1.2) s'il n'existe pas de solution locale de ce problème qui la prolonge strictement.

Définition 1.4 On dit que (I, y) est une solution globale du problème (1.1),(1.2) dans I_0 , (ou encore que y est solution du problème (1.1),(1.2)), si (I, y) est une solution locale de ce problème et si $I = I_0$.

1.3 La méthode d'EULER

Nous allons maintenant nous consacrer à la résolution du problème de Cauchy décrit précédemment en nous restreignant, ce qui n'est pas une perte de généralité, au cas où $I_0 = [t_0, t_f]$. Tout d'abord nous allons étudier la plus simple des méthodes, la méthode d'Euler connue aussi sous les noms de méthode d'Euler progressive et de méthode d'Euler explicite. Considérons donc le problème différentiel

$$\forall t \in [t_0, t_f], y'(t) = f(t, y(t)), \quad (1.3)$$

$$y(t_0) = \eta \quad \text{donné dans} \quad \mathbb{R}^m. \quad (1.4)$$

Nous nous donnons une subdivision $t_0 < t_1 < \dots < t_n < t_{n+1} < \dots < t_N = t_f$ de l'intervalle $[t_0, t_f]$, et nous posons

$$h_n = t_{n+1} - t_n \quad \text{et} \quad h = \max_{0 \leq n \leq N} h_n.$$

Les solutions de (1.3) vérifient, pour $0 \leq n < N$,

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

soit $y_0 = \eta_h$ une approximation de $y(t_0) = \eta$; nous construisons par récurrence une approximation y_n de $y(t_n)$ en remplaçant la relation précédente par

$$y_{n+1} = y_n + h_n f(t_n, y_n), \quad n = 0, 1, \dots, N-1, \quad (1.5)$$

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 (1.5) s'appelle le schéma d'Euler.

1.3.1 Majoration de l'erreur dans la méthode d'Euler

Nous cherchons à obtenir une estimation de l'erreur

$$e_n = y(t_n) - y_n$$

Posons

$$\varepsilon_n = y(t_{n+1}) - y(t_n) - h_n f(t_n, y(t_n)), \quad (1.6)$$

où $y(\cdot)$ désigne la solution de (1.3),(1.4) ; cette quantité mesure avec quelle précision la solution exacte vérifie le schéma (1.5) ; elle s'appelle l'*erreur de consistance* à l'instant t_n de la méthode d'Euler. Nous avons

$$\varepsilon_n = \int_{t_n}^{t_{n+1}} (y'(t) - y'(t_n)) dt,$$

d'où

$$|\varepsilon_n| \leq \omega(y'; h) h_n \quad (1.7)$$

où $\omega(y'; h) h_n$ désigne le module de continuité de la fonction y' ; on a, puisque $y' \in C^0[t_0, t_f]$, $\lim_{h \rightarrow 0} \omega(y'; h) = 0$. Si on suppose de plus que $y \in C^2[t_0, t_f]$, on obtient par utilisation de la formule de Taylor dans (1.6)

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

d'où

$$|\varepsilon_n| \leq h \int_{t_n}^{t_{n+1}} |y''(s)| ds. \quad (1.8)$$

On déduit de (1.5) et (1.6) la relation

$$e_{n+1} = e_n + h_n [f(t_n, y(t_n)) - f(t_n, y_n)] + \varepsilon_n$$

d'où en utilisant la condition de Lipchitz

$$|e_{n+1}| \leq (1 + h_n L) |e_n| + |\varepsilon_n|. \quad (1.9)$$

Pour en déduire une majoration de $|e_n|$, nous allons utiliser le lemme technique fondamentale suivant

Lemme 1.1 [3] Soient θ_n et α_n deux suites de réels positifs ou nuls vérifiant

$$\forall n \geq 0, \quad \theta_{n+1} \leq (1 + h_n L) \theta_n + \alpha_n,$$

alors on a

$$\forall n \geq 0, \quad \theta_n \leq e^{L(t_n - t_0) \theta_0} + \sum_{i=0}^{n-1} e^{L(t_n - t_{i+1})} \alpha_i.$$

En appliquant ce lemme à (2.10), et en utilisant (1.7), on obtient

$$\begin{aligned}
|e_n| &\leq e^{L(t_n-t_0)}|e_0| + \omega(y'; h) \sum_{i=0}^{n-1} e^{L(t_n-t_{i+1})} \int_{t_i}^{t_{i+1}} ds \\
&\leq e^{L(t_n-t_0)}|e_0| + \omega(y'; h) \sum_{i=0}^{n-1} \int_{t_i}^{t_{i+1}} e^{L(t_n-s)} ds \\
&\leq e^{L(t_n-t_0)}|e_0| + \omega(y'; h) \frac{e^{L(t_n-t_0)} - 1}{L}, \text{ (si } L \neq 0)
\end{aligned}$$

donc

$$|y(t_n) - y_n| \leq e^{L(t_n-t_0)}|\eta - \eta_h| + \omega(y'; h) \frac{e^{L(t_n-t_0)} - 1}{L}, \quad \text{si } L \neq 0. \quad (1.10)$$

(si $L = 0$, (1.10) reste vrai à condition de remplacer $\frac{e^{L(t_n-t_0)} - 1}{L}$ par $(t_n - t_0)$). Si on suppose de plus que $y \in C^2[t_0, t_f]$, on déduit de même de (2.10) et (2.9)

$$|y(t_n) - y_n| \leq e^{L(t_n-t_0)}|\eta - \eta_h| + h \int_{t_0}^{t_n} e^{L(t_n-s)} |y''(s)| ds. \quad (1.11)$$

1.4 Méthodes numériques à un pas

On appelle *méthode à un pas* une méthode permettant de calculer y_{n+1} à partir de la seule valeur antérieure y_n . Une méthode à r pas est au contraire une méthode qui utilise les r valeurs antérieurs y_n, \dots, y_{n-r+1} (valeurs qui doivent donc être mémorisées) afin de faire le calcul de y_{n+1} .

1.4.1 Définition

Les méthodes à un pas sont les méthodes de résolution numérique qui peuvent s'écrire sous la forme

$$y_{n+1} = y_n + h_n \Phi(t_n, y_n, h_n), \quad 0 \leq n \leq N,$$

où $\Phi : [t_0, t_f] \times \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ est une fonction que l'on supposera continue. Dans la pratique, la fonction $\Phi(t, y, h)$ peut n'être définie que sur une partie de la forme $[t_0, t_f] \times J \times [0, \delta]$ où J est un intervalle de \mathbb{R} (de sorte en particulier que $[t_0, t_f] \times J$ soit contenu dans le domaine de définition de l'équation différentielle).

Exemple 1.1 *La méthode d'Euler est la méthode à un pas associée à la fonction $\Phi(t, y, h) = f(t, y)$, et définie par la formule de récurrence $y_{n+1} = y_n + h_n f(t_n, y_n)$*

1.4.2 Méthode de Taylor d'ordre p

Supposons que f soit de classe C^p , alors toute solution exacte $y(\cdot)$ est de classe C^{p+1} , et sa dérivée k -ième est $y^{(k)}(t) = f^{[k-1]}(t, y(t))$. La formule de Taylor d'ordre p implique

$$y(t_n + h_n) = y(t_n) + \sum_{k=1}^p \frac{1}{k!} h_n^k f^{[k-1]}(t, y(t_n)) + O(h_n^p).$$

Lorsque h_n est assez petit, l'approximation est d'autant meilleure que p est plus grand. On est donc amené à considérer l'algorithme suivant, appelé méthode de Taylor d'ordre p :

$$\begin{cases} y_{n+1} = y_n + \sum_{k=1}^p \frac{1}{k!} h_n^k f^{[k-1]}(t_n, y_n), \\ t_{n+1} = t_n + h_n, \end{cases}$$

D'après la définition générale des méthodes à un pas, cet algorithme correspond au choix $\Phi(t, y, h) = \sum_{k=1}^p \frac{1}{k!} h^{k-1} f^{[k-1]}(t, y)$. Calculons l'erreur de consistance ε_n . En supposant $y_n = y(t_n)$, la formule de Taylor d'ordre $p+1$ donne

$$\begin{aligned} \varepsilon_n &= y(t_{n+1}) - y_{n+1} \\ &= y(t_n + h_n) - \sum_{k=0}^p \frac{1}{k!} h_n^k y^{(k)}(t_n) \\ &= \frac{1}{(p+1)!} h_n^{p+1} f^{[p]}(t_n, y_n) + O(h_n^{p+1}). \end{aligned}$$

L'erreur est donc de l'ordre de h_n^{p+1} . On dira d'une manière générale qu'une méthode est d'ordre p si l'erreur de consistance est en h_n^{p+1} chaque fois que f est de classe C^p au moins. La méthode d'Euler est le cas particulier $p=1$ de la méthode de Taylor.

Remarque 1.1 *Dans la pratique, la méthode de Taylor souffre de deux inconvénients graves qui en font généralement déconseiller l'utilisation pour $p=2$:*

- *Le calcul des quantités $f^{[k]}$ est souvent complexe et coûteux en temps machine. Il faut aussi pouvoir évaluer explicitement $f^{[k]}$, ce qui n'est pas toujours le cas (par exemple, si f est une donnée expérimentale discrétisée).*
- *la méthode suppose a priori que f soit très régulière, les erreurs risquent donc de ne pas pouvoir être contrôlées si certaines dérivées de f présentent des discontinuités ou une mauvaise continuité (pentes élevées).*

1.4.3 Méthode du point milieu

L'idée est que la corde de la fonction y sur $[t, t+h]$ a une pente voisine de $y'(t + \frac{h}{2})$, alors que dans la méthode d'Euler on approxime brutalement cette pente par $y'(t)$. On écrit donc :

$$y(t+h) \simeq y(t) + hy'(t + \frac{h}{2}) \quad (1.12)$$

Si y est de classe C^3 , il vient

$$\begin{aligned} y(t+h) &= y(t) + hy'(t) + \frac{1}{2}h^2y''(t) + \frac{1}{6}h^3y'''(t) + O(h^3); \\ y'(t + \frac{h}{2}) &= y'(t) + \frac{1}{2}hy''(t) + \frac{1}{8}h^2y'''(t) + O(h^2) \end{aligned}$$

L'erreur comise est donc :

$$e = y(t+h) - y(t) - hy'(t + \frac{h}{2}) = \frac{1}{24}h^3y'''(t) + O(h^3);$$

Soit une erreur en h^3 au lieu de h^2 dans la méthode d'Euler. On a par ailleurs

$$y'(t + \frac{h}{2}) = f(t + \frac{h}{2}, y(t + \frac{h}{2}))$$

Comme la valeur de $y(t + \frac{h}{2})$ n'est pas connue, on l'approxime par

$$y(t + \frac{h}{2}) \simeq y(t) + \frac{h}{2}f(t, y(t)), \quad (1.13)$$

d'où en définitive

$$y(t+h) \simeq y(t) + hf\left(t + \frac{h}{2}, y(t) + \frac{h}{2}f(t, y(t))\right).$$

L'algorithme du point milieu est associé au choix

$$\Phi(t, y, h) = f\left(t + \frac{h}{2}, y + \frac{h}{2}f(t, y)\right)$$

et donne lieu au schéma numérique

$$\begin{cases} y_{n+\frac{1}{2}} = y_n + \frac{h_n}{2}f(t_n, y_n), \\ p_n = f\left(t_n + \frac{h_n}{2}, y_{n+\frac{1}{2}}\right), \\ y_{n+1} = y_n + h_n p_n, \\ t_{n+1} = t_n + h_n, \end{cases}$$

En utilisant les approximations faites en (1.12) et (1.13), on calcule l'erreur de consistance, et on en déduit que la méthode du point milieu est d'ordre 2.

1.4.4 Méthodes consistantes, stables et convergentes

Définition 1.5 On dit qu'une méthode est consistante si pour toute solution exacte $y(\cdot)$ la somme des erreurs de consistances relatives à $y(\cdot)$, soit $\sum_{0 \leq n \leq N} |\varepsilon_n|$, tend vers 0 quand h_{max} tend vers 0.

Une autre notion fondamentale est la notion de stabilité. Dans la pratique, le calcul récurrent des points y_n est en effet entâché d'erreurs d'arrondi e_n . Pour que les calculs soient significatifs, il est indispensable que la propagation de ces erreurs reste contrôlable. On est amené à la définition suivante :

Définition 1.6 On dit que la méthode est stable s'il existe une constante $S \geq 0$, appelée constante de stabilité, telle que toutes suites $(y_n), (\tilde{y}_n)$ définies par

$$\begin{aligned} y_{n+1} &= y_n + h_n \Phi(t_n, y_n, h_n), & 0 \leq n \leq N \\ \tilde{y}_{n+1} &= \tilde{y}_n + h_n \Phi(t_n, \tilde{y}_n, h_n) + e_n, & 0 \leq n \leq N \end{aligned}$$

on ait

$$\max_{0 \leq n \leq N} |\tilde{y}_n - y_n| \leq S(|\tilde{y}_0 - y_0| + \sum_{0 \leq n \leq N} |e_n|).$$

Autrement dit, une petite erreur initiale $|\tilde{y}_0 - y_0|$ et de petites erreurs d'arrondis e_n dans le calcul récurrent de \tilde{y}_n provoquent une erreur finale $\max |\tilde{y}_n - y_n|$ contrôlable. Une dernière notion importante est la suivante :

Définition 1.7 On dit que la méthode est convergente si pour toute solution exacte $y(\cdot)$, la suite y_n telle que

$$\begin{aligned} y_{n+1} &= y_n + h_n \Phi(t_n, y_n, h_n) \quad \text{vérifie} \\ &\max_{0 \leq n \leq N} |y_n - y(t_n)| \rightarrow 0 \\ \text{quand } y_0 &\rightarrow y(t_0) \quad \text{et quand } h_{max} \rightarrow 0 \end{aligned}$$

1.4.5 Méthode de Runge-Kutta

Principe Général On considère un problème de Cauchy

$$\begin{cases} y' = f(t, y), & t \in [t_0, t_f], \\ y(t_0) = y_0, \end{cases}$$

et on cherche à discrétiser ce problème par rapport à une subdivision $t_0 < t_1 < \dots < t_N = t_f$. L'idée est de calculer par récurrence les points (t_n, y_n) en utilisant des points intermédiaires $(t_{n,i}, y_{n,i})$ avec

$$t_{n,i} = t_n + c_i h_n, \quad 1 \leq i \leq q, \quad c_i \in [0, 1].$$

A chacun de ces points, on associe la pente correspondante

$$p_{n,i} = f(t_{n,i}, y_{n,i})$$

Soit y une solution exacte de l'équation. On a

$$\begin{aligned} y(t_{n,i}) &= y(t_n) + \int_{t_n}^{t_{n,i}} f(t, y(t)) dt \\ &= y(t_n) + h_n \int_0^{c_i} f(t_n + uh_n, y(t_n + uh_n)) du \end{aligned}$$

grâce au changement de variable $t = t_n + uh_n$. De même

$$y(t_{n+1}) = y(t_n) + h_n \int_0^1 f(t_n + uh_n, y(t_n + uh_n)) du$$

On se donne alors pour chaque $i = 1, 2, \dots, q$ une méthode d'intégration approchée

$$\int_0^{c_i} g(t) dt \simeq \sum_{1 \leq j \leq i} a_{ij} g(c_j), \quad (1.14)$$

ces méthodes pouvant être *a priori* différentes. On se donne également une méthode d'intégration approchée sur $[0, 1]$:

$$\int_0^1 g(t) dt \simeq \sum_{1 \leq j < q} b_j g(c_j). \quad (1.15)$$

En appliquant ces méthodes à $g(u) = f(t_n + uh_n, y(t_n + uh_n))$, il vient

$$\begin{aligned} y(t_{n,i}) &\simeq y(t_n) + h_n \sum_{1 \leq j \leq i} a_{ij} f(t_{n,j}, y(t_{n,j})), \\ y(t_{n+1}) &\simeq y(t_n) + h_n \sum_{1 \leq j < q} b_j f(t_{n,j}, y(t_{n,j})) \end{aligned}$$

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

$$\left\{ \begin{array}{l} \left\{ \begin{array}{l} t_{n,i} = t_n + c_i h_n, \\ y_{n,i} = y_n + h_n \sum_{1 \leq j < i} a_{ij} p_{n,j}, \\ p_{n,i} = f(t_{n,i}, y_{n,i}) \end{array} \right. \quad 1 \leq i \leq q, \\ \\ t_{n+1} = t_n + h_n, \\ y_{n+1} = y_n + h_n \sum_{1 \leq j \leq q} b_j p_{n,j}, \end{array} \right.$$

On la représente traditionnellement par le tableau

(M_1)	c_1	0	0	...	0	0
(M_2)	c_2	a_{21}	0	...	0	0
	\vdots	\vdots	\vdots	\ddots	\vdots	\vdots
	\vdots	\vdots	\vdots		0	0
(M_q)	c_q	a_{q1}	a_{q2}	\cdots	a_{qq-1}	0
(M)		b_1	b_2	\cdots	b_{q-1}	b_q

Où les méthodes d'intégration approchées correspondent aux lignes. On pose par convention $a_{ij} = 0$ pour $j \geq i$.

Exemples

Exemple 1.2 Pour $q = 1$, le seul choix possible est

0	0
	1

On a ici

$c_1 = 0, a_{11} = 0, b_1 = 1$. L'algorithme est donné par

$$\begin{cases} p_{n,1} = f(t_n, y_n), \\ t_{n+1} = t_n + h_n, \\ y_{n+1} = y_n + h_n p_{n,1}, \end{cases}$$

Il s'agit de la méthode d'Euler.

Exemple 1.3 Pour $q = 2$, on considère les tableaux de la forme

0	0	0
a	a	0
	$1 - \frac{1}{2a}$	$\frac{1}{2a}$

L'algorithme s'écrit ici

$$\begin{cases} p_{n,1} = f(t_n, y_n), \\ t_{n,2} = t_n + \alpha h_n, \\ y_{n,2} = y_n + \alpha h_n p_{n,1}, \\ p_{n,2} = f(t_{n,2}, y_{n,2}), \\ t_{n+1} = t_n + h_n, \\ y_{n+1} = y_n + h_n \left(\left(1 - \frac{1}{2\alpha}\right) p_{n,1} + \frac{1}{2\alpha} p_{n,2} \right), \end{cases}$$

ou encore, sous forme condensée :

$$y_{n+1} = y_n + h_n \left(\left(1 - \frac{1}{2\alpha}\right) f(t_n, y_n) + \frac{1}{2\alpha} f(t_n + \alpha h_n, y_n + \alpha h_n f(t, y_n)) \right),$$

- Pour $\alpha = \frac{1}{2}$, on retrouve la méthode du point milieu

$$y_{n+1} = y_n + h_n f\left(t_n + \frac{h_n}{2}, y_n + \frac{h_n}{2} f(t_n, y_n)\right),$$

qui est basée sur la méthode d'intégration du point milieu :

$$(M) \quad \int_0^1 g(t) dt \simeq g\left(\frac{1}{2}\right).$$

- Pour $\alpha = 1$, on obtient la méthode de Heun :

$$y_{n+1} = y_n + h_n \left(\frac{1}{2} f(t_n, y_n) + \frac{1}{2} f(t_{n+1}, y_n + h_n f(t_n, y_n)) \right),$$

qui repose sur la méthode d'intégration des trapèzes :

$$(M) \quad \int_0^1 g(t) dt \simeq \frac{1}{2}(g(0) + g(1)).$$

Exemple 1.4 Méthode de Runge-Kutta "classique" :

Il s'agit de la méthode définie par le tableau

$q = 4,$	0	0	0	0	0
	$\frac{1}{2}$	$\frac{1}{2}$	0	0	0
	$\frac{1}{2}$	0	$\frac{1}{2}$	0	0
	1	0	0	1	0
		$\frac{1}{6}$	$\frac{2}{6}$	$\frac{2}{6}$	$\frac{1}{6}$

L'algorithme correspondant s'écrit

$$\left\{ \begin{array}{l} p_{n,1} = f(t_n, y_n), \\ t_{n,2} = t_n + \frac{1}{2}h_n, \\ y_{n,2} = y_n + \frac{1}{2}h_n p_{n,1}, \\ p_{n,2} = f(t_{n,2}, y_{n,2}), \\ y_{n,3} = y_n + \frac{1}{2}h_n p_{n,2}, \\ p_{n,3} = f(t_{n,2}, y_{n,3}), \quad \text{noter que } t_{n,3} = t_{n,2} \\ t_{n+1} = t_n + h_n, \quad \text{noter que } t_{n,4} = t_{n+1} \\ y_{n,4} = y_n + h_n p_{n,3}, \\ p_{n,4} = f(t_{n+1}, y_{n,4}), \\ y_{n+1} = y_n + h_n \left(\frac{1}{6}p_{n,1} + \frac{2}{6}p_{n,2} + \frac{2}{6}p_{n,3} + \frac{1}{6}p_{n,4} \right), \end{array} \right.$$

Cette méthode est d'ordre 4. Dans ce cas les méthodes d'intégration (1.14) et (1.15) utilisées sont respectivement :

$$(M_2) \quad \int_0^{\frac{1}{2}} g(t) dt \simeq \frac{1}{2}g(0) : \quad \text{rectangles à gauche,}$$

$$(M_3) \quad \int_0^{\frac{1}{2}} g(t) dt \simeq \frac{1}{2}g\left(\frac{1}{2}\right) : \quad \text{rectangles à droite,}$$

$$(M_3) \quad \int_0^1 g(t) dt \simeq g\left(\frac{1}{2}\right) : \quad \text{point milieu,}$$

$$(M) \quad \int_0^1 g(t) dt \simeq \frac{1}{6}g(0) + \frac{2}{6}g\left(\frac{1}{2}\right) + \frac{2}{6}g\left(\frac{1}{2}\right) + \frac{1}{6}g(1) : \quad \text{simpson.}$$

Ordre de la méthode de Runge-Kutta

Théorème 2 [7] La méthode de Runge-Kutta définie par le tableau des coefficients c_i, a_{ij}, b_j est

$$- \text{ d'ordre } \geq 2 \text{ ssi } \sum_j b_j c_j = \frac{1}{2}.$$

$$- \text{ d'ordre } \geq 3 \text{ ssi } \sum_j b_j c_j = \frac{1}{2}; \quad \sum_j b_j c_j^2 = \frac{1}{3}; \quad \sum_{i,j} b_i a_{ij} c_j = \frac{1}{6}.$$

– d'ordre ≥ 4 ssi

$$\begin{aligned} \sum_j b_j c_j &= \frac{1}{2}; & \sum_j b_j c_j^2 &= \frac{1}{3}; & \sum_j b_j c_j^3 &= \frac{1}{4} \\ \sum_{i,j} b_i a_{ij} c_j &= \frac{1}{6}; & \sum_{i,j} b_i a_{ij} c_j^2 &= \frac{1}{12}; & \sum_{i,j} b_i c_i a_{ij} c_j &= \frac{1}{8}; \\ & & & & \sum_{i,j,k} b_i a_{ij} a_{jk} c_k &= \frac{1}{12}. \end{aligned}$$

On voit ainsi que la méthode d'Euler est d'ordre 1, et que les méthodes de l'exemple (1.3) sont d'ordre 2. De plus, dans une méthode avec $q = 2$, il y a *a priori* un seul coefficient a_{ij} non nul, à savoir $\alpha = a_{21}$. On a alors $c_2 = \sum_{j < 2} a_{2j} = \alpha$ et la méthode est d'ordre 2 au moins ssi

$\sum b_j c_j = b_2 \alpha = \frac{1}{2}$, soit $b_2 = \frac{1}{2\alpha}$ et $b_1 = 1 - b_2 = 1 - \frac{1}{2\alpha}$. On voit donc qu'il n'y avait pas d'autres choix possibles pour une méthode d'ordre 2 avec $q = 2$. Enfin, la méthode Runge-Kutta "classique" présentée dans l'exemple (1.4) est d'ordre 4. C'est si l'on peut dire la "méthode reine" des méthodes à un pas : ordre élevé, grande stabilité. Il existe des méthodes d'ordre encore plus élevé, mais leur plus grande complexité les rend peut-être un peu moins praticables.

Contrôle du pas

La manière la plus simple d'utiliser une méthode de résolution numérique consiste à utiliser un pas constant $h_n = h$. La principale difficulté est alors de déterminer h_{max} de façon que l'erreur globale ne dépasse pas une certaine tolérance e fixée à l'avance ; on ne sait pas en effet quelle sera l'évolution de la solution étudiée, de sorte qu'il est difficile de prévoir *a priori* les erreurs de consistance. L'utilisation d'algorithmes à pas variables présente de ce point de vue deux avantages majeurs :

- l'adaptation du pas à chaque étape permet d'optimiser l'erreur commise en fonction de la tolérance prescrite e , sous réserve qu'on dispose d'une estimation "instantanée" de l'erreur de consistance ε_n .
- l'approche d'une discontinuité ou d'une singularité de l'équation différentielle ne peut se faire généralement qu'avec une réduction importante du pas. Dans cette circonstance, il convient d'arrêter l'algorithme avant de traverser cette discontinuité, faute de quoi les erreurs deviennent imprévisibles. Le calcul du pas sert alors de test d'arrêt.

1.5 Méthodes à pas multiples

On appelle *méthode numérique à $r + 1$ pas* toute méthode numérique de la forme

$$y_{n+1} = \Psi(t_n, y_n, h_n; \dots; t_{n-r}, y_{n-r}, h_{n-r})$$

L'intérêt de ces méthodes vient du fait qu'on peut obtenir un ordre élevé pour une complexité de calcul nettement inférieure à celle des méthodes de Runge-Kutta. L'un des problèmes essentiels, néanmoins, est de s'assurer que la stabilité numérique reste suffisamment bonne.

Chapitre 2

Contrôle Optimale

Introduction

La théorie du contrôle (ou commande) analyse les propriétés des systèmes dynamiques sur lesquels on peut agir au moyen d'une commande (ou contrôle). Le but est alors d'amener le système d'un état initial donné à un certain état final, en respectant éventuellement certains critères. L'objectif peut être de déterminer des solutions optimales pour un certain critère d'optimalité (contrôle optimale, ou commande optimale). Les domaines d'applications sont multiples : aérospatiale, automobile, robotique, aéronotique, internet et les communications en général, mais aussi le secteur médical, chimique, génie des procédés,...etc. Du point de vue mathématique, un système contrôle est un système dynamique dépendant d'un paramètre dynamique appelé le contrôle. Pour le modéliser, on peut avoir recours à des équations différentielles, intégrales, fonctionnelles aux différences finies, aux dérivées partielles, stochastiques,...etc.

Pour cette raison, la théorie du contrôle est à l'interconnexion de nombreux domaines mathématiques. Les contrôles sont des fonctions ou des paramètres, habituellement soumis à des contraintes. Une fois le problème de contrôlabilité résolu, on peut de plus vouloir passer de l'état initial à l'état final en minimisant un certain critère ; on parle alors d'un problème de contrôle optimale.

La théorie moderne du contrôle optimale a commencé dans les années 50, avec la formulation du principe du maximum de Pontryagin, qui généralise les équations d'Euler-Lagrange du calcul des variations. De nos jours, les systèmes automatisés font complètement partie de notre quotidien, ayant pour but d'améliorer notre qualité de vie et de faciliter certaines tâches : système de freinage ABS, assistance à la conduite, servomoteurs, thermostats, circuits frigorifiques, contrôle des flux routiers, ferroviaires, aériens, bour-

siers, photographie numérique, lecteurs CD et DVD, réseaux informatiques, moteurs de recherche internet, circuits électriques, électroniques, télécommunications en général, raffinage pétrolier, chaînes industrielles de montage, peacemakers et autres systèmes médicaux automatisés, opérations au laser, robotique, satellites, guides aérospatiaux,... la liste est infinie, les applications concernent tout système sur lequel on peut avoir une action, avec une notion de rendement optimal.

2.1 Formulation du problème

Soit un système (S) (physique, biologique, économique,...) caractérisé à chaque instant t par un état $x(t) \in \mathbb{R}^n$ et que l'on peut contrôler à l'aide d'une commande $u(t) \in \mathbb{R}^m$. Evolution du système : $\dot{x}(t) = f(t, x(t), u(t))$. On cherche la commande u qui fait passer (S) d'un état initial à un état final donné en minimisant un certain critère (objectif) (consommation, énergie,...).

2.1.1 Formulation générale (Problème de Bolza)

On appelle problème de contrôle optimal sous forme de Bolza, tout problème noté (P) du type :

$$\left\{ \begin{array}{l} \text{Min } J(x, u) = g(t_0, x(t_0), t_f, x(t_f)) + \int_{t_0}^{t_f} F(t, x(t), u(t)) dt, \\ \dot{x}(t) = f(t, x(t), u(t)), \quad t \in [t_0, t_f] \\ u(t) \in U \subset \mathbb{R}^m, \\ \Psi_0(t_0, x(t_0)) = 0, \\ \Psi_1(t_f, x(t_f)) = 0, \end{array} \right.$$

$$\Psi_0 : \mathbb{R} \times \mathbb{R}^n \rightarrow \mathbb{R}^p$$

$$\Psi_1 : \mathbb{R} \times \mathbb{R}^n \rightarrow \mathbb{R}^q$$

Le cas linéaire :

$$\left\{ \begin{array}{l} \dot{x}(t) = Ax(t) + Bu(t), \quad t \in [t_0, t_f] \\ x(t_0) = x_0 \end{array} \right.$$

avec A dans $\mathbf{M}_n(\mathbb{R})$ et B dans $\mathbf{M}_{n,m}(\mathbb{R})$

2.2 Contrôlabilité

Définition 2.1 (Ensemble Accessible) *L'ensemble des points accessibles à partir de x_0 en un temps $T > 0$ est définie par :*

$$\text{Acc}(x_0, T) = \{x_u(T) \mid u \in L^\infty([0, T], U)\}$$

où $x_u(\cdot)$ est la solution du système contrôlé associée à u . Autrement dit $Acc(x_0, T)$ est l'ensemble des extrémités des solutions du système contrôlé au temps T , lorsqu'on fait varier le contrôle u . Pour la cohérence, on pose $Acc(x_0, 0) = x_0$.

Théorème 3 [20] Soient $T > 0$ et $x_0 \in \mathbb{R}^n$. Alors pour tout $t \in [0, T]$, $Acc(x_0, t)$ est compact, convexe et varie continûment avec $t \in [t_0, t_f]$.

Proposition 2.1 [20] On suppose que $x_0 = 0$ et $U = \mathbb{R}^m$. Alors, pour tout $t > 0$, l'ensemble $Acc(0, T)$ est un sous-espace vectoriel de \mathbb{R}^n . De plus, pour tout $0 < t_1 < t_2$, on a $Acc(0, t_1) \subset Acc(0, t_2)$.

Définition 2.2 (Contrôlabilité) Le système contrôlé est dit contrôlable en temps T si $Acc(x_0, T) = \mathbb{R}^n$, i.e, pour tous $x_0, x_1 \in \mathbb{R}^n$, il existe un contrôle u tel que la trajectoire associée relie x_0 à x_1 en temps T .

Théorème 4 [20] On suppose $U = \mathbb{R}^m$ (pas de contrainte sur le contrôle). Le système $\dot{x}(t) = Ax(t) + Bu(t)$ est contrôlable en temps T quelconque si et seulement si la matrice $C = (B, AB, \dots, A^{n-1}B)$ est de rang n . La matrice C est appelée matrice de Kalman, et la condition $\text{rg } C = n$ est appelée condition de Kalman.

2.2.1 Cas avec contraintes sur le contrôle

Corollaire 2.1 [20] Sous la condition de Kalman précédente, si $0 \in U^0$ alors l'ensemble accessible $Acc(x_0, t)$ en temps t contenant un voisinage du point $\exp(tA)x_0$.

Remarque 2.1 Les propriétés de contrôlabilité globale sont reliées aux propriétés de la stabilité de la matrice A . Par exemple, il est clair que si :

1. La condition de Kalman est remplie,
2. $0 \in U^0$,
3. Toutes les valeurs propres de la matrice A sont de partie réelle strictement négative (i.e la matrice A est stable),

alors tout point de \mathbb{R}^n peut être conduit à l'origine en temps fini.

Théorème 5 [20] Soit $b \in \mathbb{R}^n$ et $U \subset \mathbb{R}$ un intervalle contenant 0 dans son intérieur. Considérons le système $\dot{x}(t) = Ax(t) + bu(t)$, avec $u(t) \in U$. Alors tout point de \mathbb{R}^n peut être conduit à l'origine en temps fini si et seulement si la paire (A, b) vérifie la condition de Kalman et la partie réelle de chaque valeur propre de A est inférieure ou égale à 0.

2.2.2 Contrôlabilité dans le cas non linéaire

Définition 2.3 *Considérons pour le système :*

$$\dot{x}(t) = f(t, x(t), u(t)); \quad x(0) = x_0 \quad (2.1)$$

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 (2.1) vérifie : $x_u(0) = x_0$, $x_u(T) = x_1$. Ceci conduit à la définition suivante : Soit $T > 0$. L'application entrée-sortie en temps T du système contrôlé (2.1) initialisé à x_0 est l'application :

$$E_T : U \rightarrow \mathbb{R}^n$$

$$u \mapsto x_u(T)$$

où U est l'ensemble des contrôles admissibles, i.e l'ensemble de contrôles u tels 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 .

Définition 2.4 *L'ensemble accessible en temps T pour le système (2.1), noté $Acc(x_0, T)$, est l'ensemble des extrémités au temps T des solutions du système partant de x_0 au temps $t = 0$. Autrement dit c'est l'image de l'application entrée-sortie en temps T .*

Définition 2.5 *Le système (2.1) est dit contrôlable (en temps quelconque) depuis x_0 si :*

$$\mathbb{R}^n = \bigcup_{T \geq 0} Acc(x_0, T)$$

Il est dit contrôlable en temps T si $\mathbb{R}^n = Acc(x_0, T)$

Proposition 2.2 [20] *Considérons le système (2.1) où $f(x_0, u_0) = 0$. Notons $A = \frac{\partial f}{\partial x}(x_0, u_0)$ et $B = \frac{\partial f}{\partial u}(x_0, u_0)$. On suppose que :*

$$rg(B, AB, \dots, A^{n-1}B) = n$$

Alors le système est localement contrôlable en x_0 .

Définition 2.6 *Soit u un contrôle défini sur $[0, T]$ tel que sa trajectoire associée x_u issue de $x(0) = x_0$ est définie sur $[0, T]$. On dit que le contrôle u (ou la trajectoire x_u) est singulier sur $[0, T]$ si la différentielle de Fréchet $dE_T(u)$ de l'application entrée-sortie au point u n'est pas surjective. Sinon on dit qu'il est régulier.*

2.3 Définitions

Définition 2.7 1. On appelle commande admissible toute fonction $u \in \text{Mes}([0, T], \mathbb{R}^m)$, $\text{Mes}([0, T], \mathbb{R}^m)$ étant l'espace des fonctions mesurables $[0, T] \rightarrow \mathbb{R}^m$, telle qu'il existe $x \in AC([0, T], \mathbb{R}^n)$, (AC étant l'espace des fonctions dites "absolument continues"), vérifiant :

(a) l'application $t \rightarrow f(t, x(t), u(t))$ appartient à $L^1([0, T], \mathbb{R}^n)$.

(b)

$$\begin{cases} \dot{x}(t) = f(t, x(t), u(t)), \\ \Psi_0(0, x(0)) = 0, \\ \Psi_1(T, x(T)) = 0, \end{cases}$$

(c) $u(t) \in U$ presque partout sur $[0, T]$.

2. la fonction x associée est appelée trajectoire admissible.

3. On appelle commande optimale, toute commande admissible solution de (P) et état optimal l'état associé.

2.4 Temps-Optimalité

2.4.1 Existence de trajectoires temps-optimales

Il faut tout d'abord formaliser, à l'aide de $Acc(x_0, t)$, la notion de temps minimal. Considérons le système contrôlé dans \mathbb{R}^n où les contrôles u sont à valeurs dans un compact d'intérieur non vide $U \subset \mathbb{R}^m$. Soient x_0 et x_1 deux points de \mathbb{R}^n . Supposons que x_1 soit accessible depuis x_0 , c'est à dire qu'il existe au moins une trajectoire reliant x_0 à x_1 , on aimerait caractériser celles qui le font en temps minimal t^* .

Si t^* est le temps minimal, alors pour tout $t < t^*$, $x_1 \notin Acc(x_0, t)$ (en effet sinon x_1 serait accessible à partir de x_0 en un temps inférieur à t^*). Par conséquent, $t^* = \inf\{t > 0 \mid x_1 \in Acc(x_0, t)\}$.

2.5 Condition nécessaire d'optimalité : principe du maximum dans le cas linéaire

Le théorème suivant donne une condition nécessaire et suffisante pour qu'un contrôle soit optimal.

Théorème 6 [20] Considérons le système de contrôle linéaire :

$$\begin{aligned} \dot{x}(t) &= A(t)x(t) + B(t)u(t) \\ x(t_0) &= x_0. \end{aligned}$$

où le domaine de contraintes $U \subset \mathcal{R}^m$ sur le contrôle est compact. Soit $t_f > 0$. Le contrôle u est optimale sur $[t_0, t_f]$ si et seulement s'il existe une solution non triviale $p(t)$ de l'équation :

$$\begin{aligned} \dot{p}(t) &= -p(t)A(t) \quad \text{telle que} \\ p(t)B(t)u(t) &= \max_{v \in U} p(t)B(t)v. \end{aligned} \quad (2.2)$$

pour presque tout $t \in [t_0, t_f]$. Le vecteur ligne $p(t) \in \mathcal{R}^m$ est appelé vecteur adjoint.

Remarque 2.2 Dans le cas mono-entrée (contrôle scalaire), et si de plus $U = [-a, a]$ où $a > 0$, la condition de maximisation implique immédiatement que $u(t) = a \times \text{signe}(p(t)B(t))$. La fonction $\varphi(t) = p(t)B(t)$ est appelée fonction de commutation, et un temps t_c auquel le contrôle optimal change de signe est appelé un temps de commutation. C'est en particulier un zéro de la fonction φ .

2.6 Théorie linéaire-quadratique

On s'intéresse aux systèmes de contrôle linéaires avec un coût quadratique. Ces systèmes sont d'une grande importance en pratique. En effet un coût quadratique est souvent très naturel dans un problème, par exemple lorsqu'on veut minimiser l'écart au carré par rapport à une trajectoire nominale (problème de poursuite). Par ailleurs même si les systèmes de contrôle sont en général non linéaires, on est souvent amené à linéariser le système le long d'une trajectoire, par exemple dans des problèmes de stabilisation.

Nous allons donc considérer un système de contrôle linéaire dans \mathbb{R}^n :

$$\dot{x}(t) = A(t)x(t) + B(t)u(t), \quad x(0) = x_0, \quad (2.3)$$

muni d'un coût quadratique du type :

$$\mathcal{C}(u) = x'(T)Qx(T) + \int_0^T (x'(t)W(t)x(t) + u'(t)U(t)u(t))dt, \quad (2.4)$$

Remarque 2.3 Ici le signe $(')$ désigne la transposée.

Où $T > 0$ est fixé, et où, pour tout $t \in [0, T]$, $U(t) \in \mathfrak{M}_m(\mathbb{R})$ est symétrique définie positive, $W(t) \in \mathfrak{M}_n(\mathbb{R})$ est symétrique positive, et $Q \in \mathfrak{M}_n(\mathbb{R})$ est une matrice symétrique positive. On suppose que les dépendances en t de A, B, W et U sont L^∞ sur $[0, T]$. Par ailleurs le coût étant quadratique, l'espace naturel des contrôles est $L^2([0, T], \mathbb{R}^m)$.

Le problème de contrôle optimal est alors le suivant, que nous appellerons problème LQ (Linéaire-Quadratique).

2.6.1 Problème LQ

Un point initial $x_0 \in \mathbb{R}^n$ étant fixé, l'objectif est de déterminer les trajectoires partant de x_0 qui minimisent le coût $\mathcal{C}(u)$. Notons que l'on n'impose aucune contrainte sur le point final $x(T)$. Pour toute la suite, on pose : $\|x(t)\|_W^2 = x'(t)W(t)x(t)$, $\|u(t)\|_U^2 = u'(t)U(t)u(t)$, et $g(x) = x'Qx$; de sorte que :

$$\mathcal{C}(u) = g(x(T)) + \int_0^T (\|x(t)\|_W^2 + \|u(t)\|_U^2) dt$$

2.6.2 Existence de trajectoires optimales

Introduisons l'hypothèse suivante sur U :

$$\exists \alpha > 0 \forall u \in L^2([0, T], \mathbb{R}^m) \int_0^T \|u(t)\|_U^2 dt \geq \alpha \int_0^T u'(t)u(t) dt \quad (2.5)$$

Par exemple cette hypothèse est vérifiée si l'application $t \mapsto U(t)$ est continue sur $[0, T]$ et $T < +\infty$, ou encore qu'il existe une constante $c > 0$ telle que pour tout $t \in [0, T]$ et pour tout vecteur $v \in \mathbb{R}^m$ on ait $v'U(t)v \geq cv'v$. On a le théorème d'existence suivant :

Théorème 7 [20] *Sous l'hypothèse (2.5), il existe une unique trajectoire minimisante pour le problème LQ.*

Remarque 2.4 (Extension du théorème) *Si la fonction g apparaissant dans le coût est une fonction continue quelconque de \mathbb{R}^n dans \mathbb{R} , et/ou si le système de contrôle est perturbé par une fonction $r(t)$, alors le théorème précédent reste vrai.*

Remarque 2.5 (Cas d'un intervalle infini) *Le théorème est encore valable si $T = +\infty$, avec $g = 0$, pourvu que le système (2.3) soit contrôlable (en temps quelconque).*

Proposition 2.3 [20] *Considérons le problème de déterminer une trajectoire solution de :*

$$\dot{x}(t) = A(t)x(t) + B(t)u(t) + r(t)$$

sur $[0, +\infty]$ et minimisant le coût :

$$C(u) = \int_0^{+\infty} (\|x(t)\|_W^2 + \|u(t)\|_U^2) dt$$

Si le système est contrôlable en un temps $T > 0$, et si l'hypothèse (2.5) est vérifiée sur $[0, +\infty]$, alors il existe une unique trajectoire minimisante.

2.6.3 Condition nécessaire et suffisante d'optimalité : principe du maximum dans le cas LQ

Théorème 8 [20] *La trajectoire x , associée au contrôle u , est optimale pour le problème LQ si et seulement s'il existe un vecteur adjoint $p(t)$ vérifiant pour presque tout $t \in [0, T]$:*

$$\dot{p}(t) = -p(t)A(t) + x'(t)W(t) \quad (2.6)$$

et la condition finale :

$$p(T) = -x'(T)Q \quad (2.7)$$

De plus le contrôle optimal u s'écrit, pour presque tout $t \in [0, T]$:

$$u(t) = U(t)^{-1}B'(t)p'(t) \quad (2.8)$$

Remarque 2.6 *Si le système de contrôle est perturbé par une fonction $r(t)$, alors le théorème précédent reste vrai. Il le reste, de même, si la fonction g apparaissant dans le coût est une fonction C^1 quelconque de \mathbb{R}^n dans \mathbb{R} , sauf que la condition finale sur le vecteur adjoint (2.7) devient :*

$$p(T) = -\frac{1}{2}\nabla g(x(T)) \quad (2.9)$$

Remarque 2.7 *Dans le cas d'un intervalle infini ($T = +\infty$), la condition devient :*

$$\lim_{t \rightarrow +\infty} p(t) = 0 \quad (2.10)$$

Remarque 2.8 *Définissons la fonction $H : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$ par :*

$$H(x, p, u) = p(Ax + Bu) - \frac{1}{2}(x'Wx + u'Uu),$$

en utilisant toujours la convention que p est un vecteur ligne de \mathbb{R}^n . Alors les équations données par le principe du maximum LQ s'écrivent :

$$\begin{aligned} \dot{x} &= \frac{\partial H}{\partial p} = Ax + Bu \\ \dot{p} &= -\frac{\partial H}{\partial x} = -pA + x'W \end{aligned}$$

et

$$\frac{\partial H}{\partial u} = 0$$

puisque $pB - u'U = 0$.

2.7 Théorie du contrôle optimal non linéaire

2.7.1 Problème de Lagrange

Ce problème simplifié est le suivant. On cherche des conditions nécessaires d'optimalité pour le système :

$$\dot{x}(t) = f(t, x(t), u(t)), \quad (2.11)$$

où les contrôles $u(\cdot) \in U$ sont définis sur $[0, T]$ et les trajectoires associées doivent vérifier $x(0) = x_0$ et $x(T) = x_1$, le problème est de minimiser un coût de la forme :

$$J(u) = \int_0^T f^0(t, x(t), u(t)) dt \quad (2.12)$$

où T est fixé. Associons au système (2.11), le système augmenté suivant :

$$\begin{aligned} \dot{x}(t) &= f(t, x(t), u(t)), \\ \dot{x}^0(t) &= f^0(t, x(t), u(t)), \end{aligned} \quad (2.13)$$

et notons $\tilde{x} = (x, x^0)$, $\tilde{f} = (f, f^0)$. Le problème revient donc à chercher une trajectoire solution de (1.13) joignant les points $\tilde{x}^0 = (x_0, 0)$ et $\tilde{x}_1 = (x_1, x^0(T))$ et minimisant la dernière coordonnée $x^0(T)$. L'ensemble des états accessibles à partir de \tilde{x}_0 pour le système (1.13) est :

$$\tilde{Acc}(\tilde{x}_0, T) = \cup_{u(\cdot)} \tilde{x}(T, \tilde{x}_0, u)$$

Théorème 9 (Principe du maximum faible) [20] *Si le contrôle u associé au système de contrôle (2.11) est optimal pour le coût (2.12), alors il existe une application $p(\cdot)$ absolument continue sur $[0, T]$, à valeurs dans \mathbb{R}^n , appelée vecteur adjoint, et un réel $p^0 \leq 0$, tels que le couple $(p(\cdot), p^0)$ est non trivial, et les équations suivantes sont vérifiées pour presque tout $t \in [0, T]$.*

$$\dot{x}(t) = \frac{\partial H}{\partial p}(t, x(t), p(t), p^0, u(t)) \quad (2.14)$$

$$\dot{p}(t) = -\frac{\partial H}{\partial x}(t, x(t), p(t), p^0, u(t)) \quad (2.15)$$

$$\frac{\partial H}{\partial u}(t, x(t), p(t), p^0, u(t)) = 0 \quad (2.16)$$

où H est l'Hamiltonien associé au système (2.11) et au coût (2.12).

$$H(t, x, p, p^0, u) = \langle p, f(t, x, u) \rangle + p^0 f^0(t, x, u) \quad (2.17)$$

Définition 2.8 *L'Hamiltonien associé au problème (P) est la fonction :*

$$\begin{aligned} H : \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} \rightarrow \mathbb{R} \\ (t, x, p, u) \mapsto H(t, x, p, u) = F(t, x, u) + \langle p, f(t, x, u) \rangle \end{aligned}$$

où $\langle \cdot, \cdot \rangle$ est le produit scalaire usuel de \mathbb{R}^n .

Le problème de contrôle optimal (P) est un problème d'optimisation en dimension infinie que l'on peut écrire :

$$(P) \quad \begin{cases} \text{Min} & J(x, u), \\ (x, u) & \in AC([0, T], \mathbb{R}^n) \times Mes([0, T], \mathbb{R}^m), \\ u & \in U_{ad} \subset Mes([0, T], \mathbb{R}^m), \\ A(x, u) & = 0, \end{cases}$$

où : U_{ad} = ensemble des commandes admissibles.

$$A(x, u) = \begin{bmatrix} S(x, u) \\ \Phi_0(x) \\ \Phi_1(x) \end{bmatrix}$$

avec :

$$S : \begin{cases} AC([0, T], \mathbb{R}^n) \times Mes([0, T], \mathbb{R}^m) \rightarrow L^1([0, T], \mathbb{R}^n), \\ (x, u) \mapsto (t \mapsto \dot{x}(t) - f(t, x(t), u(t))), \end{cases}$$

$$\Phi_0 : \begin{cases} AC([0, T], \mathbb{R}^n) \rightarrow \mathbb{R}^p, \\ x \mapsto \Psi_0(0, x(T)), \end{cases}$$

$$\Phi_1 : \begin{cases} AC([0, T], \mathbb{R}^n) \rightarrow \mathbb{R}^q, \\ x \mapsto \Psi_1(T, x(T)), \end{cases}$$

Le lagrangien associé au problème (P), s'écrit :

$$L(x, u, \lambda) = J(x, u) + \langle A(x, u), \lambda \rangle$$

Si on note $\lambda = (p, \mu_0, \mu_1)$ on a :

$$\begin{aligned} L(x, u, \lambda) &= g(0, x(0), T, x(T)) + \int_0^T l(t, x(t), u(t)) dt \\ &+ \int_0^T (\dot{x}(t) - f(t, x(t), u(t))) \backslash p(t) dt \\ &+ (\Psi_0(0, x(0))) \backslash \mu_0 + (\Psi_1(T, x(T))) \backslash \mu_1 \end{aligned}$$

Les conditions nécessaires de solution du problème (P) ont été établies par Pontryaguin en 1962.

– Si t_f^* est libre :

$$H(t_f^*, x^*(t_f^*), u^*(t_f^*), p^*(t_f^*)) + \frac{\partial \Phi}{\partial t_f}(t_0^*, x^*(t_0^*), t_f^*, x^*(t_f^*), \kappa_0^*, \kappa_1^*) = 0$$

1. On cherche les commandes H -minimales : Pour $t \in [t_0, t_f]$ fixé on résout :

$$\{ \text{Min}_{v \in U} H(t, x(t), v, p(t)) \rightarrow u^*(t) = u^*(t, x(t), p(t)),$$

2. On injecte la solution obtenue en (1) dans les systèmes différentiels en (x, p) et on obtient un système différentiel aux deux bouts. (*Two Point boundary value Problem* \equiv *TPBVP*) de $2n$ équations.

2.8 Méthodes numériques en contrôle optimale

On distingue deux types de méthodes numériques en contrôle optimal : les méthodes directes et les méthodes indirectes. Les méthodes directes consistent à discrétiser l'état et le contrôle, et réduisent le problème à un problème d'optimisation non linéaire (programmation non linéaire, ou "nonlinear programming"). Les méthodes indirectes consistent à résoudre numériquement, par une méthode de tir ("shooting method"), un problème aux valeurs limites obtenu par application du principe du maximum, il existe aussi des méthodes hybrides, qui sont un mélange des deux approches.

2.8.1 Méthodes Indirectes

Méthode de tir simple

Le principe est le suivant. Considérons le problème de contrôle optimal sous forme de Bolza, et supposons que le temps final t_f 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 $\dot{\vartheta}(t) = F(t, \vartheta(t))$, où $\vartheta(t) = (x(t), p(t))$, et les conditions initiales, finales, et les conditions de transversalité, se mettent sous la forme $R(\vartheta(t_0), \vartheta(t_f)) = 0$. Finalement, on obtient le problème aux valeurs limites

$$\begin{cases} \dot{\vartheta}(t) = F(t, \vartheta(t)), \\ R(\vartheta(t_0), \vartheta(t_f)) = 0, \end{cases} \quad (2.18)$$

Notons $\vartheta(t, \vartheta_0)$ la solution du problème de Cauchy.

$$\dot{\vartheta}(t) = F(t, \vartheta(t)), \quad \vartheta(t_0) = \vartheta_0$$

et posons $G(\vartheta_0) = R(\vartheta_0, \vartheta(t_f, \vartheta_0))$. Le problème (1.19) aux valeurs limites est alors équivalent à

$$G(\vartheta_0) = 0$$

i.e, il s'agit de *déterminer un zéro de la fonction G* . Ceci peut se résoudre par une méthode de Newton.

Remarque 2.9 *Si le temps final t_f est libre, on peut se ramener à la formulation précédante en considérons t_f comme une inconnue auxiliaire. On augmente alors la dimension de l'état en considérons l'équation supplémentaire $\frac{dt_f}{dt} = 0$. On peut utiliser le même artifice si le contrôle est bang-bang, pour déterminer les temps de commutation. Il peut cepedant s'avérer préférable, lorsque le temps final est libre, d'utiliser la condition de transversalité sur le Hamiltonien.*

Méthode de tir multiple

Par rapport à la méthode de tir simple, la méthode de tir multiple découpe l'intervalle $[t_0, t_f]$ en N intervalles $[t_i, t_{i+1}]$, et se donne comme inconnues les valeurs $\vartheta(t_i)$ au début de chaque sous-intervalle. Il faut prendre en compte des conditions de recollement en chaque temps t_i (condition de continuité). L'intérêt est d'améliorer la stabilité de la méthode.

De manière plus précise, considérons un problème de contrôle optimal générale. L'application du principe du maximum réduit le problème à un *problème aux valeurs limites* du type

$$\dot{\vartheta}(t) = F(t, \vartheta(t)) = \begin{cases} F_0(t, \vartheta(t)), & \text{si } t_0 \leq t < t_1 \\ F_1(t, \vartheta(t)), & \text{si } t_1 \leq t < t_2 \\ \vdots & \\ F_s(t, \vartheta(t)), & \text{si } t_s \leq t \leq t_f \end{cases} \quad (*)$$

où $\vartheta = (x, p) \in \mathbb{R}^{2n}$ (p est le vecteur adjoint), et $t_1, t_2, \dots, t_s \in [t_0, t_f]$.

Remarque 2.10 *A priori le temps final t_f est inconnu. Par ailleurs dans la méthode de tir multiple le nombre s de commutations doit être fixé; on le détermine lorsque c'est possible par une analyse géométrique du problème.*

La méthode de tir multiple consiste à subdiviser l'intervalle $[t_0, t_f]$ en N sous-intervalles, la valeur de $\vartheta(t)$ au début de chaque sous-intervalles étant inconnue. Plus précisément, soit $t_0 < \sigma_1 < \dots < \sigma_k < t_f$ une subdivision fixée de l'intervalle $[t_0, t_f]$. En tout point σ_j la fonction ϑ est *continue*. On peut considérer σ_j comme un point de commutation fixe, en lequel on a :

$$\begin{cases} \vartheta(\sigma_j^+) = \vartheta(\sigma_j^-), \\ \sigma_j = \sigma_j^*, \end{cases} \quad \text{fixé}$$

On définit maintenant les *noeuds*

$$\{\tau_1, \dots, \tau_m\} = \{t_0, t_f\} \cup \{\sigma_1, \dots, \sigma_k\} \cup \{t_1, \dots, t_s\} \quad (2.19)$$

Finalement on est conduit au *problème aux valeurs limites*.

$$\dot{\vartheta}(t) = F(t, \vartheta(t)) = \begin{cases} F_1(t, \vartheta(t)), & \text{si } \tau_1 \leq t < \tau_2 \\ F_2(t, \vartheta(t)), & \text{si } \tau_2 \leq t < \tau_3 \\ \vdots \\ F_{m-1}(t, \vartheta(t)), & \text{si } \tau_{m-1} \leq t \leq \tau_m \end{cases} \quad (**)$$

$$\forall j \in 2, \dots, m-1 \quad r_j(\tau_j, \vartheta(\tau_j^-), \vartheta(\tau_j^+)) = 0$$

$$r_m(\tau_m, \vartheta(\tau_1), \vartheta(\tau_m)) = 0$$

où $\tau_1 = t_0$ est fixé, $\tau_m = t_f$, et les r_j représentent les conditions intérieures ou limites précédentes.

Remarque 2.11 *On améliore la stabilité de la méthode en augmentant le nombre de noeuds. C'est là en effet le principe de la méthode de tir multiple, par opposition à la méthode de tir simple où les erreurs par rapport à la condition initiale évoluent exponentiellement en fonction de $t_f - t_0$. Bien sûr dans la méthode de tir multiple il y a beaucoup plus d'inconnues que la méthode de tir simple, mais éventuellement l'intégration du système (*) peut se paralléliser.*

Posons $\vartheta_j^+ = \vartheta(\tau_j^+)$, et soit $\vartheta(t, \tau_{j-1}, \vartheta_{j-1}^+)$ la solution du problème de Cauchy.

$$\dot{\vartheta}(t) = F(t, \vartheta(t)), \quad \vartheta(\tau_{j-1}) = \vartheta_{j-1}^+$$

On a

$$\vartheta(\tau_j^-) = \vartheta(\tau_j^-, \tau_{j-1}, \vartheta_{j-1}^+).$$

Les conditions intérieures et frontières s'écrivent

$$\begin{aligned} \forall j \in \{2, \dots, m-1\} \quad r_j(\tau_j, \vartheta(\tau_j^-, \tau_{j-1}, \vartheta_{j-1}^+), \vartheta_j^+) &= 0, \\ r_m(\tau_m, \vartheta_1^+, \vartheta(\tau_m^-, \tau_{m-1}, \vartheta_{m-1}^+)) &= 0 \end{aligned} \quad (2.20)$$

Posons maintenant

$$Z = (\vartheta_1^+, \tau_m, \vartheta_2^+, \tau_2, \dots, \vartheta_{m-1}^+, \tau_{m-1})^T \in \mathbb{R}^{(2n+1)(m-1)}$$

(où $\vartheta \in \mathbb{R}^{2n}$). Alors les conditions (1.21) sont vérifiées si :

$$G(Z) = \begin{pmatrix} r_m(\tau_m, \vartheta_1^+, \vartheta(\tau_m^-, \tau_{m-1}, \vartheta_{m-1}^+)) \\ r_2(\tau_2, \vartheta(\tau_2^-, \tau_1, \vartheta_1^+), \vartheta_2^+) \\ \vdots \\ r_{m-1}(\tau_m, \vartheta(\tau_{m-1}^-, \tau_{m-2}, \vartheta_{m-2}^+), \vartheta_{m-1}^+) \end{pmatrix} = 0 \quad (***)$$

On s'est donc ramené à déterminer un zéro de la fonction G , qui est définie sur un espace vectoriel dont la dimension est proportionnelle au nombre de points de commutation et de points de la subdivision. L'équation $G = 0$ peut alors être résolue itérativement par une méthode du type Newton.

2.8.2 Méthodes Directes

Les méthodes directes consistent à transformer la problème de contrôle optimale en un *problème d'optimisation non linéaire en dimension finie*.

Discrétisation totale

C'est la méthode la plus évidente lorsqu'on aborde un problème de contrôle, on se ramène à un problème d'optimisation non linéaire en dimension finie (ou problème de programmation non linéaire) de la forme

$$\min_{Z \in C} F(Z) \quad (2.21)$$

où $Z = (x_1, \dots, x_N, u_1, \dots, u_n)$, et

$$\begin{aligned} C = \{Z \mid & g_i(Z) = 0, i \in \{1, \dots, r\} \\ & g_j(Z) \leq 0, j \in \{r+1, \dots, m\} \end{aligned} \quad (2.22)$$

Plus précisément, la méthode consiste à choisir les contrôles dans un espace de dimension finie, et à utiliser une méthode d'intégration numérique des équations différentielles. Considérons donc une subdivision $t_0 < t_1 < \dots < t_N = t_f$ de l'intervalle $[t_0, t_f]$. Réduisons l'espace des contrôles en considérons (par exemple) des contrôles constants par morceaux selon cette subdivision. Par ailleurs, choisissons une discrétisation de l'équation différentielle, par exemple choisissons ici pour simplifier la méthode d'Euler explicite. On obtient alors, en posant $h_i = t_{i+1} - t_i$,

$$x_{i+1} = x_i + h_i f(t_i, x_i, u_i)$$

La discrétisation précédente conduit donc au problème de programmation non linéaire :

$$\begin{aligned} x_{i+1} &= x_i + h_i f(t_i, x_i, u_i), \quad i = 0, \dots, N-1 \\ \min & C(x_0, \dots, x_N, u_0, \dots, u_N), \\ u_i &\in U, i = 0, \dots, N-1 \end{aligned}$$

i.e un problème du type (2.21).

Remarque 2.12 *Cette méthode est très simple à mettre en oeuvre. De plus l'introduction d'éventuelles contraintes sur l'état ne pose aucun problème.*

Le tableau suivant résume les caractéristiques des méthodes directes et indirectes :

Méthodes Directes	Méthodes Indirectes
Mise en oeuvre simple, sans connaissance à priori	Connaissance à priori de la structure de la trajectoire optimale
Peu sensibles au choix de la condition initiale	Très sensibles au choix de la condition initiale
Facilité de la prise en compte de contraintes sur l'état	Difficulté théorique de la prise en compte de contraintes sur l'état
Contrôles (globalement) optimaux en boucle fermée	Contrôles (localement) optimaux en boucle ouverte
Précision numérique basse ou moyenne	Très grande précision numérique
Efficacité en basse dimension	Efficaces en toute dimension
Gourmandise en mémoire	Calculs parrallélisables
Problème des minima locaux	Petit domaine de convergence

Chapitre 3

Résolution d'un problème de contrôle optimale par la méthode de relaxation

3.1 Optimisation dynamique : problème de commande optimale associé à des EDO

3.1.1 Cas des systèmes linéaires avec coût quadratique : régulation d'un processus thermique

Position du problème On considère un système physique composé d'un four à moufle, dans la cheminée duquel est placé un barreau ; le but de l'étude est d'amener la température θ relevée en n points du barreau à une température $\theta_d \in \mathbb{R}^n$, en un temps fini t_f , qui représente l'horizon de la commande. On aura donc à déterminer la commande u , qui sera en fait l'intensité des courants de régulation, de telle sorte que, au bout du temps t_f , la température θ du barreau soit uniformément égale à θ_d , compte tenu d'un critère à minimiser.

notation $x \in \mathbb{R}^n$ température de la cheminée en n points

$\theta \in \mathbb{R}^n$ température du barreau en n points

$u \in \mathbb{R}^n$ intensité des courants envoyés dans chacun des n enroulements de chauffage

Un modèle a été obtenu en étudiant l'évolution du système autour d'un point de fonctionnement, on pose :

$$y = \begin{pmatrix} \theta_1 \\ x_1 \\ \vdots \\ \theta_i \\ x_i \\ \vdots \\ \theta_n \\ x_n \end{pmatrix}$$

ce qui conduit à associer un modèle mathématique de la forme suivante :

$$\begin{cases} \frac{dy}{dt} = Ay + Bu + f(t), & t \in [t_0, t_f] \\ y(t_0) = y_{t_0}, \end{cases}$$

y_0 état initial donné, $f(t)$ terme source donné, A et B deux matrices.

$$\theta = Cy$$

la variable θ modélise l'observation du système, la matrice C est appelée matrice d'observation. Le critère à minimiser est la combinaison de deux critères : d'une part, l'obtention d'une trajectoire du vecteur d'état telle que l'observation soit, en moyenne quadratique, aussi proche que possible de la consigne d'observation θ_d , d'autre part, le fait que la commande reste en moyenne quadratique aussi proche que possible de la commande nominale u_d .

$$J(u) = \alpha J_1(u) + \beta J_2(u)$$

où α et β servent de dosage entre les deux composantes du critère.

$J_1(u) = \int_{t_0}^{t_f} \|\theta - \theta_d\|_2^2 dt$, critère de précision

$J_2(u) = \int_{t_0}^{t_f} \|u - u_d\|_2^2 dt$, minimisation de la consommation d'énergie

Equations d'Hamilton-Pontryaguin

$$\begin{cases} \begin{cases} \frac{dy}{dt} = Ay + Bu + f(t), & t \in [t_0, t_f] \\ y(t_0) = y_0, \end{cases} \\ \begin{cases} -\frac{dp}{dt} = A'p + C'Cy - C'\theta_d, & t \in [t_0, t_f] \\ p(t_f) = 0, \end{cases} \\ \frac{\alpha}{\beta} B^T p + u - u_d = 0, & t \in [t_0, t_f] \end{cases}$$

3.1.2 Algorithme de relaxation associé

- approximation initiale de la commande : $u^{(0)}(t), t \in [t_0, t_f]$ donnée
- détermination de l'état $y^{(r+1)}(t)$ par intégration numérique de l'équation d'état

$$\begin{cases} \frac{dy^{(r+1)}(t)}{dt} = Ay^{(r+1)}(t) + Bu^{(r)}(t) + f(t), & t \in [t_0, t_f] \\ y^{(r+1)}(t_0) = y_0, \end{cases}$$

- détermination de l'état adjoint $p^{(r+1)}(t)$ par intégration numérique dans le sens rétrograde de l'équation d'état adjoint

$$\begin{cases} -\frac{dp^{(r+1)}(t)}{dt} = Ap^{(r+1)}(t) + C'Cy^{(r+1)}(t) - C'\theta_d, & t \in [t_0, t_f] \\ p^{(r+1)}(t_f) = 0, \end{cases}$$

- détermination de la commande :

$$u^{(r+1)}(t) = u_d - \frac{\alpha}{\beta} B'p^{(r+1)}(t), t \in [t_0, t_f]$$

3.2 Etude d'un système en anneau

Considérons un processus dont un modèle est défini par :

$$\dot{y} = f(y, u, t); \quad y(t_0) = y_0, \quad t \in [t_0, t_f] \quad \text{et} \quad y \in C([t_0, t_f]; \mathbb{R}^n)$$

où $u(t)$ est la loi de commande, $u \in C([t_0, t_f], \mathbb{R}^m)$, $t_f > t_0$. On se propose de conduire ce processus de t_0 à t_f , à partir de l'état initial y_0 , en minimisant le critère :

$$J = \int_{t_0}^{t_f} r(y, u, t) dt$$

et en respectant la contrainte : $u \in U_{ad}$, où U_{ad} est un convexe fermé représentant l'ensemble des commandes admissibles. Le principe du maximum de Pontriaguine nous permet d'affirmer que les conditions nécessaires d'optimalité sont définies par le système suivant :

$$\begin{cases} \dot{y} = f(y, u, t); & y(t_0) = y_0, & t \in [t_0, t_f] \\ -\dot{p} = \frac{\partial r}{\partial y} + \left(\frac{\partial f}{\partial y}\right)' \times p; & p(t_f) = 0, & \text{où } p(t) \text{ est l'état adjoint} \\ 0 \in \frac{\partial r}{\partial u} + \left(\frac{\partial f}{\partial u}\right)' \times p + \partial \Psi_{U_{ad}}, \end{cases}$$

Pour illustrer la méthodologie, nous considérons un exemple simple correspondant à un système à structure en "anneau", avec couplage unidirectionnel, pouvant être décomposé en n sous-systèmes scalaires identiques. Il s'agit de minimiser le critère :

$$\begin{cases} J = \frac{1}{2} \int_{t_0}^{t_f} \{\|y - y_d\|^2 + K\|u\|^2\} dt, \\ u_i \in [u_i^m, u_i^M], \end{cases} \quad i \in 1, 2 \quad (3.1)$$

sous la contrainte supplémentaire :

$$\begin{cases} \dot{y}_1 = -y_1 + ay_2 + u_1, & y_1(t_0) = 0 \\ \dot{y}_2 = ay_1 - y_2 + u_2, & y_2(t_0) = 0 \end{cases} \quad (3.2)$$

où a est une constante réelle positive. Les conditions nécessaires d'optimalité sont alors :

$$\begin{cases} (3a) \begin{cases} \dot{y}_1 = -y_1 + ay_2 + u_1, & y_1(t_0) = 0 \\ \dot{y}_2 = ay_1 - y_2 + u_2, & y_2(t_0) = 0 \end{cases} \\ (3b) \begin{cases} \dot{p}_1 = p_1 - ap_2 - y_1 + y_{1d}, & p_1(t_f) = 0 \\ \dot{p}_2 = p_2 - ap_1 - y_2 + y_{2d}, & p_2(t_f) = 0 \end{cases} \\ (3c) \begin{cases} 0 \in Ku_1 + p_1 + \partial \Psi_{U_{ad_1}}, \\ 0 \in Ku_2 + p_2 + \partial \Psi_{U_{ad_2}}, \end{cases} \end{cases} \quad (3.3)$$

où

$$\Psi_{U_{ad}} = \begin{cases} 0 & \text{si } u_i^m \leq u_i \leq u_i^M \\ +\infty & \text{si } u_i > u_i^M \\ -\infty & \text{si } u_i < u_i^m \end{cases}$$

3.2.1 Algorithmes de résolution

Pour résoudre le système (1.3), nous proposons deux types de décomposition des équations :

Méthode de l'état-adjoint par relaxation (Costate Relaxation Method)

La variante séquentielle (Sequential approach) Elle consiste à intégrer alternativement et en séquence l'équation d'état (3a) et l'équation d'état-adjoint (3b), respectivement dans le sens direct et dans le sens rétrograde, puis de résoudre les inéquations (3c).

Ceci conduit aux opérations itératives suivantes :

- au départ, choix d'une première approximation pour la loi de commande optimale, notée $u^0(t), t \in [t_0, t_f]$.
- détermination de la variable d'état $y^{(r+1)}(t)$, où r désigne le numéro de l'itération, par intégration des équations d'état à partir de l'état initial $y^{(r+1)}(t_0) = 0$:

$$\begin{cases} \dot{y}_1^{(r+1)} = -y_1^{(r+1)} + ay_2^{(r+1)} + u_1^{(r)}, & y_1^{(r+1)}(t_0) = 0 \\ \dot{y}_2^{(r+1)} = ay_1^{(r+1)} - y_2^{(r+1)} + u_2^r, & y_2^{(r+1)}(t_0) = 0 \end{cases} \quad (3.4)$$

- détermination de l'état adjoint $p^{(r+1)}(t)$ par intégration dans le sens rétrograde des équations d'état adjoint , à partir de la valeur finale $p^{(r+1)}(t_f) = 0$

$$\begin{cases} \dot{p}_1^{(r+1)} = p_1^{(r+1)} - ap_2^{(r+1)} - y_1^{(r+1)} + y_{1d}, \\ \dot{p}_2^{(r+1)} = p_2^{(r+1)} - ap_1^{(r+1)} - y_2^{(r+1)} + y_{2d}, \\ p_1^{(r+1)}(t_f) = 0 \quad p_2^{(r+1)}(t_f) = 0, \end{cases} \quad (3.5)$$

- résolution des inéquations :

$$\begin{cases} 0 \in Ku_1^{(r+1)} + p_1^{(r+1)} + \partial\Psi_{U_{ad1}}, \\ 0 \in Ku_2^{(r+1)} + p_2^{(r+1)} + \partial\Psi_{U_{ad2}}, \end{cases} \quad (3.6)$$

La variante parallèle asynchrone (Asynchronous Parallel Approach)

Elle consiste à intégrer simultanément les équations d'état (3a) et les équations d'état adjoint (3b) à l'aide de deux processeurs, chacun communiquant à l'autre le résultat de ses calculs par paquets de composantes, chaque paquet correspondant en fait à l'ensemble des valeurs sur un intervalle

$[\tau_h, \tau_{h+1}] \subset [t_0, t_f]$. La détermination de la loi de commande (équation (3c)) est confiée indifféremment à l'un ou l'autre processeur, voir même à un troisième. Précisons que l'asynchronisation résulte du fait que chaque processeur effectue ses calculs à son rythme propre avec les données les plus fraîches dont il dispose.

Ceci peut se traduire par les équations suivantes :

$$\left\{ \begin{array}{l} \left\{ \begin{array}{l} \dot{y}_1^{(h)} = -y_1^{(h)} + ay_2^{(h)} + v_1^{(h)}, \\ y_1^{(h)}(\tau_h) = \varpi_1^{(h-1)}(\tau_h), \\ \dot{y}_2^{(h)} = ay_1^{(h)} - y_2^{(h)} + v_2^{(h)}, \\ y_2^{(h)}(\tau_h) = \varpi_2^{(h-1)}(\tau_h), \end{array} \right. \\ \\ \left\{ \begin{array}{l} \dot{p}_1^{(k)} = p_1^{(k)} - ap_2^{(k)} - \varpi_1^{(k)} + y_{1d}, \\ p_1^{(k)}(\tau_{k+1}) = q_1^{(k)}(\tau_{k+1}), \\ \dot{p}_2^{(k)} = p_2^{(k)} - ap_1^{(k)} - \varpi_2^{(k)} + y_{2d}, \\ p_2^{(k)}(\tau_{k+1}) = q_2^{(k)}(\tau_{k+1}), \end{array} \right. \\ \\ \left\{ \begin{array}{l} 0 \in ku_1^{(l)} + q_1^{(l)} + \partial\Psi_{U_{ad1}}, \\ 0 \in ku_2^{(l)} + q_2^{(l)} + \partial\Psi_{U_{ad2}}, \end{array} \right. \end{array} \right. \quad (3.7)$$

où $\varpi_i^{(k)}, q_i^{(l)}$ et $v_i^{(k)}$ représentent les données disponibles à l'instant d'intégration considéré.

Méthode de décentralisation et coordination par relaxation (Relaxation Decentralisation Method)

Décomposition en sous-problèmes Lorsque U_{ad} se laisse décomposer en un produit cartésien de n convexes fermés U_{adi} , on peut décomposer le vecteur de commande u en n sous-vecteurs u_i , et effectuer une décomposition semblable du vecteur d'état y . Le problème (1.3) peut alors se réécrire comme une famille de n sous-problèmes tels que :

$$\left\{ \begin{array}{l} \dot{y}_1 + y_i - u_i = ay_i, \quad y_i(t_0) = 0 \\ -\dot{p}_i + p_i - y_i = -ap_j - y_{id}, \quad p_i(t_f) = 0 \\ 0 \in ku_i + p_i + \partial\Psi_{U_{adi}}, \end{array} \right. \quad (3.8)$$

Ces sous-problèmes sont couplés, comme l'expriment les seconds membres des équations de (1.8), la relaxation du sous-problème n° i consiste à le découpler des autres, en considérant les variables de couplages (y_j, p_j, u_j) comme connues, et à résoudre séparément sur ces bases.

Coordination des sous-problèmes Nous appelons application de Jacobi, associée à la résolution du problème décentraliser l'application \bar{F} qui à tout v fait correspondre $w = \bar{F}(v)$ tel que pour tout $i, w_i = \bar{F}_i(v)$ est la solution du i ème sous-problème relaxé :

$$D_i(w_i) + L_i(w_i) = -L_j(v_j)$$

où, dans le cadre de l'exemple illustratif :

$$D_i(w_i) = \left[\frac{dy_i}{dt}, -\frac{dp_i}{dt}, \partial\Psi_{U_{adi}}(u_i) \right]'$$

$$L_i(w_i) = \begin{bmatrix} y_i - u_i \\ p_i - y_i \\ p_i + ku_i \end{bmatrix}$$

$$L_j(v_j) = \begin{bmatrix} -ay_j \\ -ap_j + y_{id} \\ 0 \end{bmatrix}$$

Dans ces conditions, la solution $x(t)$ du problème global

$$D(x) + L(x) = 0 \quad (3.9)$$

peut être considérée comme le point fixe de l'application de Jacobi associée :

$$x(t) = \bar{F}[x(t)] \quad (3.10)$$

De nombreux algorithmes itératifs peuvent être mis en oeuvre dans le recherche de ce point fixe à partir d'un vecteur initial $x^{(0)}$, et constituent, s'ils convergent, autant de méthodes de coordination.

Ici encore, deux variantes peuvent être envisagées, une variante séquentielle où on applique, pour chaque sous-problème, la méthode de l'état adjoint par relaxation, une variante parallèle asynchrone où chaque sous-problème est résolu par un processeur échangeant ses infos avec les autres. Dans les deux cas, la coordination est effectuée en utilisant les données disponibles.

3.2.2 Analyse de la convergence

Ayant associé aux algorithmes précédents une équation de point fixe du type (3.10), on peut envisager deux méthodes distinctes d'analyse de la convergence.

Contraction en norme vectorielle

On appelle norme vectorielle, et on note $\| \cdot \|$, le vecteur des normes scalaires des sous-vecteurs x_i :

$$\|x\| = (\|x_1\|, \dots, \|x_i\|, \dots, \|x_n\|)'$$

avec $x = (x_1, x_2, \dots, x_i, \dots, x_n)'$. Soit F une application, de domaine de définition de $F \subset D(F)$, à valeurs dans un espace vectoriel normé E , on suppose que :

$$D(F) \neq \emptyset \quad \text{et} \quad F[D(F)] \subset D(F) \quad (3.11)$$

$$F \quad \text{admet un point fixe unique} \quad \hat{x} \in D(F) \quad (3.12)$$

On dit que l'application F est contractante en \hat{x} pour la norme vectorielle $\| \cdot \|$ s'il existe une matrice G de type $n \times n$ non négative et de rayon spectrale $\rho(G) < 1$ tel que :

$$\|F(x) - F(\hat{x})\| \leq G\|x - \hat{x}\| \quad \forall x \in D(F) \quad (3.13)$$

On peut alors énoncer le résultat suivant[10] : Les hypothèses (3.11), (3.12) et (3.13) étant vérifiées, tout algorithme de relaxation, partant de $x^{(0)} \in D(F)$, converge vers \hat{x} point fixe de F .

Contraction en norme scalaire

On dit que l'application F est contractante en \hat{x} , pour la norme scalaire $\|\cdot\|$, s'il existe un nombre réel $g \in]0, 1[$ tel que :

$$\|F(x) - F(\hat{x})\| \leq g\|x - \hat{x}\|, \quad \forall x \in D(F) \quad (3.14)$$

On peut alors énoncer le résultat suivant :

$$\|x\|_{g,G} = \max_{i \in \{1,2,\dots,n\}} \frac{\|x_i\|}{\Gamma_i^g}$$

où le vecteur Γ^g de composantes Γ_i^g est le vecteur propre de la matrice G associé à la plus grande valeur propre g de la matrice G . On obtient un résultat de convergence analogue à celui énoncé au paragraphe précédent. Enfin, signalons que les critères de convergence énoncés précédemment sont liés entre eux par le fait que si l'application F est contractante en norme vectorielle, alors on peut construire une norme scalaire pour laquelle F sera également contractante.

3.3 Application numérique : Résolution d'un problème de contrôle optimale à entrée libre

3.3.1 Position du problème

On considère le problème suivant :

$$\begin{cases} \text{Déterminer } u \in U_{ad} \text{ tel que,} \\ J(u) \leq J(v), \quad \forall v \in U_{ad}; \end{cases}$$

où

$$J(u) = \frac{1}{2} \int_0^{t_f} ((x - x_d)^2 + ku^2) dt$$

x_d correspond à un état désiré, u est la commande et k sert à doser deux critères distincts à minimiser, un critère de précision et un autre de minimisation d'énergie et U_{ad} est l'ensemble des commandes admissibles. sous les contraintes suivantes :

$$\begin{cases} \dot{x}_1 = -bx_1 + ax_2, \\ \dot{x}_2 = ax_1 - bx_2 + u, \\ x_0 = z \in X_0 = \{z \in \mathfrak{R}^2 : Gz = \gamma, -2 \leq z \leq 2\}, \\ k^* x(t_f) = g, \end{cases}$$

avec $G = (1, 2)$, $\gamma = 3$, $k^* = (1, 0)$, $g = 2$, $t_f = 2$, $a > 0, b > 0$.

On peut formuler le problème d'une autre manière :

$$\begin{cases} \dot{x}_1 = -bx_1 + ax_2, \\ \dot{x}_2 = ax_1 - bx_2 + u, \\ x_1(0) = z_1, x_2(0) = z_2 \quad \text{tel que} \quad z_1 + 2z_2 = 3, \quad -2 \leq z_1 \leq 2, \quad -2 \leq z_2 \leq 2, \\ x_1(t_f) = 2, \end{cases} \quad (3.15)$$

La forme matricielle du système d'état s'écrit :

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} -b & a \\ a & -b \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u,$$

avec

$$J(u) = \frac{1}{2} \int_0^{t_f} \left[(x_1 - x_{1d}, x_2 - x_{2d}) \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 - x_{1d} \\ x_2 - x_{2d} \end{pmatrix} + ku^2 \right] dt.$$

L'Hamiltonien est donné par :

$$\begin{aligned} H(x(t), p(t), u(t), t) &= \frac{1}{2} ((x_1 - x_{1d})^2 + (x_2 - x_{2d})^2 + ku^2) \\ &\quad + (p_1(t), p_2(t)) \begin{pmatrix} -b & a \\ a & -b \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + (p_1(t), p_2(t)) \begin{pmatrix} 0 \\ u \end{pmatrix} \\ &= \frac{1}{2} ((x_1 - x_{1d})^2 + (x_2 - x_{2d})^2 + ku^2) + p_1(t)(-bx_1 + ax_2) \\ &\quad + p_2(t)(ax_1 - bx_2) + p_2(t)u. \end{aligned}$$

Les équations d'optimalité sont données :

$$\begin{cases} \begin{cases} \frac{dx_1}{dt} = \frac{\partial H}{\partial p_1} = -bx_1 + ax_2, \\ \frac{dx_2}{dt} = \frac{\partial H}{\partial p_2} = ax_1 - bx_2 + u, \\ x(0) = x_0, \end{cases} \\ \begin{cases} -\frac{dp_1}{dt} = \frac{\partial H}{\partial x_1} = -bp_1 + ap_2 + x_1 - x_{1d}, & p_1(0) = a_1, p_1(t_f) = a_2 \\ -\frac{dp_2}{dt} = \frac{\partial H}{\partial x_2} = ap_1 - bp_2 + x_2 - x_{2d}, & p_2(0) = a_3, p_2(t_f) = a_4 \end{cases} \\ \frac{\partial H}{\partial u} = 0 = ku + p_2(t), \end{cases}$$

où $a_i, i = \overline{1, 4}$ seront précisées ci-dessous.

On adjoint au système précédent les conditions de transversalité : $\exists(\kappa_0, \kappa_1) \neq 0$ tel que :

$$\varphi(0, z, t_f, x(t_f), \kappa_0, \kappa_1) = (\Psi_0(0, z)/\kappa_0) + (\Psi_1(t_f, x(t_f))/\kappa_1),$$

avec

$$\Psi_0(0, z) = Gz - \gamma, \quad \Psi_1(t_f, x(t_f)) = k^*x(t_f) - g, \quad (3.16)$$

$$\varphi(0, z, t_f, x(t_f), \kappa_0, \kappa_1) = (Gz - \gamma \backslash \kappa_0) + (k^*x(t_f) - g \backslash \kappa_1). \quad (3.17)$$

$$\begin{cases} p(0) = \frac{\partial \varphi}{\partial z}(0, z, t_f, x(t_f), \kappa_0, \kappa_1) = G' \kappa_0, \\ p(t_f) = -\frac{\partial \varphi}{\partial x(t_f)}(0, z, t_f, x(t_f), \kappa_0, \kappa_1) = -k'^* \kappa_1, \end{cases} \quad (3.18)$$

$$\begin{cases} (p_1(0), p_2(0)) = (1, 2)\kappa_0 = (\kappa_0, 2\kappa_0), \\ (p_1(t_f), p_2(t_f)) = -(1, 0)\kappa_1 = (-\kappa_1, 0), \end{cases}$$

Les équations d'Hamilton-Pontryaguin s'écrivent :

$$\begin{cases} \begin{cases} \dot{x}_1 = -bx_1 + ax_2, \\ \dot{x}_2 = ax_1 - bx_2 + u, \\ x(0) = x_0, \end{cases} \\ \begin{cases} -\dot{p}_1 = -bp_1 + ap_2 + x_1 - x_{1d}, & p_1(0) = \kappa_0, p_1(t_f) = -\kappa_1 \\ -\dot{p}_2 = ap_1 - bp_2 + x_2 - x_{2d}, & p_2(0) = 2\kappa_0, p_2(t_f) = 0 \end{cases} \\ 0 = ku + p_2(t), \end{cases}$$

D'où la forme matricielle :

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ -\dot{p}_1 \\ -\dot{p}_2 \\ 0 \\ -p_1(0) \\ p_1(t_f) \end{pmatrix} = \begin{pmatrix} -b & a & 0 & 0 & 0 & 0 & 0 \\ a & -b & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & -b & a & 0 & 0 & 0 \\ 0 & 1 & a & -b & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & k & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ p_1 \\ p_2 \\ u \\ \kappa_0 \\ \kappa_1 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ -x_{1d} \\ -x_{2d} \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Dans le système précédent, notons que $p_1(t_f)$, $p_1(0)$, κ_0 , κ_1 sont des inconnues dans les relations suivantes $p_1(t_f) = -\kappa_1$, $p_1(0) = \kappa_0$, $p_2(0) = 2\kappa_0$. Pour déterminer la solution du problème de contrôle optimale à entrée libre, on considère la méthode de tir [20] qui permet de déterminer $p(0)$ tel que la condition sur l'état final $x_1(t_f) = 2$ soit satisfaite, et de déterminer $p(t_f)$ tel que la condition sur l'état initial $x_1(0) + 2x_2(0) = 3$ soit satisfaite.

Ainsi, on aura un problème aux deux bouts qu'on peut formuler de cette manière :

$$\left\{ \begin{array}{l} \dot{x}_1 = -bx_1 + ax_2, \\ \dot{x}_2 = ax_1 - bx_2 + u, \\ x(0) = x_0, \\ -\dot{p}_1 = -bp_1 + ap_2 + x_1 - x_{1d}, \quad p_1(0), p_1(t_f) \\ -\dot{p}_2 = ap_1 - bp_2 + x_2 - x_{2d}, \quad p_2(0), p_2(t_f) \\ \kappa_0 = p_1(0), \\ \kappa_1 = -p_1(t_f), \\ u = -\frac{p_2(t)}{k}, \end{array} \right.$$

La méthode de relaxation revient à résoudre de manière itérative, le problème de point fixe suivant :

$$\left\{ \begin{array}{l} \dot{x}_1 = -bx_1 + ax_2, \\ \dot{x}_2 = ax_1 - bx_2 + u, \\ x_1(0) = 1, x_2(0) = 1 \\ \dot{p}_1 = bp_1 - ap_2 - x_1 + x_{1d}, \quad p_1(0) = \kappa_0, p_1(t_f) = -\kappa_1 \\ \dot{p}_2 = -ap_1 + bp_2 - x_2 + x_{2d}, \quad p_2(0) = 2\kappa_0, p_2(t_f) = 0 \\ u = -\frac{p_2(t)}{k}, \end{array} \right.$$

3.3.2 La solution exacte

On utilise la méthode de dérivation au niveau des équations. On obtient :

$$\dot{p}_1 = bp_1 - ap_2 - x_1 + x_{1d}, \quad (3.19)$$

$$\ddot{p}_1 = b\dot{p}_1 - a\dot{p}_2 - \dot{x}_1, \quad (3.20)$$

$$\ddot{p}_1 = b(bp_1 - ap_2 - x_1 + x_{1d}) - a(-ap_1 + bp_2 - x_2 + x_{2d}) - (-bx_1 + ax_2), \quad (3.21)$$

$$\ddot{p}_1 = b^2p_1 - abp_2 - bx_1 + bx_{1d} + a^2p_1 - abp_2 + ax_2 - ax_{2d} + bx_1 - ax_2, \quad (3.22)$$

$$\ddot{p}_1 = (a^2 + b^2)p_1 - 2abp_2 + bx_{1d} - ax_{2d}, \quad (3.23)$$

d'où

$$\ddot{p}_1 = (a^2 + b^2)p_1 - 2abp_2 + bx_{1d} - ax_{2d}$$

De la même manière :

$$\dot{p}_2 = -ap_1 + bp_2 - x_2 + x_{2d}$$

$$\ddot{p}_2 = -a\dot{p}_1 + b\dot{p}_2 - \dot{x}_2,$$

$$\ddot{p}_2 = -a(bp_1 - ap_2 - x_1 + x_{1d}) + b(-ap_1 + bp_2 - x_2 + x_{2d}) - (ax_1 - bx_2 - \frac{p_2}{k}),$$

$$\ddot{p}_2 = -abp_1 + a^2p_2 + ax_1 - ax_{1d} - abp_1 + b^2p_2 - bx_2 + bx_{2d} - ax_1 + bx_2 - ax_1 + bx_2 + \frac{p_2}{k},$$

$$\ddot{p}_2 = -2abp_1 + (a^2 + b^2 + \frac{1}{k})p_2 - ax_{1d} + bx_{2d}$$

D'où

$$\ddot{p}_2 = (a^2 + b^2 + \frac{1}{k})p_2 - abp_1 - ax_{1d} + bx_{2d} \quad (3.24)$$

Dérivons 2 fois l'équation (3.24), on obtient :

$$p_1^{(4)} = (a^2 + b^2)\ddot{p}_1 - 2ab\ddot{p}_2, \quad (3.25)$$

$$p_1^{(4)} = (a^2 + b^2)\ddot{p}_1 - 2ab(-2abp_1 + (a^2 + b^2 + \frac{1}{k})p_2 - ax_{1d} + bx_{2d}), \quad (3.26)$$

$$p_1^{(4)} = (a^2 + b^2)\ddot{p}_1 + 4a^2b^2p_1 - 2ab(a^2 + b^2 + \frac{1}{k})p_2 + 2a^2bx_{1d} - 2ab^2x_{2d}; \quad (3.27)$$

D'où

$$p_1^{(4)} = (a^2 + b^2)\ddot{p}_1 + 4a^2b^2p_1 - 2ab(a^2 + b^2 + \frac{1}{k}) + 2a^2bx_{1d} - 2ab^2x_{2d}, \quad (3.28)$$

(3.28) entraîne :

$$2abp_2 = (a^2 + b^2)p_1 - \ddot{p}_1 + bx_{1d} - ax_{2d}, \quad (3.29)$$

En injectant (3.29) dans (3.28), on obtient :

$$p_1^{(4)} = (a^2 + b^2)\ddot{p}_1 + 4a^2b^2p_1 - (a^2 + b^2 + \frac{1}{k})((a^2 + b^2)p_1 - \ddot{p}_1 + bx_{1d} - ax_{2d}) + 2a^2bx_{1d} - 2ab^2x_{2d}$$

$$p_1^{(4)} = (a^2 + b^2)\ddot{p}_1 + 4a^2b^2p_1 - ((a^2 + b^2)^2 + \frac{1}{k}(a^2 + b^2))p_1 + (a^2 + b^2 + \frac{1}{k})\ddot{p}_1 - b(a^2 + b^2 + \frac{1}{k})x_{2d} + 2a^2bx_{1d} - 2ab^2x_{2d}$$

$$p_1^{(4)} = (2a^2 + 2b^2 + \frac{1}{k})\ddot{p}_1 - ((a^2 + b^2)^2 + \frac{a^2 + b^2}{k} - 4a^2b^2)p_1 + (-b^3 - \frac{b}{k} + a^2b)x_{1d} + (a^3 - ab^2 + \frac{a}{k})x_{2d},$$

d'où

$$p_1^{(4)} - (2a^2 + 2b^2 + \frac{1}{k})\ddot{p}_1 - ((a^2 + b^2)^2 + \frac{a^2 + b^2}{k} - 4a^2b^2)p_1 = (-b^3 - \frac{b}{k} + a^2b)x_{1d} + (a^3 - ab^2 + \frac{a}{k})x_{2d} \quad (3.30)$$

L'équation caractéristique correspondante à l'équation (3.30) s'écrit :

$$C^4 - 2(a^2 + b^2 + \frac{1}{2k})C^2 + ((a^2 + b^2)^2 + \frac{a^2 + b^2}{k} - 4a^2b^2) = 0$$

$$\Delta = \frac{1}{4k^2} + 4a^2b^2$$

Les racines de l'équation caractéristique sont données par :

$$C_1^2 = (a^2 + b^2 + \frac{1}{2k}) - \sqrt{\frac{1}{4k^2} + 4a^2b^2}, \quad (3.31)$$

$$C_2^2 = (a^2 + b^2 + \frac{1}{2k}) + \sqrt{\frac{1}{4k^2} + 4a^2b^2}.$$

D'où

$$p_1(t) = \lambda e^{C_1 t} + \beta e^{-C_1 t} + \mu e^{C_2 t} + \alpha e^{-C_2 t} + \nu. \quad (3.32)$$

Détermination de ν , on a :

$$\begin{aligned} \dot{p}_1(t) &= \lambda C_1 e^{C_1 t} - \beta C_1 e^{-C_1 t} + \mu C_2 e^{C_2 t} - \alpha C_2 e^{-C_2 t}, \\ \ddot{p}_1(t) &= \lambda C_1^2 e^{C_1 t} + \beta C_1^2 e^{-C_1 t} + \mu C_2^2 e^{C_2 t} + \alpha C_2^2 e^{-C_2 t}, \\ p_1^{(3)}(t) &= \lambda C_1^3 e^{C_1 t} - \beta C_1^3 e^{-C_1 t} + \mu C_2^3 e^{C_2 t} - \alpha C_2^3 e^{-C_2 t}, \\ p_1^{(4)}(t) &= \lambda C_1^4 e^{C_1 t} + \beta C_1^4 e^{-C_1 t} + \mu C_2^4 e^{C_2 t} + \alpha C_2^4 e^{-C_2 t}. \end{aligned}$$

On remplace dans (3.30), et on obtient :

$$\begin{aligned} &\lambda [C_1^4 - ((a^2 + b^2)^2 + \frac{1}{k}(a^2 + b^2) - 4a^2b^2) - 2(a^2 + b^2 + \frac{1}{2k})C_1^2] e^{C_1 t} \\ &+ \beta [C_1^4 - ((a^2 + b^2)^2 + \frac{1}{k}(a^2 + b^2) - 4a^2b^2) - 2(a^2 + b^2 + \frac{1}{2k})C_1^2] e^{-C_1 t} \\ &+ \mu [C_2^4 - ((a^2 + b^2)^2 + \frac{1}{k}(a^2 + b^2) - 4a^2b^2) - 2(a^2 + b^2 + \frac{1}{2k})C_2^2] e^{C_2 t} \\ &+ \alpha [C_2^4 - ((a^2 + b^2)^2 + \frac{1}{k}(a^2 + b^2) - 4a^2b^2) - 2(a^2 + b^2 + \frac{1}{2k})C_2^2] e^{-C_2 t} \\ &+ ((a^2 + b^2)^2 + \frac{1}{k}(a^2 + b^2) - 4a^2b^2)\nu = (-b^3 - \frac{b}{k} + a^2b)x_{1d} + (a^3 - ab^2 + \frac{a}{k})x_{2d} \end{aligned}$$

Par identification, on obtient :

$$\nu = \frac{(-b^3 - \frac{b}{k} + a^2b)x_{1d} + (a^3 - ab^2 + \frac{a}{k})x_{2d}}{((a^2 + b^2)^2 + \frac{a^2+b^2}{k} - 4a^2b^2)} \quad (3.33)$$

On en déduit aisement $p_2(t)$:

De (3.29), on obtient :

$$p_2(t) = \frac{1}{2ab} [(a^2 + b^2)p_1 - \ddot{p}_1 + bx_{1d} - ax_{2d}2ab],$$

$$p_2(t) = \frac{a^2 + b^2}{2ab} [\lambda e^{C_1t} + \beta e^{-C_1t} + \mu e^{C_2t} + \alpha e^{-C_2t} + \nu] - \frac{1}{2ab} [\lambda C_1^2 e^{C_1t} + \beta C_1^2 e^{-C_1t} + \mu C_2^2 e^{C_2t} + \alpha C_2^2 e^{-C_2t}] + \frac{1}{2a} x_{1d} - \frac{1}{2b} x_{2d},$$

D'où

$$p_2(t) = \lambda \left[\frac{a^2 + b^2 - C_1^2}{2ab} \right] e^{C_1t} + \beta \left[\frac{a^2 + b^2 - C_1^2}{2ab} \right] e^{-C_1t} + \mu \left[\frac{a^2 + b^2 - C_2^2}{2ab} \right] e^{C_2t} + \alpha \left[\frac{a^2 + b^2 - C_2^2}{2ab} \right] e^{-C_2t} + \frac{a^2 + b^2}{2ab} \nu + \frac{1}{2a} x_{1d} - \frac{1}{2b} x_{2d}. \quad (3.34)$$

De même compte tenu des équations :

$$\begin{aligned} x_1 &= bp_1 - ap_2 - \dot{p}_1 + x_{1d}, \\ x_2 &= -ap_1 + bp_2 - \dot{p}_2 + x_{2d}. \end{aligned}$$

On obtient les résultats suivants :

$$\begin{aligned} x_1(t) &= b[\lambda e^{C_1t} + \beta e^{-C_1t} + \mu e^{C_2t} + \alpha e^{-C_2t} + \nu] + a[\lambda \left(\frac{a^2 + b^2 - C_1^2}{2ab} \right) e^{-C_1t} \\ &+ \mu \left(\frac{a^2 + b^2 - C_2^2}{2ab} \right) e^{C_2t} + \alpha \left(\frac{a^2 + b^2 - C_2^2}{2ab} \right) e^{-C_2t} + \frac{a^2 + b^2}{2ab} \nu + \frac{1}{2a} x_{1d} - \frac{1}{2b} x_{2d}] \\ &- [\lambda C_1 e^{C_1t} - \beta C_1 e^{-C_1t} + \mu C_2 e^{C_2t} - \alpha C_2 e^{-C_2t}] + x_{1d}. \end{aligned}$$

D'où

$$\begin{aligned} x_1(t) &= \lambda \left[\frac{b^2 - a^2 + C_1^2}{2b} - C_1 \right] e^{C_1t} + \beta \left[\frac{b^2 - a^2 + C_1^2}{2b} + C_1 \right] e^{-C_1t} + \mu \left[\frac{b^2 - a^2 + C_2^2}{2b} - C_2 \right] e^{C_2t} \\ &+ \alpha \left[\frac{b^2 - a^2 + C_2^2}{2b} + C_2 \right] e^{-C_2t} + \frac{b^2 - a^2}{2ab} \nu + \frac{1}{2} x_{1d} + \frac{a}{2b} x_{2d}. \quad (3.35) \end{aligned}$$

$$\begin{aligned} x_2(t) &= -a[\lambda e^{C_1t} + \beta e^{-C_1t} + \mu e^{C_2t} + \alpha e^{-C_2t} + \nu] + b[\lambda \left(\frac{a^2 + b^2 - C_1^2}{2ab} \right) e^{C_1t} + \beta \left(\frac{a^2 + b^2 - C_1^2}{2ab} \right) e^{-C_1t} \\ &+ \mu \left(\frac{a^2 + b^2 - C_2^2}{2ab} \right) e^{C_2t} + \alpha \left(\frac{a^2 + b^2 - C_2^2}{2ab} \right) e^{-C_2t} + \frac{a^2 + b^2}{2ab} \nu + \frac{1}{2a} x_{1d} - \frac{1}{2d}] \\ &- [\lambda C_1 \left(\frac{a^2 + b^2 - C_1^2}{2ab} \right) e^{C_1t} - \beta C_2 \left(\frac{a^2 + b^2 - C_1^2}{2ab} \right) e^{-C_1t} + \mu C_2 \left(\frac{a^2 + b^2 - C_1^2}{2ab} \right) e^{C_2t} \\ &- \alpha C_2 \left(\frac{a^2 + b^2 - C_1^2}{2ab} \right) e^{-C_2t}] + x_{2d} \end{aligned}$$

D'où

$$\begin{aligned}
x_2(t) = & \lambda \left[\frac{b^2 - a^2 - C_1^2}{2a} - C_1 \left(\frac{b^2 + a^2 - C_1^2}{2ab} \right) \right] e^{C_1 t} + \beta \left[\frac{b^2 - a^2 - C_1^2}{2a} + C_1 \left(\frac{b^2 + a^2 - C_1^2}{2ab} \right) \right] e^{-C_1 t} \\
& + \mu \left[\frac{b^2 - a^2 - C_2^2}{2a} - C_2 \left(\frac{b^2 + a^2 - C_2^2}{2ab} \right) \right] e^{C_2 t} + \alpha \left[\frac{b^2 - a^2 - C_2^2}{2a} + C_2 \left(\frac{b^2 + a^2 - C_2^2}{2ab} \right) \right] e^{-C_2 t} \\
& + \frac{a^2 - b^2}{2a} \nu + \frac{b}{2a} x_{1d} + \frac{1}{2} x_{2d}. \tag{3.36}
\end{aligned}$$

Les constantes étant déterminées par les conditions aux limites suivantes :

$$\begin{aligned}
x_1(0) = 1, \quad x_2(0) = 1, \\
p_1(0) = \kappa_0, \quad x_1(t_f) = 2, \\
p_1(t_f) = -\kappa_1, \quad p_2(t_f) = 0,
\end{aligned}$$

On résoud le système linéaire numériquement :

$$\left\{ \begin{aligned}
& \kappa_0 = \lambda + \beta + \mu + \alpha + \nu, \\
& -\kappa_1 = \lambda e^{C_1 t_f} + \beta e^{-C_1 t_f} + \mu e^{C_2 t_f} + \alpha e^{-C_2 t_f} + \nu, \\
& 0 = \lambda \left(\frac{a^2 + b^2 - C_1^2}{2ab} \right) e^{C_1 t_f} + \beta \left(\frac{a^2 + b^2 - C_1^2}{2ab} \right) e^{-C_1 t_f} + \mu \left(\frac{a^2 + b^2 - C_2^2}{2ab} \right) e^{C_2 t_f} + \alpha \left(\frac{a^2 + b^2 - C_2^2}{2ab} \right) e^{-C_2 t_f} \\
& + \left(\frac{b^2 + a^2}{2ab} \right) \nu + \frac{1}{2a} x_{1d} - \frac{1}{2b} x_{2d}, \\
& 1 = \lambda \left(\frac{b^2 - a^2 + C_1^2}{2b} - C_1 \right) + \beta \left(\frac{b^2 - a^2 + C_1^2}{2b} + C_1 \right) + \mu \left(\frac{b^2 - a^2 + C_2^2}{2b} - C_2 \right) + \alpha \left(\frac{b^2 - a^2 + C_2^2}{2b} + C_2 \right) + \frac{b^2 - a^2}{2b} \nu \\
& + \frac{1}{2} x_{1d} + \frac{a}{2b} x_{2d}, \\
& 1 = \lambda \left[\frac{b^2 - a^2 - C_1^2}{2a} - C_1 \left(\frac{b^2 + a^2 - C_1^2}{2ab} \right) \right] + \beta \left[\frac{b^2 - a^2 - C_1^2}{2a} + C_1 \left(\frac{b^2 + a^2 - C_1^2}{2ab} \right) \right] + \mu \left[\frac{b^2 - a^2 - C_2^2}{2a} - C_2 \left(\frac{b^2 + a^2 - C_2^2}{2ab} \right) \right] \\
& + \alpha \left[\frac{b^2 - a^2 - C_2^2}{2a} + C_2 \left(\frac{b^2 + a^2 - C_2^2}{2ab} \right) \right] + \left(\frac{a^2 - b^2}{2a} \right) \nu + \frac{b}{2a} x_{1d} + \frac{1}{2} x_{2d}, \\
& 2 = \lambda \left(\frac{b^2 - a^2 + C_1^2}{2b} - C_1 \right) e^{C_1 t_f} + \beta \left(\frac{b^2 - a^2 + C_1^2}{2b} + C_1 \right) e^{-C_1 t_f} + \mu \left(\frac{b^2 - a^2 + C_2^2}{2b} - C_2 \right) e^{C_2 t_f} \\
& + \alpha \left(\frac{b^2 - a^2 + C_2^2}{2b} + C_2 \right) e^{-C_2 t_f} + \frac{b^2 - a^2}{2b} \nu + \frac{1}{2} x_{1d} + \frac{a}{2b} x_{2d},
\end{aligned} \right.$$

Ceci est équivalent au système :

$$\left\{ \begin{array}{l} \lambda + \beta + \mu + \alpha - \kappa_0 = -\nu, \\ \lambda e^{C_1 t_f} + \beta e^{-C_1 t_f} + \mu e^{C_2 t_f} + \alpha e^{-C_2 t_f} + \kappa_1 = -\nu, \\ \lambda \left(\frac{a^2 + b^2 - C_1^2}{2ab} \right) e^{C_1 t_f} + \beta \left(\frac{a^2 + b^2 - C_1^2}{2ab} \right) e^{-C_1 t_f} + \mu \left(\frac{b^2 + a^2 - C_2^2}{2b} \right) e^{C_2 t_f} + \alpha \left(\frac{a^2 + b^2 - C_2^2}{2ab} \right) e^{-C_2 t_f} = - \left(\frac{a^2 + b^2}{2ab} \right) \nu \\ - \frac{1}{2a} x_{1d} + \frac{1}{2b} x_{2d}, \\ \lambda \left(\frac{b^2 - a^2 + C_1^2}{2b} - C_1 \right) + \beta \left(\frac{b^2 - a^2 + C_1^2}{2b} + C_1 \right) + \mu \left(\frac{b^2 - a^2 + C_2^2}{2b} - C_2 \right) + \alpha \left(\frac{b^2 - a^2 + C_2^2}{2b} + C_2 \right) = 1 - \frac{b^2 - a^2}{2b} \nu \\ - \frac{1}{2} x_{1d} - \frac{a}{2b} x_{2d}, \\ \lambda \left[\frac{b^2 - a^2 - C_1^2}{2a} - C_1 \left(\frac{b^2 + a^2 - C_1^2}{2ab} \right) \right] + \beta \left[\frac{b^2 - a^2 - C_1^2}{2a} + C_1 \left(\frac{b^2 + a^2 - C_1^2}{2ab} \right) \right] + \mu \left[\frac{b^2 - a^2 - C_2^2}{2a} - C_2 \left(\frac{b^2 + a^2 - C_2^2}{2ab} \right) \right] \\ + \alpha \left[\frac{b^2 - a^2 - C_2^2}{2a} + C_2 \left(\frac{b^2 + a^2 - C_2^2}{2ab} \right) \right] = 1 - \left(\frac{a^2 - b^2}{2a} \right) \nu - \frac{b}{2a} x_{1d} - \frac{1}{2} x_{2d}, \\ \lambda \left(\frac{b^2 - a^2 + C_1^2}{2b} - C_1 \right) e^{C_1 t_f} + \beta \left(\frac{b^2 - a^2 + C_1^2}{2b} + C_1 \right) e^{-C_1 t_f} + \mu \left(\frac{b^2 - a^2 + C_2^2}{2b} - C_2 \right) e^{C_2 t_f} \\ + \alpha \left(\frac{b^2 - a^2 + C_2^2}{2b} + C_2 \right) e^{-C_2 t_f} = 2 - \frac{b^2 - a^2}{2b} \nu - \frac{1}{2} x_{1d} - \frac{a}{2b} x_{2d}, \end{array} \right.$$

Soit encore sous la forme matricielle :

$$\begin{pmatrix} 1 & 1 & 1 & 1 & -1 & 0 \\ e^{C_1 t_f} & e^{-C_1 t_f} & e^{C_2 t_f} & e^{-C_2 t_f} & 0 & 1 \\ a_{31} & a_{32} & a_{33} & a_{34} & 0 & 0 \\ a_{41} & a_{42} & a_{43} & a_{44} & 0 & 0 \\ a_{51} & a_{52} & a_{53} & a_{54} & 0 & 0 \\ a_{61} & a_{62} & a_{63} & a_{64} & 0 & 0 \end{pmatrix} \times \begin{pmatrix} \lambda \\ \beta \\ \mu \\ \alpha \\ \kappa_0 \\ \kappa_1 \end{pmatrix} = \begin{pmatrix} -\nu \\ -\nu \\ -(\frac{a^2+b^2}{2ab})\nu - \frac{1}{2a}x_{1d} + \frac{1}{2b}x_{2d} \\ 1 - \frac{b^2-a^2}{2b}\nu - \frac{1}{2}x_{1d} - \frac{a}{2b}x_{2d} \\ 1 - (\frac{a^2-b^2}{2a})\nu - \frac{b}{2a}x_{1d} - \frac{1}{2}x_{2d} \\ 2 - \frac{b^2-a^2}{2b}\nu - \frac{1}{2}x_{1d} - \frac{a}{2b}x_{2d} \end{pmatrix}$$

avec :

$$a_{31} = \left(\frac{a^2 + b^2 - C_1^2}{2ab}\right)e^{C_1 t_f}, \quad a_{32} = \left(\frac{a^2 + b^2 - C_1^2}{2ab}\right)e^{-C_1 t_f}$$

$$a_{33} = \left(\frac{a^2 + b^2 - C_2^2}{2ab}\right)e^{C_2 t_f}, \quad a_{34} = \left(\frac{a^2 + b^2 - C_2^2}{2ab}\right)e^{-C_2 t_f}$$

$$a_{41} = \frac{b^2 - a^2 + C_1^2}{2b} - C_1, \quad a_{42} = \frac{b^2 - a^2 + C_1^2}{2b} + C_1$$

$$a_{43} = \frac{b^2 - a^2 + C_2^2}{2b} - C_2, \quad a_{44} = \frac{b^2 - a^2 + C_2^2}{2b} + C_2$$

$$a_{51} = \frac{b^2 - a^2 + C_1^2}{2a} - C_1 \left(\frac{b^2 + a^2 - C_1^2}{2ab}\right),$$

$$a_{52} = \frac{b^2 - a^2 - C_1^2}{2a} + C_1 \left(\frac{b^2 + a^2 - C_1^2}{2ab}\right)$$

$$a_{53} = \frac{b^2 - a^2 - C_2^2}{2a} - C_2 \left(\frac{b^2 + a^2 - C_2^2}{2ab}\right)$$

$$a_{54} = \frac{b^2 - a^2 - C_2^2}{2a} + C_2 \left(\frac{b^2 + a^2 - C_2^2}{2ab}\right)$$

$$a_{61} = \left(\frac{b^2 - a^2 + C_1^2}{2b} - C_1\right)e^{C_1 t_f}$$

$$a_{62} = \left(\frac{b^2 - a^2 + C_1^2}{2b} + C_1\right)e^{-C_1 t_f}$$

$$a_{63} = \left(\frac{b^2 - a^2 + C_2^2}{2b} - C_2\right)e^{-C_2 t_f}$$

$$a_{64} = \left(\frac{b^2 - a^2 + C_2^2}{2b} + C_2\right)e^{-C_2 t_f}$$

3.3.3 Résultats numériques

Pour ce dernier exemple, nous avons simulé la méthode de relaxation qui utilise la méthode numérique de Runge Kutta pour l'intégration du système différentiel. Nous avons déterminé les multiplicateurs de Lagrange κ_0 et κ_1 en utilisant la méthode de tir.

Les résultats obtenus pour $b = 3$, $a = 1$, $k = 2$ grace à Matlab sont donnés ci-dessous :

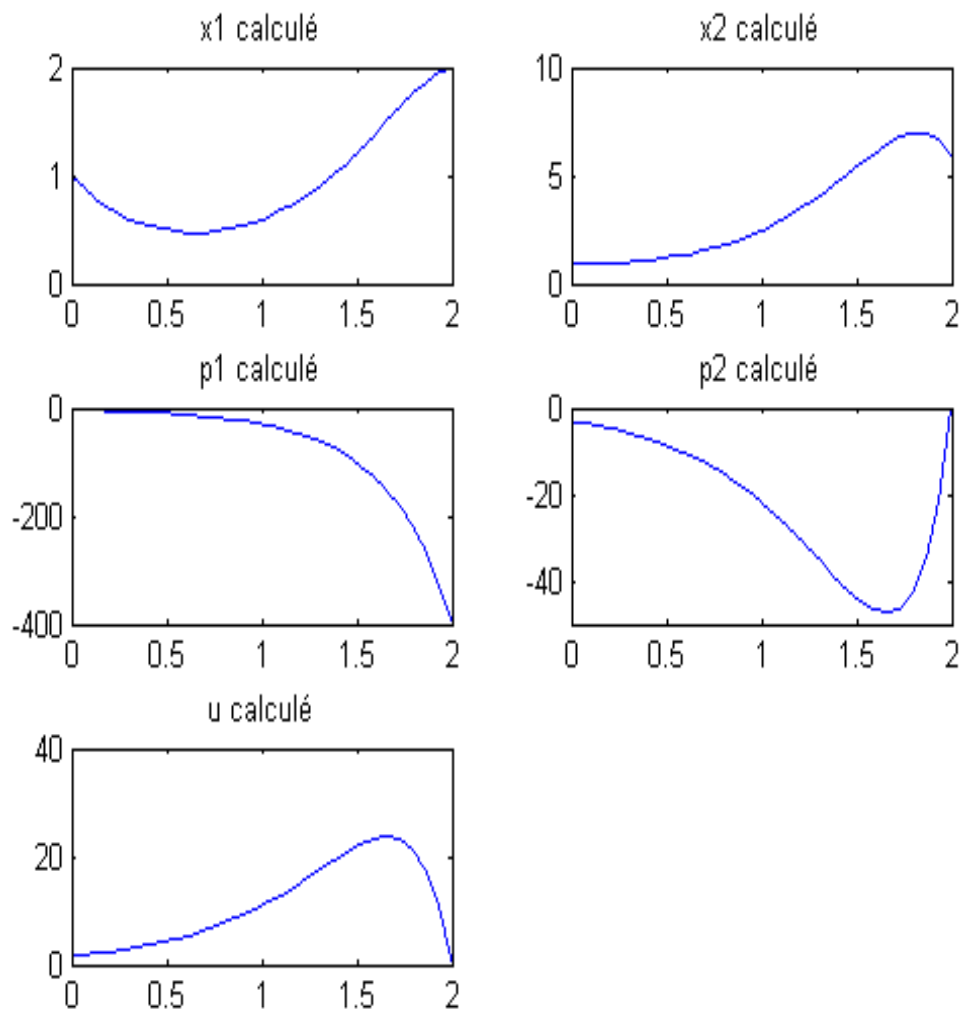


FIG. 3.1 – Solution calculé

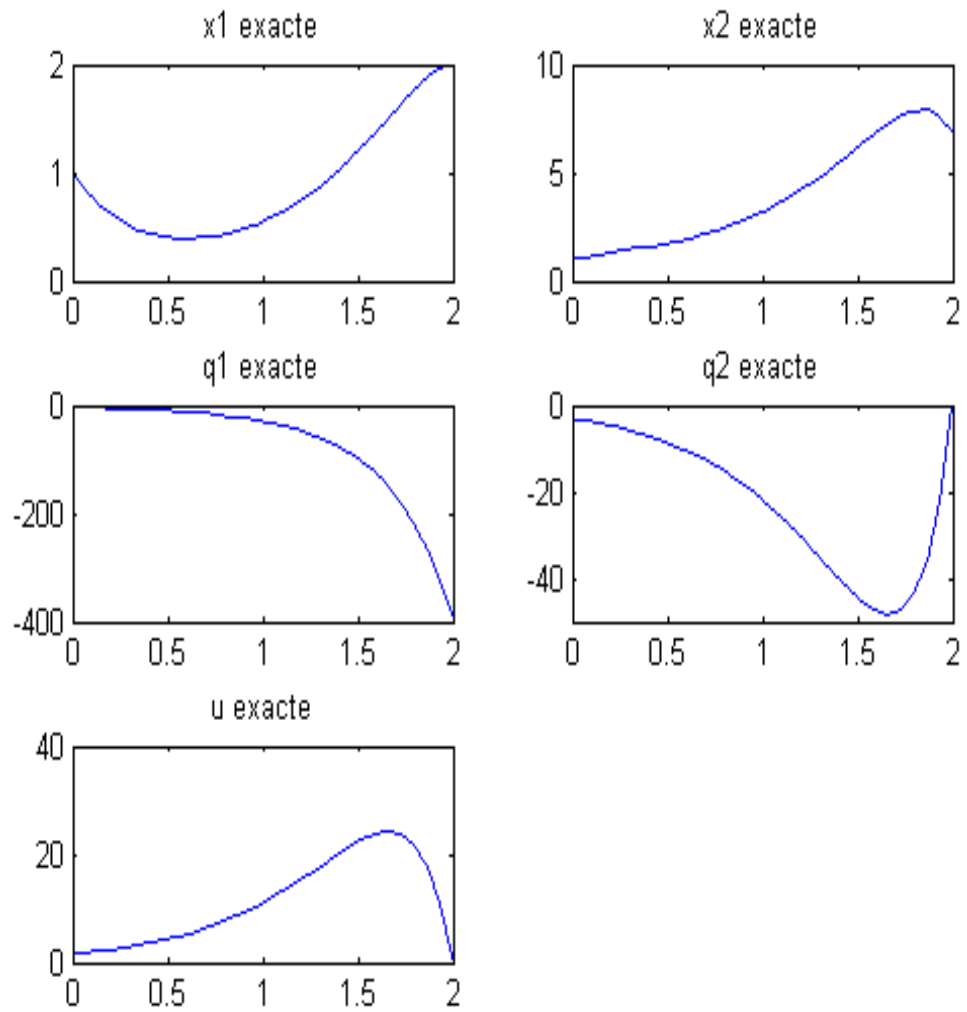


FIG. 3.2 – Solution exacte

Le tableau ci-dessous présente le nombre d'itérations ainsi que les temps de convergence de notre programme en faisant varier le paramètre k :

k	$kakon$	temps-CPU
0.5	12	1.1094
1	9	0.5625
1.5	8	0.5625
2	7	0.5469
2.5	7	0.5938
3	7	0.6250
3.5	7	0.6250
4	6	0.6719
4.5	6	0.6406
5	6	0.5156
5.5	6	0.5156
6	6	0.5469
6.5	6	0.5938
7	6	0.5781
7.5	6	0.5781
8	6	0.6563
8.5	6	0.6875
9	6	0.7031
9.5	6	0.6791
10	6	0.6875

Où $kakon$ est le nombre d'itérations et temps-CPU est le temps d'exécution.

3.4 Conclusion

On a résolu un problème de contrôle optimale à entrée libre par la méthode de relaxation, on a utilisé la résolution numérique et la résolution analytique, on a constaté que les résultats obtenus sont les mêmes. Plus encore, nous avons trouvé la solution pour différents k , on a pris en considération le nombre d'itérations et le temps d'exécution. Nous avons conclu que, l'algorithme converge vite quand k augmente, c'est ce paramètre qui règle la vitesse de convergence.

Si par exemple les conditions suivantes sont vérifiées :

$$b > |a| + 1$$

alors la matrice A est une M -matrice et dans ces conditions on est assuré de la convergence des algorithmes de relaxation appliqués à la recherche de la solution du système d'équations 3.19.

Notons que : Une matrice A est une M -matrice si toutes les valeurs propres de la matrice A sont positives.

Conclusion générale et perspectives

Dans ce mémoire, nous avons donné quelques méthodes numériques d'intégration des équations différentielles, on a choisi parmi ces méthodes la méthode de Runge Kutta pour simuler la méthode de relaxation.

Le problème de contrôle optimal à entrée libre, peut être interpréter comme un problème pour lequel on cherche un meilleur départ appartenant à un ensemble prédéfini X_0 et une commande u qui nous permet de ramener le système de $x(0)$ vers $x(t_f)$.

Ensuite, nous nous sommes intéressé à la résolution de ce problème par la méthode de relaxation numériquement et analytiquement. La solution exacte que nous avons obtenue est identique à la solution numérique.

La méthode de relaxation est une méthode qui converge vite quand k est grand. Pour conclure, il apparait que l'étude et l'application des méthodes de relaxation présentent au moins deux avantages important. D'une part, les calculs des lois de commande optimale peuvent être effectué rapidement en temps réel, notamment par l'utilisation de calculateurs parallèles, d'autre part, les critères de convergences présentés précédemment fournissent des critères d'arrêt des itérations, d'autant plus sûrs que la convergence est plus lente.

Il existe dans la littérature de nombreux travaux concernant la méthode de relaxation. Compte tenu des développements actuels des moyens de calculs, il est utile de s'intéresser aux possibilités de calcul parallèle synchrone et asynchrone pour la résolution de systèmes différentielles avec conditions aux deux bouts.

A ce sujet, nous trouvons dans [16], une variante asynchrone de la méthode de l'état adjoint par relaxation, dans laquelle des processus parallèles échangent entre eux des informations en cours de calculs, dès que les résultats sont disponibles, par ce biais, chaque processeur effectue de nouvelles itérations avec les données les plus fraîches concernant les fonctions d'interactions.

Nous y trouvons également, une analyse du comportement de ces algorithmes qui est basée sur une adaptation de la notion de H-accrétivité, une notion complexe qui est défini dans [16], et qui permet d'obtenir une propriété de contraction en

norme scalaire.

Cette notion peut par ailleurs, être utilisée pour analyser la convergence d'algorithmes de relaxation séquentiels et parallèles (synchrone et asynchrones), liés à la résolution de certains problèmes aux limites.

Enfin, nous y trouvons des applications concernant la régulation de processus thermiques, pour lesquelles sont données des expérimentations basées sur des simulations d'exécutions parallèles des algorithmes de calcul de la loi de commande.

Une autre application des méthodes de relaxation concerne la résolution de problèmes multicritères [11], l'analyse de la convergence y est étudié en utilisant la notion de H-accrétivité, certains résultats de convergence d'algorithmes de relaxation appliqués à la résolution de problèmes de commande mono-critère sont étendus au cas de problèmes multicritères dans le cadre de la recherche d'un équilibre de Nash ainsi que la recherche d'une solution pareto-optimale.

Les critères de convergences obtenus sont ensuite appliqués au cas de systèmes linéaires avec contraintes sur la commande et critères quadratiques.

On peut également mentionner une application à la résolution de problèmes de commande optimale dans le cas de systèmes singulièrement perturbés [11], il n'est pas rare d'être en présence d'un processus possédant des dynamiques différentes et notamment d'un processus à deux échelles de temps. Une façon de faire apparaître ces deux dynamiques au niveau du modèle est d'introduire un petit paramètre, on obtient alors un système d'équations singulièrement perturbées.

Les modèles singulièrement perturbés considérés dans [12], sont les cas où un retour d'état optimal ne peut être effectué simplement, notamment lorsque les commandes sont contraintes. Après avoir étendu certains résultats de convergence d'algorithmes de relaxation à cette classe de problèmes, une méthodologie de calcul en ligne de la loi de commande optimale y est proposée.

En ce qui concerne les axes de recherche à suivre après ce mémoire, on peut citer entre autre :

1. La généralisation de cette méthode à une matrice A d'ordre plus grand et la détermination des critères de convergence.
2. L'application de cette méthode pour un problème de contrôle optimale à entrée libre avec une commande polyédrale ou une commande vectorielle.
3. L'application de cette méthode pour un problème de contrôle optimale avec une commande polyédrale ou une commande vectorielle.

Bibliographie

- [1] Aidene Mohamed, Louadj Kahina *Un problème de contrôle optimale avec une entrée libre*. Exposé SMAI'2011.
- [2] G.M. Baudet. *Asynchronous iterative methods for multiprocesseurs*. Journal of ACM, 25, pp 226-244. 1978.
- [3] D.P. Bertsekas and J.Tsitsiklis *Parallel and Distributed computation*. Numerical Methods, Englewood cliffs : Prentice Hall. 1989.
- [4] P.Borne, G.Dauphin-Tanguy, J.P Richard, F.Rotella, I.Zambettakis. *Commande et optimisation des processus*. Editions Technip, 1990.
- [5] D.Chazan and Miranker. *Chaoric Relaxation*. Linear Algebra Appl, 2, pp 199-222. 1969.
- [6] M.Crouzeix, A.L Mignot. *Analyse numérique des équations différentielles*. 2ème Edition Masson. 1989.
- [7] Jean-Pierre Denailly. *Analyse numérique et équations différentielles*. EDP Sciences. 2006.
- [8] Richard Epenoy. *Contrôle optimale - Application à la mécanique spatiale*.
- [9] B.Lang, J.C Miellou and P.Spiteri. *Asynchronous relaxation algorithms for optimal control*. Mathematics ans computers in simulation, Elsevier Science Publishers B.V (North Halland). 1986.
- [10] B.Lang, P.Spiteri. *Algorithmes de relaxation appliqués à la résolution de problèmes de commande optimale : Etude d'un système en anneau*. Commande optimale de systèmes complexes. 1983.
- [11] B.Lang, P.Spiteri. *Application des méthodes de relaxation à la résolution de problèmes multicritères*. Commandes optimales de systèmes complexes. 1983.
- [12] B.Lang, P.Spiteri. *Application des méthodes de relaxation à la résolution de problèmes de commande optimale dans le cas des systèmes singulièrement perturbés*. Commande optimale de systèmes complexes. 1983.
- [13] Louadj Kahina, Aidene Mohamed, Spiteri Pierre, Messine Frederic. *Résolution d'un problème de contrôle optimale avec la méthode de relaxation*. ISOR'11. 2011.
- [14] Pierre Martinon. *Résolution numérique de problèmes de contrôle optimal par une méthode homotopique simpliciale* . Thèse, Chapitre 1 "Méthode de tir et homotopie". 2005.
- [15] J.C Miellou and P.Spiteri. *A parallel asynchronous relaxation algorithm for optimal control problems*. Mathematical analysis ans its applications, Pergamon Press. 1985.

- [16] J.C Miellou, P.Spiteri. *La notion de H-accrétivité : application à la résolution de problèmes aux deux bouts*. Commande Optimale de systèmes complexes. 1982.
- [17] F.Robert, M.Charnay et F.Musy. *Itérations chaotiques série-parrallèle pour des équations non linéaires de point fixe*. Aplikace Mathematik, 20, pp 1-38. 1975.
- [18] F.Robert. Convergence locale d'itération chaotique non linéaire. C.R.A.S série A, 284, p 679-682. 1977.
- [19] P.Spiteri ans B.Lang. *Optimal control decomposition and coordination using asynchronous iteration*. Systems and control encyclopedia, Pergamon Press.
- [20] Emanuel Trélat. *Contrôle optimal : théorie et applications*. Vuibert, Collection Mathématiques concrètes. 2005.