

N° d'ordre : 2276

Année 2005

THESE

présentée pour obtenir le titre de

DOCTEUR DE L'INSTITUT NATIONAL POLYTECHNIQUE DE TOULOUSE

Ecole Doctorale : Informatique et Télécommunications
Spécialité : Programmation et Systèmes

par

Pierre Martinon

Résolution numérique de problèmes de contrôle optimal par une méthode homotopique simpliciale

Soutenue publiquement le 4 Novembre 2005 devant le jury composé
de :

M. E. ALLGOWER	<i>Rapporteur</i>
M. E. HAIRER	<i>Rapporteur</i>
M. J. NOAILLES	<i>Directeur de thèse</i>
M. J. GERGAUD	<i>Co-directeur de thèse</i>
M. R. EPENY	<i>Examineur</i>
M. P. LEGENDRE	<i>Examineur</i>

“- Comment puis-je m’empêcher de voir ce qui est devant mes yeux ? Deux et deux font quatre.

- Parfois, Winston. Parfois ils font cinq. Parfois ils font trois. Parfois ils font tout à la fois. Il faut essayer plus fort. Il n’est pas facile de devenir sensé.”

(G. Orwell, 1984)

“J’admets que ”deux fois deux : quatre” est une chose excellente, mais s’il faut tout louer, je vous dirai que ”deux fois deux : cinq” est aussi parfois une petite chose bien charmante.”

(F.M. Dostoïevski, Le sous-sol)

“ $2+2=5$... for extremely large values of 2.”

(inconnu)

Collaborations & sigles

Ce travail de thèse a été effectué au sein de l'équipe APO¹ du LIMA², situé à l'ENSEEIH³ et rattaché à l'IRIT⁴. Il s'est déroulé dans le cadre d'une bourse BDI⁵ accordée par le CNRS⁶, et dans la suite d'une longue et fertile collaboration avec le CNES⁷.

¹Algorithmes Parallèles et Optimisation

²Laboratoire d'Informatique et de Mathématiques Appliquées

³Ecole Nationale Supérieure d'Electronique, d'Electrotechnique, d'Informatique, d'Hydraulique et de Télécommunications de Toulouse

⁴Institut de Recherche en Informatique de Toulouse, UMR CNRS 5505

⁵Bourse de Docteur Ingénieur

⁶Centre National de la Recherche Scientifique

⁷Centre National d'Etudes Spatiales

Remerciements

Je tiens tout d'abord à remercier mon directeur de thèse, le professeur Joseph Noailles, pour m'avoir accordé sa confiance et permis d'effectuer cette thèse au sein de l'équipe APO. C'est à lui que je dois mon entrée dans le monde de la recherche, ce dont je lui suis très reconnaissant.

Vient ensuite Joseph Gergaud, qui m'a accompagné et conseillé tout au long de ce travail de thèse et m'a, véritablement, donné le goût de la recherche, du problème à sa résolution. De nombreuses parties du code SIMPLICIAL n'auraient pu voir le jour sans ses suggestions et remarques.

Pour poursuivre avec l'équipe APO, je ne peux manquer de remercier Jean-Baptiste Caillau, pour ses conseils en toute circonstance, et qui a bien souvent éclipsé mes retards chroniques aux réunions.

Je tiens également à remercier Richard Epenoy et Paul Legendre du CNES, qui ont accepté d'être examinateurs de ma thèse, et à qui nous devons cette famille de problèmes de transferts orbitaux, dont l'étude s'est révélée si riche au fil des années et des thèses.

Je garde bien sûr une pensée pour mes camarades du thèse : Dorin Preda, expert en systèmes linux, Thomas Haberkorn, qui a quitté Toulouse et ses satellites pour Hawaii et ses sous-marins, Ming Chau, qui soutint sa thèse la veille de la mienne, Romain Dujol, dont la connaissance de L^AT_EX/BEAMER n'en finit jamais de m'émerveiller, et Ahmed Touhami, pour la relecture de cette page. Je pense également aux occupants du bureau voisin, qui nous éclairèrent de leurs lumières (et au sens littéral!) : Frédéric Messine, dont le duo avec Ming me redonna envie d'apprendre la guitare, et le professeur Pierre Spitéri, mon *Parrain*.

Par ailleurs, j'aimerais ici plus généralement remercier tous les membres du LIMA pour l'ambiance sympathique qui y règne, en particulier son directeur Michel Daydé (et sa disponibilité pour les signatures à la dernière minute!), l'équipe AlgoProg dirigée par Patrick Sallé, et Marc Pantel, qui m'a permis de rejoindre le groupe des correcteurs de l'épreuve Informatique des CCP.

Enfin, je suis très reconnaissant aux professeurs Eugene Allgower et Ernst Hairer d'avoir accepté d'être les rapporteurs de cette thèse, et les remercie encore une fois pour leurs conseils éclairés sur les méthodes simpliciales et l'intégration numérique.

Table des matières

1	Méthode de tir et homotopie	3
1.1	Principe et premières applications	3
1.2	Résultats de convergence	16
2	Méthode homotopique simpliciale	23
2.1	Principe des méthodes simpliciales	23
2.2	Homotopie de jonction	31
2.3	Triangulation adaptative	36
2.4	Raffinage de solution	50
3	Problèmes de Transfert Orbital	55
3.1	Présentation du problème	56
3.2	Approche de résolution	59
3.3	Comparaison de différents intégrateurs	64
3.4	Etude de la triangulation adaptative	73
4	Problèmes d’Arcs Singuliers	87
4.1	Présentation des problèmes	88
4.2	Perturbation quadratique	92
4.3	Approche BVP discrétisé	101
4.4	Tir structuré	111
4.5	Formulation Contrôle discrétisé	121
4.6	Approche par méthode directe	126
A	Présentation du package Simplicial	133
A.1	Utiliser le package Simplicial	135
A.2	Structure général du code Simplicial	144
B	Options principales	147
B.1	Classe de problème	147
B.2	Etiquetages	147
B.3	Intégrateurs	148

C Exemples de fichiers de problèmes	151
C.1 Problème de démonstration	151
C.2 Transfert orbital	155
C.3 Problème de pêche	158
C.4 Régulateur quadratique	160

Introduction

Ce travail de thèse porte sur la résolution numérique de problèmes de contrôle optimal de faible régularité. Les méthodes indirectes, basées sur le Principe du Maximum de Pontriaguine, sont réputées pour leur rapidité et leur précision dans le traitement des problèmes de contrôle optimal. Toutefois, leur mise en oeuvre peut en pratique rencontrer certaines difficultés lorsque la structure de contrôle est de type bang-bang (on entend ici par bang-bang un contrôle de norme soit nulle soit maximale), ou présente des arcs singuliers par exemple. En effet, ces méthodes transforment le problème de contrôle originel en la résolution d'un système d'équations non linéaires. Dans les deux cas précités, ce système est peu régulier, et sa résolution numérique est en particulier très sensible au choix du point initial. Nous traitons ici cette difficulté par une démarche homotopique (ou continuation), et notre choix se porte plus précisément sur les méthodes simpliciales en raison de leur robustesse.

Nous commençons le chapitre I par une présentation rapide du principe général des méthodes indirectes, et plus précisément du tir simple, qui consiste à transformer le problème de contrôle optimal en la recherche de zéro de la fonction de tir associée. Nous illustrons cette méthode sur un exemple simple, afin de souligner certaines des difficultés que l'on s'attend à rencontrer dans la cas d'un contrôle de type bang-bang. On introduit alors la méthode de continuation, dont le principe est de s'appuyer sur la résolution d'un problème plus simple que celui de départ. On construit à cet effet une famille de problèmes en paramétrant l'objectif du problème initial, puis on démarre la résolution en partant d'un cas facile (plus formellement, il s'agit de suivre le chemin de zéros de l'homotopie correspondant à cette famille de problèmes). Nous concluons en rappelant certains résultats de convergence concernant la continuation, qui sont également valables dans le cas multi-valué (pour les arcs singuliers par exemple).

Le chapitre II est consacré à la description de la méthode simpliciale, dont le principal avantage est le faible niveau d'hypothèses à vérifier par le chemin de zéros. Nous rappelons d'abord le principe général de la méthode, avant de présenter certains points algorithmiques que nous avons introduit

dans ce cadre d'études, comme la triangulation adaptative ou le raffinage de solution.

On s'intéresse au chapitre III à l'étude d'une famille de problèmes de transfert orbital à poussée faible fournis par le CNES, dont les résultats coïncident avec ceux obtenus par la continuation différentielle (cf [26]). Nous comparons ensuite sur ces problèmes différents intégrateurs à pas variable, ainsi que plusieurs configurations pour le mécanisme de triangulation adaptative.

Dans le chapitre IV nous étudions en parallèle deux problèmes d'arcs singuliers, pour garantir une meilleure généralité des méthodes employées. Pour ces deux problèmes, la structure du contrôle optimal consiste en une succession d'arcs bang-bang et d'arcs singuliers. On cherche dans un premier temps à obtenir suffisamment d'informations sur cette structure, en utilisant des continuations bien choisies. Plus précisément, on souhaite déterminer d'une part le nombre et la localisation des arcs singuliers, et d'autre part la valeur du contrôle sur les arcs bang-bang. On passe ensuite à la résolution précise des problèmes, pour laquelle on utilise des méthodes dérivées du tir multiple, et adaptées au cas des arcs singuliers. Pour finir, nous présentons les résultats numériques préliminaires obtenus avec une nouvelle formulation, qui considère le contrôle discrétisé comme partie de l'inconnue (ce qui est plus dans l'état d'esprit des méthodes directes).

Enfin, après la conclusion et les perspectives futures de ce travail, nous présentons brièvement en annexe le programme *Simplicial*, dans lequel nous avons implémenté les diverses continuations et formulations utilisées pour les expérimentations numériques.

Chapitre 1

Méthode de tir et homotopie

Pour commencer, nous rappelons le principe général de la méthode de tir, qui transforme le problème de contrôle optimal d'origine en la résolution d'une équation non linéaire. L'application pratique de cette méthode est ensuite détaillée sur un exemple simple, pour illustrer certaines des difficultés attendues pour les problèmes étudiés aux chapitres 3 et 4. Ces difficultés nous conduisent à introduire une démarche homotopique (ou *continuation*), qui est décrite plus en détails au chapitre 2. On rappelle également à la fin de ce chapitre certains résultats théoriques, qui ont en particulier guidés les choix algorithmiques faits au chapitre 4 pour l'étude des problèmes avec arcs singuliers.

1.1 Principe et premières applications

1.1.1 Méthode du Tir simple

La méthode du tir simple fait partie des méthodes *indirectes*, et est basée sur le Principe du Maximum de Pontriaguine (PMP) (nous renvoyons les lecteurs intéressés par ces méthodes à [32, 18, 4, 35], pour ne citer que quelques-uns). On rappelle que les méthodes *directes*, quant à elles, impliquent traditionnellement la discrétisation totale ou partielle du problème, et utilisent ensuite diverses approches (SQP et techniques de point intérieur par exemple) pour résoudre le problème résultant. Les méthodes directes sont donc supposées robustes, mais sont en contrepartie relativement peu précises, et peuvent mener à de grandes tailles de problème suivant le pas de discrétisation utilisé. Cela rend ces méthodes moins adaptées à certains cas particuliers, comme les problèmes étudiés au chapitre 3, qui présentent un contrôle de type bang-bang avec un très grand nombre de commutations.

Le tir simple, par contre, consiste à trouver un zéro de la *fonction de tir* associée au problème original. Il n'y a pas ici de discrétisation explicite, même si la méthode requiert tout de même l'intégration du système. Il s'agit

d'une méthode rapide et de haute précision, qui ne requiert pas d'hypothèses sur la structure du contrôle.

Toutefois, un handicap majeur de ces méthodes est leur besoin d'un point initial correct : comme elles consistent typiquement à appliquer un algorithme de quasi-Newton à la fonction de tir, le rayon de convergence peut être très faible, suivant la régularité du problème. Ceci est particulièrement vrai par exemple pour les transferts orbitaux à contrôle bang-bang étudiés au chapitre 3, ainsi que pour les problèmes du chapitre 4 qui présentent des arcs singuliers. C'est pourquoi on introduit ensuite une approche homotopique, détaillée plus tard au chapitre 2.

Conditions nécessaires, PMP et Problème aux deux bouts

Notation/Remarque : pour plus de clarté, le temps t sera fréquemment omis dans les formules, sauf en cas d'ambiguïté.

On considère un problème général de contrôle optimal, sous la forme de Bolza :

$$(P) \begin{cases} \text{Min } g(t_0, x(t_0), t_f, x(t_f)) + \int_{t_0}^{t_f} l(t, x, u) dt & \text{Objectif} \\ \dot{x} = f(t, x, u) & \text{Dynamique} \\ u \in U & \text{Contrôles admissibles} \\ \psi_0(t_0, x(t_0)) = 0 & \text{Conditions initiales} \\ \psi_1(t_f, x(t_f)) = 0 & \text{Conditions finales} \end{cases}$$

On utilise ici et dans toute la suite les notations : x pour l'état, u pour le contrôle, U l'ensemble des contrôles admissibles, et f pour la dynamique de l'état. On introduit l'état adjoint p , de même dimension que l'état x , et on définit le **Hamiltonien** par

$$\mathcal{H}(t, x, p, u) = l(t, x, u) + (p|f(t, x, u)).$$

*Remarque : on suppose donc dans toute la suite que l'on est dans le cas dit **normal**, c'est-à-dire que l'état adjoint p_0 associé à l'objectif intégral l est non nul, et peut ainsi être ramené à 1, en divisant tout l'état adjoint p par p_0 ...*

Le Principe du Maximum de Pontriaguine, présenté à l'origine dans [32], dit alors que, sous les hypothèses :

- $\exists (x, u)$ admissible pour (P) , avec x absolument continu et u mesurable.
- f et l sont continues par rapport à u et C^1 par rapport à t et x .
- g, ψ_0, ψ_1 sont C^1 par rapport à x .

Soit (\bar{x}, \bar{u}) une paire optimale pour (P) , alors

(i) $\exists \bar{p} \neq 0$ absolument continu vérifiant le système Hamiltonien

$$\begin{cases} \dot{\bar{x}} = \frac{\partial \mathcal{H}}{\partial p}(t, \bar{x}, \bar{p}, \bar{u}) \\ \dot{\bar{p}} = -\frac{\partial \mathcal{H}}{\partial x}(t, \bar{x}, \bar{p}, \bar{u}) \end{cases}$$

(ii) \bar{u} est solution de $\text{Min}_{w \in U} \mathcal{H}(t, \bar{x}, \bar{p}, w)$ pp sur $[t_0, t_f]$.

(iii) “Conditions de Transversalité” : $\exists (\bar{\mu}_0, \bar{\mu}_1)$ tels que

$$(TC) \begin{cases} \psi_0(t_0, x(t_0)) = 0 \\ \bar{p}(t_0) = -\frac{\partial \Phi}{\partial x_0}(t_0, \bar{x}(t_0), t_f, \bar{x}(t_f), \bar{\mu}_0, \bar{\mu}_1) \\ \psi_1(t_f, x(t_f)) = 0 \\ \bar{p}(t_f) = \frac{\partial \Phi}{\partial x_f}(t_0, \bar{x}(t_0), t_f, \bar{x}(t_f), \bar{\mu}_0, \bar{\mu}_1) \end{cases}$$

avec

$$\Phi : (t_0, x_0, t_f, x_f, \mu_0, \mu_1) \mapsto g(t_0, x_0, t_f, x_f) + (\psi_0(t_0, x_0)|\mu_0) + (\psi_1(t_f, x_f)|\mu_1)$$

Remarque : ceci explique pourquoi les méthodes indirectes sont parfois appelées méthodes par conditions nécessaires.

On note désormais $y = (x, p)$, et φ la dynamique du couple état-état adjoint donnée par le système Hamiltonien. Alors si Γ désigne l’application multivaluée qui à y associe l’expression du contrôle optimal issue de la minimisation du Hamiltonien, alors résoudre (P) équivaut à résoudre le problème aux deux bouts ([Two Point] Boundary Value Problem) suivant :

$$(BVP) \begin{cases} \dot{y} \in \Phi(y) = \varphi(y, \Gamma(y)) & \text{pp sur } [t_0, t_f] \\ c_0(t_0, y(t_0)) = 0 & \text{Conditions aux limites en } t_0 \\ c_1(t_f, y(t_f)) = 0 & \text{Conditions aux limites en } t_f \end{cases}$$

Note : ces conditions aux limites c_0 et c_1 correspondent aux conditions de transversalité mentionnées ci-dessus, qui contiennent les conditions initiales et finales de (P) , en plus des conditions sur l’état adjoint p .

Problème à valeur initiale et méthode de tir

On suppose ici que l’expression du contrôle optimal donnée par les conditions nécessaires (Γ) est en fait une fonction, notée γ . Il est possible d’intégrer $y = (x, p)$ si l’on fixe la valeur de $y(t_0)$, et l’on obtient alors le problème à valeur initiale (Initial Value Problem) suivant

$$(IVP) \begin{cases} \dot{y} = \Phi(y) = \varphi(y, \gamma(y)) & \text{pp sur } [t_0, t_f] \\ y(t_0) = \beta & \text{Valeur Initiale} \end{cases}$$

On introduit maintenant une application appelée **fonction de tir**, qui à la valeur initiale β associe la valeur des conditions aux limites en t_f pour la solution correspondante de (IVP). En pratique, les conditions initiales ψ_0 du problème (P) fixent déjà une partie de $y(t_0)$, et l'inconnue de la fonction de tir est donc réduite à la partie "manquante", que l'on note z . Une situation fréquente est celle où les conditions initiales déterminent l'état initial $x(t_0)$, z étant alors finalement la valeur de l'état adjoint initial $p(t_0)$. La valeur de la fonction de tir est alors donnée par les conditions en t_f pour la solution \hat{y} de (IVP) avec $y(t_0) = [x_0, z]$

$$(\text{Fonction de tir}) \quad S : z \mapsto c_1(\hat{y}(t_f)).$$

Trouver un zéro de la fonction de tir ("shooting function") S est alors équivalent à la résolution de (BVP), et donne ainsi une solution de (P). La "méthode de tir" consiste donc à résoudre l'équation $S(z) = 0$.

Remarque : on ne prend ici pas en compte le cas des contraintes sur l'état, qui soulèvent des difficultés particulières avec les méthodes indirectes (voir par exemple [12], ou [28], où sont données plusieurs formulations du Principe du Maximum pour les contraintes sur l'état). Toutefois, il s'agit là d'une intéressante classe de problèmes, que nous souhaitons étudier à l'avenir.

Un premier exemple simple : contrôle bang-bang

Nous présentons maintenant un premier exemple simple, destiné à illustrer l'application pratique de la méthode du tir simple. Ce problème de démonstration nous permet par ailleurs de souligner les difficultés liées à la structure bang-bang du contrôle, ce qui est le cas des problèmes étudiés plus tard au chapitre 3.

1.1.2 Position du problème

On considère le problème de contrôle optimal suivant :

$$(P) \left\{ \begin{array}{l} \text{Min } \int_0^2 |u| dt \\ \dot{x}_1 = x_2 \\ \dot{x}_2 = u \\ |u| \leq 1 \\ x(0) = (0, 0) \\ x(2) = (0.5, 0) \end{array} \right.$$

Le Hamiltonien est défini par

$$\mathcal{H} : (t, x, p, u) \mapsto |u| + p_1 x_2 + p_2 u$$

et le contrôle optimal par

$$\begin{cases} u = -\text{sgn}(p_2) & \text{si } |p_2| > 1 \\ u = 0 & \text{si } |p_2| < 1 \\ u = -\alpha p_2 & \text{avec } \alpha \in [0, 1] \text{ sinon.} \end{cases}$$

Ainsi le contrôle est discontinu, de type bang-bang.

Si l'on définit la *fonction de commutation* correspondante ψ par

$$\psi : (t, x, p) \mapsto 1 - |p_2|,$$

alors les commutations de la structure bang-bang sont les zéros de ψ .

Nous avons le problème aux deux bouts :

$$(BVP) \begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = u \\ \dot{p}_1 = 0 \\ \dot{p}_2 = -p_1 \\ x(0) = (0, 0) \\ x(2) = (0.5, 0) \\ p(0) \text{ libre} \\ p(2) \text{ libre} \end{cases}$$

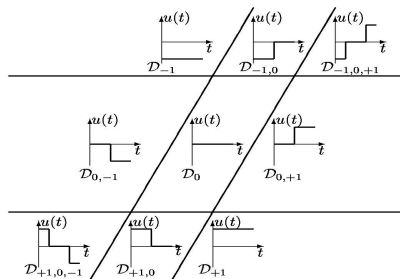
avec la fonction de tir

$$S(z) = \hat{x}(2) - (0.5, 0)$$

où \hat{x} correspond à l'intégration du problème à valeur initiale

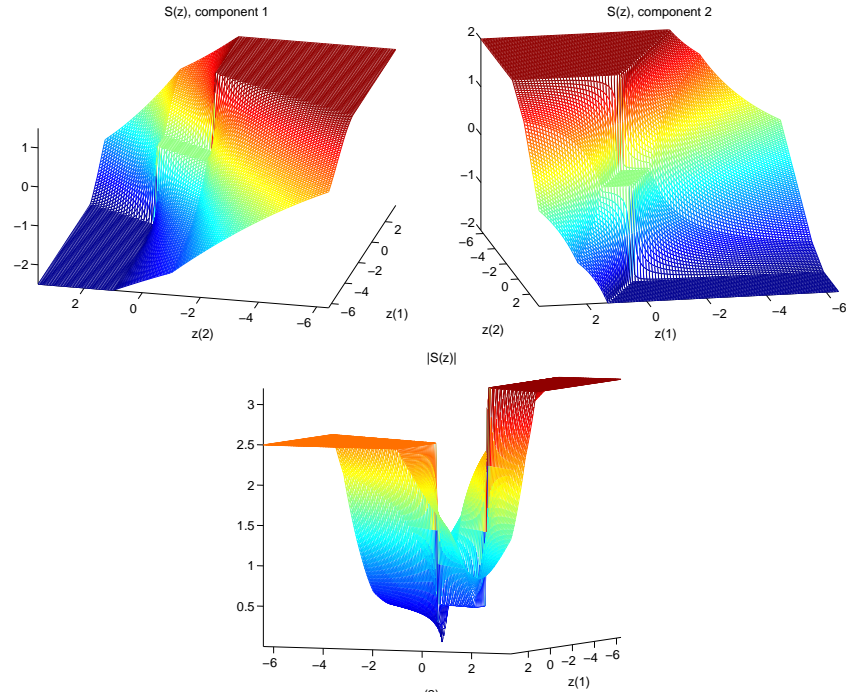
$$(IVP) \begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = u \\ \dot{p}_1 = 0 \\ \dot{p}_2 = -p_1 \\ x(0) = (0, 0) \\ p(0) = z. \end{cases}$$

Si l'on examine les différentes valeurs possibles de l'état adjoint initial $p(0) = z \in \mathbf{R}^2$, on constate que la dynamique de l'état adjoint et l'expression du contrôle optimal mènent à 9 structures de contrôle possibles :



Structures de contrôle correspondant à la valeur de $p(0) \in \mathbf{R}^2$

Si l'on représente la fonction de tir S , on reconnaît ces 9 régions



Fonction de tir pour (P)

Il est à noter que S n'est pas différentiable à la frontière de ces régions, et n'est même pas définie en $(0, -1)$ et $(0, 1)$. En conséquence, des difficultés risquent de survenir lorsqu'on essaie de résoudre $S(z) = 0$ à l'aide d'un algorithme de type Newton (plus précisément, on utilise ici le solveur HYBRD, qui implémente une méthode de Powell hybride), en particulier si le point initial n'est pas dans la bonne région. De fait, même sur cet exemple très simple, résoudre directement $S(z) = 0$ depuis un point de départ quelconque n'est pas trivial. On indique ci-dessous les résultats du tir simple à partir d'un point de chacune des 9 régions (la convergence est notée "CV" (avec entre parenthèses le nombre d'appels de la fonction de tir par le solveur) en cas de succès, et (-) sinon).

$p(0) = (0, 2)$ -	$p(0) = (1, 2)$ -	$p(0) = (2, 2)$ CV(61)
$p(0) = (-1, 0)$ CV(30)	$p(0) = (0, 0)$ -	$p(0) = (1, 0)$ CV(63)
$p(0) = (-2, -2)$ CV(39)	$p(0) = (-1, -2)$ CV(26)	$p(0) = (0, -2)$ -

Convergence du tir simple suivant la région du point de départ

Bien sur, il n'y a ici que peu de régions à explorer, et l'on peut en fait atteindre la convergence depuis un point situé dans une "mauvaise" région (avec davantage d'itérations toutefois), mais ceci est dû à la simplicité du

problème (P). En fait, le tir simple diverge déjà dans 4 cas sur 9, et l'on comprend qu'il ne soit pas réaliste d'espérer résoudre directement de cette façon des problèmes plus difficiles. C'est pourquoi nous avons eu recours à une démarche homotopique pour obtenir un point initial satisfaisant.

1.1.3 Démarche homotopique

Comme nous venons de le voir, l'application de la méthode de tir est parfois délicate, en raison de la très grande sensibilité au point initial. Ceci est dû au fait qu'elle consiste typiquement à utiliser un algorithme de type quasi-Newton pour trouver un zéro de la fonction de tir. Suivant les problèmes, cette fonction de tir peut être fortement non linéaire ou peu régulière, d'où un rayon de convergence parfois très faible pour la méthode de tir. C'est en particulier le cas pour les problèmes à structure bang-bang ou présentant des arcs singuliers, la fonction de tir devenant une application multivaluée dans ce dernier cas.

Une voie possible pour contourner cette difficulté est alors d'utiliser une démarche homotopique pour trouver une bonne initialisation de la méthode de tir. A la base, le principe d'une méthode homotopique (ou *continuation*) est de résoudre un problème difficile en partant de la solution connue d'un problème apparenté, mais plus facile. On entend par apparenté l'existence d'une application H , appelée *homotopie*, qui connecte les deux problèmes, avec de bonnes propriétés.

Notation : dans toute la suite, on réserve la notation H à l'homotopie, tandis que \mathcal{H} désigne le Hamiltonien.

DÉFINITION 1 *Soient r et f deux applications de \mathbf{R}^n dans \mathbf{R}^n . On appelle une homotopie connectant r et f toute application H :*

$$\begin{aligned} H : \overline{\Omega} \times [0, 1] &\rightarrow \mathbf{R}^n \\ (z, \lambda) &\mapsto H(z, \lambda) \end{aligned}$$

avec Ω un ouvert borné de \mathbf{R}^n et H continue, telle que

$$H(\cdot, 0) = r \quad \text{et} \quad H(\cdot, 1) = f$$

La première chose à faire est donc de trouver une telle application H pour le problème de contrôle optimal considéré. Pour cela, on choisit d'introduire le paramètre homotopique λ dans l'objectif du problème original. Cette paramétrisation est faite de telle manière que le cas $\lambda = 1$ corresponde au problème non modifié, et que le cas $\lambda = 0$ soit plus régulier et puisse être résolu directement par tir simple. On obtient alors une famille de problèmes

aux deux bouts $(BVP)_\lambda$ paramétrée par $\lambda \in [0, 1]$, et l'on peut utiliser les fonctions de tirs associées (S_λ) pour la continuation, en définissant l'homotopie

$$H : (z, \lambda) \mapsto S_\lambda(z).$$

Maintenant, supposons connaître une solution z_0 du problème $(BVP)_0$, c'est-à-dire un zéro de $H(\cdot, 0)$. Le principe de la méthode de continuation est de partir de cette solution en $\lambda = 0$ pour atteindre $\lambda = 1$, où l'on a une solution du problème originel.

Continuation discrète

La première idée qui vient à l'esprit est tout simplement d'essayer de résoudre une suite de problèmes $(BVP)_{\lambda_k}$, avec une suite (λ_k) allant de 0 à 1. Chaque problème intermédiaire utilise alors la solution précédente comme point initial. Cependant, trouver en pratique une telle suite (λ_k) convenable est souvent problématique, la question du pas suivant λ étant en particulier laissée à l'expérimentateur, ce qui implique souvent de nombreux tâtonnements. C'est pourquoi il est souvent préférable de recourir à une méthode de suivi de chemin complet.

Continuation différentielle et méthode simpliciale

Contrairement à la continuation discrète, les méthodes de suivi de chemin visent à suivre complètement le chemin de zéros, et peuvent être classifiées en deux grandes familles, les méthodes de continuation différentielle (ou *Predictor-Corrector (PC)*), et les méthodes simpliciales (ou *Piecewise Linear (PL)*). Une documentation étendue sur les deux classes peut être trouvée dans les ouvrages de référence de E.Allgower et K.Georg, cf [1, 2] en particulier.

Les méthodes de continuation différentielle suivent le chemin de zéros en le considérant comme une courbe différentiable. Elles sont généralement rapides, car le pas suivant le paramètre homotopique λ est calculé automatiquement, et le suivi peut parfois ainsi être complété en un très petit nombre de pas. La contrepartie est que la continuation différentielle requiert certaines propriétés de régularité pour le chemin (typiquement, C^2 -différentiabilité et Jacobien de rang maximal). Ensuite, le calcul du Jacobien peut poser certaines difficultés pratiques, H étant ici une fonction de tir dont l'évaluation implique l'intégration d'un problème à valeur initiale. Une méthode Prédicteur-Correcteur efficace est implémentée par exemple dans le package HOMPACK de Watson (cf [38]), qui a été utilisé par T.Haberkorn pour résoudre la famille de problèmes de transfert orbital étudiés au chapitre 3, voir [26].

Les méthodes simpliciales, quant à elles, suivent en pratique une approximation affine par morceaux du chemin de zéros (d'où le nom plus général de *piecewise linear methods*), et sont censées être plus robustes, quoique plus lentes, que la continuation différentielle. Le présent travail est basé sur l'application de ces méthodes à des problèmes de contrôle optimal ayant une faible régularité, plus spécifiquement présentant une structure bang-bang ou des arcs singuliers. On décrit au chapitre 2 le principe général de ces méthodes, ainsi que certaines améliorations que nous avons tenté d'introduire dans ce contexte particulier.

Nous revenons maintenant à l'exemple simple vu précédemment, pour illustrer la démarche de continuation servant à obtenir un point initial fiable pour le tir.

Retour sur l'exemple simple : continuation sur l'objectif

On paramètre ici l'objectif par $\lambda \in [0, 1]$:

$$J_\lambda = \int_0^2 \lambda |u| + (1 - \lambda) |u|^2 dt$$

Ceci donne une famille de problèmes (P_λ) telle que (P_1) est le problème (P) original. L'application du PMP à (P_λ) donne la dynamique (inchangée) de l'état et état adjoint

$$\varphi \begin{cases} \dot{x}_1(t) = x_2(t) \\ \dot{x}_2(t) = u(t) \\ \dot{p}_1(t) = 0 \\ \dot{p}_2(t) = -p_1(t) \end{cases}$$

ainsi que l'expression du contrôle optimal

$$\lambda < 1$$

$$\begin{array}{ll} \text{si } |p_2(t)| \leq \lambda & \text{alors } u^*(t) = 0 \\ \text{si } |p_2(t)| > 2 - \lambda & \text{alors } u^*(t) = -\text{sgn}(p_2(t)) \\ & \text{sinon } u^*(t) = -\text{sgn}(p_2(t)) \frac{|p_2(t)| - \lambda}{2(1 - \lambda)} \end{array}$$

$$\lambda = 1$$

$$\begin{array}{ll} \text{si } |p_2(t)| < 1 & \text{alors } u^*(t) = 0 \\ & \text{sinon } u^*(t) = -\text{sgn}(p_2(t)) \end{array}$$

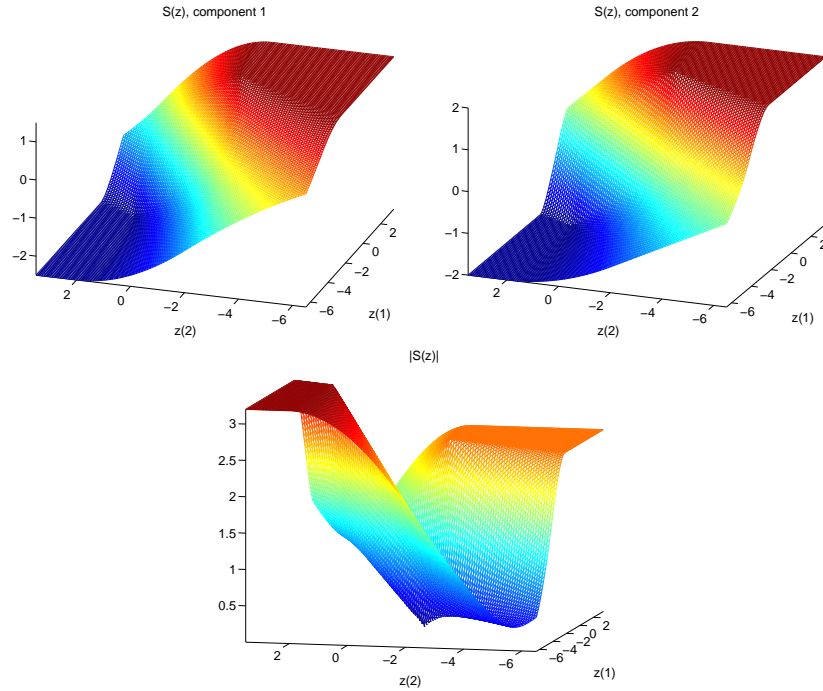
avec les conditions aux limites sur l'état adjoint p :

- $p(0)$ est libre (donc $z = p(0)$).
- $p(2)$ est libre, car $x(2)$ est fixé.

Si l'on note S_λ la fonction de tir associée à (P_λ) , notre but est maintenant de suivre le chemin de zéros de l'homotopie

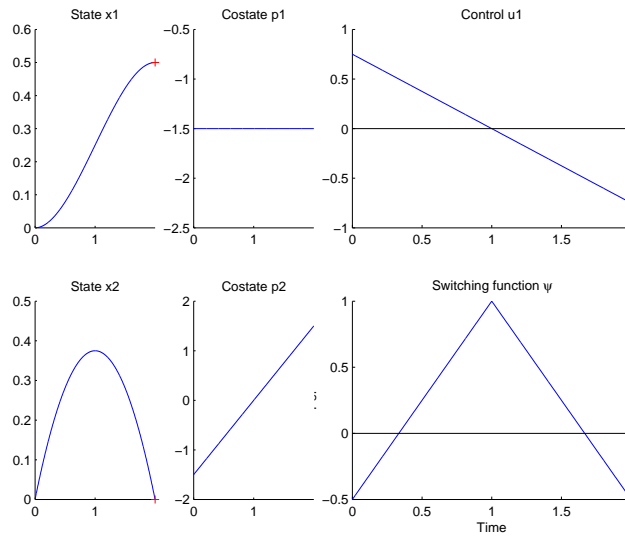
$$H : (z, \lambda) \mapsto S_\lambda(z)$$

de $\lambda = 0$ jusqu'à $\lambda = 1$. Pour $\lambda = 0$ le problème (P_0) est bien plus régulier, car le contrôle optimal est continu, au lieu de la structure bang-bang discontinue pour $\lambda = 1$. La fonction de tir correspondante S_0 est ainsi plus lisse, comme on le voit sur les graphes ci-dessous.



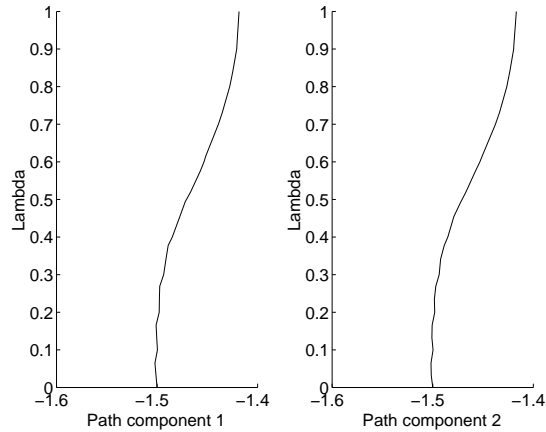
Fonction de tir pour (P_0) à $\lambda = 0$

En fait, pour (P_0) le tir simple converge depuis n'importe laquelle des 9 initialisations testées précédemment, et donne la solution suivante :



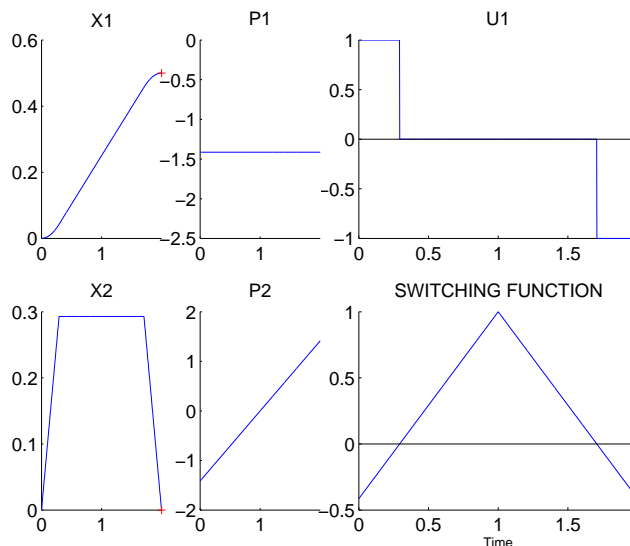
Solution pour (P_0) - $z^* = (-1.5, -1.5)$

il ne nous reste plus qu'à effectuer le suivi de chemin de $\lambda = 0$ à $\lambda = 1$, ce qui est réalisé par l'algorithme simplicial décrit au chapitre 2. On représente ici les deux composantes du chemin de zéros (qui pour cet exemple particulier sont identiques, ce qui n'est évidemment pas le cas en général).



Chemin de zéros pour le problème (P)

Maintenant, si l'on prend comme point initial la solution z_1 obtenue à la fin du chemin (pour $\lambda = 1$), le tir simple converge immédiatement. Voici la solution, avec les commutations lorsque la fonction de commutation $\psi = 1 - |p_2|$ s'annule.



Solution pour (P) - $z^* = (-\sqrt{2}, -\sqrt{2})$.

Un second exemple simple : arcs singuliers

On considère ici un second exemple simple, pour illustrer le cas des arcs singuliers, qui est étudié plus tard au chapitre 4. On définit la famille de

problèmes

$$(P_\lambda) \begin{cases} \text{Min } \int_0^2 \lambda |u(t)| + (1-\lambda)u^2(t) dt \\ \dot{x} = u \\ |u| \leq 1 \\ x(0) = 0 \\ x(2) = 0.5 \end{cases}$$

L'application du Principe du Maximum donne l'expression du contrôle optimal

$\lambda < 1$

$$\begin{array}{ll} \text{si } |p(t)| \leq \lambda & \text{alors } u^*(t) = 0 \\ \text{si } |p(t)| > 2 - \lambda & \text{alors } u^*(t) = -\text{sgn}(p(t)) \\ & \text{sinon } u^*(t) = -\text{sgn}(p(t)) \frac{|p(t)| - \lambda}{2(1-\lambda)} \end{array}$$

$\lambda = 1$

$$\begin{array}{ll} \text{si } |p(t)| < 1 & \text{alors } u^*(t) = 0 \\ \text{sinon si } |p(t)| > 1 & \text{alors } u^*(t) = -\text{sgn}(p(t)) \\ \text{sinon} & u(t) = -\alpha(t) \text{sgn}(p(t)) \quad , \alpha(t) \in [0, 1]. \end{array}$$

On remarque que lorsque $|p(t)| = 1$, le contrôle optimal est multivalué. Si cela se produit sur un intervalle non réduit à un point, nous avons un *arc singulier*.

Voici le problème aux deux bouts

$$(BVP_\lambda) \begin{cases} \dot{x}(t) = u(t) \\ \dot{p}(t) = 0 \\ x(0) = 0 \\ x(2) = 0.5 \\ p(0) \text{ libre} \\ p(2) \text{ libre} \end{cases}$$

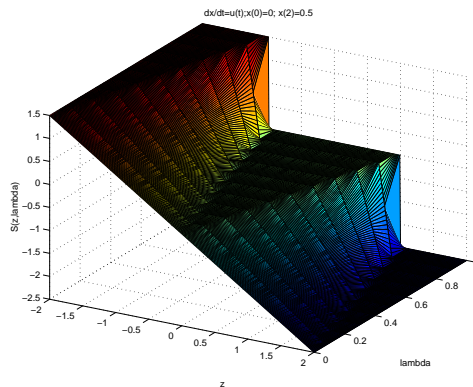
et la fonction de tir S :

$$S_\lambda(z) = \hat{x}(2) - 0.5$$

où \hat{x} est solution du problème à valeur initiale

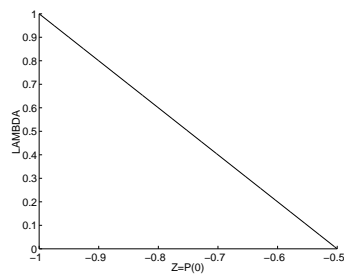
$$(IVP_\lambda) \begin{cases} \dot{x}(t) = u(t) \\ \dot{p}(t) = 0 \\ x(0) = 0 \\ p(0) = z. \end{cases}$$

On définit l'homotopie $H : (z, \lambda) \mapsto S_\lambda(z)$, comme précédemment. Pour $\lambda = 0$, la solution correspond à un contrôle constant $u = 0.25$, avec $z^* = p^*(0) = -0.5$. Pour $\lambda = 1$, tout contrôle non négatif vérifiant $\int_0^2 u(t) dt = 0.5$ est optimal, ce qui correspond à un arc singulier arc sur $[0, 2]$, avec $z^* = p^*(0) = -1$. On représente ici l'évolution de la fonction de tir S .



Evolution de la fonction de tir sur un cas simple d'arc singulier

Comme nous le voyons, S n'est pas définie à la solution $(z, \lambda) = (-1, 1)$. Si on tente d'utiliser ici une continuation différentielle pour suivre le chemin, l'algorithme stoppe juste avant $\lambda = 1$, ce qui n'est pas surprenant. La méthode simpliciale, par contre, converge correctement à la solution $z^* = -1$ en $\lambda = 1$:



Suivi de chemin sur un cas simple d'arc singulier

1.2 Résultats de convergence

Pour conclure ce chapitre, nous présentons ici quelques résultats théoriques portant sur la démarche homotopique appliquée à la méthode de tir. Afin de pouvoir prendre en compte le cas des arcs singuliers, on se place dans le cadre des inclusions différentielles, plus précisément on considère que l'application donnant le contrôle optimal peut être multivaluée. Comme on va le voir, ce choix d'hypothèses volontairement faibles limite fortement (à notre connaissance) les résultats de convergence obtenus.

1.2.1 Propriétés de minimisation du Hamiltonien

Nous commençons par redonner certaines propriétés concernant la minimisation du Hamiltonien, qui ont été présentées par J.Gergaud dans [24] (voir aussi [29]). Le résultat principal porte sur la semicontinuité supérieure (voir ci-après) de l'application multivaluée donnant le contrôle optimal.

Notations : on garde la notation $y = (x, p)$, avec y de dimension n (ie deux fois la dimension de l'état). De même, \mathcal{H} désigne le Hamiltonien, par opposition à H pour l'homotopie.

On rappelle tout d'abord un résultat standard :

THÉORÈME 1 *On suppose que $U \subset \mathbf{R}^m$ est un compact convexe d'intérieur non vide, et que le Hamiltonien $\mathcal{H} : [a, b] \times \mathbf{R}^n \times U \rightarrow \mathbf{R}$ est continu et convexe par rapport au contrôle u . On note $\Gamma(t, x, p)$ l'ensemble des solutions de $\min_{u \in U} \mathcal{H}(t, x, p, u)$ Alors Γ est une application multivaluée (ou multi-application) à valeurs compactes convexes non vides.*

Ainsi que les définitions suivantes :

DÉFINITION 2 *Une multi-application $F : \mathbf{R}^n \rightarrow \mathbf{R}^m$ est dite semi continue supérieurement (scs) (au sens de Berge, [8] page 114) en x_0 ssi pour tout ouvert O de Y contenant $F(x_0)$, il existe un voisinage U de x_0 tel que $F(U) \subset O$. F est dite semi continue supérieurement ssi F est scs en tout point de \mathbf{R}^n .*

DÉFINITION 3 *Une multi-application $F : \mathbf{R}^n \rightarrow \mathbf{R}^m$ est dite semi continue inférieurement (sci) en x_0 ssi pour tout $y_0 \in F(x_0)$ et toute suite (x_k) qui converge vers x_0 , il existe une suite (y_k) , $y_k \in F(x_k)$ qui converge vers y_0 . F est dite semi continue inférieurement ssi F est sci en tout point de \mathbf{R}^n .*

DÉFINITION 4 *Une fonction $f : \mathbf{R}^m \rightarrow \overline{\mathbf{R}}$ est dite inf-compacte (au sens de [33]) ssi*

$$\forall (y, a) \in \mathbf{R}^m \times \mathbf{R}, \quad \{u \in \mathbf{R}^m : f(u) - (u|y) \leq a\} \quad \text{est compact.}$$

($\overline{\mathbf{R}}$ désigne la droite réelle étendue et $(\cdot|\cdot)$ le produit scalaire usuel sur \mathbf{R}^m .)

LEMME 1 Une application à valeurs compactes G est scs si et seulement si pour toute suite (x_k) qui converge vers x , $(G(x_k))$ converge vers $G(x)$ suivant : $\forall \epsilon > 0, \exists k_0 > 0$ tel que $\forall k > k_0, G(x_k) \subset G(x) + \epsilon B(0, 1)$.¹

Preuve. voir [30] page 66.

Dans le lemme suivant, on entend par fonction propre une fonction à valeur réelle qui ne prend jamais la valeur $-\infty$ et qui n'est pas identiquement égale à $+\infty$.

LEMME 2 Soit $(f_k)_{k \in \mathbf{N}}$ une suite de fonctions propres convexes semicontinues inférieurement définies sur \mathbf{R}^m . Sous les hypothèses suivantes

(i) $(f_k)_{k \in \mathbf{N}}$ converge simplement vers f

(ii) $\text{int}(\text{dom } f) \neq \emptyset$

(iii) f est inf-compacte

Alors,

$$\lim_{k \rightarrow \infty} \inf_{u \in \mathbf{R}^m} f_k(u) = \inf_{u \in \mathbf{R}^m} f(u)$$

et, $\forall \epsilon > 0$, il existe $k_0 \in \mathbf{N}$ tel que

$$\text{argmin}_{u \in \mathbf{R}^m} f_k(u) \subset \text{argmin}_{u \in \mathbf{R}^m} f(u) + \epsilon B(0, 1) \quad \forall k \geq k_0.$$

Preuve. voir [33], page I.3.54.

THÉORÈME 2 On considère les mêmes hypothèses que pour le Théorème 1.

Alors Γ vérifie les propriétés de convergence suivantes :

Si (t_k, x_k, p_k) est une suite qui converge vers (t, x, p) alors

(i) $\inf_{u \in \mathbf{R}^m} \mathcal{H}(t_k, x_k, p_k, u) \rightarrow \inf_{u \in \mathbf{R}^m} \mathcal{H}(t, x, p, u)$ quand $k \rightarrow +\infty$

(ii) $\forall \epsilon > 0, \exists k_0 > 0$ tel que $\forall k > k_0, \Gamma(t_k, x_k, p_k) \subset \Gamma(t, x, p) + \epsilon B(0, 1)$.

Preuve. Soit (t_k, x_k, p_k) une suite qui converge vers (t, x, p) .

On note $f_k(u) = \mathcal{H}(t_k, x_k, p_k, u) + \delta(u/U)$ et $f(u) = \mathcal{H}(t, x, p, u) + \delta(u/U)$ (avec $\delta(u/U) = 0$ si $u \in U$, et $+\infty$ si $u \notin U$).

Pour les problèmes étudiés dans la suite, l'expression du Hamiltonien indique que les (f_k) sont convexes et semicontinues inférieurement. Vérifions les hypothèses du Lemme 2 :

(i) Si $u \notin U$ alors $f_k(u) = f(u) = +\infty \forall k$

si $u \in U$ alors $f_k(u) = \mathcal{H}(t_k, x_k, p_k, u)$, et comme \mathcal{H} est continue on a $\mathcal{H}(t_k, x_k, p_k, u) \rightarrow \mathcal{H}(t, x, p, u)$, d'où $f_k(u) \rightarrow f(u)$.

(ii) $\text{int}(\text{dom } f) = \text{int}(U) \neq \emptyset$.

(iii) Soit $v \in \mathbf{R}^m, a \in \mathbf{R}$, alors $\{u \mid \mathcal{H}(t, x, p, u) + \delta(u/U) - (u|v) \leq a\} = U \cap \{u \mid \mathcal{H}(t, x, p, u) - (u|v) \leq a\}$. Il s'agit d'un sous ensemble fermé de \mathbf{R}^m inclus dans U compact, il est donc compact. Ceci donne l'inf-compactité de f . Et le Lemme 2 prouve le théorème.

¹ $B(0, 1)$ désigne la boule unité fermée.

COROLLAIRE 1 *On considère les mêmes hypothèses qu'au Théorème 1. Alors Γ est semicontinue supérieurement.*

Preuve. les Théorème 1, Lemme 1 et Théorème 2 donnent le résultat.

Remarque : Dans le cas où \mathcal{H} est strictement convexe, on a la propriété bien connue (voir par exemple [23](Théorème 6.1 p.75) et [10]) que u^* est une fonction continue (Γ étant alors une fonction continue).

1.2.2 Propriétés de convergence

On s'intéresse maintenant aux propriétés de convergence de l'homotopie proprement dite. Ces résultats ont été présentés par J.Gergaud dans [15], et dérivent principalement des ouvrages [5] par J.P. Aubin et A. Cellina, et [22] par A.F. Filippov, dont on conserve les notations.

Notations :

- soit $K \subset \mathbf{R}^n$, $\overline{\text{co}} K$ désigne l'enveloppe convexe fermée de K
- soit $M \subset \mathbf{R}^n$, $M^\delta = \{m \text{ tel que } d(m, M) \leq \delta\}$

On rappelle ici la définition d'une δ -solution au sens de [22] (p. 76).

DÉFINITION 5 *Une fonction y est appelée δ -solution de $\dot{y}(t) \in F(t, y(t))$, avec $F : [a, b] \times \mathbf{R}^n \rightarrow \mathbf{R}^n$ une application multivaluée scs², si sur un intervalle $[a, b]$ y est absolument continue et*

$$\dot{y}(t) \in F_\delta(t, y(t)) = [\overline{\text{co}} F(t^\delta, y^\delta)]^\delta$$

avec

$$F(t^\delta, y^\delta) = \cup_{s \in t^\delta, z \in y^\delta} F(s, z).$$

LEMME 3 *Soit (y_k) une suite de $AC_n([a, b])^3$ telle que :*

(i) $\forall t \in [a, b]$, $\{y_k(t)\}_k$ est relativement compact.

(ii) $\exists l$ tel que $|\dot{y}_k(t)| \leq l$ presque partout sur $[a, b]$.

*Alors il existe une sous-suite, encore notée (y_k) , qui converge uniformément vers une fonction $y : [a, b] \rightarrow \mathbf{R}^n$ absolument continue, et pour laquelle la suite (\dot{y}_k) converge *-faiblement vers \dot{y} dans $L_n^\infty([a, b])$.*

Preuve. la preuve suit le principe de la démonstration du Théorème 4, [5] p.14-15. La suite (y_k) est équicontinue car

$$|y_k(t') - y_k(t'')| = \left| \int_{t'}^{t''} \dot{y}_k(t) dt \right| \leq l|t' - t''|.$$

²semicontinue supérieurement

³AC désigne absolument continu.

Le théorème d'Arzelá-Ascoli implique l'existence d'une sous-suite, encore notée (y_k) , qui converge uniformément vers y dans $C_n([a, b])$. De plus, $\dot{y}_k \in \overline{B}(0, l) \subset L_n^\infty([a, b])$, et $L_n^\infty([a, b])$ est le dual de $L_n^1([a, b])$. Alors le théorème d'Alaoglu implique que cette boule fermée est *-faiblement compact. Comme $L_n^1([a, b])$ est séparable cette boule fermée est métrisable pour la topologie faible-* (cf [11]). Alors il existe une sous-suite, encore notée (\dot{y}_k) , qui converge *-faiblement vers z dans $L_n^\infty([a, b])$.

On veut maintenant prouver que y est absolument continue et que $\dot{y} = z$. Tout d'abord, (y_k) est absolument continue, donc

$$y_k(t') - y_k(t'') = \int_{t'}^{t''} \dot{y}_k(s) ds \quad (*)$$

(y_k) converge (uniformément) vers y , donc le membre de gauche converge vers $y(t') - y(t'')$. Comme (\dot{y}_k) converge *-faiblement vers z , pour chaque composante i on a

$$\langle \mathbf{1}, \dot{y}_{k,i} \rangle_{L^1, L^\infty} = \int_a^b \dot{y}_{k,i}(s) ds \rightarrow \langle \mathbf{1}, z_i \rangle_{L^1, L^\infty} = \int_a^b z_i(s) ds$$

(avec $\mathbf{1}$ l'application constante égale à 1). Donc le membre de droite de (*) converge vers $\int_{t'}^{t''} z(s) ds$. On a ainsi

$$y(t') - y(t'') = \int_{t'}^{t''} z(s) ds$$

avec z dans $L_n^\infty([a, b])$, donc dans $L_n^1([a, b])$. Cela signifie que y est absolument continue et que $\dot{y}(t) = z(t)$ presque partout.

THÉORÈME 3 *Soit (y_k) une suite de $AC_n([a, b])$ qui converge vers y et vérifie $\dot{y}_k(t) \in K$ pour tout k et t , avec K compact. Alors y est absolument continue et $\dot{y}_k(t) \in \overline{\text{co}} K$ pour tout t .*

Preuve. La preuve est basée sur le Lemme 13 de Filippov, [22] p.64.

THÉORÈME 4 *Soit F une application à valeurs convexes compactes non vides, définie sur un ouvert $\Omega \subset \mathbf{R}^{n+1}$. Soit (y_k) une suite de δ_k -solutions définies sur $[a, b]$ qui converge uniformément vers $y : [a, b] \rightarrow \mathbf{R}^n$ quand $\delta_k \rightarrow 0$, et telle que le graphe de y est dans Ω . Alors y est solution de l'inclusion différentielle $\dot{y}(t) \in F(t, y(t))$.*

Preuve. voir le Lemme 1 de Filippov, [22] p.76.

LEMME 4 *Soit $\varphi : \Omega \times [0, 1] \rightarrow \mathbf{R}^n$, avec Ω un ouvert de \mathbf{R}^n , une application multivaluée vérifiant*

(i) φ est semicontinue supérieurement et à valeurs convexes compactes non vides.

(ii) $\varphi_\lambda = \varphi(\cdot, \lambda)$ est une fonction usuelle et est C^1 par morceaux pour $0 \leq \lambda < 1$.

Supposons que les solutions de $\dot{y}_\lambda(t) = \varphi_\lambda(y(t))$ restent dans un compact fixé K et sont définies sur un intervalle $[0, t_f]$. Alors y_λ est une δ -solution de l'inclusion différentielle $\dot{y}(t) \in \varphi(y(t), 1)$, et δ tend vers 0 quand $\lambda \rightarrow 1$.

Preuve. φ est semicontinue supérieurement en $(y^*, 1)$ pour tout $(y^*, 1) \in K$, donc pour tout $\epsilon = \delta$, il existe η tel que pour tout $|y - y^*| < \eta_{y^*} < \delta$, $|\lambda - 1| < \eta_{y^*} < \delta$, on a $\varphi_\lambda(y, \lambda) \in \varphi(y^*, 1)^\epsilon$.

Donc $K \subset \cup_{y^* \in K} B(y^*, \eta_{y^*})$ et comme K est compact, on a $K \subset \cup_{i=1}^q B(y_i, \eta_i)$.

$\forall \epsilon = \delta, \forall y_i, \exists \eta_i |y - y_i| < \eta_i < \delta$ et $|\lambda - 1| < \eta_i < \delta$, $\varphi_\lambda(y) \in \varphi(y_i, 1)^\epsilon$.

Pour tout $y \in K$, il existe y_i tel que $y \in B(y_i, \eta_i)$ et ainsi

$$\varphi_\lambda(y) \in (\varphi(y_i, 1))^\epsilon \subset (\varphi(y_i^\eta, 1))^\epsilon \subset (\varphi(y^\delta, 1))^\epsilon.$$

Alors pour tout λ tel que $|\lambda - 1| < \eta$ et pour tout $y \in K$, on a $\varphi_\lambda(y) \in (\varphi(y^\delta, 1))^\delta$.

THÉORÈME 5 *Supposons que les solutions de $(BVP)_\lambda$ restent dans un compact fixé de $[0, t_f] \times K$, $K \subset \Omega$, avec Ω un ouvert de \mathbf{R}^n . Alors de toute suite (y_{λ_k}) de solutions de $(BVP)_{\lambda_k}$, telle que $\lambda_k \rightarrow 1$ quand $k \rightarrow +\infty$, on peut extraire une sous-suite (y_k) vérifiant :*

(i) (y_k) converge uniformément vers y solution de $(BVP)_1$.

(ii) (\dot{y}_k) converge *-faiblement vers \dot{y} dans $L_n^\infty([0, t_f])$.

Preuve. $\varphi(y, \lambda)$ est scs donc $\varphi(K, [1 - \epsilon, 1])$ est compact. Il existe l tel que $|\dot{y}_\lambda(t)| < l$ pour $\lambda \in [1 - \epsilon, 1]$. Les y_λ sont absolument continues, et le Lemme 3 dit que l'on peut extraire une sous-suite (y_k) qui converge uniformément vers y , et telle que (\dot{y}_k) converge *-faiblement vers \dot{y} dans $L_n^\infty([0, t_f])$. D'après le Lemme 4, (y_k) est une δ_k -solution. $\varphi(y, 1)$ est à valeurs convexes compactes non vides, et le Théorème 3 dit que y est une solution de l'inclusion différentielle $\dot{y}(t) \in \varphi(y(t), 1)$. Les conditions initiales et finales peuvent s'exprimer comme $h_0(y(0)) = 0$ et $h_f(y(t_f)) = 0$ avec h_0 et h_f continues. La convergence uniforme de (y_k) implique que y vérifie les conditions aux limites, donc y est une solution de $(BVP)_1$.

COROLLAIRE 2 *Sous les hypothèses du Théorème 5, supposons que $\dot{x} = f(t, x, u)$ fournit une expression du contrôle sous la forme $u = S(t, x) + R(t, x)\dot{x}$, avec R et S continues et R linéaire par rapport à x . On considère la sous-suite $(y_k) = (x_k, p_k)$ du Théorème 5, et on note $(u_k) = S(t, x_k) + R(t, x_k)\dot{x}_k$. Alors (u_k) converge *-faiblement dans $L_n^\infty([0, t_f])$.*

Preuve. voir [17], preuve de la Proposition 3.2, page 551-552.

Il est à noter que le Théorème 5 et le Corollaire 2 n'établissent la convergence que d'une sous-suite extraite de la suite des solutions (y_{λ_k}) , ceci provenant du théorème d'Arzelà-Ascoli dans le Lemme 3. A notre connaissance, sous les hypothèses choisies, on ne peut pas affirmer la convergence de la suite des solutions elle-même.

Chapitre 2

Méthode homotopique simpliciale

On rappelle ici le principe général des algorithmes simpliciaux, avant de détailler certains points algorithmiques que nous avons introduits dans le cadre de ce travail. A ce propos, nous souhaitons souligner que nous avons essayé de rester cohérents avec l'esprit des problèmes étudiés, caractérisés par une faible régularité. Nous voulons en particulier conserver la robustesse des méthodes simpliciales, c'est pourquoi les modifications introduites ne devraient pas requérir d'hypothèses supplémentaires sur le chemin (qui ne seraient d'ailleurs vraisemblablement pas satisfaites en pratique). C'est aussi la raison pour laquelle on utilise à la base une version très dépouillée d'algorithme simplicial, sans prendre en compte certaines améliorations existantes (comme l'introduction de pas de Newton par exemple).

2.1 Principe des méthodes simpliciales

Comme mentionné avant, les lecteurs intéressés par les méthodes simpliciales et/ou la continuation différentielle peuvent se référer en particulier aux ouvrages par E.Allgower et K.Georg ([1, 2, 3]), ainsi que ([36, 37]) par M.Todd pour les méthodes simpliciales plus spécifiquement.

Les méthodes simpliciales suivent en fait le chemin de zéros de l'homotopie H en construisant une approximation affine par morceaux de H , d'où le nom plus général de *Piecewise Linear continuation methods*. Cette approximation est bâtie sur une certaine subdivision de l'espace de recherche, le plus souvent sous la forme d'une *triangulation* en *simplexes*, à l'origine du nom des méthodes simpliciales.

L'avantage principal de ces méthodes est qu'elles ne requièrent qu'une faible régularité de l'homotopie H . Comme elles n'utilisent pas de dérivées, il

n'est pas nécessaire que l'homotopie soit différentiable comme dans le cas de la continuation différentielle. La continuité est en fait suffisante, et ne devrait même pas être nécessaire dans tous les cas. De plus, cette classe de méthodes est plus facilement adaptable au cas multivalué, ce qui est intéressant pour l'étude des problèmes avec arcs singuliers. Leur plus grand inconvénient est qu'elles sont nettement plus lentes que la continuation différentielle, quand cette dernière converge bien sûr.

Simplexes et triangulations

On considère dans la suite une homotopie $H : \mathbf{R}^{n+1} \rightarrow \mathbf{R}^n$. Commençons par plusieurs définitions utiles.

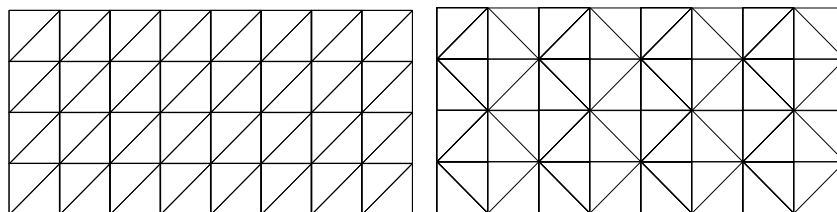
DÉFINITION 6 (SIMPLEXES ET FACES) *Un simplexe est l'enveloppe convexe de $n + 2$ points de \mathbf{R}^{n+1} affinement indépendants (appelés sommets), tandis qu'une k -face d'un simplexe est l'enveloppe convexe de k sommets du simplexe (note : k sera omis dans la suite pour les $(n + 1)$ -faces, qui seront juste appelées faces).*

DÉFINITION 7 (TRIANGULATION) *On appellera triangulation toute famille dénombrable T de simplexes de \mathbf{R}^{n+1} vérifiant :*

- *L'intersection de deux simplexes de T est soit une face soit vide.*
- *T est localement finie (un sous-ensemble compact de \mathbf{R}^{n+1} rencontre un nombre fini de simplexes).*

En pratique, une triangulation est décrite par la forme de ses simplexes (c'est-à-dire les relations liant les sommets), ainsi que par ses "règles de pivotage", qui permettent de construire les simplexes adjacents à un simplexe donné.

Les graphes suivants illustrent deux célèbres triangulations, la K_1 de Freundenthal, et la J_1 ("Union Jack") de Todd. Il s'agit dans les deux cas de triangulations "uniformes", dans le sens où la taille des simplexes reste la même sur tout l'espace, par opposition à certaines triangulations "raffinées" montrées ci-après.

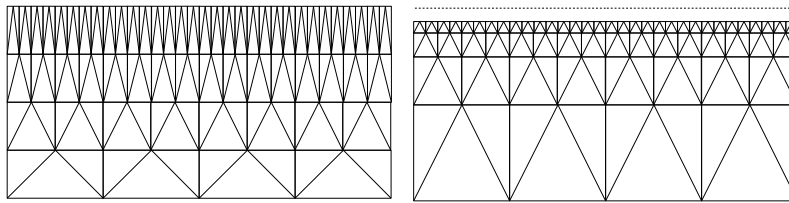


Triangulations uniformes K_1 et J_1 de $\mathbf{R} \times [0, 1]$

Ces deux triangulations sont très simples à manipuler, car les règles de pivotage peuvent s'exprimer sous la forme de réflexions géométriques (voir

[2] 13.3 par exemple). Cette simplicité est la raison pour laquelle nous avons naturellement commencé les expérimentations avec ces deux triangulations, qui se sont finalement avérées très robustes dans leur comportement. D'autres triangulations plus sophistiquées ont bien sûr été présentées, mais leurs règles de pivotage sont généralement plus complexes. Nous avons par exemple également testé la triangulation D_1 décrite par Dang dans [21] (cette triangulation est similaire à la J_1 en dimension 2, mais on peut voir la différence sur les graphes page 27), mais n'avons pas noté d'amélioration significative par rapport aux deux précédentes.

Par ailleurs, il existe également des triangulations "raffinées", pour lesquelles la taille des simplexes diminue lorsqu'on approche de la convergence. L'idée derrière ces triangulations est de combiner une bonne rapidité au début du chemin, ou le suivi est supposé plus facile, avec une bonne précision finale, pour obtenir une solution précise à la convergence. Voici par exemple deux évolutions raffinées de la J_1 , dues à Todd.



Triangulations raffinées : J_4 et J_3 (de $\mathbf{R} \times [0, 1[$ pour la dernière)

Remarque : on voit sur le graphe ci-dessus que pour la J_3 la taille des simplexes tend vers zéro lorsque le paramètre homotopique λ tend vers la valeur de convergence (1 par exemple), mais sans jamais l'atteindre. Il faut donc utiliser ici un critère d'arrêt adapté, c'est-à-dire quand le paramètre homotopique franchit un certain seuil $1 - \epsilon$, avec $\epsilon = 10^{-6}$ or 10^{-9} par exemple.

Toutefois, nous avons constaté que ces triangulations plus évoluées sont assez délicates à utiliser en pratique. On rencontre ainsi souvent des difficultés soit au début du chemin (en particulier pour la J_3 , dont les premiers simplexes sont trop grands pour démarrer correctement le suivi), où à la convergence, les simplexes devenant alors trop petits. Ceci est probablement dû à la nature particulière des problèmes étudiés : les homotopies sont des fonctions de tir, qui peuvent être fortement non linéaires même au début du chemin, qui n'est pas si facile à suivre. De plus, comme on est en général peu régulier vers $\lambda = 1$, le fait d'avoir des simplexes très petits tend à rendre l'algorithme plus sensible aux difficultés numériques près de la convergence. Par ailleurs, l'objectif principal de la méthode simpliciale n'est pas ici de résoudre précisément le problème, mais de fournir un point initial acceptable pour la méthode de tir. Ainsi, l'idée d'utiliser de petits simplexes pour

obtenir une solution plus précise n'est pas forcément pertinente dans notre cas.

Étiquetage et suivi de chemin

En plus du choix de la triangulation, il nous faut construire l'approximation affine par morceaux de l'homotopie. Ceci est réalisé par "l'étiquetage" des sommets des simplexes en fonction de la valeur de l'homotopie en ces points. On utilise ici l'étiquetage vectoriel standard rappelé ci-dessous, mais il existe d'autres possibilités, comme par exemple l'étiquetage entier mentionné en Annexe B.2, page 147.

DÉFINITION 8 (ÉTIQUETAGE) *On appelle étiquetage une application l qui associe une valeur aux sommets v_i d'un simplexe. On étiquette ici les simplexes par l'homotopie $H : l(v^i) = H(z^i, \lambda^i)$, avec $v^i = (z^i, \lambda^i)$.*

Une interpolation affine sur les sommets donne ainsi une approximation affine par morceaux de l'homotopie H sur la triangulation T . On note H_T cette approximation, dont on veut suivre le chemin de zéros. On rappelle pour cela la notion de "face complètement étiquetée".

DÉFINITION 9 (FACE COMPLÈTEMENT ÉTIQUETÉE) *Une face $[v_1, \dots, v_{n+1}]$ est dite complètement étiquetée si et seulement si elle contient un zéro de H_T , cette propriété restant vraie à une certaine perturbation près (ceci pour éliminer le cas dégénéré où ce zéro appartiendrait en fait à une k -face de dimension inférieure à $n+1$). Plus précisément, la face doit contenir une solution v_ϵ de l'équation $H_T(v) = \vec{\epsilon}$ pour tout $\epsilon > 0$ suffisamment petit (avec $\vec{\epsilon} = (\epsilon, \dots, \epsilon^n)$).*

Il existe une façon plus implémentable de caractériser une telle face, donnée par exemple dans [2], chapitre 12.3, page 159.

DÉFINITION 10 (MATRICE D'ÉTIQUETAGE) *La matrice d'étiquetage L d'une face $[v_1, \dots, v_{n+1}]$ est définie par*

$$L = \begin{pmatrix} 1 & \dots & 1 \\ H(v_1) & \dots & H(v_{n+1}) \end{pmatrix}.$$

Et on a la propriété suivante.

LEMME 5 *Une face $[v_1, \dots, v_{n+1}]$ est complètement étiquetée si et seulement si sa matrice d'étiquetage L est non singulière et L^{-1} est lexicographiquement positive, ie la première composante non nulle de chaque ligne de L^{-1} est positive.*

Preuve. voir [2], proposition 12.3.3, page 159-160

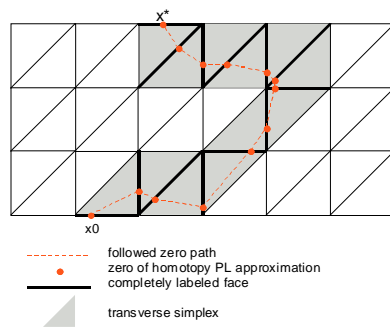
On rappelle maintenant la propriété fondamentale sur laquelle est basé le coeur d'un algorithme simplicial.

LEMME 6 *Tout simplexe possède soit aucune, soit exactement deux faces complètement étiquetées (on l'appelle dans ce dernier cas un simplexe transversal).*

Preuve. voir [2], chapitre 12.4, page 163-165.

Une preuve constructive de cette propriété, qui donne la seconde face complètement étiquetée (face de sortie) d'un simplexe en possédant déjà une, est souvent appelée *PL step*, *linear programming step*, ou *test lexicographique*. Une fois encore, nous renvoyons à [2] (pp 163-165), ou [24] (pp 69-70), pour une description détaillée de ces mécanismes. Alors, de par les propriétés des triangulations, il existe un unique simplexe transversal qui partage cette seconde face complètement étiquetée, qui peut être construit par les règles de pivotage spécifiques à la triangulation.

Un algorithme simplicial parcourt donc essentiellement une suite de simplexes transversaux, depuis un simplexe de départ donné avec une face complètement étiquetée en $\lambda = 0$, jusqu'à un simplexe final avec une face complètement étiquetée en $\lambda = 1$ (ou $1 - \epsilon$ pour certaines triangulations raffinées n'atteignant jamais 1, comme la J_3), qui contient une solution approchée de $H(z, 1) = 0$.

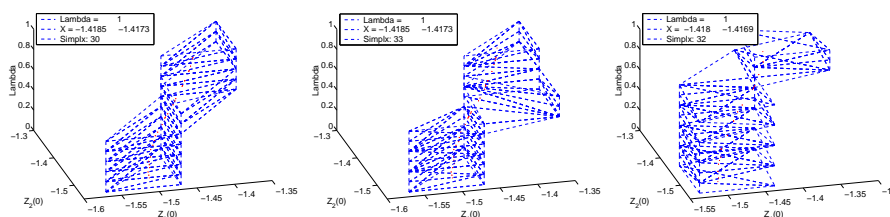


Suivi de chemin schématique pour la triangulation K_1 en dimension 2

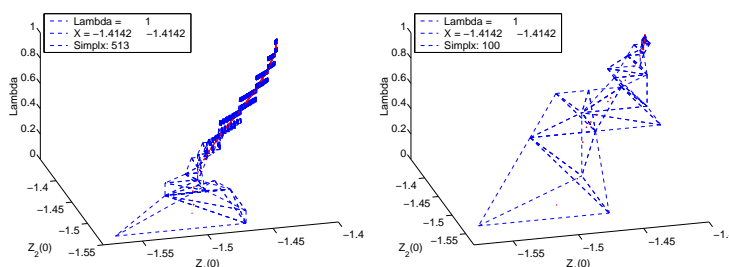
Exemples de suivis de chemin

On illustre ici le suivi de chemin du problème simple (P) décrit au chapitre 1, pour diverses triangulations (dont la description complète peut être trouvée dans [2, 37, 21] par exemple).

Triangulation	Simplexes	Solution	Norme finale
$K_1(10^{-1})$	30	(-1.418,-1.417)	$2.13 \cdot 10^{-3}$
$J_1(10^{-1})$	33	(-1.418,-1.417)	$2.13 \cdot 10^{-3}$
$D_1(10^{-1})$	32	(-1.418,-1.417)	$1.94 \cdot 10^{-3}$
$J_3(10^{-1})$	68	(-1.414,-1.414)	$7.17 \cdot 10^{-12}$
$J_4(10^{-1})$	513	(-1.414,-1.414)	$1.31 \cdot 10^{-9}$



Suivi de chemin de $\lambda = 0$ à $\lambda = 1$ - triangulations K_1, J_1 et D_1



Suivi de chemin de $\lambda = 0$ à $\lambda = 1$ - triangulations J_4 et J_3

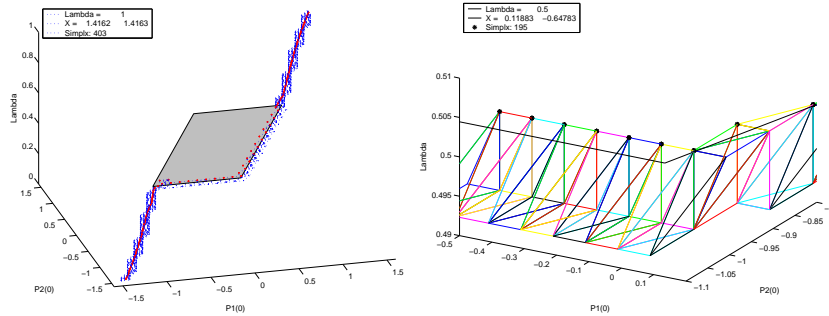
Suivi d'un chemin de rang non maximal

Incidentement, on peut illustrer sur cet exemple simple la robustesse des méthodes simpliciales, et plus spécifiquement la possibilité de suivre un chemin qui n'est pas de rang maximal (contrairement à la continuation différentielle par exemple). On considère ici une autre continuation pour le problème (P) , dans laquelle on modifie les conditions finales suivant λ , ce qui donne la famille de problèmes

$$(P_\lambda) \begin{cases} \text{Min } \int_0^2 |u| dt \\ \dot{x}_1 = x_2 \\ \dot{x}_2 = u \\ |u| \leq 1 \\ x(0) = (0, 0) \\ x(2) = (0.5 - \lambda, 0) \end{cases}$$

On retrouve ici le problème (P) pour $\lambda = 0$, tandis que $\lambda = 1$ correspond au problème "miroir", avec le contrôle optimal opposé. Pour $\lambda = 0.5$, les conditions initiales et terminales coïncident, avec $x(0) = x(2) = 0$. Dans ce cas, le contrôle nul est optimal, ce qui correspond à toute valeur de $p(0)$ dans la région centrale sur le graphe page 7. Ainsi, le chemin de zéros n'est pas

de rang maximal en $\lambda = 0.5$. Néanmoins, l'algorithme simplicial converge vers la solution correcte, comme on le voit ci-dessous.



Traversée de la région centrale en $\lambda = 0.5$

On constate que l'algorithme parvient à traverser la région centrale en $\lambda = 0.5$ en suivant la frontière, avec des simplexes ayant des sommets de part et d'autre. On note également que suivant la triangulation et/ou la taille de simplexes employée, l'algorithme peut tout aussi bien suivre l'autre frontière, plutôt que celle "bas-droite" comme sur le graphe ci-dessus.

Stabilité numérique des simplexes et matrices d'étiquetage

Comme on l'a dit plus haut, à chaque étape d'un algorithme simplicial on met à jour le simplexe transversal courant. Plus précisément, il est construit à partir du simplexe précédent, avec qui il partage une face complètement étiquetée, ce qui signifie qu'il n'y a qu'un seul sommet à modifier. Ceci est effectué par le biais des règles de pivotages, une fois que le test lexicographique a déterminé quel sommet du simplexe précédent doit être remplacé (ie celui qui ne fait pas partie de la face complètement étiquetée de sortie). Dans le même temps, on met directement à jour l'inverse de la matrice d'étiquetage du nouveau simplexe, au lieu de réinverser la nouvelle matrice d'étiquetage.

L'on peut donc s'inquiéter du risque d'accumulation d'erreurs numériques lorsque le nombre de simplexes traversés devient grand. Pour vérifier cela, on recalcule périodiquement tous les sommets à partir de l'expression formelle du simplexe, et on reconstruit puis inverse la matrice d'étiquetage.

Les tableaux ci-dessous montrent l'évolution de l'erreur sur les sommets des simplexes et l'inverse de la matrice d'étiquetage. Les tests sont réalisés sur l'exemple simple du chapitre 1, ainsi que les deux problèmes d'arcs singuliers étudiés au chapitre 4 (formulation BVP discrétisé). On indique dans chaque cas la triangulation utilisée, la dimension de z , et le nombre total de simplexes parcourus.

Problème simple : $K_1(10^{-6})$, dimension 2, > 3 000 000 simplexes

Fréquence de test	Erreur simplexe (abs/rel)	Erreur matrice (abs/rel)
10 000	0 / 0	$10^{-8} - 10^{-12} / 10^{-15} - 10^{-17}$
100 000	0 / 0	$10^{-9} - 10^{-11} / 10^{-16} - 10^{-18}$
1 000 000	0 / 0	$10^{-10} / 10^{-16}$

Problème d'arc 1 : $K_1(10^{-2})$, dimension 101, > 200 000 simplexes

Fréquence de test	Erreur simplexe (abs/rel)	Erreur matrice (abs/rel)
1 000	$10^{-15} - 0 / 10^{-17} - 0$	$10^{-9} - 10^{-11} / 10^{-13} - 10^{-15}$
10 000	$10^{-15} - 0 / 10^{-17} - 0$	$10^{-9} - 10^{-11} / 10^{-13} - 10^{-14}$
100 000	0 / 0	$10^{-10} - 10^{-11} / 10^{-14}$

Problème d'arc 2 : $K_1(10^{-2})$, dimension 82, > 100 000 simplexes

Fréquence de test	Erreur simplexe (abs/rel)	Erreur matrice (abs/rel)
1 000	$10^{-15} - 0 / 10^{-17} - 0$	$10^{-9} - 10^{-11} / 10^{-13} - 10^{-15}$
10 000	$10^{-17} - 0 / 10^{-19} - 0$	$10^{-9} - 10^{-11} / 10^{-13} - 10^{-15}$
100 000	0 / 0	$10^{-10} / 10^{-14}$

On constate que l'erreur sur les simplexes est numériquement négligeable, et parfois même indétectable en double précision. Quant à l'erreur sur l'inverse de la matrice d'étiquetage, elle semble raisonnable, en particulier pour les deux problèmes de dimension élevée, compte tenu du fait que le calcul de cette matrice implique normalement un appel de l'homotopie par sommet, en plus de l'inversion. Finalement, il semble donc qu'il n'y ait pas lieu de trop s'inquiéter à propos de ces erreurs survenant lors des mises à jour.

Propriété de convergence

Pour une homotopie H multivaluée, on a la propriété de convergence suivante.

THÉORÈME 6 *On considère un algorithme simplicial utilisant une sélection de H comme étiquetage et une triangulation raffinée de $\mathbf{R}^n \times [0, 1[$ (J_3 par exemple). On fait les deux hypothèses suivantes sur le suivi de chemin :*

(i) *toutes les faces générées par l'algorithme restent dans $K \times [0, 1]$, avec K compact.*

(ii) *l'algorithme ne redescend pas en $\lambda = 0$.*

Alors si H est scs et à valeurs convexes compactes, l'algorithme génère une suite (z_i, λ_i) telle que $\lambda_i \rightarrow 1$, et il existe une sous-suite encore notée (z_i, λ_i) qui converge vers $(z, 1)$ tel que $0 \in H(z, 1)$.

Preuve. la preuve vient de [1], chapitre 4, page 56.

Quelques points algorithmiques

On décrit dans la suite quelques améliorations que nous avons tenté d'introduire lors de nos expérimentations numériques avec la méthode simpliciale (de plus amples détails sur l'implémentation du programme Simplicial

sont disponibles en Annexe A). Durant leur élaboration, on a tâché de garder à l'esprit deux hypothèses en accord avec le contexte de ce travail. Tout d'abord, on suppose que l'essentiel du coût numérique total provient des évaluations de l'homotopie, tandis que les opérations de la partie purement simpliciale sont négligeables. Ceci est bien vérifié en pratique à cause de la nature particulière des homotopies considérées (fonctions de tir). Ensuite, comme on s'attend à une faible régularité des problèmes étudiés, on souhaite que ces modifications conservent autant que possible la robustesse de la méthode simpliciale.

2.2 Homotopie de jonction

Pour un algorithme simplicial, l'initialisation du suivi de chemin consiste à trouver une première face complètement étiquetée sur la frontière (typiquement $\lambda = 0$). A partir de cette face on peut alors construire le premier simplexe transversal du chemin, puis effectuer une suite de tests lexicographiques et de pivotages, comme décrit précédemment. Mais il se trouve que l'obtention d'une telle face n'est pas toujours triviale.

Comme cette face est supposée contenir un zéro de l'approximation affine par morceaux de l'homotopie, une idée naturelle est de trouver un zéro de $H(\cdot, 0)$, et d'essayer de construire une face centrée sur ce point. La première étape, trouver un zéro de l'homotopie au début de la continuation, est censée être facile : cela fait justement partie du principe de la continuation.

Cependant, la seconde partie, construire une face centré sur cette solution, ne garantit pas que cette face soit complètement étiquetée. En effet, la face doit contenir un zéro de *l'approximation affine de H sur la face*, ce qui n'est pas la même chose qu'un zéro de H elle-même. En pratique, suivant la régularité de l'homotopie aux environs du zéro de départ, une face suffisamment petite peut se révéler complètement étiquetée (des faces plus grandes signifient en général une approximation moins précise de H , et sont donc moins susceptibles d'être complètement étiquetées). Toutefois, un point ennuyeux est que l'utilisation d'une petite triangulation mène à un grand nombre de simplexes parcourus, d'où un grand temps de calcul au final. Plus important, Cette stratégie ne fonctionne pas toujours, parfois même des tailles ridiculement faibles ne donnent pas de face complètement étiquetée. Nous aimerions donc une méthode qui permette d'obtenir une face complètement étiquetée pour une taille de triangulation donnée. Pour l'initialisation du chemin l'on cherche une face en $\lambda = 0$, mais l'on verra que plus généralement, trouver une face complètement étiquetée à un certain niveau λ_j peut être également utile.

2.2.1 Principe

Le principe d'une homotopie de jonction suit l'idée (cf par exemple [2], Lemme 13.2.6 page 183) que si l'étiquetage est affine, alors il est bien plus aisé de trouver une face complètement étiquetée. Plus précisément, on considère une homotopie H_j qui se déroule à un λ_j fixé par rapport au suivi de chemin principal, et qui utilise son propre paramètre $\lambda' \in [0, 1]$. On essaye de connecter une application $z \mapsto A(z - z_j)$ à $H(\cdot, \lambda_j)$, avec la matrice A choisie de manière à fournir une approximation affine acceptable de $H(\cdot, \lambda_j)$ près du point de départ z_j de la jonction. Dans le cas de l'initialisation, pour $\lambda = 0$, ce point de départ est typiquement la solution z_0 obtenue par un solveur non-linéaire appliqué à $H(\cdot, 0) = 0$. La différence avec l'homotopie principale est que construire une première face complètement étiquetée pour l'étiquetage affine est bien plus simple, comme il coïncide avec son approximation sur la face. Ainsi si A est non singulière, la face centrée sur z_j est complètement étiquetée. La taille de triangulation pour cette homotopie de jonction est la taille δ souhaitée pour la face complètement étiquetée, à l'exception de la taille suivant le paramètre homotopique λ' , dont on reparle plus bas.

Cette homotopie intermédiaire est censée être rapide, ne requérant qu'un faible nombre de simplexes, pourvu que nous ayons une matrice A acceptable. Dans cette optique, on essaye de minimiser le nombre d'évaluations de H en utilisant une triangulation "abrupte", avec une taille de 1 suivant λ' . Ainsi, les sommets de la jonction vérifient soit $\lambda' = 0$ ou $\lambda' = 1$ (les deux correspondant toujours à λ_j sur le chemin principal), et on définit l'étiquetage pour H_j :

$$\begin{cases} H_j(z, \lambda') = A(z - z_j) & \text{si } \lambda' = 0 \\ H_j(z, \lambda') = H(z, \lambda_j) & \text{si } \lambda' = 1. \end{cases}$$

Seuls les simplexes au niveau $\lambda' = 1$ requièrent une évaluation de H , tandis que les autres en $\lambda' = 0$ ne mettent en jeu que certaines opérations matrices/vecteurs.

Il est bien sûr possible de fixer un pas plus petit suivant λ' , et d'utiliser un étiquetage de la forme"

$$H_j(z, \lambda') = (1 - \lambda') A(z - z_j) + \lambda' H(z, \lambda_j)$$

Toutefois, ceci nécessite davantage d'appels à H , ce que l'on souhaite éviter puisque l'essentiel du coût numérique provient de ces appels (ceci vient du fait que H est dans notre cas une fonction de tir, avec intégration d'un (IVP)). De même, utiliser le premier étiquetage avec un pas plus petit que 1 pour λ' a été abandonné, pour les mêmes raisons. En fait, comme ils conduisent à des jonctions plus longues, la seule justification de ces

étiquetages moins abrupts serait de permettre des jonctions plus difficiles, qui échoueraient autrement. Le cas des jonctions difficiles a finalement plutôt été résolu en amont, en imposant certaines restrictions sur l'utilisation des jonctions dans le cas de la triangulation adaptative, comme on le verra plus tard.

2.2.2 Applications

Comme on l'a dit, les homotopies de jonction ont d'abord été introduites pour l'initialisation du suivi de chemin, c'est-à-dire trouver la première face complètement étiquetée.

Une autre possibilité offerte par ces jonctions est de changer la taille de la triangulation au cours du suivi de chemin. L'idée de changer la taille des simplexes le long du chemin de zéros pour améliorer le suivi n'est pas nouvelle. Elle est en particulier illustrée par les triangulations raffinées, qui utilisent typiquement de grands simplexes au début pour une progression rapide, puis de plus en plus petits pour une bonne précision de la solution à la convergence. Dans ce cas, la taille variable des simplexes est une propriété structurelle de la triangulation, qui conduit souvent à des règles de pivotage assez complexes, les changements de "niveaux" devant être pris en compte. Dans des circonstances favorables ces triangulations peuvent donner des résultats impressionnants (voir 1.1.1 page 6). Cependant, leur utilisation sur les problèmes de contrôle optimal que nous étudions s'est révélée assez décevante, c'est pourquoi nous avons tenté autre chose. A la base, on choisit d'interrompre le suivi de chemin à certains points, et on essaye de déterminer une "meilleure" taille de triangulation (cette étape est détaillée plus loin dans 2.3, page 36). On procède alors à une homotopie de jonction pour obtenir une première face complètement étiquetée pour cette nouvelle triangulation, avant de reprendre le suivi de chemin à partir de là.

Pour finir, une troisième application des jonctions peut être le raffinement de solution, où on essaye de trouver une meilleure solution que celle obtenue à la fin du suivi de chemin. Une façon possible de procéder est de tenter une suite de jonctions à la fin du chemin, avec des tailles de triangulation décroissantes (voir 2.4.2 page 52 pour plus de détails).

Remarque : il est à noter que l'implémentation des homotopies de jonction est telle que la triangulation utilisée pour la jonction peut être d'un type différent de celui de la triangulation pour le suivi de chemin principal. La seule restriction est qu'une jonction doit utiliser une triangulation uniforme, puisque son objectif est justement de fournir une face complètement étiquetée pour une taille donnée. Par exemple, si on choisit une triangulation uniforme (comme K_1 , J_1 ou D_1), les jonctions utiliseront le même type

de triangulation, pour faciliter les raccordements entre jonctions et homotopie principale. Par contre, la jonction d'initialisation en $\lambda = 0$ pour un suivi avec une triangulation raffinée (J_3 ou J_4) doit utiliser une triangulation uniforme pour trouver la première face. Ceci implique une étape de conversion de la face obtenue par la jonction vers le type de triangulation du chemin principal, ce qui fort heureusement se fait assez simplement en pratique.

2.2.3 Formulations pratiques

On rappelle que nous cherchons pour la jonction une approximation de $H(\cdot, \lambda_j)$ sous la forme $z \mapsto A(z - z_j)$. La première idée que nous avons implémentée est tout simplement de prendre pour A une approximation de la Jacobienne de $H(\cdot, \lambda_j)$, calculée grossièrement par différences finies centrées.

Remarques : Il existe d'autres possibilités pour obtenir directement la Jacobienne de H , comme utiliser l'équation variationnelle du problème de tir, que l'on rappelle brièvement. On considère le problème à valeur initiale

$$(IVP) \begin{cases} \dot{y} = \phi(y) \\ y(0) = \beta \end{cases}$$

On suppose que ϕ est continue, et continûment différentiable par rapport à y . Si $\hat{y}(\cdot, \beta)$ est une solution de (IVP), alors $\frac{d\hat{y}}{d\beta}$ est une solution de

$$(Eq. Var.) \begin{cases} \dot{Y} = \frac{\partial \phi}{\partial y}(y) Y \\ Y(0) = Id \end{cases}$$

Cette propriété peut servir à obtenir les dérivées de la fonction de tir, mais la régularité requise est trop importante dans notre contexte. Ceci dit, il existe également des moyens plus sophistiqués pour calculer ces dérivées, en particulier la "différentiation numérique interne" (IND) présentée par Bock dans [9]. Toutefois, comme on s'attend à une faible régularité des problèmes étudiés, notre objectif est en fait de trouver une formulation qui ne s'appuie pas sur l'usage des dérivées. C'est pourquoi nous n'avons pas vraiment concentré nos efforts sur un calcul intelligent de ces dérivées, et avons simplement utilisé les différences finies jusqu'à ce que l'on trouve une meilleure formulation pour les jonctions.

Le principal désavantage de cette première formulation est qu'il faut choisir avec soin le pas de différences finies. Dans notre cas, l'homotopie est une fonction de tir, dont l'évaluation requiert l'intégration d'un problème à valeur initiale. Cette intégration donne lieu à une certaine erreur, suivant le type d'intégrateur utilisé, ainsi que le nombre de pas (à pas fixe) ou les tolérances (à pas variable). Si le pas est trop grand, l'approximation de la

Jacobienne peut être très mauvaise, voire inutilisable. D'un autre côté, un pas trop petit par rapport à la précision de l'intégration peut produire du "bruit numérique" tout aussi gênant en différenciant les erreurs.

De plus, il y a la question de la sensibilité de l'homotopie par rapport à certaines composantes de l'inconnue. Lié avec la mise à l'échelle de l'inconnue z , ceci peut rendre plus adaptée l'utilisation de différents pas $h_i, i = 1, \dots, n$ au lieu d'un pas uniforme h pour toutes les dimensions. De même, la taille de la face courante peut également intervenir, comme on voudrait que l'approximation soit valide au voisinage de la face de départ de la jonction (les jonctions sont censées trouver une face complètement étiquetée près de leur point de départ, sans trop s'éloigner). On essaye alors de modifier l'idée précédente pour refléter ce lien entre le pas de différence finies et la taille de triangulation. On fixe ainsi le pas en divisant la taille δ des simplexes par un certain coefficient k , avec

$$h_i = \frac{\delta_i}{k}, \forall i \in [1..n].$$

Les expérimentations numériques suggèrent que des valeurs de k entre 10 et 1000 donnent généralement les meilleurs résultats. Cette variante est un peu meilleure que la précédente, mais souffre toujours de la sensibilité au pas (ie le choix de k ici), ce qui semble une conséquence inévitable de l'utilisation des différences finies. Quoi qu'il en soit, on s'attend de toute façon à rencontrer des fonctions de tir qui risquent de ne pas être différentiables.

C'est pourquoi on souhaite trouver une autre méthode, qui ne fasse pas appel aux dérivées. Une possibilité est d'utiliser directement l'approximation de $H(\cdot, \lambda_j)$ par interpolation sur les sommets de la face de départ de la jonction. Ceci introduit tout d'abord une légère modification par rapport à la formulation précédente du type $z \mapsto A(z - z_j)$, car on préfère ici travailler avec les coordonnées barycentriques de z et z_j dans la face de départ $[v_1, \dots, v_{n+1}]$. On pose donc

$$F = \begin{pmatrix} 1 & \dots & 1 \\ v_1 & \dots & v_{n+1} \end{pmatrix},$$

et les coordonnées barycentriques de z et z_j (notées \hat{z} et \hat{z}_j) sont données par

$$\hat{z} = F^{-1} \begin{pmatrix} 1 \\ z \end{pmatrix} \quad \text{et} \quad \hat{z}_j = F^{-1} \begin{pmatrix} 1 \\ z_j \end{pmatrix}.$$

On rappelle la matrice d'étiquetage de la face $[v_1, \dots, v_{n+1}]$

$$L = \begin{pmatrix} H(v_1, \lambda_j) & \dots & H(v_{n+1}, \lambda_j) \end{pmatrix}.$$

et on définit l'étiquetage affine de la jonction par

$$z \mapsto L(\hat{z} - \hat{z}_j).$$

Cette expression s'annule en $z = z_j$, et coïncide avec $H(\cdot, \lambda_j)$ sur les sommets v_i , $\forall i \in [1..n+1]$, et nous pouvons l'utiliser pour la jonction. L'amélioration majeure de cette seconde formulation est de ne pas utiliser de dérivées, comme on s'attend à une faible régularité des homotopies. Un autre avantage (quoique mineur) est que l'initialisation de la fonction ne requiert que n appels de l'homotopie, contre $2n$ auparavant.

Résumé :

- *Première formulation : $A \approx \text{Jac}(H(\cdot, \lambda_j))$ par différences finies, avec un pas h uniforme ou proportionnel à la taille δ .*
- *Seconde formulation : utilise l'approximation affine de $H(\cdot, \lambda_j)$ sur la face de départ, pas de dérivées.*

Une comparaison numérique de ces formulations sur les problèmes de transferts orbitaux est détaillée en 3.4.2, page 81. Globalement, les tests indiquent que la seconde formulation est effectivement plus stable que la première. Elle est donc le mode de jonction utilisé par défaut dans le code `Simplicial`.

2.3 Triangulation adaptative

2.3.1 Principe

Une des raisons généralement admises de la lenteur des algorithmes simpliciaux par rapport à la continuation différentielle est la difficulté d'adapter la triangulation au chemin suivi. Contrairement aux méthodes Prédicteur-Correcteur, où la direction de recherche et la pas de progression sont calculés automatiquement pour suivre au mieux le chemin, les méthodes simpliciales sont intrinsèquement privées de tels mécanismes, de par la taille fixée des simplexes.

Certes, les triangulations raffinées sont une façon élégante d'introduire des simplexes de taille variable, mais cette taille ne prend pas en compte le chemin réellement suivi, et n'est donc pas nécessairement bien adaptée. De fait, nous avons constaté que les problèmes que nous étudions se prêtent mal à ce type de triangulation. En bref, la sensibilité de l'homotopie empêche l'usage de grands simplexes au démarrage, tandis que de petits simplexes peuvent rendre l'algorithme plus vulnérable aux difficultés numériques si la régularité du chemin est faible vers la fin. A part dans le cas de l'exemple simple du 1.1.1, la plupart du temps nous ne sommes pas parvenus à utiliser

de façon satisfaisante les triangulations raffinées (J_3, J_4) .

De plus, même avec une K_1 basique, que l'on a finalement retenue pour la plupart des expérimentations en raison de la fiabilité dont elle a fait preuve, le choix de la taille se révèle une question non triviale. Trop grande, et l'approximation est trop loin du chemin de zéros pour effectuer avec succès le suivi de chemin. Trop petite, et le suivi est extrêmement long. Et bien sûr, il n'est guère efficace de devoir essayer plusieurs suivis avec diverses tailles juste pour trouver la mieux adaptée à un problème particulier.

On souhaiterait donc adapter dynamiquement la triangulation au suivi de chemin, pour améliorer la rapidité et/ou la précision de la continuation. Pour des raisons pratiques, on choisit de n'effectuer cette étape d'adaptation que lorsque l'algorithme atteint un certain "niveau" $\lambda = \lambda_i$, c'est-à-dire quand on a une face complètement étiquetée dont tous les sommets appartiennent à $\mathbf{R}^n \times \lambda_i$. A ce moment-là on calcule la nouvelle taille (voir 2.3.2), d'après la taille courante et le chemin parcouru depuis le niveau précédent $\lambda = \lambda_{i-1}$, comme décrit dans la suite. On procède ensuite à une homotopie de jonction (cf 2.2) pour obtenir une face complètement étiquetée pour cette nouvelle taille de triangulation. Le suivi de chemin peut alors continuer à partir de cette face avec la nouvelle triangulation, jusqu'au prochain niveau λ_{i+1} . Ces étapes adaptatives étaient à l'origine effectuées dans les deux sens, c'est-à-dire également quand on revenait au niveau précédent $\lambda = \lambda_{i-1}$. Cependant, il se trouve que dans tous les problèmes étudiés ici les chemins semblent être monotones, avec un λ croissant. Dans ce cas, un retour en arrière indique simplement que l'on s'écarte localement du chemin, et doit plutôt être simplement ignoré, pour ne pas perturber davantage le suivi. Le plus souvent, ces "redescentes" ne durent de toute façon que le temps de quelques simplexes, après quoi les choses rentrent dans l'ordre et le suivi se poursuit à λ croissant.

*Remarque : cette procédure, bien qu'assez simple dans son principe, est en pratique un peu délicate et requiert une implémentation soignée pour tenir compte de certains cas particuliers. Il s'agit en fait de la partie la plus technique du code *Simplicial*, et une description détaillée ligne par ligne serait extrêmement fastidieuse (et d'un intérêt discutable...) pour le lecteur. Toutefois, on souhaite mentionner ici rapidement certains points qui nous semblent dignes d'intérêt.*

- Tout d'abord, il est parfois possible d'éviter complètement l'homotopie de jonction, et de juste redémarrer le suivi avec la nouvelle triangulation. Ceci se produit lorsque la face de départ de la jonction est en fait déjà complètement étiquetée pour l'homotopie principale $H(\cdot, \lambda_i)$ (la raison d'être de la jonction étant justement de trouver une telle face). On teste

donc cette possibilité juste avant de commencer la jonction, et on la supprime le cas échéant. Cette situation apparaît réellement en pratique, même si elle ne semble pas si fréquente que cela. Comme le test consiste simplement à vérifier si une matrice est lexicographiquement positive, et a le même coût que l'initialisation de la jonction (et nettement moins que celui d'une jonction complète), je pense qu'il est plus avantageux de traiter ce cas.

Un autre cas de figure est celui où le calcul de la nouvelle triangulation aboutit en fait à garder la même taille jusqu'au prochain niveau. Visiblement il n'y a alors pas besoin d'une jonction, le suivi doit juste continuer normalement. Toutefois, ceci doit être détecté séparément, car on ne rentre pas dans le cas précédent : on avait bien sûr déjà une face complètement étiquetée avant la jonction, mais la face de départ de la jonction est elle construite avec (z_i^*, λ_i) (le zéro courant de l'approximation H_T de H) comme isobarycentre. Ainsi les sommets de cette face ne coïncident pas avec les précédents, même si la taille reste inchangée (en théorie, ils *pourraient* coïncider, si z_i^* était *exactement* l'isobarycentre de la face...). Avec un brin de malchance, il est tout à fait possible que cette face de départ ne soit pas complètement étiquetée pour $H(\cdot, \lambda_i)$, comme cette propriété dépend de l'approximation sur la face. On traite donc ce cas à part, pour éviter le risque d'une jonction et d'un pseudo changement de taille inutiles. L'économie est non négligeable, car cette situation est assez fréquente en pratique, en raison de l'inertie incorporée au calcul de la nouvelle taille (pour limiter les modifications intempestifs, voir plus loin).

Note : on rappelle juste qu'en cas d'augmentation de la taille, la nouvelle face n'a pas de raisons particulières d'être complètement étiquetée, même si elle contient une telle face. On ne peut donc pas ici se passer de la jonction directement (on peut toutefois encore se retrouver dans le premier cas mentionné plus haut).

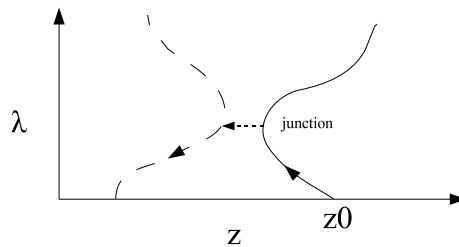
- Un autre point est que, malheureusement, il n'est pas garanti qu'une jonction converge, et ce qui est plus important, en un temps raisonnable. Etant donné qu'un des objectifs de la triangulation adaptative est d'accélérer le suivi de chemin, il semble évident que le coût des jonctions doit être contrôlé d'une façon ou d'une autre. Sinon, on risque de passer plus de temps à effectuer les jonctions que ce qu'on gagne sur le chemin principal avec l'adaptatif. Cette inquiétude est en pratique tout à fait justifiée, comme le montrent les expériences sur les problèmes de transferts orbitaux au 3.4, page 73.

Je n'ai pas trouvé de panacée pour cette difficulté, et ce qui m'a semblé le plus efficace est de borner relativement bas le nombre de simplexes autorisés pour les jonctions, sachant qu'elles sont supposées être courtes de

toute façon. Si la jonction ne parvient pas à s'effectuer dans cette limite (ou si elle échoue pour tout autre raison d'ailleurs), on annule simplement le changement de taille et on garde l'actuelle jusqu'au prochain niveau. Cette limitation devrait être fixée de telle sorte qu'elle ne bride pas la complétion des jonctions dans la plupart de cas, comme on veut simplement régler le cas isolé des jonctions à problèmes à un coût raisonnable. Les tests numériques suggèrent que cette approche fonctionne bien avec une limite de 100 simplexes par exemple. Accessoirement, fixer une limite très basse (50 ou moins) cause en fait plus de mal que de bien, en bloquant la plupart des jonctions, ce qui dégrade fortement tout le mécanisme adaptatif. D'un autre côté, une contrainte trop lâche (au-delà de 500) est peu efficace, car elle permet aux jonctions ratées de "gaspiller" davantage de simplexes.

Une idée un peu plus sophistiquée est d'annuler une jonction avant la limite si sa progression est insuffisante (par exemple si le paramètre λ' n'a pas atteint une certaine valeur après un certain nombre de simplexes). Cependant, le choix pratique de ces valeurs semble en pratique très hasardeux, et les tests réalisés sur cette idée n'ont pas été très convaincants pour l'instant.

- Un phénomène intéressant (mais heureusement rare) qui se produit parfois est une "redescente" du suivi en $\lambda = 0$ après une jonction. Une explication probable est alors l'existence de deux chemins de zéros distincts, proches l'un de l'autre à un certain niveau λ_i . Une jonction qui se produit à cet endroit peut dévier et aboutir à une face complètement étiquetée correspondant à l'autre chemin. L'algorithme peut alors suivre celui-ci jusqu'en $\lambda = 0$, ce qui accessoirement donne un autre zéro de $H(\cdot, 0)$. Si on trace les projections du chemin (avec λ en ordonnée), ceci est indiqué par une sorte de "saut" horizontal au niveau de la jonction.



Une façon simple et efficace de prévenir ce type de comportement est tout simplement de limiter le nombre de simplexes des jonctions, ce qui est en fait déjà réalisé pour des raisons de performance, comme indiqué précédemment. Cependant, on doit alors abandonner le changement de taille, et cette solution atteint de toute façon ses limites si les deux branches sont très proches (peut-être par exemple si l'on est au voisinage d'une bifurcation). Dans ce

cas, empêcher les “sauts” peut demander une limite si basse que cela revient à bloquer la plupart des jonctions le long du chemin...

Incidentement, ces sauts ne mènent pas nécessairement ensuite en $\lambda = 0$, car le suivi peut très bien continuer sur la deuxième branche et finalement converger. Un exemple de ce comportement est décrit au chapitre 3, plus précisément au 3.4 (page 73).

- Une dernière remarque concernant le changement de triangulation est que l’on perd théoriquement la propriété selon laquelle un algorithme simplicial ne boucle pas. Toutefois, nous n’avons pas rencontré ce cas de figure lors des nombreux tests effectués, et ce risque semble donc assez hypothétique.

2.3.2 Calcul de la nouvelle taille

Supposons donc que nous ayons atteint un certain niveau λ_i , comment pouvons-nous adapter la taille courante δ pour améliorer le suivi de chemin ? On décrit ici les deux mécanismes d’adaptation que nous utilisons en pratique.

Le premier, que l’on appelle “contrôle de déviation”, se sert de la norme de l’homotopie H à la solution courante z_i pour mesurer la justesse du suivi de chemin. En comparant cette norme à un certain seuil de tolérance, on détermine s’il est approprié ou non d’étendre ou réduire uniformément la taille de triangulation.

Le second, appelé “déformation anisotrope”, est davantage en rapport avec la direction globale prise par le chemin de zéros. On regarde le poids relatif de chaque dimension dans la progression le long du chemin depuis le niveau précédent, et on peut alors décider d’augmenter ou de diminuer la taille séparément pour chaque dimension (d’où le nom du procédé). Les expérimentations numériques indiquent que la première approche est mieux adaptée au gain de performance, tandis que la seconde est plus orientée vers une meilleure précision. On essaye naturellement de combiner au mieux ces deux aspects, ce que l’on désigne par “mode adaptatif complet”.

Remarque : dans ces deux mécanismes, la dernière composante de la taille, δ_{n+1} , qui correspond au paramètre homotopique λ , est traité à part. La raison en est que ce pas a une forte influence sur le suivi de chemin global, car il donne la “hauteur” du prochain niveau, auquel le prochain changement de triangulation peut se produire. C’est pourquoi nous sommes plus restrictifs lorsqu’il s’agit de modifier ce pas, pour garder un meilleur contrôle du mode adaptatif. De plus amples détails sur ce cas particulier sont donnés dans la suite, en particulier dans la description du mode adaptatif complet.

Contrôle de déviation

A la base, la première méthode que l'on utilise consiste à adapter la taille de triangulation pour le prochain niveau en fonction de la qualité du suivi au niveau courant λ_i . Si l'on estime que le suivi se passe bien, on peut raisonnablement augmenter la taille pour améliorer la performance. Au contraire, si l'on semble s'éloigner du chemin de zéros, réduire la taille peut aider à revenir sur les rails. La question est, bien sûr, comment évaluer la précision du suivi de chemin en l'état courant.

Une façon simple d'avoir une telle indication est de tester la validité de l'approximation affine de l'homotopie sur la face courante. Plus précisément, on considère la solution courante (z^i, λ_i) qui est censée être un zéro de H_T , et on évalue la véritable homotopie H en ce point. Si la norme de $H(z^i, \lambda_i)$ est suffisamment basse, le suivi est considéré assez précis, et l'on peut augmenter la taille. Si la norme est trop élevée, le suivi est probablement trop grossier, et l'on devrait réduire la taille pour se rapprocher du chemin de zéros. Entre les deux, on ne fait rien, la situation n'étant ni mauvaise au point de requérir un ralentissement, ni assez bonne pour autoriser une accélération sûre. Ce dernier point est important, car le processus doit avoir une sorte d'inertie pour éviter par exemple une suite d'expansion/réduction si les normes testées restent au voisinage de la tolérance.

Une remarque préventive devrait être faite ici concernant l'idée de réduire la taille pour améliorer la précision. Ceci est raisonnablement vrai en général, d'un point de vue qualitatif. Toutefois, il n'y pas en pratique de garantie que l'on puisse garder la norme de $H(z^i, \lambda_i)$ en-dessous de seuils arbitrairement bas, même avec des tailles extrêmement petites. De fait, le terme de "tolérance" peut être un peu trompeur ici, car il suggère un contrôle strict de la norme de l'homotopie. Il s'agit en pratique plutôt d'une valeur indicative "acceptable" pour la norme, que l'on essaye (de manière assez lâche) de respecter. Voyons maintenant d'un peu plus près cette histoire de déviation.

On utilise ici les notations suivantes :

- z^i désigne le noeud du chemin de zéro au niveau courant, qui est donc le zéro de $H_T(\cdot, \lambda_i)$ contenu dans la face complètement étiquetée du niveau λ_i .
- h_{tol} est la tolérance (indicative) par rapport à laquelle on détermine l'éventuelle réduction ou expansion de la taille de triangulation.

Algorithme :

Si $|H(z^i, \lambda_i)| < h_{tol} \times 10^{-1}$

Alors $\delta_j \leftarrow \delta_j \times 2$, $j \in [1..n + 1]$

Simon si $|H(z^i, \lambda_i)| > h_{tol} \times 10$

alors $\delta_j \leftarrow \delta_j/2$, $j \in [1..n + 1]$

Simon si $|H(z^i, \lambda_i)| > h_{tol}$

Alors $\delta_j \leftarrow \delta_j/2$, $j \in [1..n]$

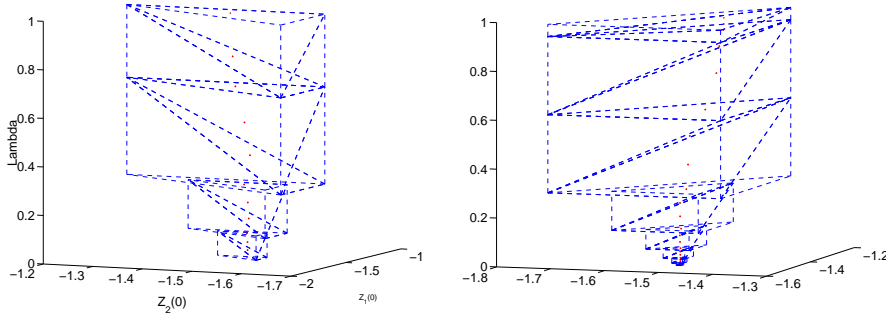
Remarques :

- Le pas δ_{n+1} est traité différemment, car il a un impact sur l'ensemble du mode adaptatif. A part ce cas particulier, toutes les autres dimensions sont traitées de la même façon, contrairement à la déformation anisotrope décrite ci-après.

- Les essais numériques tendent à indiquer que ce procédé est plus approprié au gain de performance, par une augmentation contrôlée de la taille de triangulation. A l'inverse, vouloir satisfaire une tolérance trop basse sur la norme mène rapidement à une taille très faible, avec un coût prohibitif en termes de simplexes parcourus.

- En raison de la mise à l'échelle appliquée à z et $y = (x, p)$, l'ordre de grandeur des valeurs de H le long du chemin n'est généralement pas trop étendu en pratique. En conséquence, la plage de valeurs pertinentes pour le paramètre h_{tol} est heureusement également limitée. Typiquement, le réglage par défaut à 10^{-1} a été choisi pour obtenir un équilibre satisfaisant entre le gain de performance et la perte de précision. Une valeur de 1 correspond à un suivi plus agressif, et peut donner une plus grande rapidité, au prix de la stabilité. D'un autre côté, une tolérance de 10^{-2} impose un comportement plus prudent, ce qui peut avoir un coût élevé. D'après nos expérimentations, choisir une tolérance au-delà de ces valeurs est soit trop laxiste ou restrictif, et donne des résultats médiocres dans les deux cas, avec un suivi respectivement très instable ou très lent.

Voici un exemple d'application du contrôle de la déviation, appliqué à l'exemple simple du 1.1.1. Sur ce problème très facile, on voit une augmentation régulière de la taille de la triangulation le long du chemin, jusqu'à atteindre la convergence en $\lambda = 1$ (on note que le dernier niveau est redimensionné suivant λ pour ne pas dépasser $\lambda = 1$).



Contrôle de déviation sur le problème démo - taille initiale 10^{-1} et 10^{-2}

Déformation anisotrope

Le second mécanisme adaptatif tente de prendre en compte l'anisotropie du chemin suivant les différentes composantes de l'inconnue z . Pour la j^{eme} dimension, si la taille correspondante δ_j est trop grande, les sommets risquent de s'éloigner du chemin de zéros. Cela n'est pas nécessairement néfaste, par exemple si l'on ne perd qu'un peu de précision tout en ayant un suivi rapide. Cependant, si l'homotopie est sensible par rapport à cette composante particulière de z , ces sommets éloignés peuvent obtenir des étiquetages très hasardeux, voire même causer des erreurs numériques lors de l'évaluation de l'homotopie. Cette difficulté est bien réelle dans notre cas, où l'homotopie est une fonction de tir et z l'inconnue du problème à valeur initiale. A cause de l'intégration numérique effectuée pour évaluer l'homotopie, des variations relativement faibles de l'état adjoint initial peuvent conduire à des trajectoires complètement différentes, ou pire quitter le domaine de définition de la fonction de tir. Incidemment, ce phénomène peut être encore amplifié par la mise à l'échelle de z , qui doit parfois être relâchée pour réduire la sensibilité de la fonction de tir (c'est en particulier le cas pour le problème de transfert orbital étudié au chapitre 3, où l'intégration peut être très longue).

Clairement, rétrécir la taille de triangulation rapproche mécaniquement les sommets du chemin de zéros, mais implique également davantage de simplexes, et un suivi plus long. C'est le même problème que si l'on choisit la taille initiale trop petite au départ, où on n'a aucune idée a priori de l'allure du chemin. C'est pourquoi si la taille δ_j est trop faible par rapport à la progression du chemin suivant la j^{eme} dimension, on est inutilement proche du chemin pour cette dimension. Dans ce cas, si on augmente légèrement δ_j , on doit pouvoir progresser plus rapidement sans trop s'éloigner du chemin.

L'un dans l'autre, il s'agit d'une approche assez qualitative : on essaye de contracter et/ou dilater la triangulation suivant certaines dimensions pour mieux correspondre au chemin de zéros, et on espère ainsi améliorer à la fois

la précision et la rapidité du suivi. On donne ci-dessous une vue schématique de cette déformation anisotrope en dimension 2. On considère un chemin “déséquilibré”, qui progresse nettement plus suivant la deuxième dimension que la première, donc l’idée générale est d’accroître la taille δ_2 et de réduire δ_1 . On représente d’abord les simplexes transversaux qui peuvent être traversés par le suivi avec une taille isotrope ($\delta_1 = \delta_2$). Les deux graphes suivants illustrent l’effet attendu pour δ_2 doublé et δ_1 réduit de moitié respectivement, tandis que le dernier graphe combine les deux modifications.

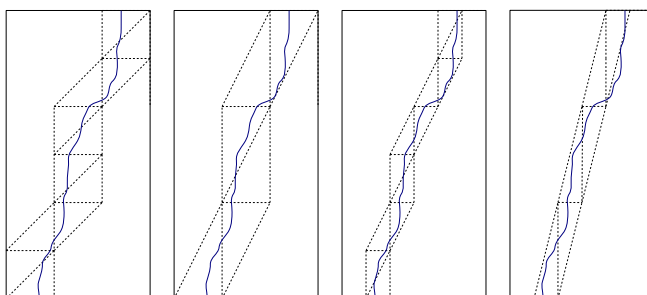


Illustration schématique du mécanisme de déformation anisotrope
 $\delta_1 = \delta_2$, $\delta_2 \leftarrow \delta_2 * 2$, $\delta_1 \leftarrow \delta_1 / 2$, $(\delta_1, \delta_2) \leftarrow (\delta_1 / 2, \delta_2 * 2)$

On voudrait donc que les réductions de taille permettent d’éviter les sommets trop éloignés du chemin, tandis que les augmentations réduisent le nombre global de simplexes parcourus. Ces deux aspects sont naturellement antagonistes, et l’on espère pouvoir les combiner de façon à profiter des deux avantages.

En pratique, on utilise l’information sur la progression réalisée depuis le niveau précédent (de λ_{i-1} à λ_i) pour déterminer si certaines composantes de la taille doivent être modifiées pour le prochain niveau. Bien sûr, on s’attend implicitement à ce que le chemin de zéros se comporte plus ou moins de la même façon sur $[\lambda_i, \lambda_{i+1}]$, ce qui explique pourquoi la déformation anisotrope fonctionne généralement mieux avec des niveaux relativement petits.

De plus, il faut être prudent avec ces modifications de taille : des changements brutaux peuvent avoir un effet désastreux, passant d’une taille trop grande à une trop faible, et vice-versa. C’est pourquoi nous avons finalement limité les changements à un facteur entre 0.5 et 2, et même abandonné les valeurs intermédiaires : on se contente de doubler ou réduire de moitié la taille δ_j quand on l’estime judicieux, et on la laisse inchangée sinon. Ce dernier point est important, car il introduit une forme d’inertie, comme pour le contrôle de déviation décrit précédemment. Cet aspect est nécessaire pour les mêmes raisons, en particulier pour éviter un phénomène de yoyo, lorsqu’une taille δ_j subit une série d’augmentations/diminutions après qu’une valeur bien adaptée au chemin ait été trouvée.

On utilise ici les notations suivantes :

- δ est la taille de triangulation, un vecteur de dimension $n + 1$; δ_{n+1} est donc le pas suivant λ
- z^{i-1} et z^i désignent les noeuds du chemin aux niveaux précédent et courant. Cela signifie que z^{i-1} est un zéro de $H_T(\cdot, \lambda_{i-1})$ et z^i un zéro de $H_T(\cdot, \lambda_i)$
- Δ est la progression depuis le dernier niveau, remis à l'échelle de la taille, ie

$$\Delta_j = \frac{|z_j^i - z_j^{i-1}|}{\delta_j}, \quad j \in [1..n]$$

- $progress_{up} > 1 > progress_{low} > progress_{toolow}$ sont trois seuils dont les valeurs par défaut sont 2, 0.2 et 0.1. Pour déterminer si la taille δ_i doit être augmentée, réduite ou laissée inchangée, on compare le ratio “progression suivant j-ème dimension divisée par la progression moyenne” à ces seuils, comme indiqué ci-dessous.

Algorithme : pour chaque dimension j , $j \in [1..n]$

Si $\Delta_j \geq \frac{\sum_k \Delta_k}{n} \times progress_{up}$

Alors $\delta_j \leftarrow \delta_j \times 2$

Sinon si $\Delta_j \leq \frac{\sum_k \Delta_k}{n} \times progress_{low}$ et $\Delta_j \geq \frac{\sum_k \Delta_k}{n} \times progress_{toolow}$

Alors $\delta_j \leftarrow \delta_j / 2$

Suivant le principe expliqué ci-dessus, $progress_{up}$ devrait indiquer une progression clairement supérieure à la moyenne, et $progress_{low}$ une progression nettement inférieure à la moyenne. De plus, l'écart entre ces deux seuils délimite une zone d'inertie, dans laquelle on ne modifie pas la taille. Cette inertie est nécessaire pour une meilleure stabilité de la déformation, comme pour le contrôle de déviation vu auparavant.

Le troisième seuil $progress_{toolow}$ a été introduit spécifiquement pour empêcher le cas de figure ennuyeux où une taille δ_j tend vers 0 quand il n'y a presque plus de progression suivant une certaine dimension j (ce qui correspond à un chemin presque “vertical” sur représentation de la j-ème composante). Ce type de situation se rencontre effectivement en pratique, ce qui nous a conduit à cette mesure préventive.

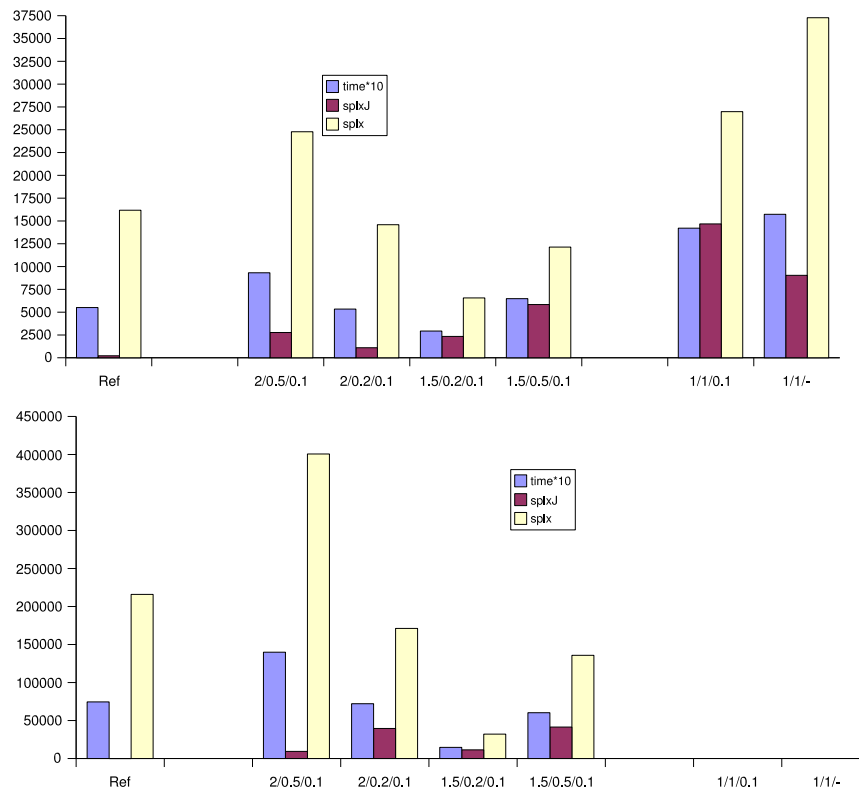
Remarques :

- La composante δ_{n+1} n'est ici pas modifiée, pour éviter d'interférer avec le contrôle de déviation (qui lui peut modifier ce pas), et garder un certain

contrôle sur l'évolution de ce pas.

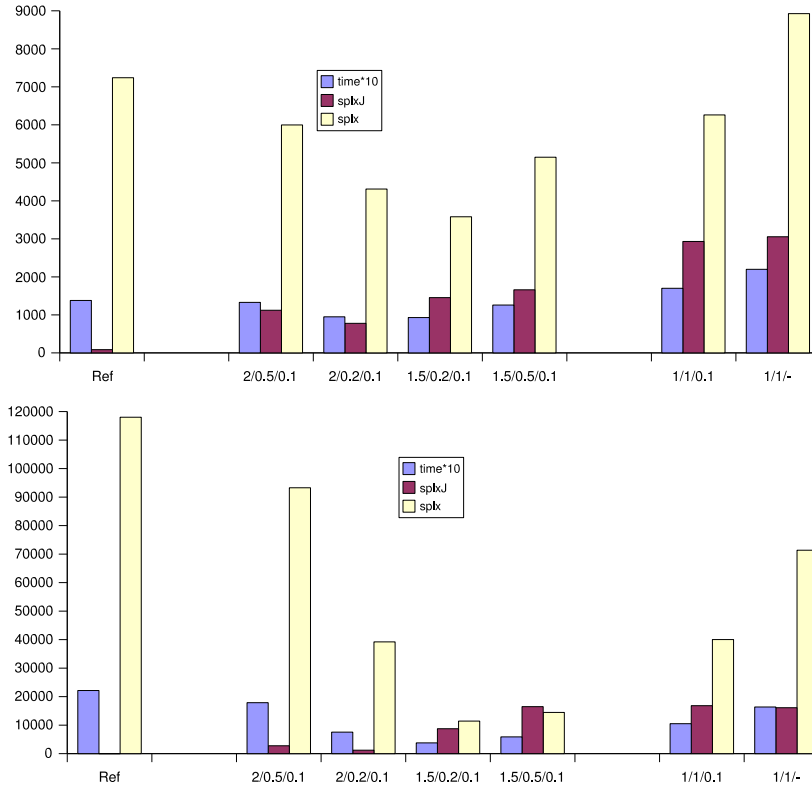
• En ce qui concerne le double objectif de la déformation anisotrope, les schémas ci-dessus décrivent bien sûr une situation idéale. On constate en pratique que ce mécanisme est mieux adapté à l'amélioration de la précision du suivi, plutôt qu'au gain de performance. Ceci n'est pas vraiment contraire, car il complète ainsi à point nommé les effets du contrôle de déviation.

Voici une comparaison de différents réglages pour les trois seuils anisotropes $progress_{up,low,toolow}$, menée sur la formulation BVP discrétisé des problèmes d'arcs singuliers étudiés au chapitre 4. Pour chaque réglage les colonnes indiquent le temps total du suivi en secondes (à un facteur 10 près pour une meilleure lisibilité), le nombre de simplexes de jonction, et le nombre de simplexes du chemin principal. Le premier groupe de colonnes à gauche correspond à la triangulation uniforme de référence. Les quatre groupes au milieu correspondent aux réglages $(2, 0.5, 0.1)$, $(2, 0.2, 0.1)$, et $(1.5, 0.2, 0.1)$, $(1.5, 0.5, 0.1)$. Les deux derniers groupes à droite correspondent aux réglages sans inertie $(1, 1, 0.1)$, $(1, 1, -)$ (qui ne convergent pas pour la $K_1(0.01)$ sur le Problème 1).



Comparaison de réglages anisotropes - Problème 1 (dim 101)

Triangulations $K_1(0.1)$ et $K_1(0.01)$.



Comparaison de réglages anisotropes - Problème 2 (dim 82)
Triangulations $K_1(0.1)$ et $K_1(0.01)$.

Tout d'abord, les mauvais résultats (et même l'échec pour le Problème 1 avec la $K_1(0.01)$) des deux groupes à droite illustrent le besoin d'une plage d'inertie. Le réglage $(1, 1, -)$, qui modifie systématiquement *toutes* les composantes de la taille, conduit à un nombre affreux de simplexes pour les jonctions et le chemin principal (parfois même davantage que pour la référence !), avec un temps très long. Le $(1, 1, 0.1)$ est légèrement moins mauvais, car il limite la réduction abusive des composantes avec une faible progression. Toutefois, l'égalité de $progress_{up}$ et $progress_{low}$ handicape toujours grandement cette configuration.

Si l'on examine les quatre configurations du milieu, on retrouve la même allure générale sur les quatre séries de tests. Le temps d'exécution apparaît meilleur pour les trois à droite, spécialement pour $(1.5, 0.2, 0.1)$, avec un nombre de simplexes principaux particulièrement bas. Cela n'est pas surprenant, dans la mesure où il s'agit du réglage le plus agressif : on augmente les composantes qui ne dépassent que de 50% la progression moyenne, tandis qu'on ne réduit que celles en dessous d'un cinquième de la moyenne. A

l'inverse, le réglage le plus prudent, $(2, 0.5, 0.1)$, est logiquement plus lent, avec un nombre de simplexes principaux assez élevé.

Les deuxième et quatrième configurations, $(2, 0.2, 0.1)$ et $(1.5, 0.5, 0.1)$, correspondent à la plus grande et plus petite zone d'inertie respectivement. Ceci est cohérent avec le nombre de simplexes de jonctions observé, qui tend à être minimal (resp. maximal) pour ces réglages. Les deux donnent des performances moyennes en temps d'exécution, entre les première et troisième (la plus rapide) configurations.

Voici les résultats numériques, avec pour chaque configuration le nombre de simplexes pour le chemin **principal**, les simplexes de **jonction**, le **temps** d'exécution (en secondes) et la **norme** de l'homotopie à la fin du chemin.

Problème 1, $K_1(10^{-1})$ et $K_1(10^{-2})$								
Réglages	Princ.	Jonc.	Temps	$ H $	Princ.	Jonc.	Temps	$ H $
2/0.5/0.1	24782	2774	932	0.122	400673	9453	13991	0.133
2/0.2/0.1	14589	1101	534	0.078	171224	39554	7214	0.098
1.5/0.2/0.1	6566	2345	293	0.077	32156	11441	1473	0.211
1.5/0.5/0.1	12137	5840	648	0.100	135807	41415	6031	0.083
1/1/0.1	26986	14673	1421	0.074	xxx	xxx	xxx	xxx
1/1/-	37277	9041	1573	0.158	xxx	xxx	xxx	xxx

Problème 2, $K_1(10^{-1})$ et $K_1(10^{-2})$								
Réglages	Princ.	Jonc.	Temps	$ H $	Princ.	Jonc.	Temps	$ H $
2/0.5/0.1	5998	1122	133	0.965	93281	2778	1789	0.288
2/0.2/0.1	4311	778	95	0.894	39208	1221	755	0.763
1.5/0.2/0.1	3582	1454	93	1.040	11424	8743	379	1.125
1.5/0.5/0.1	5149	1660	126	0.890	14462	16503	589	1.016
1/1/0.1	6261	2932	170	0.829	40031	16833	1052	0.432
1/1/-	8926	3056	220	0.812	71370	16122	1637	0.317

Bien que le réglage $(1.5, 0.2, 0.1)$ soit le plus rapide, on voit qu'il dégrade assez fortement la norme finale. Au contraire, la configuration plus inertielle $(2, 0.2, 0.1)$ donne de meilleurs résultats, et semble un bon candidat pour la combinaison avec le contrôle de déviation. Accessoirement, les deux configurations sans inertie ont une bonne norme finale (quand elles convergent...), ce qui est simplement une conséquence de la petite taille de simplexes à laquelle elles aboutissent...

Mode adaptatif complet

On souhaite maintenant combiner ces deux mécanismes de la façon la plus constructive possible. Comme mentionné précédemment, le principal bénéfice que l'on peut tirer du contrôle de la déviation est le gain de performance. C'est pourquoi on choisit de configurer la déformation anisotrope avec des réglages assez conservatifs, pour contrebalancer la perte de

précision qui en découle. La formulation obtenue permet généralement un gain intéressant sur le nombre de simplexes suivis, sans trop dégrader la précision finale. En fait, on parvient même souvent à avoir une meilleure norme à la solution qu’avec une triangulation uniforme de base, en plus de l’accélération du suivi.

Résumé :

En pratique, on utilise typiquement les réglages suivants pour le mode adaptatif complet :

- $h_{tol} = 0.1$ pour le contrôle de déviation
- $progress_{up,low,toolow} = (2, 0.2, 0.1)$ pour la déformation anisotrope

Ces valeurs tendent à donner les meilleurs résultats sur les différents problèmes testés, même si de meilleurs réglages peuvent être trouvés pour chaque problème individuellement, avec un peu de pratique.

Pour finir, voici un certain nombre de remarques à propos de la triangulation adaptative dans son ensemble, et en particulier du pas suivant λ .

L’adaptation de la taille est faite pour le prochain niveau, en se basant sur le suivi depuis le niveau précédent. On rappelle que la “hauteur” de ces niveaux est donnée par la dernière composante de la taille, δ_{n+1} . Si ce pas est trop grand, il n’y a que peu de niveaux au cours du chemin, et l’adaptation est trop limitée pour être efficace. A l’inverse, si le pas est trop petit, on risque de perdre du temps en modifiant la taille à de très courts intervalles. Il faut donc trouver un juste milieu, avec deux conséquences pratiques.

Tout d’abord, il faut être prudent avec les modifications de δ_{n+1} lors du processus adaptatif. C’est pourquoi on l’a finalement retiré de la déformation anisotrope, pour éviter des interférences potentielles avec les modifications pouvant survenir lors du contrôle de déviation. Il est alors plus facile de garder la main sur son évolution au cours du suivi, et on introduit de plus une borne inférieure δ_{min} sur δ_{n+1} pour l’empêcher de devenir trop faible, par exemple si la tolérance pour la déviation est réglée trop bas. De plus, il est parfois nécessaire de réajuster la valeur de δ_{n+1} pour éviter l’apparition de sommets au-dessus de $\lambda = 1$ ou en-dessous de $\lambda = 0$. Finalement, on applique à la fin du calcul de la nouvelle taille de triangulation la contrainte

$$\delta_{min} \leq \delta_{n+1} \leq \min(\lambda, 1 - \lambda),$$

avec par exemple

$$\delta_{min} = 10^{-6}.$$

Une illustration visuelle de ce point peut être trouvée sur les graphes page 42, où l’on voit que le dernier niveau est redimensionné pour respecter la limite en $\lambda = 1$. En fait, les premières valeurs de δ_{n+1} sont sur ces

exemples également modifiées par rapport à la borne en $\lambda = 0$, mais ceci n'est pas visible sur les graphes à cause de la petite taille initiale.

Ensuite, ceci explique pourquoi lors des expérimentations numériques le mode adaptatif donne de meilleurs résultats (comparé à la triangulation uniforme figée) avec un δ_{n+1} initial de 10^{-2} au lieu de 10^{-1} (voir en particulier 3.4). Pour les problèmes étudiés, choisir une valeur initiale de δ_{n+1} en dehors de $[10^{-2}, 10^{-1}]$ donne généralement de mauvais résultats de toute façon, avec un suivi bien trop long ou très instable. C'est pourquoi on utilise finalement presque exclusivement ces deux réglages, avec 10^{-1} dans le cas ordinaire et 10^{-2} pour les problèmes rapides (comme les problèmes d'arcs singuliers du chapitre 4 par exemple).

Pour terminer, il est à noter que les difficultés liées aux jonctions ont en pratique fortement handicapé la triangulation adaptative pendant un long moment. Au début, les formulations pour les jonctions faisaient encore appel à des dérivées, et les règles de calcul de la nouvelle taille de triangulation étaient bien moins restrictives que dans la version actuelle. Sans même parler de la perte importante de stabilité, le taux désespérément élevé d'échec lors des jonctions, et le coût numérique important des rares tentatives réussies, étaient tels que le mode adaptatif était inutilisable la plupart du temps. Une première amélioration notable fut d'imposer de strictes limitations sur les changements de taille possibles, avec un facteur 2 ou 0.5 seulement à la fois. Ceci permit de revenir à une stabilité normale, en plus de grandement faciliter les jonctions, ajouté à l'introduction de la troisième formulation. Le mode adaptatif devint alors enfin intéressant en pratique, même si les expérimentations menées au chapitre 3 sur les problèmes de transfert orbital ont montré que le coût numérique des jonctions n'était pas à négliger.

2.4 Raffinage de solution

On s'intéresse ici à la question de la précision finale de l'algorithme simplicial. Dans notre contexte, l'objectif premier de la continuation est de fournir un point initial satisfaisant pour la méthode de tir. De ce fait, la précision de la solution à la fin du chemin peut sembler secondaire au premier abord. Cependant, on constate dans certains cas que le tir final prend beaucoup de temps, presque autant que le suivi de chemin en entier ! On peut alors se demander si un meilleur point initial pourrait permettre une convergence plus rapide du tir, comme il utilise à la base une méthode de quasi-Newton. Dans d'autres situations, il se peut que la méthode de tir ne parvienne pas à converger, à cause de difficultés intrinsèques telles que des arcs singuliers par exemple. Dans ce cas de figure, on aimerait malgré tout pouvoir obtenir une meilleure solution que celle qu'on a normalement à la

fin du suivi.

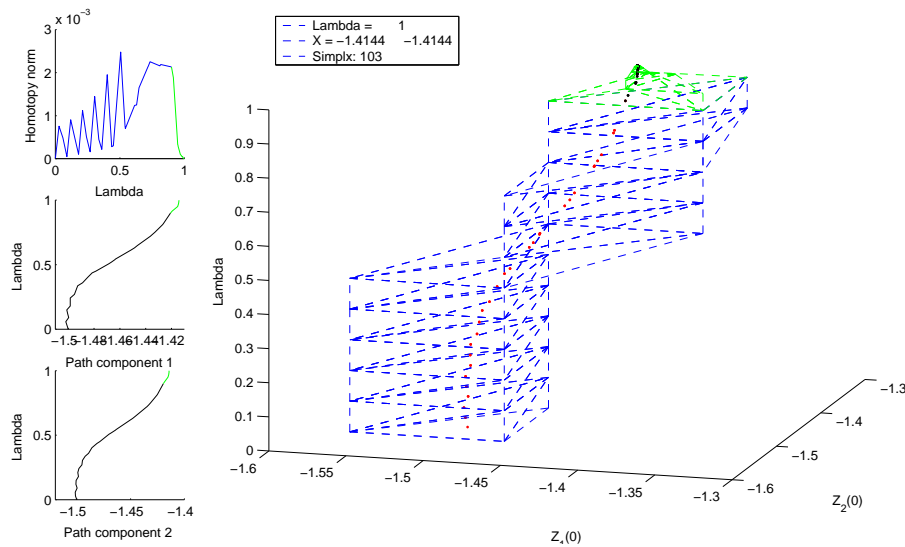
L'objectif est donc ici d'améliorer la précision de la solution donnée par la continuation, ce pour quoi nous avons essayé deux méthodes. La première vise à utiliser une triangulation raffinée avant d'arriver en $\lambda = 1$. La seconde est basée sur le principe de l'algorithme de redémarrage de Merrill, et essaye plutôt de raffiner la solution après la convergence.

2.4.1 Raffinage avant la convergence

L'idée est ici de réutiliser le principe des triangulations raffinées, mais seulement à l'approche de la convergence. On espère ainsi éviter les difficultés rencontrées par ces triangulations (surtout J_3) au début du chemin. En bref, on commence le suivi avec une triangulation uniforme, K_1 par exemple, de $\lambda = 0$ jusqu'à une certaine valeur λ_{refine} . A ce moment là, on change de triangulation, et on continue le suivi avec une J_3 (ou J_4), jusqu'à la convergence.

Remarque : le changement de triangulation en $\lambda = \lambda_{refine}$ ne requiert pas d'homotopie de jonction en pratique. On construit plutôt la première face complètement étiquetée pour la triangulation raffinée directement à partir de la dernière face complètement étiquetée de la triangulation uniforme. Ceci est possible car on garde la même taille de triangulation lors du changement.

Le graphe suivant illustre ce procédé sur le problème de démonstration présenté en 1.1.1.



Raffinage de solution : K_1 jusqu'à $\lambda = 0.9$, suivie par un raffinement J_3

Solution	Triangulation	Simplexes	Norme finale
$\lambda = 0.9$	$K_1(0.1)$	30	$2.13 \cdot 10^{-3}$
$\lambda = 1$	$J_3, (\epsilon = 10^{-9})$	90	$4.47 \cdot 10^{-13}$

L'effet du changement de triangulation en $\lambda = 0.9$ est bien visible sur le premier graphe en haut à droite, qui montre l'évolution de la norme de l'homotopie le long du chemin de zéros. On observe une brusque chute de la norme une fois le raffinement J_3 commencé, avec une norme finale très faible à $1.47 \cdot 10^{-13}$. On note toutefois également que sur cet exemple simple, le raffinement prend déjà trois fois plus de simplexes que le suivi principal jusqu'à $\lambda = 0.9$, ce qui est un des défauts de cette approche (même si le temps global d'exécution reste ici en-dessous de 2 secondes).

Si les choses se passent bien ici, cette formulation souffre en fait encore de certains inconvénients des triangulations raffinées, c'est-à-dire un coût élevé en termes de simplexes associé à certaines difficultés de convergence. Ce manque relatif de succès confirme à nos yeux que les triangulations raffinées ne sont pas vraiment adaptées à notre cadre d'études.

2.4.2 Raffinement après la convergence

La seconde idée est basée sur le principe de l'algorithme de redémarrage de Merrill (voir [2], p181 ou [24], p74 par exemple). Plus précisément, en partant de la solution z_1^* à la fin du chemin, on essaye d'effectuer une suite de jonctions au niveau $\lambda = 1$, avec des tailles de triangulation décroissantes. En pratique, on divise la taille par 2 à chaque tentative, de façon à ce que les jonctions ne soient pas trop difficiles à compléter. Cette approche ne souffre pas des problèmes de convergence mentionnés précédemment : si l'une des jonctions échoue, on l'ignore simplement et on passe à la suivante. On choisit d'arrêter le processus après un nombre fixé de jonctions de raffinement, car un critère d'arrêt portant sur la norme de $H(\cdot, 1)$ à la solution ne garantirait pas la terminaison. Ainsi, on peut grossièrement prédire la taille finale de triangulation utilisée, suivant la taille initiale et le nombre maximal de jonctions. Par exemple, en partant d'une taille $\delta = 10^{-1}$, 10 raffinages mènent à $\delta \approx 10^{-4}$, et 20 à $\delta \approx 10^{-7}$. Cette prédiction n'est en fait approchée que si le mode adaptatif est activé, la taille à la fin du chemin ayant alors probablement évolué depuis la valeur initiale.

Voici les résultats de cette approche sur le problème de démonstration : on effectue le suivi normalement avec une $K_1(0.1)$, puis on essaye 20 raffinages successifs.

Solution	Taille	Norme
Arrivée en $\lambda = 1$	0.1	$2.13 \cdot 10^{-3}$
Raffinage 10	$1.95 \cdot 10^{-4}$	$1.92 \cdot 10^{-5}$
Raffinage 20	$1.91 \cdot 10^{-7}$	$4.29 \cdot 10^{-10}$

La diminution de la norme est ici très régulière, ce qui n'est pas le cas en général. De plus, les raffinages sont ici quasiment instantanés (avec un temps total de 1 seconde pour le suivi principal plus les 20 jonctions en $\lambda = 1$), ce qui une fois encore n'est pas un comportement général.

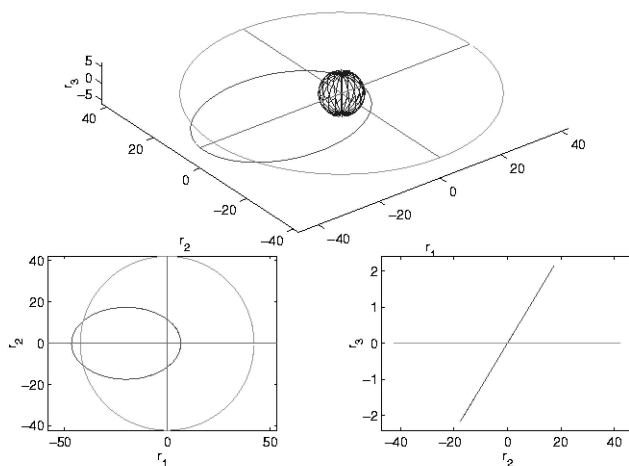
En fait, contrairement aux jonction adaptatives qui se produisent au cours du suivi, ces jonctions de raffinement qui ont lieu après la convergence requièrent souvent davantage de simplexes, comme l'homotopie est moins régulière. En conséquence, le nombre maximal de simplexes autorisé pour les jonctions doit être augmenté significativement, jusqu'à 10000 par exemple, suivant les problèmes.

Enfin, il est à noter que le coût numérique de ces jonctions de raffinement est généralement important, ce qui n'est pas à négliger. C'est pourquoi cette méthode semble plus appropriée en tant que substitut au tir final, quand celui-ci ne converge pas. Dans ce cas nous avons peu d'autres options pour obtenir une meilleure solution, à part refaire tout le suivi de chemin avec une taille de triangulation plus fine (ou bien régler le mode adaptatif plus prudemment). Ce raffinement de solution peut dans ce cas se révéler intéressant en termes de ratio précision finale / coût total. Un exemple d'application de ce procédé, en plus du mode adaptatif, est donné dans l'étude des problèmes d'arcs singuliers, plus précisément en 4.3.3, page 108.

Chapitre 3

Problèmes de Transfert Orbital

On s'intéresse ici à un problème de transfert orbital terrestre, étudié dans la cadre d'une collaboration avec le CNES (contrat R&T 02/CNES/0257/00-DPI 500, voir [16]). On souhaite transférer un satellite d'une orbite initiale elliptique et faiblement inclinée par rapport à l'équateur, à l'orbite équatoriale géosynchrone, tout en maximisant la masse finale (charge utile).



Transfert orbital (l'inclinaison sur le troisième graphe est redimensionnée)

L'objectif choisi, la maximisation de la masse finale, conduit à une structure de type bang-bang pour le contrôle optimal, avec un très grand nombre de commutations à poussée faible. En conséquence, il n'est pas possible en pratique de résoudre ce problème directement par le tir simple. L'étude complète (avec une méthode de continuation différentielle) ainsi que les interprétations physiques de ce problème sont décrites dans la thèse de Thomas Haberkorn ([26]), certains de ces résultats ayant déjà été présentés dans [25].

Note : Dans la suite, on utilisera fréquemment le terme “critère” pour désigner l’objectif du problème.

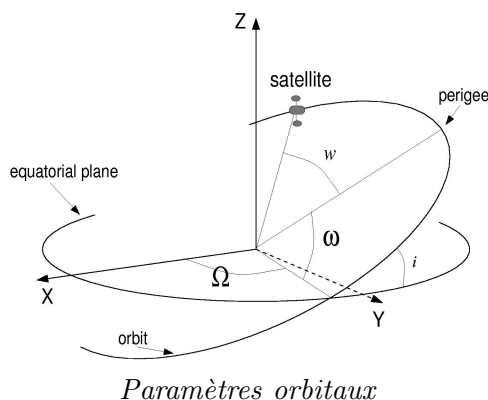
3.1 Présentation du problème

Dans la modélisation choisie, on considère que l’accélération subie par le satellite est due au potentiel terrestre ainsi qu’à la poussée exercée par les moteurs, plus précisément

$$\ddot{r} = -\mu_0 \frac{r}{|r|^3} + \frac{u}{m}$$

avec r le vecteur position, u la poussée (c’est à dire le contrôle), m la masse du satellite et μ_0 la constante gravitationnelle de la Terre ($\mu_0 = \mathcal{G}m_T$, avec \mathcal{G} la constant de gravitation universelle et m_T la masse de la Terre), et $|\cdot|$ la norme euclidienne.

Le vecteur d’état du système comprend la position, vitesse et masse du satellite à un instant t . Toutefois, à cause du grand nombre de révolutions attendu à poussée faible, exprimer la position et la vitesse en coordonnées cartésiennes géocentriques mènerait à d’importantes oscillations de l’état, préjudiciables à la stabilité numérique. C’est pourquoi on préfère utiliser un ensemble modifié d’éléments orbitaux classiques, qui décrit le mouvement du satellite d’un point de vue plus lié à l’orbite. On utilise les cinq premières composantes de l’état pour caractériser l’orbite osculatrice (que décrirait le satellite en l’absence de toute poussée), tandis que la sixième indique la position du satellite sur cette orbite. Comme la déformation de l’orbite est très régulière, en particulier pour les transferts à poussée faible, cela garantit une très bonne stabilité numérique de l’état, ce qui ne serait pas vrai en cartésien. Ce choix particulier de coordonnées est illustré ci-dessous.



avec

- P et e : paramètre et excentricité de l’ellipse

- θ : anomalie vraie
- Ω : longitude du noeud ascendant
- ω : argument du périégée
- i : inclinaison

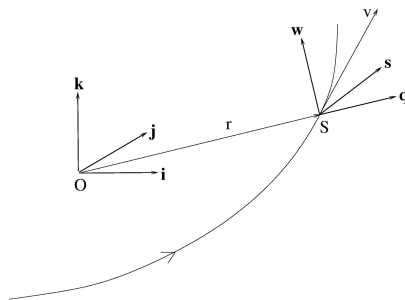
On définit maintenant l'état dans \mathbf{R}^7 :

- Paramètre de l'orbite P
- Vecteur excentricité (e_x, e_y) , dans le plan orbital, orienté vers le périégée
- Vecteur rotation (h_x, h_y) , dans le plan équatorial, colinéaire à l'intersection des plans orbitaux et équatoriaux
- Longitude vraie L
- Masse m

Avec les notations précédentes, ces paramètres sont définis par :

$$\begin{aligned} e_x &= e \cos(\Omega + \omega) & , & & e_y &= e \sin(\Omega + \omega) \\ h_x &= \tan(i/2) \cos \Omega & , & & h_y &= \tan(i/2) \sin \Omega \\ L &= \Omega + \omega + \theta \end{aligned}$$

Quant au contrôle tridimensionnel, on choisit de l'exprimer dans le repère mobile attaché au satellite, comme illustré ci-dessous.



Expression du contrôle dans le repère mobile (q, s, w)

Le contrôle normalisé u (tel que la poussée $\overrightarrow{T}(t) = u(t) T_{max}$) est alors exprimé dans \mathbf{R}^3 comme : poussée radiale q , poussée ortho-radiale et poussée normale w .

Si l'on note T_{Max} la poussée maximale et I_{sp} l'impulsion spécifique du propulseur, l'approximation choisie des forces gravitationnelles donne la dy-

namique du problème :

$$\left\{ \begin{array}{l} \dot{P}(t) = \frac{2T_{Max}}{m(t)} \sqrt{\frac{P^3(t)}{\mu_0}} \frac{s(t)}{Z(t)} \\ \dot{e}_x(t) = \frac{T_{Max}}{m(t)} \sqrt{\frac{P(t)}{\mu_0}} \frac{1}{Z(t)} [Z(t) \sin(L(t)) q(t) + A_1(t) s(t) \\ \quad - e_y(t) (h_x(t) \sin(L(t)) - h_y(t) \cos(L(t))) w(t)] \\ \dot{e}_y(t) = \frac{T_{Max}}{m(t)} \sqrt{\frac{P(t)}{\mu_0}} \frac{1}{Z(t)} [-Z(t) \cos(L(t)) q(t) + A_2(t) s(t) \\ \quad + e_x(t) (h_x(t) \sin(L(t)) - h_y(t) \cos(L(t))) w(t)] \\ \dot{h}_x(t) = \frac{T_{Max}}{2m(t)} \sqrt{\frac{P(t)}{\mu_0}} \frac{X(t)}{Z(t)} \cos(L(t)) \cdot w(t) \\ \dot{h}_y(t) = \frac{T_{Max}}{2m(t)} \sqrt{\frac{P(t)}{\mu_0}} \frac{X(t)}{Z(t)} \sin(L(t)) \cdot w(t) \\ \dot{L}(t) = \sqrt{\frac{\mu_0}{P^3(t)}} Z^2(t) + \frac{1}{m(t)} \sqrt{\frac{P(t)}{\mu_0}} \\ \quad \times \frac{1}{Z(t)} (h_x(t) \sin(L(t)) - h_y(t) \cos(L(t))) w(t) \\ \dot{m}(t) = -\frac{T_{Max}}{I_{sp} g_0} |(q(t), s(t), w(t))| \end{array} \right.$$

Avec

$$\left\{ \begin{array}{l} Z(t) = 1 + e_x(t) \cos(L(t)) + e_y(t) \sin(L(t)) \\ A_1(t) = e_x(t) + (1 + Z(t)) \cos(L(t)) \\ A_2(t) = e_y(t) + (1 + Z(t)) \sin(L(t)) \\ X(t) = 1 + h_x^2(t) + h_y^2(t) \\ I_{sp} = 2000s, \quad , \quad g_0 = 9.8ms^{-2} \quad \text{et} \quad \mu_0 = 398600.47km^3 \cdot s^{-2} \end{array} \right.$$

Si l'on note $x = (P, e_x, e_y, h_x, h_y, L)$ et $u = (q, s, w)$ on obtient la formulation suivante du problème de transfert avec maximisation de la masse finale, avec les conditions initiales et finales correspondant aux deux orbites :

$$(P_{m_f}) \left\{ \begin{array}{l} Max \ m(t_f) \\ \dot{x} = a(x) + \frac{T_{Max}}{m} B(x)u \\ \dot{m} = -\frac{T_{Max}}{I_{sp} g_0} |u| \\ |u| \leq 1 \\ CI : x(t_0) = (11625; 0.75; 0; 0.0612; 0; \pi), \ m(t_0) = 1500 \\ CF : x(t_f) = (42165; 0; 0; 0; 0; libre), \ m(t_f) \text{ libre} \\ t_0 = 0 \\ t_f \text{ fixe} \end{array} \right.$$

Contrairement aux problèmes en temps minimal, pour lesquels le temps de transfert est l'objectif à minimiser, et l'instant final t_f est donc libre, on considère ici un transfert à temps final fixé. La raison en est qu'il n'est pas évident que le problème de maximisation de la masse à temps final libre

admette une solution.

Par ailleurs, le critère réel utilisé pour la maximisation de la masse utile n'est pas $Max m(t_f)$ mais (ce qui est équivalent de par la dynamique de la masse)

$$Min \int_{t_0}^{t_f} |u(t)| dt.$$

Le Hamiltonien est donc défini par (t est ici omis par souci de clarté)

$$\mathcal{H}(x, m, p, p_m, u) = \left(1 - \frac{T_{Max}}{I_{sp}g_0} p_m\right) |u| + \frac{T_{Max}}{m} (B(x)u|p|) + (a(x)|p|).$$

Si $B^t(x)p \neq 0$ on définit la **fonction de commutation** ψ :

$$\psi(x, m, p, p_m) = 1 - \frac{T_{Max}}{I_{sp}g_0} p_m - \frac{T_{Max}}{m} |B^t(x)p|.$$

L'application du Principe du Maximum de Pontriaguine donne alors l'expression du contrôle optimal

$$\begin{cases} u = -\frac{B^t(x)p}{|B^t(x)p|} & si \ \psi(x, m, p, p_m) < 0 \\ u = -\alpha \frac{B^t(x)p}{|B^t(x)p|} & \alpha \in [0, 1] \quad si \ \psi(x, m, p, p_m) = 0 \\ u = 0 & si \ \psi(x, m, p, p_m) > 0 \end{cases}$$

Sinon si $B^t(x)p = 0$ on a

$$\begin{cases} u \in S(0, 1) & si \ 1 - \frac{T_{Max}}{I_{sp}g_0} p_m < 0 \\ u \in B(0, 1) & si \ 1 - \frac{T_{Max}}{I_{sp}g_0} p_m = 0 \\ u = 0 & si \ 1 - \frac{T_{Max}}{I_{sp}g_0} p_m > 0 \end{cases}$$

On voit que ce contrôle possède une structure bang-bang, comme sa norme commute entre 0 et 1 aux zéros de la fonction de commutation ψ . On fait maintenant deux hypothèses, qui sont vérifiées numériquement :

- On suppose que $B^t(x)p$ ne s'annule pas sur $[t_0, t_f]$
- Il n'y a pas **d'arc singulier**, c'est-à-dire que l'on n'a pas $\psi(x, m, p, p_m) = 0$ sur un intervalle non réduit à un point.

3.2 Approche de résolution

Comme mentionné plus tôt, la méthode de résolution de cette famille de problèmes est décrite en détails dans [26], et on en rappelle juste ici l'idée générale. Alors que la maximisation de la masse donne un problème difficile, à cause de la nature discontinue du contrôle optimal, le critère quadratique

$\text{Min} \int_{t_0}^{t_f} |u(t)|^2 dt$ (qui équivaut à minimiser une énergie) conduit à un problème bien plus régulier, avec un contrôle continu. On effectue donc une continuation de la forme “énergie vers masse”, en considérant la famille de problèmes (BVP_λ) munie du critère

$$J_\lambda = \int_{t_0}^{t_f} \lambda |u(t)| + (1 - \lambda) |u(t)|^2 dt.$$

et en définissant l’homotopie comme la fonction de tir correspondante

$$H : (z, \lambda) \mapsto S_\lambda(z).$$

On note que pour $\lambda = 1$, on retrouve bien $J_1 = \int_{t_0}^{t_f} |u(t)| dt$, qui est le critère correspondant à la maximisation de la masse finale. Cependant, les premières expérimentations numériques montrèrent l’existence de solutions locales pénalisantes. Elles furent éliminées en utilisant la longitude au lieu du temps comme variable d’intégration, ce qui mena finalement à une formulation à temps et longitude finale fixés.

Note : une conséquence de cette formulation qui utilise la longitude comme variable d’intégration est que la fonction de tir devient extrêmement sensible par rapport à l’état adjoint p_L associé à la longitude. En conséquence, la mise à l’échelle usuelle de cette composante de l’inconnue de l’(IVP) pose des difficultés numériques. Comme on veut garder la méthode aussi peu dépendante du problème que possible, on introduit une mise à l’échelle moins stricte au lieu d’ajuster manuellement cette composante. Ce “soft scaling” ne ramène les valeurs que sur l’intervalle $[10^{-2}, 10]$ au lieu de $[10^{-1}, 1]$.

En ce qui concerne l’intégration de l’(IVP), on s’attend à un contrôle bang-bang avec un nombre élevé de commutations, et on utilise principalement un intégrateur à pas variable (RKF45), qui se montre en pratique très satisfaisant. On a également essayé plusieurs schémas de Runge-Kutta à pas fixe, qui ne sont pas parvenus à compléter le suivi de chemin. La cause en est peut-être l’apparition progressive de la structure bang-bang, qui est difficile à traiter correctement à pas fixe (les transferts à poussée faible sont très longs, et comportent plusieurs centaines de commutations).

3.2.1 Initialisation

Pour cette famille de problèmes, l’initialisation de la continuation (ie résoudre le problème pour $\lambda = 0$) devient non trivial lorsque la poussée maximale T_{max} devient faible. Cette difficulté est contournée dans [26] en introduisant une homotopie préliminaire sur les conditions initiales, mais nous avons trouvé ici une autre possibilité.

Plus précisément, on essaye de résoudre le problème en $\lambda = 0$ par tir simple, mais en utilisant une intégration à pas fixe au lieu de RKF45. L'initialisation pour ce tir est très grossière : on fixe $p(0) = 0$. Cependant, on fixe la longueur et le temps final (qui sont plus sensibles) en accord avec les lois empiriques présentées dans [15], selon lesquelles ces deux grandeurs sont inversement proportionnelles à la poussée maximale. Ensuite, le nombre (fixé) de pas d'intégration est également choisi inversement proportionnel à T_{max} , comme indiqué ci-dessous.

T_{max}	L_f	t_f (h)	pas Rk4
10N	29.89539	150	300
5N	56.64919	300	600
1N	270.67959	1 500	3 000
0.5N	538.21759	3 000	6 000
0.2N	1340.83159	7 500	30 000
0.1N	2678.52159	15 000	50 000

Initialisation du tir en $\lambda = 0$ - Intégration RK4 pas fixe

Voici les résultats de ce tir à pas fixe, qui sont très satisfaisants.

T_{max}	t_f (h)	Norme à la solution	appels H	Temps(s)
10N	153.87	$6.75 \cdot 10^{-15}$	113	1
5N	304.76	$2.31 \cdot 10^{-15}$	169	2
1N	1 529.65	$3.74 \cdot 10^{-15}$	147	7
0.5N	3 065.80	$6.05 \cdot 10^{-15}$	145	13
0.2N	7 659.83	$2.20 \cdot 10^{-14}$	183	86
0.1N	15 315.79	$2.14 \cdot 10^{-14}$	123	96

Résultats du tir en $\lambda = 0$ - Intégration RK4 pas fixe

Le point intéressant est que ces solutions, obtenues pour $\lambda = 0$ avec une intégration à pas fixe, nous permettent en fait d'initialiser correctement l'algorithme simplicial, bien que l'intégration de l'(IVP) le long du chemin soit réalisée avec RKF45. On n'a donc plus besoin d'effectuer la continuation préliminaire sur les conditions initiales, ce qui est un gain de temps appréciable, surtout à poussée faible.

Remarque : il serait intéressant de vérifier si cette solution à pas fixe est également suffisante pour initialiser la continuation différentielle.

3.2.2 Suivi de chemin et solution

Comme dit plus tôt, le suivi de chemin utilise RKF45 pour l'intégration, avec une précision modérée (on fixe 10^{-8} et 10^{-6} pour les tolérances absolues et relatives). Après la convergence en $\lambda = 1$, on procède à un tir final avec une précision plus élevée ($10^{-12}/10^{-10}$). On présente les résultats de cette approche pour des poussées allant de 10N à 0.1N. Les colonnes indiquent

- **Norme initiale** : la norme de H à la solution à pas fixe (ici réintégré avec RKF45).

- **Simplexes** : le nombre de simplexes suivi pour le chemin

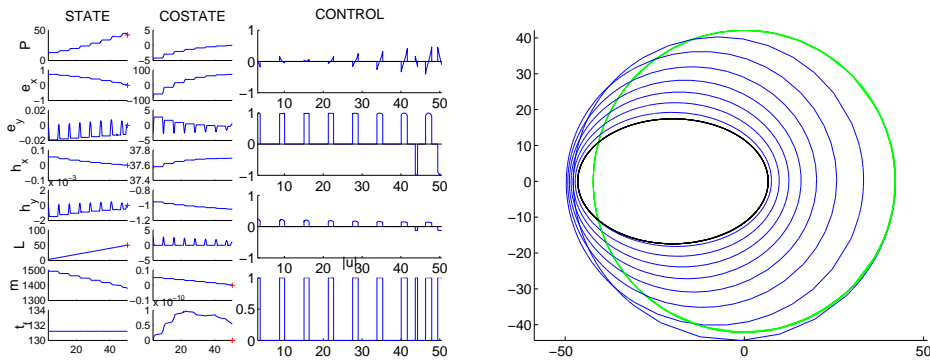
- **Temps** : le temps d'exécution en secondes.
- **Objectif** : la valeur de l'objectif (consommation de carburant en kg).
- **Norme finale** : la norme de H à la solution du tir final.

T_{max}	Norme initiale	Simplexes	Temps	Objectif	Norme finale
10N	$3.29 \cdot 10^{-4}$	1336	20	121.21	$3.43 \cdot 10^{-10}$
5N	$1.55 \cdot 10^{-4}$	1909	51	121.58	$4.63 \cdot 10^{-10}$
1N	$3.09 \cdot 10^{-3}$	840	132	121.78	$8.02 \cdot 10^{-11}$
0.5N	$3.58 \cdot 10^{-3}$	2028	548	121.69	$1.12 \cdot 10^{-9}$
0.2N	$6.34 \cdot 10^{-3}$	912	661	121.71	$3.21 \cdot 10^{-9}$
0.1N	$8.33 \cdot 10^{-3}$	775	1252	121.70	$1.75 \cdot 10^{-6}$

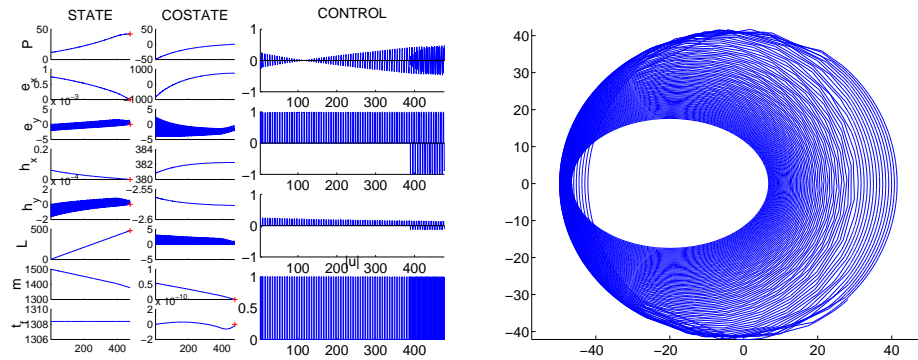
En ce qui concerne le suivi, on constate que l'initialisation se passe bien, avec une première jonction qui demande généralement moins de 100 simplexes malgré le changement d'intégrateur. Ceci est cohérent avec la norme correcte (autour de 10^{-3}) obtenue à la réintégration de la solution à pas fixe. Le suivi en lui-même converge normalement pour une triangulation basique $K_1(10^{-1})$, avec entre 1000 et 2000 simplexes. Pour le tir final, on note que la précision se dégrade pour les poussées T_{max} plus faibles, tandis que le temps pris par le tir augmente.

Par comparaison avec les résultats de la continuation différentielle utilisée en [26], les normes finales sont du même ordre, et les valeurs de l'objectif coïncident. On trouve en fait les mêmes solutions, ce qui est bien sûr rassurant. En termes de performance, le temps d'exécution de la méthode simpliciale est environ 3 fois plus long que celui de la continuation différentielle. Cela n'est pas surprenant, les méthodes Prédicteur-Correcteur étant connues pour être plus rapides que les méthodes simpliciales quand elles convergent.

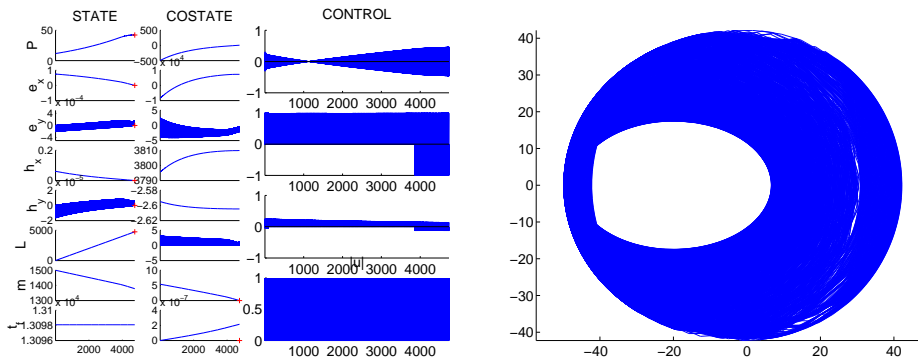
Voici les solutions pour $T_{max} = 10, 1$ et $0.1N$. Les deux colonnes à gauche représentent les composantes de l'état et de l'état adjoint, la colonne du milieu les trois composantes et la norme du contrôle, et finalement à droite la trajectoire (en 2D) depuis l'orbite elliptique initiale jusqu'à la géostationnaire. Les commutations du contrôle bang-bang sont bien visibles (pour $10N$ en tout cas...) sur la norme du contrôle, ie sur le graphe en bas au milieu.



Solution et trajectoire pour $T_{max} = 10N$



Solution et trajectoire pour $T_{max} = 1N$



Solution et trajectoire pour $T_{max} = 0.1N$

Remarque : si l'on essaye de réintégrer ces solutions à pas fixe, avec un second tir utilisant RK4 par exemple, on a du mal à obtenir la même précision, même avec un très grand nombre de pas. Ceci pourrait confirmer l'inadéquation d'une intégration à pas fixe (uniforme du moins) pour ces contrôles bang-bang avec un grand nombre de commutations.

En fait, le traitement correct de ce contrôle bang-bang (avec plusieurs centaines de commutations à poussée faible), sans connaissance a priori de

la structure, est une réussite remarquable de l'approche combinée continuation/tir simple. Par ailleurs, nous aimerions mentionner que la résolution de ces problèmes à mis en lumière certaines propriétés et interprétations physiques intéressantes, et renvoyons une fois encore le lecteur intéressé à la thèse de Thomas Haberkorn ([26]).

3.3 Comparaison de différents intégrateurs

On compare ici pour le tir final de RKF45 avec deux autres intégrateurs à pas variable, à savoir les codes DOP853 et ODEX par E.Hairer et G.Wanner, décrits en détail dans [27]. On utilise les mêmes tolérances absolues et relatives (ie 10^{-12} et 10^{-10}) pour tous les tirs, tout en gardant le RKF45 (10^{-8} , 10^{-6}) pour le chemin. On compare d'abord les résultats des tirs, avant d'examiner plus en détails les trajectoires intégrées.

3.3.1 Résultats du tir final

On indique ici les résultats des tirs pour les trois intégrateurs (bien sûr, ceux de RKF45 n'ont pas changé).

- **Pas** est le nombre de pas d'intégration à la solution
- **Norme finale** est la norme de H à la solution du tir final
- **Objectif** est la valeur de l'objectif à la solution
- **Tir (s)** donne le temps d'exécution (en secondes) du tir
- **Tir (H)** donne le nombre d'évaluations de l'homotopie durant le tir

T_{max}	Pas	Norme finale	Objectif	Tir (s)	Tir (H)
10N	1 552	$3.43 \cdot 10^{-10}$	121.21	4	55
5N	2 932	$4.63 \cdot 10^{-10}$	121.58	11	87
1N	15 170	$8.02 \cdot 10^{-11}$	121.78	38	51
0.5N	27 768	$1.12 \cdot 10^{-9}$	121.69	105	74
0.2N	67 769	$3.21 \cdot 10^{-9}$	121.71	161	49
0.1N	134 528	$1.75 \cdot 10^{-6}$	121.70	350	49

Solutions pour 10,5,1,0.5,0.2 et 0.1N - intégrateur RKF45

T_{max}	Pas	Norme finale	Objectif	Tir (s)	Tir (H)
10N	670	$2.86 \cdot 10^{-10}$	121.21	5	57
5N	1 328	$1.52 \cdot 10^{-11}$	121.58	15	88
1N	7 095	$5.82 \cdot 10^{-10}$	121.78	55	54
0.5N	12 805	$3.51 \cdot 10^{-9}$	121.69	164	96
0.2N	31 894	$1.13 \cdot 10^{-8}$	121.71	194	45
0.1N	63 810	$1.92 \cdot 10^{-5}$	121.70	570	65

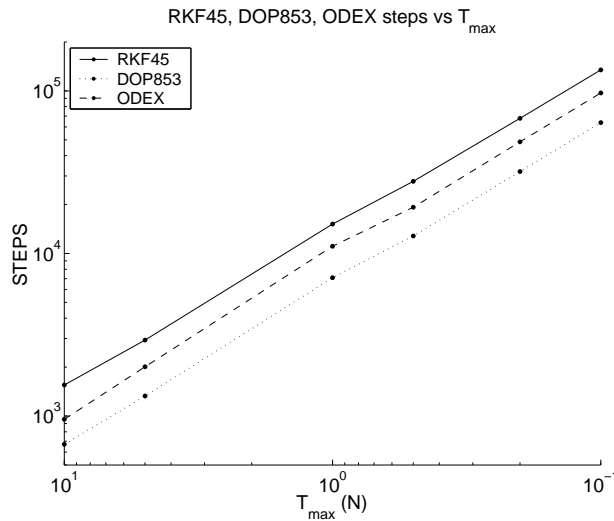
Solutions pour 10,5,1,0.5,0.2 et 0.1N - intégrateur DOP853

T_{max}	Pas	Norme finale	Objectif	Tir (s)	Tir (H)
10N	954	$3.96 \cdot 10^{-10}$	121.21	6	58
5N	2 008	$1.07 \cdot 10^{-11}$	121.58	13	69
1N	11 072	$9.61 \cdot 10^{-10}$	121.78	58	51
0.5N	19 242	$7.77 \cdot 10^{-10}$	121.69	216	109
0.2N	48 591	$1.49 \cdot 10^{-8}$	121.71	365	76
0.1N	97 104	$1.13 \cdot 10^{-2}$	121.70	381	37

Solutions pour 10,5,1,0.5,0.2 et 0.1N - intégrateur ODEX

Les solutions sont similaires (identiques jusqu'à 3 à 5 chiffres significatifs suivant les composantes de z pour 0.1N par exemple), avec les mêmes valeurs de l'objectif. Les normes finales sont également très proches, à l'exception du mauvais résultat d'ODEX pour 0.1N (le solveur stoppe ici prématurément, après seulement 37 appels de H). En termes de temps de tir l'ordre est moins régulier, mais globalement RKF45 tire mieux son épingle du jeu, suivi par DOP853 et ODEX, probablement parce que son ordre plus faible le pénalise moins face aux commutations..

Si l'on observe d'un peu plus près le nombre de pas d'intégration en fonction de la poussée maximale T_{max} , on voit plusieurs choses intéressantes, comme indiqué sur le graphe ci-dessous.



Pas d'intégration en fonction de T_{max} - RKF45, DOP853 et ODEX (échelle logarithmique sur les deux axes)

Pour les trois intégrateurs, le nombre de pas d'intégration à la solution et numériquement inversement proportionnel à la poussée maximale, ce qui illustre bien l'extrême régularité de cette famille de problèmes par rapport à T_{max} .

$$\text{Pas d'intégration} \times T_{max} \approx C^{te}.$$

Cette loi empirique est à rapprocher de résultats similaires concernant le temps de transfert et la longitude finale, obtenus lors de travaux précédents

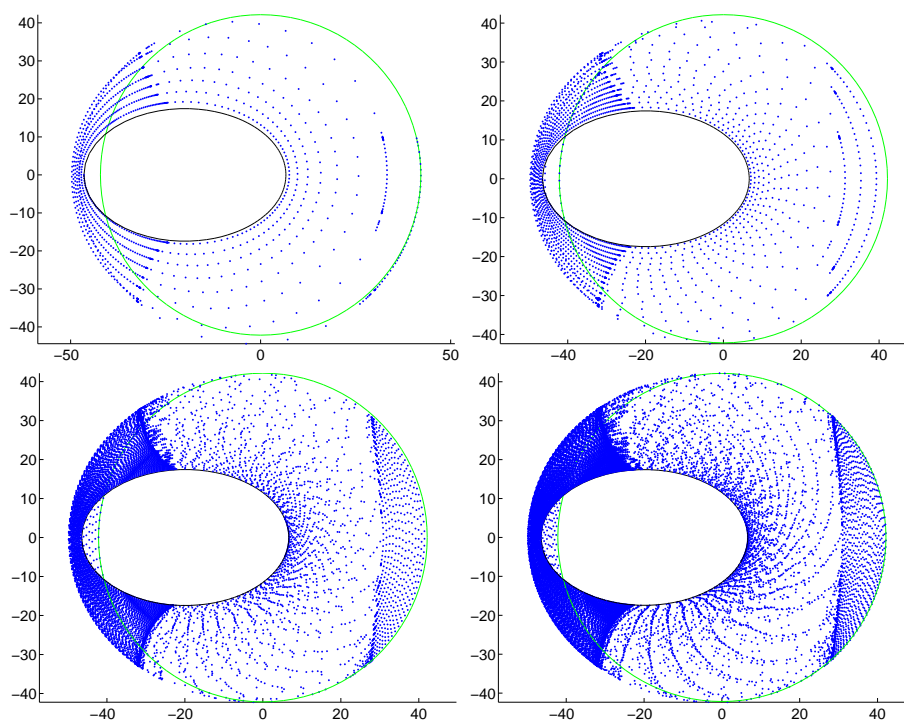
(cf [26]).

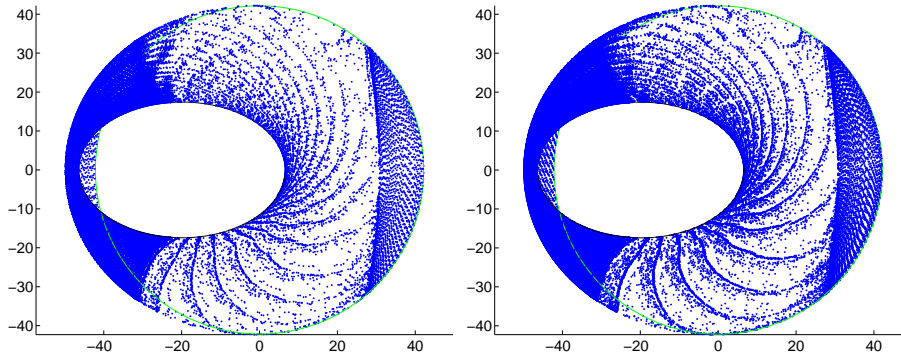
3.3.2 Trajectoires des solutions

Regardons maintenant d'un peu plus près les trajectoires obtenues par les trois intégrateurs à pas variable. Les trajectoires représentées dans la suite correspondent à la solution donnée par le tir final à la fin du suivi de chemin, pour les poussées $T_{max} = 10, 5, 1, 0.5, 0.2$ et $0.1N$. Sur chaque graphes sont dessinées les orbites initiales et finales, en plus des points pour chaque pas d'intégration. L'espacement entre les points indique donc le pas d'intégration (on rappelle que la variable d'intégration est ici toujours la longitude L , même si elle est parfois abusivement dénommée "time" sur certains graphes).

Intégrateur RKF45

On commence par les trajectoires intégrées par RKF45.





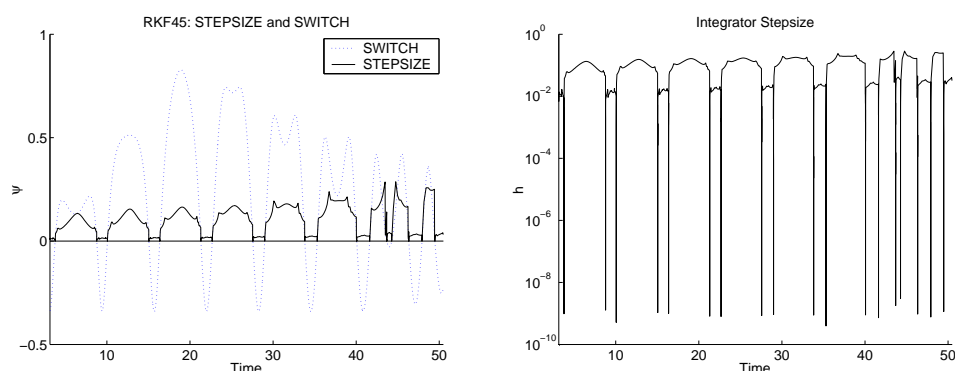
Trajectoires pour 10,5,1,0.5,0.2 et 0.1N - intégrateur RKF45

La première chose que l'on remarque à propos de ces trajectoires est une indication supplémentaire de l'extrême régularité du problème par rapport à la poussée T_{max} . On retrouve en effet la même disposition des noeuds d'intégration pour chaque transfert, avec simplement plus de révolutions pour les poussées plus faibles. Concernant la répartition des noeuds, ils semblent à première vue régulièrement espacés tout au long de l'orbite, mais avec un pas clairement plus faible sur les arcs de poussée (centrés sur les apogées et derniers périgées) que sur les arcs sans poussée.

On examine maintenant l'évolution du pas d'intégration en relation avec la fonction de commutation ψ . On rappelle que ψ et le contrôle optimal u^* sont étroitement liés, par :

- $|u^*(t)| = 1$ quand $\psi(t) < 0$, t est alors dans un arc de poussée.
- $u^*(t) = 0$ quand $\psi(t) > 0$, t est alors dans un arc sans poussée.
- u^* est discontinue en t quand $\psi(t) = 0$, t est alors un instant de commutation.

Note : les graphes suivants montrant le pas d'intégration correspondent au transfert à 10N. On observe le même comportement qualitatif pour les poussées inférieures, mais les graphes ont peu d'intérêt visuel à cause du grand nombre de commutations. Le premier graphe montre la fonction de commutation ψ et le pas d'intégration h , le second seulement h , avec une échelle logarithmique pour mieux distinguer les trois plages de valeurs prises (qui correspondent aux arcs de poussée / sans poussée, et aux commutations).



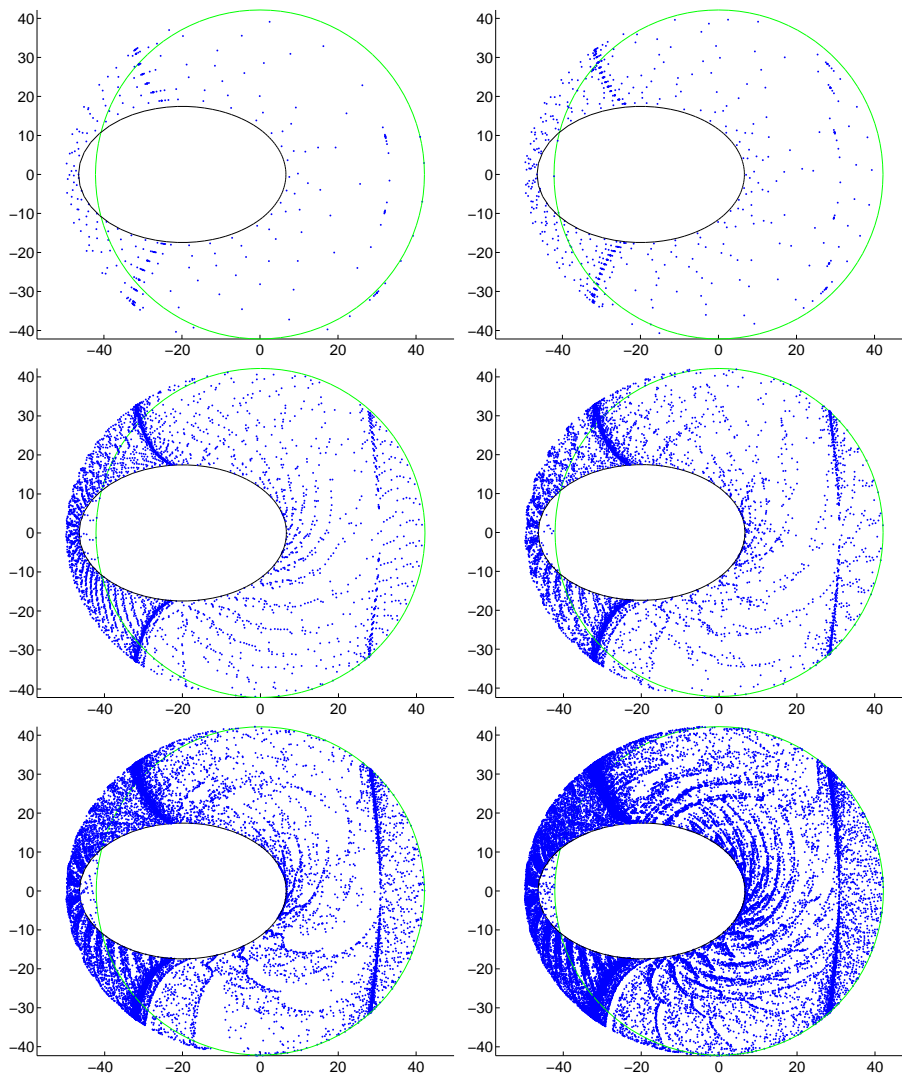
Pas d'intégration et fonction de commutation (10N) - RK45

L'échelle logarithmique sur le graphe de droite donne une meilleure idée de l'ordre de grandeur du pas, et l'on distingue clairement trois cas, qui correspondent remarquablement bien au signe de la fonction de commutation ψ . Tout d'abord, on voit que le pas devient très faible (jusqu'à 10^{-9}) aux commutations, lorsque ψ s'annule. Ceci indique que l'intégrateur détecte correctement les discontinuités du second membre liées aux commutations du contrôle, mais ces pas extrêmement faibles sont certainement pénalisants en termes de temps de calcul.

Sur les arcs de poussée, le pas est stable et relativement bas, autour de 10^{-2} , et augmente légèrement d'une révolution à l'autre. La différence entre le pas des arcs de poussée au début et à la fin du transfert est nettement visible sur la partie gauche des graphes représentant les trajectoires en dessous de $1N$: les arcs de fin se superposent à ceux du début, avec des points plus espacés (comme pour les arcs de poussée sur la partie gauche des graphes). Sur les arcs sans poussée, le pas est environ un ordre de grandeur au-dessus, autour de 10^{-1} , et est là aussi assez régulier. Cette différence d'ordre de grandeur s'explique probablement par la dynamique du système. Avec l'état exprimé en fonction des paramètres orbitaux, le satellite soumis à un contrôle nul dérive simplement sur son orbite, qui reste elle inchangée. ainsi sur l'état $[P, e_x, e_y, h_x, h_y, L, m]$, seule la longitude varie, tandis que tous les autres éléments orbitaux et la masse demeurent constants. C'est pourquoi il semble logique que l'intégration puisse se faire avec des pas plus grands sur les arcs sans poussée. Concernant l'augmentation modérée du pas sur les arcs de poussée successifs, elle est peut-être liée au changement de forme de l'orbite, qui devient de plus en plus circulaire, avec une vitesse angulaire du satellite plus faible que pour les apogées du début, ce qui permettrait une intégration plus rapide suivant la longitude.

Intégrateur DOP853

On passe maintenant à l'intégrateur DOP853, qui appartient à la même classe que RKF45 (méthodes de Runge Kutta imbriquées), mais d'ordre plus élevé, 8 au lieu de 5.

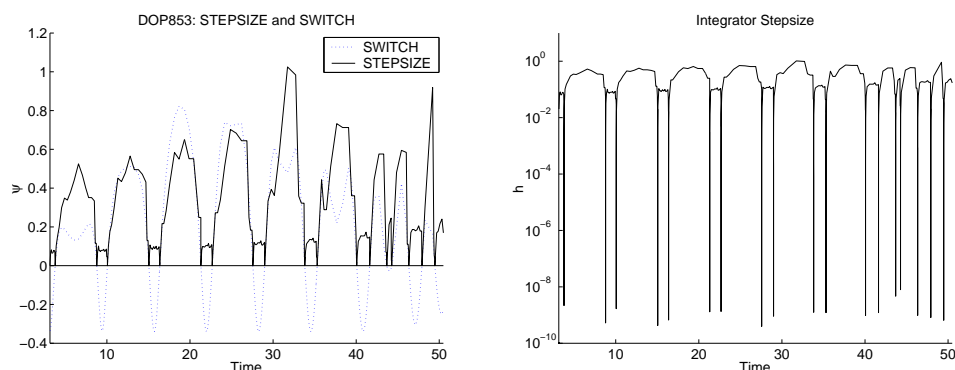


Trajectoires pour 10,5,1,0.5,0.2 et 0.1N - intégrateur DOP853

Comme auparavant, on constate que les trajectoires ont la même allure pour toutes les poussées. Les similarités avec RKF45 sont la répartition régulière des noeuds d'intégration sur les arcs avec et sans poussée, encore une fois avec un pas plus faible sur les premiers. On remarque aussi le même phénomène sur la différence de pas entre les apogées du début et celles de la fin, ou les périgées. Une autre différence notable est la présence d'une concentration importante de noeuds autour des commutations entre les arcs

avec et sans poussée, particulièrement visible sur les graphes $1N$ et $0.5N$ par exemple. Il y avait également de très petits pas aux commutations pour RKF45, mais on ne voyait pas un tel contraste avec les arcs de poussée.

Voyons maintenant plus en détails l'évolution du pas.



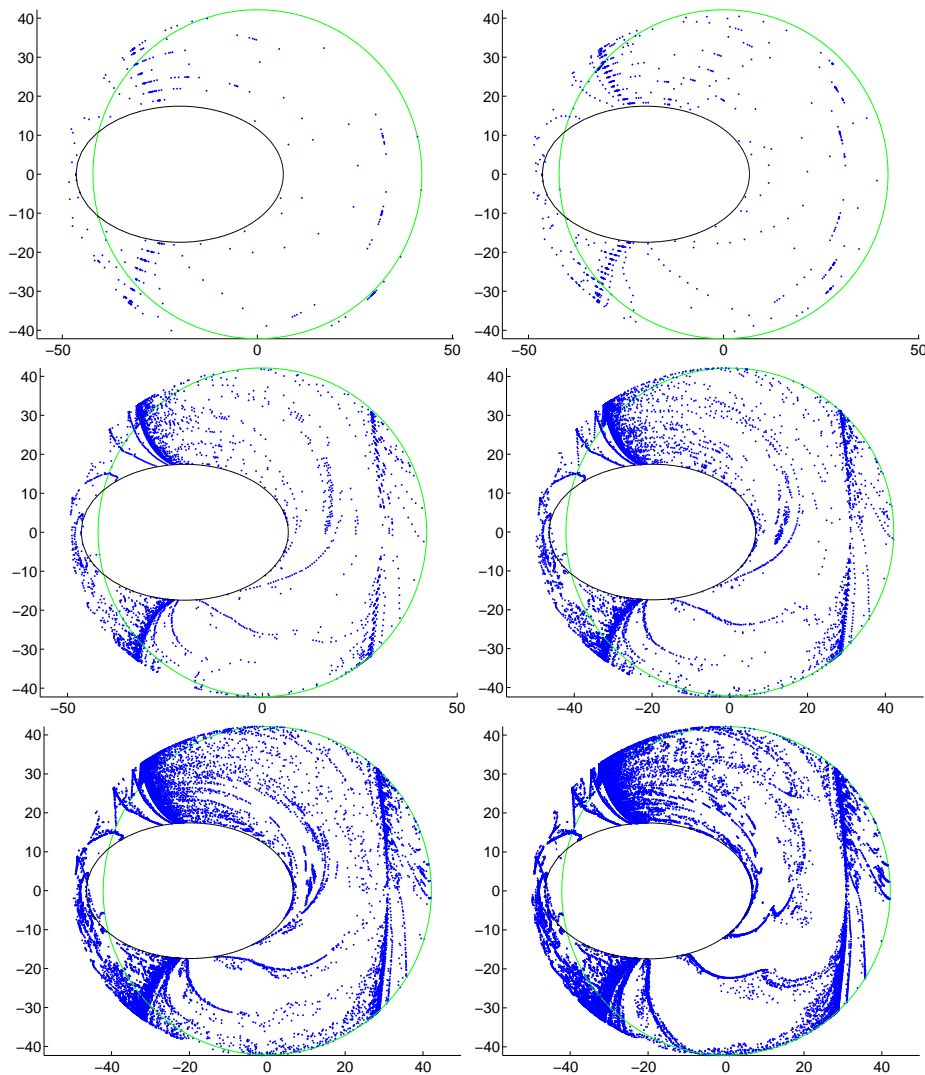
Pas d'intégration et fonction de commutation (10N) - DOP853

Se confirme ici l'importante similarité de comportement entre DOP853 et RKF45, car l'on retrouve trois plages distinctes de valeurs, qui correspondent encore parfaitement au signe de la fonction de commutation. Pour commencer, le pas devient comme avant très faible (10^{-9} encore) au niveau des commutations, où l'intégrateur doit faire avec les discontinuités du second membre. Pour les arcs de poussée, le pas est encore régulier et assez bas, autour de 10^{-1} . En dehors, les pas sont plus grands, entre 0.1 et 41 environ.

Finalement, la principale différence avec RKF45 semble être le pas globalement plus grand (sauf aux commutations, d'où peut-être l'impression visuelle d'une accumulation de points), ainsi que le moindre écart entre le pas sur les arcs avec et sans poussée à la fin du transfert. Le pas plus grand est vraisemblablement une conséquence de l'ordre plus élevé de DOP853, tandis que le second point indique peut-être une meilleure performance de l'intégrateur lorsque l'orbite devient plus circulaire. On a par contre le même comportement qualitatif pour les deux intégrateurs, ce qui est cohérent avec le fait qu'ils implémentent tous les deux des méthodes de Runge Kutta imbriquées, mais d'ordre plus élevé pour DOP853.

Intégrateur ODEX

On termine par l'intégrateur ODEX, qui contrairement aux deux autres utilise une méthode d'extrapolation, d'ordre variable en plus du pas.

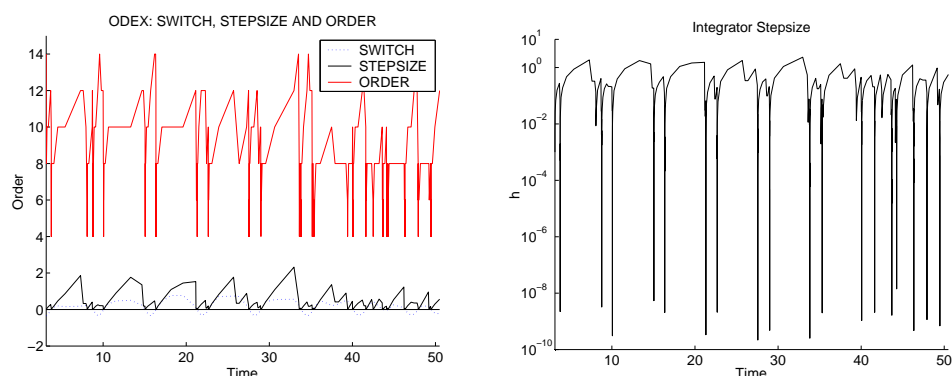


Trajectoires pour $10, 5, 1, 0.5, 0.2$ et $0.1N$ - intégrateur ODEX

Comparé aux deux autres intégrateurs, on note une différence évidente dans le motif des noeuds d'intégration pour ODEX. Contrairement à RKF45 et DOP853, leur répartition est fortement irrégulière, avec de grands "blancs", visibles aussi bien sur les arcs avec et sans poussée. Ceci indique de très grands pas d'intégration, qui sont effectivement un trait des méthodes d'extrapolation. Ce qui est un peu surprenant est l'apparente absence de discrimination entre les arcs avec et sans poussée en termes de pas, particulièrement visible sur les apogées du début (partie gauche des graphes). On retrouve tout de même l'accumulation de points aux commutations, et plus précisément ici *avant* les commutations proprement dites. C'est un peu différent du cas DOP853 par exemple, où les très petits étaient davantage

centrés sur les commutations.

Pour ODEX l'ordre de l'intégration est également variable, et il est donc représenté en plus du pas et de la fonction de commutation.



Ordre et pas d'intégration, fonction de commutation (10N) - ODEX

Une fois encore, on voit que le pas chute lors des commutations (en-dessous de 10^{-9}). Il est intéressant de voir que l'ordre descend également à cette occasion à sa valeur minimale (4), ce qui est censé permettre une prise en compte efficace des discontinuités, comme mentionné dans [27], II.9 page 237. A part cela, les pas sont encore globalement plus grands sur les arcs sans poussée que les arcs de poussée, et sont assez irréguliers dans les deux cas. En particulier, l'augmentation du pas après une commutation semble moins abrupte que pour DOP853 et RKF45, pour lesquels l'évolution du pas sur chaque arc de poussée est presque symétrique. Ceci est peut-être dû au fait que le mécanisme de contrôle du pas est différent pour ODEX, ce qui pourrait aussi expliquer l'atténuation de l'écart de pas entre les arcs avec et sans poussée à la fin du transfert.

Comme on pouvait s'en douter au vu des trajectoires, le pas est globalement plus grand que pour DOP853, entre 10^{-1} et 1 sur les arcs de poussée, et au-dessus de 1 en dehors. Par contre, on note des difficultés accrues au niveau des commutations, avec une accumulation de noeuds avant d'arriver aux instants de commutation, ce qui n'était pas le cas pour les deux autres. L'ordre est quant à lui compris entre 4 (valeur minimale possible, atteinte aux commutations) et 14, ce qui est effectivement plus élevé que pour les deux intégrateurs précédents. A ce stade, il semble que la méthode d'extrapolation ne soit pas forcément un bon choix pour ce problème, car son ordre élevé rend le franchissement des commutations plus pénalisant en termes de temps de calcul.

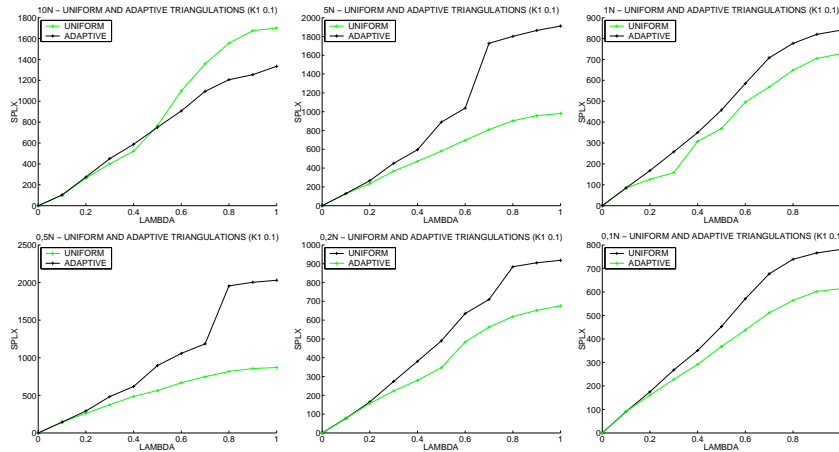
3.4 Etude de la triangulation adaptative

On souhaite maintenant étudier sur cette famille de problèmes le comportement de la triangulation adaptative décrite en 2.3. On choisit de faire porter les tests sur la configuration par défaut du mode adaptatif complet, à savoir une tolérance de 10^{-1} pour le contrôle de déviation, et $(2, 0.2, 0.1)$ comme seuils de déformation anisotrope. On compare d'abord les suivis sans et avec le mode adaptatif, pour mesurer l'impact sur le nombre de simplexes suivis. On s'intéresse ensuite aux résultats des tirs à la fin du chemin, pour voir comment ils sont affectés par la précision des solutions en $\lambda = 1$.

Suivi de chemin

On représente ici le nombre de simplexes suivis le long du chemin. Les courbes sont évidemment croissantes, partant de 0 pour atteindre le nombre total de simplexes en $\lambda = 1$. On trace ensemble les simplexes pour la triangulation uniforme de référence et pour le mode adaptatif, pour $T_{max} = 10, 5, 1, 0.5, 0.2$ et $0.1N$.

Note : le nombre maximal de simplexes autorisé pour les jonctions est fixé à 100, sauf pour $0.2N$ où on l'augmente à 500 en raison des difficultés de jonction rencontrées (voir plus loin).



*Simplexes suivis - Triangulation uniforme et adaptative
10,5,1,0.5,0.2 et 0.1N - triangulation $K_1(10^{-1})$*

Tout d'abord, on remarque un curieux "saut" de simplexes entre les niveaux $\lambda = 0.6, 0.7$ ou $\lambda = 0.7, 0.8$ sur certains chemins, comme 5, 0.5 et $0.2N$, que l'on étudiera plus en détails dans la suite. Concernant le gain en simplexes du mode adaptatif, il ne semble pas si important que cela, excepté dans deux cas (5 et $0.5N$) où ces sauts sont évités. Dans les autres cas, le gain existe mais est assez modéré (à l'exception du $10N$, pour lequel l'adaptatif prend en fait davantage de simplexes!). Plus généralement, il semble

qu'on ait moins à attendre du mode adaptatif lorsque le suivi se passe déjà bien avec une grande triangulation uniforme comme la $K_1(10^{-1})$, car il y a moins de marge pour l'amélioration que pour un suivi qui rencontre des difficultés. Maintenant que l'on a une idée du gain en simplexes, on examine l'amélioration de la précision, et ses conséquences sur le tir final.

Tir final

On s'intéresse donc de plus près aux résultats du tir après les suivis uniformes et adaptatifs. Tous les tirs utilisent ici le RKF45(10^{-12} , 10^{-10}) habituel, et on retrouve donc les résultats précédents pour la $K_1(10^{-1})$ uniforme. Les tableaux suivants indiquent

- $|H_1|$: la norme de l'homotopie à la fin du chemin, **avant** le tir (ceci mesure grossièrement la qualité du point initial du tir).
- **Tir (H)** et **Norme finale** : comme avant, le nombre d'appels de l'homotopie, et norme finale pour le tir.
- **Total (H)** et **Total (s)** : nombre total d'appels de l'homotopie, et temps d'exécution global (en secondes).

Note : il est à noter que les évaluations de l'homotopie effectués durant le tir sont plus coûteuses que celles du suivi de chemin, car on demande une plus grande précision d'intégration. C'est pourquoi le temps d'exécution global n'est pas exactement proportionnel au nombre total d'appels de l'homotopie (pour un T_{max} donné), quand bien même l'essentiel du coût numérique provient de ces appels. La différence n'est cependant pas très importante, à cause du faible nombre d'appels "de tir" comparé au nombre d'appels "de chemin".

T_{max}	$ H_1 $	Tir (H)	Norme finale	Total (H)	Total (s)
10N	0.24	55	$3.43 \cdot 10^{-10}$	1553	20
5N	0.76	87	$4.63 \cdot 10^{-10}$	2117	51
1N	0.05	51	$8.02 \cdot 10^{-11}$	943	132
0.5N	0.74	74	$1.12 \cdot 10^{-9}$	2226	548
0.2N	0.24	49	$3.21 \cdot 10^{-9}$	1050	661
0.1N	0.04	49	$1.75 \cdot 10^{-6}$	920	1252

Triangulation $K_1(10^{-1})$ - Mode uniforme (référence)

On remarque que le tir semble avoir besoin de davantage d'appels à l'homotopie lorsque la norme de la solution à la fin du chemin est plus élevée (5 et 0.5N). Voici maintenant les résultats du mode adaptatif.

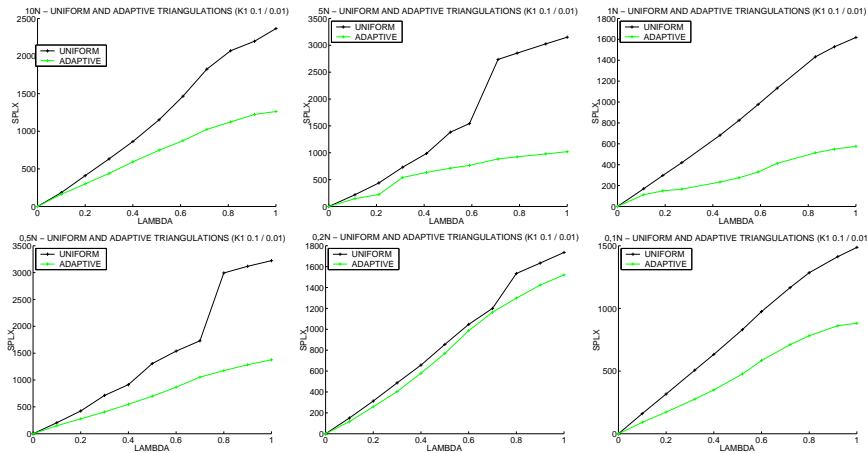
T_{max}	$ h_1 $	Tir (H)	Norme finale	Total (H)	Total (s)
10N	0.01	43	$1.93 \cdot 10^{-10}$	2252	27
5N	0.002	66	$2.07 \cdot 10^{-10}$	1425	35
1N	0.02	64	$4.11 \cdot 10^{-11}$	1226	166
0.5N	0.01	44	$4.89 \cdot 10^{-10}$	1277	305
0.2N	0.02	44	$2.16 \cdot 10^{-7}$	1116	696
0.1N	0.02	70	$9.85 \cdot 10^{-5}$	1018	1486

Triangulation $K_1(10^{-1})$ - Mode adaptatif

On constate tout d'abord que les normes à la fin du chemin en $\lambda = 1$ sont meilleures qu'avec le mode uniforme. Toutefois, l'amélioration sur les tirs n'est pas éblouissante : ils sont meilleurs dans deux cas (5 et $0.5N$ encore une fois), mais restent sinon comparables, voir se dégradent (1 et $0.1N$). Finalement, d'après les résultats médiocres en termes de performance (nombre total d'appels à l'homotopie et temps d'exécution), il semble que le mode adaptatif ne tire pas vraiment son épingle du jeu avec la triangulation $K_1(10^{-1})$ utilisée.

3.4.1 Une triangulation plus adaptée

Les premiers tests, menés avec une triangulation $K_1(10^{-1})$, confirment le fait que le mode adaptatif ne fonctionne pas bien avec un pas aussi grand suivant λ . En effet, une valeur de 0.1 signifie seulement 10 jonctions adaptatives (environ) le long du chemin, ce qui est trop peu pour bénéficier pleinement de la triangulation adaptative. C'est pourquoi on essaye de reprendre cette série de tests, avec une taille légèrement modifiée, plus exactement $\delta_i = 0.1, \forall i \in [1..n]; \delta_{n+1} = 0.01$, que l'on note dans la suite $K_1(10^{-1}/10^{-2})$. Cette seconde triangulation, bien que donnant un suivi plus lent que la $K_1(10^{-1})$ standard, permet de mieux étudier les effets du mode adaptatif. On répète donc les tests précédents, mais avec un pas de 10^{-2} suivant λ .



*Simplexes suivis - Triangulation uniforme (référence) et adaptative
10,5,1,0.5,0.2 et 0.1N - Triangulation $K_1(10^{-1}/10^{-2})$*

Un point intéressant est que l'on retrouve les mêmes "sauts" qu'auparavant (nous verrons pourquoi dans la suite). Le mode adaptatif est encore ici particulièrement efficace pour 5 et $0.5N$, mais s'en sort cette fois beaucoup mieux pour 10, 1 et $0.1N$. Seul le $0.2N$ ne semble pas très bon, et l'on étudiera ce cas plus en détails dans la suite. Il semble donc bien que le pas de 0.1 suivant λ pénalise le mode adaptatif dans la première série de tests.

On regarde maintenant les résultats des tir pour la $K_1(10^{-1}/10^{-2})$.

T_{max}	$ h_1 $	Tir (H)	Norme finale	Total (H)	Total (s)
10N	0.24	80	$1.30 \cdot 10^{-10}$	2685	34
5N	0.76	71	$4.11 \cdot 10^{-10}$	3425	77
1N	0.05	36	$1.53 \cdot 10^{-10}$	1793	223
0.5N	0.74	95	$1.55 \cdot 10^{-9}$	3546	855
0.2N	0.24	83	$6.33 \cdot 10^{-9}$	2030	1293
0.1N	0.04	58	$3.49 \cdot 10^{-5}$	1726	2170

Triangulation $K_1(10^{-1}/10^{-2})$ - Mode uniforme (référence)

Comme auparavant, on note un lien grossier entre la norme à la fin du chemin et le nombre d'appels de l'homotopie pour le tir, avec ici deux cas plus rapides pour 1 et 0.1N. Comparons avec les résultats pour le mode adaptatif.

T_{max}	$ h_1 $	Tir (H)	Norme finale	Total (H)	Total (s)
10N	0.01	48	$2.28 \cdot 10^{-10}$	3736	49
5N	0.01	43	$1.78 \cdot 10^{-10}$	2677	72
1N	0.02	33	$4.70 \cdot 10^{-9}$	1742	209
0.5N	0.03	34	$9.78 \cdot 10^{-10}$	2090	489
0.2N	0.03	49	$2.10 \cdot 10^{-9}$	2552	1510
0.1N	0.02	46	$6.77 \cdot 10^{-5}$	1772	2051

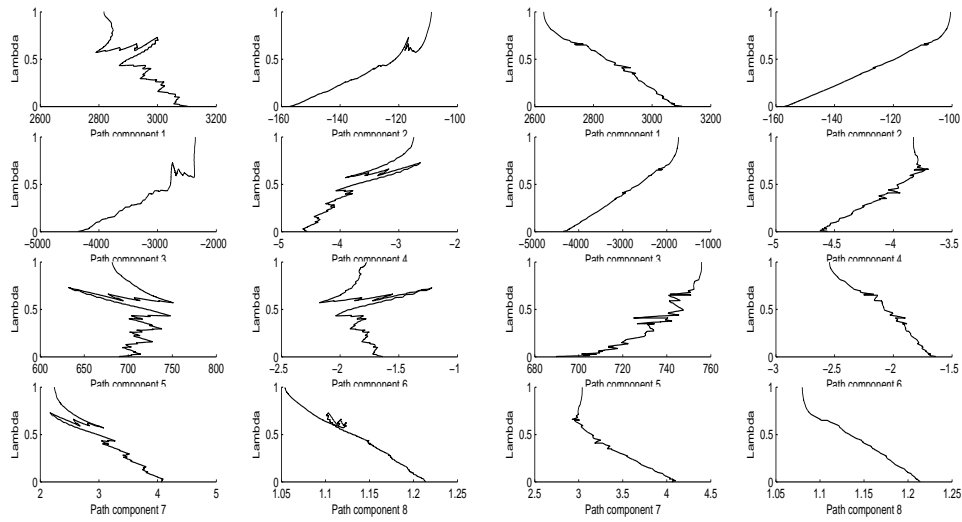
Triangulation $K_1(10^{-1}/10^{-2})$ - Mode adaptatif

Cette fois les choses sont plus satisfaisantes, avec une nette amélioration des normes à la fin des chemins, et surtout également au niveau des tirs (sauf pour les 1 et 0.1N qui étaient déjà rapides). Maintenant tous les tirs effectuent entre 30 et 50 appels de l'homotopie. Si l'on ajoute ces bons résultats à ceux sur le nombre de simplexes parcourus, il est désormais clair que le mode adaptatif a besoin d'un pas relativement faible suivant λ pour être efficace. Cependant, on remarque que malgré le gain en simplexes et les tirs plus rapides, les performances globales sont un peu décevantes, car le coût numérique des jonctions adaptatives contrebalance souvent ces améliorations.

On se concentre maintenant un peu plus sur l'aspect qualitatif du suivi de chemin, et plus particulièrement pour 0.5 et 0.2N. Le suivi pour 0.5N présente un "saut" important, et on note une très bonne performance du mode adaptatif. Au contraire, pour le 0.2N, qui présente aussi un saut (moins marqué), le gain de l'adaptatif est insignifiant, et l'on observe de grandes difficultés pour les jonctions (le nombre maximal de simplexes de jonction ayant du être porté à 500).

Un cas favorable

Commençons par le cas favorable à 0.5N.



Chemin de zéros pour $0.5N$ - Modes uniforme et adaptatif

On voit que le chemin de gauche, correspondant à la triangulation uniforme, est très irrégulier, avec de fortes oscillations sur les composantes 1,5 et 6 en particulier. En regardant plus précisément vers $\lambda = 0.7$, on trouve l'explication du "saut" de simplexes constaté précédemment. On constate que le suivi redescend après avoir atteint $\lambda = 0.7$, et revient à $\lambda = 0.6$ avant de repartir jusqu'à $\lambda = 1$. C'est pourquoi il faut un si grand nombre de simplexes pour passer du niveau $\lambda = 0.7$ au niveau $\lambda = 0.8$. D'après l'allure générale du chemin avant et après cet incident, ce comportement pourrait être lié à l'existence de deux chemins distincts (ou peut-être une bifurcation).

A l'inverse, le chemin de droite, qui correspond au mode adaptatif, à l'air bien plus lisse. Les oscillations des composantes 1,5,6 sont fortement réduites, et plus important, il n'y a pas de signe de redescente en $\lambda = 0.7$ (cependant, une inflexion du chemin est clairement visible sur les composantes 4,7 par exemple). Ce meilleur comportement se traduit par la réduction significative du nombre de simplexes parcourus (1377 contre 3222). Les jonctions adaptatives totalisent 1286 simplexes, ce qui est presque autant que le chemin principal (on rappelle toutefois que tous les simplexes de jonction me requièrent pas l'évaluation de l'homotopie, cf plus loin).

On regarde maintenant l'évolution de la taille de la triangulation au cours du suivi. On ne représente δ que pour des valeurs de λ multiples de 0.1 pour plus de clarté, même si la hauteur réelle des niveaux est en fait $\delta_{n+1} = 0.01$ (et reste inchangée durant tout le suivi).

λ	δ_1	δ_2	δ_3	δ_4	δ_5	δ_6	δ_7	δ_8
0	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
0.1	0.1	0.1	0.4	0.1	0.4	0.1	0.2	0.05
0.2-0.3	0.1	0.1	0.4	0.1	0.4	0.1	0.2	0.025
0.4	0.1	0.1	0.4	0.2	0.4	0.2	0.1	0.025
0.5-0.6	0.1	0.1	0.4	0.1	0.4	0.2	0.1	0.025
0.7-1	0.05	0.05	0.2	0.1	0.2	0.1	0.05	0.0125

Evolution de δ au cours du suivi - $0.5N$ ($\delta_{n+1} = 0.01$)

On remarque immédiatement qu'une réduction globale de la taille survient entre $\lambda = 0.6$ et $\lambda = 0.7$ (à $\lambda = 0.65$ plus précisément), ce qui coïnciderait avec la déviation et redescente observées avec la triangulation uniforme. L'algorithme semble à juste titre détecter des difficultés à cet endroit (il s'agit en pratique de la norme de l'homotopie qui dépasse le seuil de 10^{-1}), et adopte une triangulation plus fine pour un suivi plus prudent. Ceci se révèle être un choix judicieux, car on évite ainsi le mauvais comportement du mode uniforme. De plus, le fait qu'on garde cette petite taille de triangulation jusqu'à la fin est cohérent avec l'aspect presque "vertical" du chemin pour $\lambda \in [0.7, 1]$.

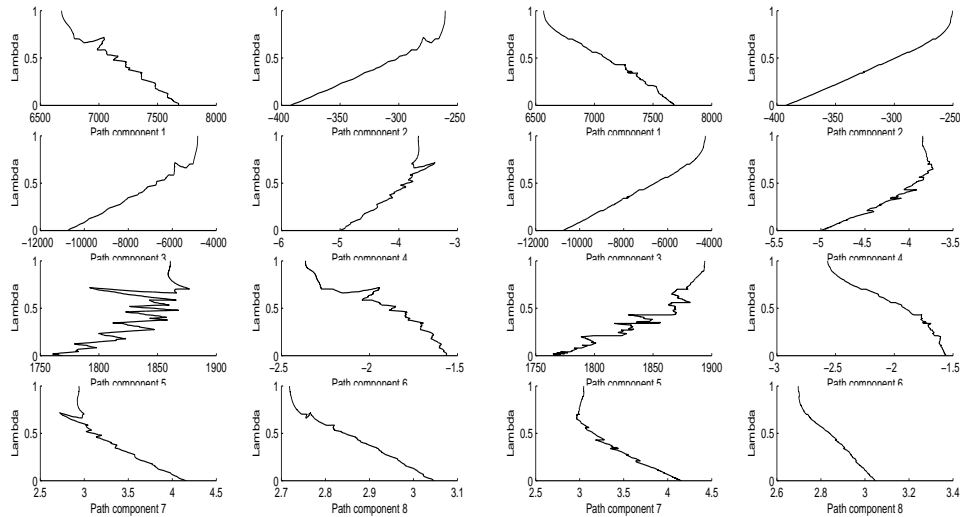
En ce qui concerne les différentes tailles pour chaque dimension, on constate que les composantes 3 et 5 tendent à avoir un pas plus grand, tandis que la composante 8 a un pas plus faible. Ceci est cohérent avec le fait que la composante 3 est celle ayant la plus forte variation relative (avec plus d'un facteur 3 entre les valeurs initiales et finales), tandis que la composante 8 varie très peu (de 1.21 à 1.08 environ). Le cas de la composante 5 est moins clair, car même en tenant compte de la mise à l'échelle, sa variation ne semble pas si importante que cela. Toutefois, on remarque que cette composante est une des plus irrégulières dans le cas uniforme, et la moins lissée dans le cas adaptatif. De plus, il se trouve que les difficultés du transfert à $0.2N$ semblent également liées à cette composante particulière (voir juste après).

Pour finir, le bénéfice additionnel du mode adaptatif, à savoir un tir final plus rapide, est ici significatif également. La norme à la fin du chemin (avant le tir) passe à 0.03 au lieu de 0.74, et le tir ne nécessite plus que 34 évaluations de l'homotopie (contre 95 dans le cas uniforme). Cela n'est pas négligeable, d'autant que les évaluations lors du tir sont plus coûteuses que lors du suivi de chemin (où on intègre avec une précision moindre).

Ces deux facteurs combinés conduisent au final à un gain important, avec un temps d'exécution total de 489s contre 855s (2090 et 3546 appels de l'homotopie respectivement).

Un cas défavorable

On souhaiterait ici mieux comprendre ce qui se passe dans le cas du $0.2N$, pour lequel l'utilisation de la triangulation adaptative conduit à un temps d'exécution supérieur à la référence. Concernant le tir final, il y a pourtant bel et bien amélioration, avec 49 appels de l'homotopie contre 83 (la norme de H à la fin du chemin étant de 0.03 au lieu de 0.24). Le problème se situe donc au niveau du suivi en lui-même, sur lequel on ne gagne quasiment pas de simplexes en adaptatif. Ceci est certainement lié aux difficultés rencontrées par les jonctions adaptatives, pour lesquelles on a augmenté le nombre de simplexes autorisés de 100 à 500. Sans cela, les jonctions échouent à répétition, et on garde pratiquement la même taille de triangulation tout du long. Pourtant même avec ce réglage plus souple, le gain en simplexes est très limité par rapport au coût des jonctions. On examine donc les deux chemins de zéros pour essayer d'y voir plus clair.



Chemin de zéros pour $0.2N$ - Modes uniforme et adaptatif

En premier lieu, on remarque que le chemin est plus lisse dans le cas adaptatif, comme pour $0.5N$. Cependant, si on observe en particulier la composante 5 de z , on note la présence de “sauts” horizontaux, qui correspondent à des jonctions qui parcourent beaucoup de simplexes. Ceci illustre le phénomène décrit en 2.3.1, quand une jonction fait sauter le suivi de chemin d'une branche à une autre. On peut alors ensuite redescendre en $\lambda = 0$, ou bien comme ici continuer vers $\lambda = 1$. Le problème est que ceci n'est pas du tout le rôle d'une jonction adaptative !

En effet, l'on est censé avoir calculé une nouvelle taille de triangulation mieux adaptée au chemin que l'on est en train de suivre. Si on saute sur une autre branche au lieu de continuer normalement ce chemin, rien

ne garantit que la nouvelle taille sera adaptée à la nouvelle branche, ce qui ruine complètement le mécanisme adaptatif. De plus, on court le risque d'une perte de précision du suivi lors de ces sauts, lorsque la jonction rejoint l'autre branche. Bien sûr, les conséquences de ce phénomène sont aggravées si cela se produit à plusieurs reprises. En plus du risque de pertes répétées de précision, on se retrouve la plupart du temps avec une taille inadaptée, car déterminée à partir de la progression *avant* le saut. Dans ces circonstances, un moindre mal est peut-être en effet d'abandonner le mode adaptatif et de garder la triangulation uniforme.

Il semblerait que l'on soit ici dans une telle situation, ce qui explique les difficultés de l'adaptatif. Imposer une limite plus stricte (50 simplexes) sur les jonctions ne sert à rien : un grand nombre d'entre elle échouent, et on parcourt au final plus de simplexes qu'en uniforme. De même, essayer un réglage plus agressif (tolérance de déviation à 1) se révèle trop instable, tandis qu'un réglage plus prudent (10^{-2}) ne fait pas mieux qu'avant.

Remarque : accessoirement, si l'on regarde l'évolution de la taille au cours du suivi, on note que le pas pour la composante 5 devient très petit, en-dessous de 10^{-2} . Ceci est probablement une des raisons pratiques du médiocre résultat de l'adaptatif sur ce chemin, et confirme que le comportement de cette inconnue (c'est-à-dire $p_{h_y}(0)$) semble très sensible.

Premières conclusions sur la triangulation adaptative

Globalement, ces tests montrent un comportement qualitatif satisfaisant de la triangulation adaptative, à deux niveaux. D'abord, on observe dans la plupart des cas une réduction significative du nombre de simplexes parcourus sur le chemin principal, avec un suivi généralement plus lisse. Le gain est remarquable dans certains cas favorables, lorsqu'on parvient à éviter certaines irrégularités du suivi. De plus, on note également une amélioration générale de la précision du suivi, ce qui mène à des tirs de fin plus rapides, avec moins d'évaluations de l'homotopie. Cependant, si les jonctions adaptatives sont le plus souvent rapides (moins de 50 simplexes), on peut aussi rencontrer des situations problématiques, peut-être liées à l'existence de plusieurs branches du chemin de zéros. Dans ce cas les jonctions causent en fait plus de mal que de bien, et il semble préférable de complètement désactiver le mode adaptatif.

Néanmoins, la comparaison des temps d'exécution globaux souligne le fait que le coût numérique des jonctions adaptatives est en pratique loin d'être négligeable. En fait, ce coût contrebalance souvent les deux gains mentionnés ci-dessus, menant au final à une amélioration limitée des performances. Généralement parlant, il y a moins à attendre des suivis qui se

déroulent déjà bien avec une grande triangulation uniforme, tandis qu'on peut espérer une bonne amélioration pour les suivis plus irréguliers (sauf dans les cas problématiques mentionnés ci-dessus).

Remarque : ces expérimentations confirment qu'un pas de 10^{-2} pour λ est préférable pour le mode adaptatif, un pas de 10^{-1} étant trop grossier.

Accessoirement, l'utilisation du raffinage de solution décrit en 2.4 fonctionne sur ces problèmes, mais souffre du même inconvénient. Le gain réalisé sur le tir final (grâce à un meilleur point de départ en $\lambda = 1$ après les raffinages) est inférieur au coût des jonctions de raffinage, et l'on a finalement un temps d'exécution plus long au total.

3.4.2 Formulations des homotopies de jonction

Pour finir, on veut comparer sur cette famille de problèmes les différentes formulations décrites en 2.2 pour les homotopies de jonction. En particulier, on souhaite souligner que la première formulation (qui utilise une approximation de la Jacobienne de H par différences finies) requiert certains réglages numériques, ce qui est une forte limitation comparé à la seconde formulation (sans dérivées). La série de tests consiste à effectuer le suivi seul (pas de tir final), avec le mode adaptatif complet. On garde les réglages par défaut, à savoir 0.1 pour la tolérance de déviation et (2, 0.2, 0.1) pour les seuils de déformation anisotrope. Comme on s'attend à des jonctions plus ou moins difficiles suivant les réglages de la première formulation, et pour garder un environnement de test cohérent, on augmente le nombre maximal de simplexes pour les jonctions à 500 pour tous les tests. Enfin, on utilise une triangulation $K_1(10^{-1}/10^{-2})$ au lieu d'une $K_1(10^{-1})$, pour avoir davantage de jonctions adaptatives et éviter de brider le mode adaptatif.

Il est à noter que si le nombre de simplexes de jonction diffère naturellement d'une formulation à l'autre, le nombre de simplexes pour le chemin principal peut lui aussi être différent. Ceci est dû au fait que les jonctions peuvent amener à différentes faces complètement étiquetées, ou bien parfois tout simplement échouer (contrairement à un échec de la première jonction, un échec de jonction adaptative ne provoque pas l'arrêt du suivi : on garde simplement la même taille qu'avant).

Première formulation

On commence par la première formulation, qui utilise des différences finies centrées, avec à la base un pas uniforme h . On rappelle que l'intégration numérique le long du chemin est réalisée par RKF45, avec des tolérances absolues et relatives de 10^{-8} et 10^{-6} .

Un pas uniforme de 10^{-6} donne de très mauvais résultats, ne convergeant que dans 2 cas sur 6, avec trois fois le nombre d'appels à H usuel. Visiblement, il faut choisir un pas plus grand. En effet, des valeurs de 10^{-3} et 10^{-1} donnent de meilleurs résultats, comme indiqué dans les tableaux ci-dessous :

- (i) **Prin.** le nombre de simplexes pour le chemin principal
- (ii) **Jonc.** le nombre total de simplexes pour les jonctions
- (iii) **Tot (H)** le nombre total d'évaluations de l'homotopie
- (iv) **Tps** le temps total (en secondes) du suivi de chemin

T_{max}	Prin.	Jonc.	Tot (H)	Tps	T_{max}	Prin.	Jonc.	Tot (H)	Tps
10N	1218	2148	4233	44	10N	1218	2398	4451	48
5N	913	1940	2901	63	5N	1143	7185	4201	92
1N	605	1546	1738	216	1N	773	662	1818	190
0.5N	1079	7971	4524	887	0.5N	1316	3725	3908	830
0.2N	1140	3027	3253	1745	0.2N	1397	871	2793	1411
0.1N	-	-	-	-	0.1N	904	425	1800	1794

Jonctions pour la première formulation, avec $h = 10^{-3}$ et $h = 10^{-1}$

On considère maintenant la première formulation modifiée, dans laquelle le pas est proportionnel à la taille de la triangulation : $h_i = \frac{\delta_i}{k}$, $i = 1, \dots, n$.

Avec $k = 1000$ et $k = 100$, on a comme auparavant de mauvais résultats, avec convergence dans 3 cas sur 6 seulement, et des temps jusqu'à deux fois la référence. Avec des valeurs de 10 et 1, c'est nettement mieux, on obtient des résultats comparables à ceux de la seconde formulation.

T_{max}	Prin.	Jonc.	Tot (H)	Tps	T_{max}	Prin.	Jonc.	Tot (H)	Tps
10N	1218	2134	4236	47	10N	1218	2490	4464	48
5N	913	1928	2904	63	5N	761	2316	2711	56
1N	547	466	1418	171	1N	559	567	1476	154
0.5N	1019	1252	2591	578	0.5N	1260	2243	3210	629
0.2N	1395	707	2776	1594	0.2N	1404	670	2696	1314
0.1N	1450	736	2810	3198	0.1N	846	586	1906	1892

Jonctions pour la première formulation, avec $k = 10$ et $k = 1$

Seconde formulation (référence)

Voici maintenant les résultats de la seconde formulation, qui ne fait pas intervenir les dérivées de l'homotopie. Ce point était important dans notre contexte, car on veut pouvoir travailler sur des homotopies peu régulières. Cette formulation est celle employée par défaut dans le code *Simplicial*, et il n'y a ici aucun paramètre numérique à régler.

T_{max}	Prin.	Jonc.	Tot (H)	Tps
10N	1383	3785	4188	45
5N	792	4129	2614	60
1N	586	447	1339	161
0.5N	1044	1069	2386	493
0.2N	1500	371	2421	1422
0.1N	1527	505	2545	2812

Jonctions pour la seconde formulation

On compare ici les différents réglages de la première formulation ($h = 10^{-3}, 10^{-1}$ et $k = 10, 1$) à la seconde formulation, en termes de nombre total d'appels à l'homotopie et de temps de suivi :

- **ratio (H)** désigne $\frac{\text{appels de H pour formulation 1}}{\text{appels de H pour formulation 2}}$

- **ratio (s)** désigne $\frac{\text{temps pour formulation 1}}{\text{temps pour formulation 2}}$

T_{max}	(H)	(s)	(H)	(s)	T_{max}	(H)	(s)	(H)	(s)
10N	1.01	0.98	1.06	1.07	10N	1.01	1.04	1.06	1.07
5N	1.11	1.05	1.61	1.53	5N	1.11	1.05	1.04	0.93
1N	1.30	1.34	1.36	1.18	1N	1.06	1.06	1.10	0.96
0.5N	1.90	1.80	1.64	1.68	0.5N	1.09	1.17	1.35	1.28
0.2N	1.34	1.23	1.15	0.99	0.2N	1.15	1.12	1.11	0.92
0.1N	-	-	0.71	0.64	0.1N	1.10	1.14	0.75	0.67

Comparaison : première ($h = 10^{-3}, 10^{-1}$ et $k = 10, 1$) / seconde formulation

On constate que pour les formulations ($h = 10^{-3}, k = 10$), à chaque T_{max} les valeurs des deux ratios sont proches, ce qui est cohérent avec le fait que l'essentiel du coût numérique vient des appels de l'homotopie. Toutefois, la mesure du temps d'exécution est en pratique plus délicate (pour des raisons matérielles, et dépend notamment de la machine de test), c'est pourquoi le nombre d'évaluations de l'homotopie est un critère plus fiable.

A l'inverse, comparer uniquement les simplexes du chemin principal ne permet pas de juger les mérites relatifs des différentes formulations. On le voit pour 0.2N, où les premières formulations utilisent moins de simplexes principaux, mais avec un grand nombre de simplexes de jonction, et au final de moins bonnes performances. Ceci vient bien sûr de fait que les jonctions font aussi des appels de H , et un suivi qui met beaucoup d'efforts dans ses jonctions peut donc être inférieur à un autre avec plus de simplexes mais des jonctions plus légères.

Cependant, il ne suffit pas non plus de compter en plus le nombre de simplexes de jonction, comme le montrent les suivis pour 10 et 5N. La seconde formulation utilise ici presque deux fois plus de simplexes de jonction, et à peu près autant de simplexes principaux, pour un nombre total

d'évaluations de H inférieur ! Ceci vient de la nature des homotopies de jonction, qui utilisent un pas de 1 sur leur paramètre homotopique λ' (cf 2.2). Les sommets en $\lambda' = 0$ ne requièrent pas d'appel à H , seulement quelques opérations matrice-vecteur, dont le coût est bien moindre. C'est pourquoi compter seulement le nombre de simplexes sur le chemin principal et les jonctions donne une comparaison faussée des différentes formulations.

Ainsi, on voit qu'en dehors du temps d'exécution, seul le nombre total d'évaluations de l'homotopie permet d'avoir une idée précise de l'efficacité relative de ces différentes formulations. Finalement, cette série de tests indique qu'il est possible de trouver des réglages corrects pour la première formulation, mais aussi que de mauvais choix peuvent être catastrophiques (échecs des jonctions d'initialisation en particulier). Ceci est ennuyeux dans la mesure où les bons réglages sont vraisemblablement dépendants des problèmes étudiés, et même de la méthode et/ou précision d'intégration. Trouver les paramètres corrects risque donc de passer par une phase de tâtonnements, ce qui est bien sûr pénible. Sur cet aspect la seconde formulation est clairement plus stable, et donne de plus souvent de meilleurs résultats que la première (sauf avec les meilleurs réglages).

Conclusion

Avec la continuation sur l'objectif et l'initialisation fournie par le tir initial à pas fixe, la méthode simpliciale nous a permis de résoudre le problème de transfert jusqu'à une poussée de $0.1N$. Les solutions obtenues coïncident avec celles de la continuation différentielle utilisée dans [26], mais avec un temps d'exécution jusqu'à trois fois plus long, ce qui était prévisible.

On procède ensuite à une comparaison de trois intégrateurs à pas variable (RKF45, DOP853 et ODEX), ce qui est l'occasion de vérifier l'extrême régularité de cette famille de problèmes par rapport à la poussée maximale. Concernant le traitement de la structure bang-bang et du grand nombre de commutations associées, il semblerait que RKF45 offre la meilleure combinaison vitesse/précision. Ceci dépend bien sûr du cadre de test choisi, et il existe peut-être des configurations plus adaptées aux deux autres intégrateurs (d'ordre plus élevé).

Les expérimentations numériques portant sur la triangulation adaptative montrent un comportement qualitatif satisfaisant, avec un suivi de chemin plus lisse. Dans les cas favorables cela conduit à un double gain, avec à la fois une réduction du nombre de simplexes suivis et un tir final plus rapide. Cependant, on observe également que le coût numérique des jonctions adaptatives est loin d'être négligeable. En fait, si le suivi se passe déjà bien avec

la triangulation uniforme, le gain global de performance peut ainsi être insignifiant. Un autre problème particulier semble être la présence de plusieurs branches proches du chemin de zéros, qui peut perturber les jonctions au point de rendre le mode adaptatif inutilisable.

Ceci suggère que la meilleure voie pour améliorer la triangulation adaptative est probablement de trouver une formulation moins coûteuse pour les jonctions, ou bien une autre façon d'effectuer les changements de taille. Pour finir, les tests indiquent également la supériorité de la formulation sans dérivées pour les jonctions. En effet, les autres formulations (qui approximent la Jacobienne de l'homotopie par différences finies) apparaissent très sensibles aux réglages numériques du pas, et sont vraisemblablement moins adaptées aux problèmes peu réguliers de toute façon.

Chapitre 4

Problèmes d'Arcs Singuliers

Parmi les problèmes de contrôle optimal, les problèmes d'arcs singuliers sont intéressants et difficiles à résoudre par les méthodes indirectes, car ils impliquent des contrôles multivalués et des inclusions différentielles. En bref, les arcs singuliers se produisent lorsque les conditions nécessaires données par la minimisation du Hamiltonien ne suffisent pas pour déterminer de façon unique le contrôle optimal sur un intervalle non réduit à un point. Dans ce cas, appliquer le Principe du Maximum de Pontriaguine conduit à un problème aux deux bouts comportant une inclusion différentielle. On garde les notations usuelles : x l'état, p l'état adjoint, $y = (x, p)$ et φ la dynamique du couple état-état adjoint. Alors si Γ désigne l'expression multivaluée du contrôle optimal, on a

$$(BVP) \begin{cases} \dot{y} \in \Phi(y) = \varphi(y, \Gamma(y)) & pp \text{ sur } [t_0, t_f] \\ \text{Conditions aux limites} \end{cases}$$

Le tir multiple peut être un moyen efficace de résoudre ce type de problèmes, mais requiert typiquement des hypothèses a priori sur la structure du contrôle (nombre et localisation approximative des arcs singuliers en particulier). Une autre possibilité, décrite par exemple dans [6], est d'utiliser une discrétisation du (BVP) avec des techniques de pénalisation intérieure (ce qui peut être vu comme une forme de continuation discrète).

L'objectif primaire de la continuation est donc ici de fournir une détection fiable de la structure singulière et une bonne initialisation, afin de pouvoir résoudre ensuite le problème par une méthode dérivée du tir multiple. La continuation est basée sur la régularisation du problème en ajoutant un terme quadratique à l'objectif (on se limite ici au cas où le Hamiltonien est linéaire par rapport au contrôle).

Remarque : les problèmes d'arcs singuliers présentent parfois une difficulté particulière appelée "chattering", ie une concaténation infinie d'arcs

singuliers et/ou bang-bang qui se produit sur un intervalle de temps fini. Cependant, nous n'avons pas rencontré ce cas dans les expérimentations sur les deux problèmes étudiés ici.

Note : on introduit dans ce chapitre plusieurs variantes des méthodes de tir simple ou multiple, dont certaines impliquent la discrétisation du couple état-état adjoint ou du contrôle. On garde alors le terme de "fonction de tir", qui peut sembler quelque peu inapproprié pour les formulations discrétisées. Toutefois, ces formulations utilisent toutes l'état adjoint p , et prennent en compte les conditions nécessaires sur le contrôle. C'est pourquoi il me semble justifié de leur garder un nom en rapport avec la classe des méthodes indirectes.

4.1 Présentation des problèmes

On étudie dans ce chapitre deux problèmes d'arcs singuliers en parallèle : un modèle de récolte de pêche optimale dû à Clark ([20]), et un problème de régulateur quadratique étudié par exemple par Chen et Huang dans [19]. On fait cela pour pouvoir observer les similarités et/ou différences de comportement par rapport à la méthode, de façon à mieux cerner ce qui est lié à la méthode ou à un problème en particulier. On espère ainsi mettre au point une méthode qui ne dépende pas trop du problème, en essayant de traiter les deux problèmes dans les mêmes conditions. Ceci est particulièrement important pour la résolution numérique, car les expérimentations montrent que les réglages donnant les meilleurs résultats pour l'un des problèmes peuvent être mal adaptés à l'autre. En fait, nous aimerions aboutir à une démarche relativement générale qui pourrait être appliquée à divers problèmes, plutôt que trouver les meilleurs paramètres dans un cas particulier.

Notations : dans ce chapitre, on utilise souvent les indices 1 et 2 pour les notations spécifiques aux Problèmes 1 et 2, en gardant les notations sans indices pour le cas général.

4.1.1 Récolte de pêche optimale

Le premier des deux problèmes étudiés est le modèle de pêche décrit par Clark dans [20], qui est également étudié par exemple par Schilling dans [34]¹. L'état $x(t) \in \mathbf{R}$ représente la population de poisson, et le contrôle $u(t) \in \mathbf{R}$ est l'activité de pêche. L'objectif est de maximiser le produit de la pêche sur une certaine période de temps, l'activité de pêche étant bornée par une certaine valeur U_{max} . Le modèle du produit de la pêche et de l'évolution

¹où le (BVP) est interprété dans le cadre des points fixes pour opérateurs multivalués

de la population est décrit dans la formulation originelle du problème :

$$(P) \begin{cases} \text{Max} \int_0^{10} (E - \frac{c}{x(t)}) u(t) dt \\ \dot{x}(t) = rx(t) (1 - \frac{x(t)}{k}) - u(t) \\ 0 \leq u(t) \leq U_{max} \quad \forall t \in [0, 10] \\ x(0) = 70 \cdot 10^6 \quad x(10) \text{ libre} \end{cases}$$

avec $E = 1$, $c = 17.5 \cdot 10^6$, $r = 0.71$, $k = 80.5 \cdot 10^6$ et $U_{max} = 20 \cdot 10^6$.

Il est à noter que les valeurs numériques des constantes ci-dessus sont telles que le terme $(E - \frac{c}{x(t)})$ est en pratique toujours positif, ce qui garantit un produit de pêche positif. Incidemment, on pourrait vouloir ajouter une contrainte d'état du type $x(t) > 0$, $\forall t \in [0, 10]$, car une population de poisson négative semble avoir peu de sens... Cependant, dans toutes les expérimentations numériques dans la suite on ne rencontre jamais ce cas de figure. Il va d'ailleurs sans dire que prendre en compte explicitement cette contrainte rendrait le problème nettement plus compliqué.

On modifie maintenant légèrement cette modélisation pour obtenir une formulation similaire à celle des autres problèmes étudiés. En premier lieu on transforme le problème de maximisation en un problème de minimisation d'objectif opposé (on rappelle que $(\frac{c}{x(t)} - E < 0)$, ce qui interviendra dans la suite, voir 4.2 page 92). On remplace ensuite l'expression du contrôle $u(t)$ par $u(t) U_{max}$, en récrivant la contrainte sur le contrôle sous la forme $0 \leq u(t) \leq 1 \quad \forall t \in [0, 10]$. On travaille donc en pratique sur le problème

$$(P_1) \begin{cases} \text{Min} \int_0^{10} (\frac{c}{x(t)} - E) u(t) U_{max} dt \\ \dot{x}(t) = rx(t) (1 - \frac{x(t)}{k}) - u(t) U_{max} \\ 0 \leq u(t) \leq 1 \quad \forall t \in [0, 10] \\ x(0) = 70 \cdot 10^6 \quad x(10) \text{ libre} \end{cases}$$

On applique le Principe du Maximum de Pontriaguine, avec l'expression du Hamiltonien

$$\mathcal{H}_1(t, x, p, u) = \left(\frac{c}{x(t)} - E \right) u(t) U_{max} + p(t) \left(rx(t) \left(1 - \frac{x(t)}{k} \right) - u(t) U_{max} \right)$$

qui donne la dynamique de l'état et état adjoint

$$\begin{cases} \dot{x}(t) = r x(t) (1 - \frac{x(t)}{k}) - u(t) U_{max} \\ \dot{p}(t) = \frac{c}{x^2(t)} u(t) U_{max} - p(t) r \left(1 - \frac{2x(t)}{k} \right). \end{cases}$$

On a les conditions aux limites

$$\begin{cases} x(0) = 70 \cdot 10^6 & p(0) \text{ libre} \\ x(10) \text{ libre} & p(10) = 0. \end{cases}$$

Pour le contrôle optimal u^* , la minimisation du Hamiltonien mène à la fonction de commutation ψ :

$$\psi(t) = \frac{c}{x(t)} - E - p(t),$$

avec le contrôle optimal bang-bang correspondant

$$\begin{cases} u^*(t) = 0 & \text{si } \psi(t) > 0 \\ u^*(t) = 1 & \text{si } \psi(t) < 0 \\ u^*(t) \in [0, 1] & \text{si } \psi(t) = 0. \end{cases}$$

Le dernier cas indique normalement une commutation, où le contrôle est discontinu et passe d'un arc a un autre quand ψ s'annule. Cela n'est pas très gênant en soi si cela n'arrive qu'en des instants isolés, mis a part qu'un nombre élevé de commutations peut rendre l'intégration numérique du (BVP) délicate et ainsi entrainer des difficultés pour l'application de la méthode de tir (voir 1.1.1). Néanmoins, on a vu au chapitre 3 que l'utilisation d'un intégrateur à pas variable et d'une continuation bien choisie pouvait donner de bons résultats.

Cependant, on étudie ici précisément le cas des arcs singuliers, et l'on s'attend donc à ce que la fonction de commutation ψ s'annule sur des intervalles non triviaux. Les conditions nécessaires données par la minimisation du Hamiltonien ne permettent alors pas à elles seules de déterminer le contrôle sur ces arcs singuliers, ce qui est bien sûr plus problématique qu'un nombre fixé de discontinuités.

On essaye donc naturellement d'extraire des informations sur le contrôle à partir du fait que les dérivées successives de la fonction de commutation doivent s'annuler sur un arc singulier.

Remarque : on rappelle que t est omis par souci de clarté dans les équations suivantes, où il n'apparaît pas en lui-même de toute façon. On note ainsi x, p, u, ψ au lieu de $x(t), p(t), u(t), \psi(t)$.

La dérivée première de la fonction de commutation est de la forme

$$\dot{\psi} = r \left(-\frac{c}{x} + \frac{c}{k} + p - \frac{2 p x}{k} \right).$$

Il se trouve que u a disparu de cette expression, et la relation $\dot{\psi} = 0$ ne donne donc pas d'information sur le contrôle. On regarde alors la dérivée seconde

$$\ddot{\psi} = r^2 \left(-\frac{2}{kr} \left(\frac{c}{x} - p \right) u U_{max} + \frac{c}{x} - \frac{c}{k} - p + \frac{2 p x}{k} - \frac{2 p x^2}{k^2} \right).$$

Cette fois heureusement, la relation $\dot{\psi} = 0$ donne l'expression du contrôle singulier

$$u_{\text{singulier}}^* = \frac{k r}{2 \left(\frac{c}{x} - p\right) U_{\text{max}}} \left(\frac{c}{x} - \frac{c}{k} - p + \frac{2px}{k} - \frac{2px^2}{k^2} \right).$$

Tout le problème est bien sûr que cette expression n'est utilisable que si l'on *sait* que l'on est sur un arc singulier. Un autre point est qu'il n'est pas évident d'après cette formule que $u_{\text{singulier}}^*$ vérifie toujours la contrainte $0 \leq u(t) \leq 1, \forall t \in [0, 10]$. En pratique toutefois, cette expression algébrique du contrôle singulier est seulement utilisée pour la résolution précise du problème décrite plus loin en 4.4. Dans notre cas l'initialisation fournie par la continuation (cf 4.2 et 4.3 ci-après) est en fait suffisamment proche de la solution pour que l'on ne rencontre pas de tels contrôles singuliers "hors limites".

Remarque : incidemment, on obtient en combinant les trois équations $\psi, \dot{\psi}, \ddot{\psi} = 0$ les relations $\dot{x} = 0$ et $\dot{p} = 0$. Ceci indique que le l'état et l'état adjoint sont constants sur un arc singulier, et donc le contrôle singulier l'est aussi. Cette propriété est bien évidemment spécifique au problème, et ne constitue en aucun cas un résultat général.

4.1.2 Régulateur quadratique

Le second problème d'arc singulier que l'on étudie ici est un problème de régulateur quadratique étudié par exemple dans [19] par Chen et Huang (qui utilisent également une approche par perturbation, mais avec une continuation discrète basée sur l'équation différentielle de Davidenko). L'état est cette fois de dimension 2, tandis que le contrôle est toujours de dimension 1. L'objectif est de minimiser le carré de la norme L_2 de l'état.

$$(P_2) \begin{cases} \text{Min } \frac{1}{2} \int_0^5 (x_1^2(t) + x_2^2(t)) dt \\ x_1(t) = x_2(t) \\ \dot{x}_2(t) = u(t) \\ -1 \leq u(t) \leq 1 \quad \forall t \in [0, 5] \\ x(0) = (0, 1) \quad x(5) \text{ libre.} \end{cases}$$

On a l'expression du Hamiltonien

$$\mathcal{H}_2(t, x, p, u) = \frac{1}{2}(x_1^2(t) + x_2^2(t)) + p_1(t)x_2(t) + p_2(t)u(t),$$

qui donne la dynamique de l'état et de l'état adjoint

$$\begin{cases} \dot{x}_1(t) = x_2(t) \\ \dot{x}_2(t) = u(t) \\ \dot{p}_1(t) = -x_1(t) \\ \dot{p}_2(t) = -p_1(t) - x_2(t). \end{cases}$$

Les conditions aux limites sont

$$\begin{cases} x(0) = (0, 1) & p(0) \text{ libre} \\ x(5) \text{ libre} & p(5) = (0, 0). \end{cases}$$

La minimisation du Hamiltonien conduit immédiatement à la fonction de commutation

$$\psi(t) = p_2(t),$$

et au contrôle optimal bang-bang correspondant

$$\begin{cases} u^*(t) = - \text{sign } p_2(t) & \text{si } \psi(t) \neq 0 \\ u^*(t) \in [-1, 1] & \text{si } \psi(t) = 0. \end{cases}$$

Une fois encore, on s'attend à ce que ψ s'annule sur des intervalles non triviaux, et l'on essaye d'obtenir l'expression du contrôle singulier à partir des dérivées successives de ψ . Ce qui donne au premier ordre

$$\dot{\psi}(t) = -p_1(t) - x_2(t).$$

Comme pour le Problème 1, cette équation ne donne pas d'information sur le contrôle, et on passe au second ordre

$$\ddot{\psi}(t) = x_1(t) - u(t),$$

qui donne immédiatement l'expression du contrôle optimal via $\ddot{\psi} = 0$:

$$u_{\text{singular}}^*(t) = x_1(t).$$

On fait ici la même remarque que précédemment concernant la contrainte sur le contrôle, qui n'est pas automatiquement vérifiée par cette expression. Cependant, on est en pratique dans la même situation que pour le problème (P_1) , et on ne rencontre pas de contrôles hors limites durant la résolution.

4.2 Perturbation quadratique

Comme pour le problème de transfert orbital étudié au chapitre précédent, la résolution directe de ces problèmes par le tir simple n'est pas possible, ici à cause de la présence d'arcs singuliers. De plus, l'utilisation de méthodes plus évoluées comme le tir multiple nécessiterait certaines informations qualitatives sur la structure de contrôle, plus précisément le nombre et la localisation approchée des arcs singuliers. Il nous faudrait également une bonne initialisation pour les noeuds du tir multiple, en particulier pour l'état adjoint, qui n'a pas d'interprétation physique évidente, et est donc difficile à estimer a priori. On essaye donc d'utiliser une démarche de continuation comme précédemment, mais cette fois plus spécifiquement pour déterminer l'allure de la structure singulière, et pour avoir une approximation correcte de l'état et état adjoint à la solution. On pourra alors utiliser cette information pour résoudre le problème avec précision par des méthodes de type tir multiple.

4.2.1 Formulation

Le contexte est ici un peu différent des transferts orbitaux, où la difficulté principale était le nombre élevé de commutations à poussée faible. A la fois (P_1) et (P_2) sont à temps final fixé, et la difficulté réside dans les arcs singuliers. L'objectif de la continuation est donc de régulariser le problème pour se débarrasser des arcs singuliers, sachant que l'on veut tout de même obtenir des informations fiables sur la structure singulière à la solution. On essaye alors d'ajouter une perturbation quadratique à l'objectif originel, comme fait par exemple par Chen et Huang pour (P_2) dans [19] (la différence étant que l'on utilise ici une méthode simpliciale pour mener la continuation, au lieu d'une continuation discrète basée sur l'équation différentielle de Davidenko). Une conséquence de cette perturbation est que le Hamiltonien devient strictement convexe par rapport au contrôle, ce qui empêche l'apparition des arcs singuliers. On espère qu'en utilisant le paramètre de régularisation comme paramètre homotopique, la continuation nous permettra de prédire l'apparition des arcs singuliers.

Dans toute la suite on essaye de traiter de la même façon (P_1) et (P_2) , l'idée étant d'éviter d'avoir une démarche trop dépendante du problème. Voici les objectifs régularisés que l'on utilise pour la continuation :

$$\text{Min} \int_0^{10} \left(\frac{c}{x(t)} - E \right) (u(t) - (1 - \lambda) u^2(t)) U_{max} dt, \quad \lambda \in [0, 1]$$

et

$$\text{Min} \frac{1}{2} \int_0^5 (x_1^2(t) + x_2^2(t)) + (1 - \lambda)u^2(t) dt, \quad \lambda \in [0, 1].$$

Pour le problème (P_1) , comme mentionné précédemment le terme $\frac{c}{x(t)} - E$ est toujours négatif, donc le signe moins devant $(1 - \lambda) u^2(t) U_{max}$ revient bien à "ajouter" un terme quadratique, comme pour (P_2) . On obtient alors deux familles de problèmes aux deux bouts paramétrées par λ , notées $(BVP_1)_\lambda$ et $(BVP_2)_\lambda$ respectivement. L'objectif originel (non perturbé) correspond bien au cas $\lambda = 1$.

Si l'on revient aux résultats du 1.2.1 sur la minimisation du Hamiltonien, on a les domaines suivants pour les contrôles :

$$U_1 = [0, 1] \quad \text{et} \quad U_2 = [-1, 1],$$

et les Hamiltoniens sont (t est omis pour plus de clarté) :

$$\mathcal{H}_1(t, x, p, u) = \left(\frac{c}{x} - E \right) (u - (1 - \lambda)u^2)U_{max} + p(rx \left(1 - \frac{x}{k} \right) - u U_{max})$$

et

$$\mathcal{H}_2(t, x, p, u) = \frac{1}{2}(x_1^2 + x_2^2 + (1 - \lambda)u^2) + p_1x_2 + p_2u.$$

On voit que \mathcal{H}_1 et \mathcal{H}_2 sont tous deux continus, et convexes par rapport à u (car $\frac{c}{x} - E < 0$). Donc pour les deux problèmes (P_1) et (P_2) , le Théorème 1 et Corollaire 1 s'appliquent, donc Γ_1 et Γ_2 sont semicontinues supérieurement, et à valeurs convexes compactes non vides. On note également que pour $\lambda < 1$, dans les deux cas le Hamiltonien est strictement convexe, donc le contrôle optimal est une fonction continue, d'après le Corollaire 1.

On rappelle la dynamique de l'état et état adjoint pour $(BVP_1)_\lambda$ et $(BVP_2)_\lambda$:

$$\varphi_1(y, u, \lambda) = \left(\begin{array}{c} r x (1 - \frac{x}{k}) - u U_{max} \\ \frac{c}{x^2} (u - (1 - \lambda)u^2)U_{max} - p r (1 - \frac{2x}{k}) \end{array} \right)$$

et

$$\varphi_2(y, u, \lambda) = \left(\begin{array}{c} x_2 \\ u \\ -x_1 \\ -p_1 - x_2 \end{array} \right).$$

On considère la dynamique multivaluée $\Phi(y, \lambda) = \varphi(y, \Gamma(y), \lambda)$. D'après l'expression de φ_1 et φ_2 , et le fait que Γ_1 et Γ_2 sont scs et à valeurs convexes compactes non vides, on a que Φ_1 et Φ_2 sont aussi scs et à valeurs convexes compactes non vides. On suppose maintenant que les solutions de $(BVP_1)_\lambda$ et $(BVP_2)_\lambda$ restent dans des compacts fixés. D'après le Théorème 5, on sait que d'une suite de solutions (y_{λ_k}) telle que λ_k tende vers 1, il est possible d'extraire une sous-suite (y_k) convergeant uniformément vers une solution y du (BVP) , avec (\dot{y}_k) convergeant *-faiblement vers \dot{y} dans $L_n^\infty([0, t_f])$.

De plus, comme on a les expressions du contrôle suivantes :

$$\begin{cases} u_\lambda(t) = \frac{1}{U_{max}}(r x_\lambda(t) (1 - \frac{x_\lambda(t)}{k}) - \dot{x}_\lambda(t)) & \text{pour } (P_1) \\ u_\lambda(t) = \dot{x}_{2_\lambda}(t) & \text{pour } (P_2). \end{cases}$$

le Corollaire 2 s'applique et donne également la convergence *-faible pour la sous-suite du contrôle (u_k) . Toutefois, l'existence de ces sous-suites convergentes ne suffit pas à garantir en pratique le succès de la continuation, comme on le constate lors des expérimentations numériques.

Concernant le Théorème 6 (cf page 30), on fait les hypothèses sur le suivi de chemin que toutes les faces générées par l'algorithme restent dans $K \times [0, 1]$ avec K compact, et que l'algorithme ne redescend pas en zéro. Dans les deux cas l'homotopie H est semicontinue supérieurement, à valeurs compactes et acycliques. Ce dernier point implique que H est à valeurs connexes, ce qui donne la convexité en dimension 1 mais pas en dimension

2. Les hypothèses sur l’homotopie sont donc vérifiées pour (P_1) , mais pas pour (P_2) a priori. Ce point mérite probablement une étude plus approfondie, mais ceci impliquerait une bonne maîtrise du degré topologique... En pratique, les suivis de chemins convergent pour les deux problèmes, mais on rencontre néanmoins des difficultés liées à la présence d’arcs singuliers, comme on va le voir.

Remarque : accessoirement, l’on pourrait se demander ce qui se passe avec le problème (P_1) si on met un signe plus devant la perturbation, “comme pour le transfert orbital et (P_2) ”. Et bien, le Hamiltonien correspondant est concave, et non convexe, et conduit à un contrôle discontinu. La continuation n’est alors pas très intéressante en pratique, pour ne pas dire plus...

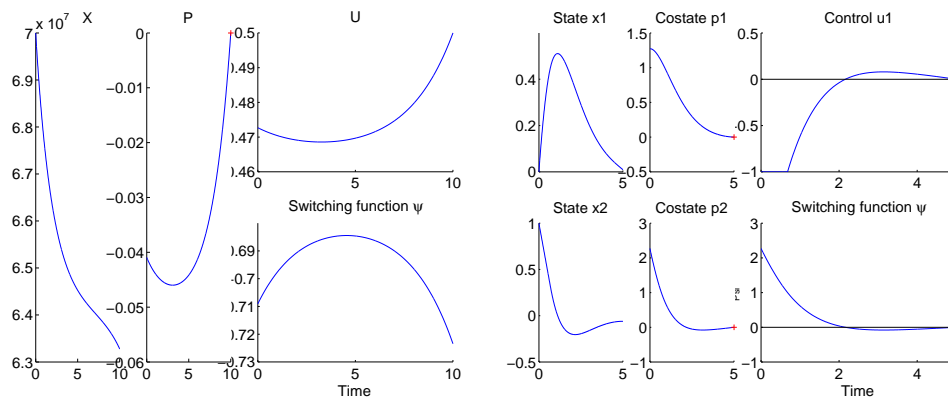
4.2.2 Suivi de chemin

Initialisation

Pour initialiser la continuation, on doit résoudre les deux problèmes pour $\lambda = 0$, ce qui est facilement réalisé par tir simple. On utilise ici une intégration de type RK4 à pas fixe (1000), les résultats étant similaires avec des intégrateurs à pas variable. La convergence du tir simple est immédiate dans les deux cas, avec un point initial trivial, c’est-à-dire $z_0 = 0.1$ et $z_0(0, 0)$ respectivement. Voici les résultats du tir et les solutions correspondantes en $\lambda = 0$.

	z_0	z^*	$ S_0(z^*) $	objectif	iter	temps
Problème 1	0.1	$-4.0935 \cdot 10^{-2}$	$3.6295 \cdot 10^{-16}$	69374046	39	< 1s
Problème 2	(0, 0)	(1.2733, 2.2715)	$3.3596 \cdot 10^{-14}$	0.4388	134	< 1s

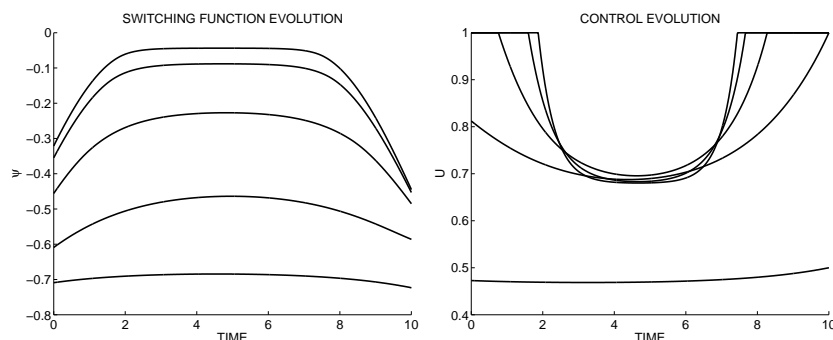
Résultats du tir simple en $\lambda = 0$.



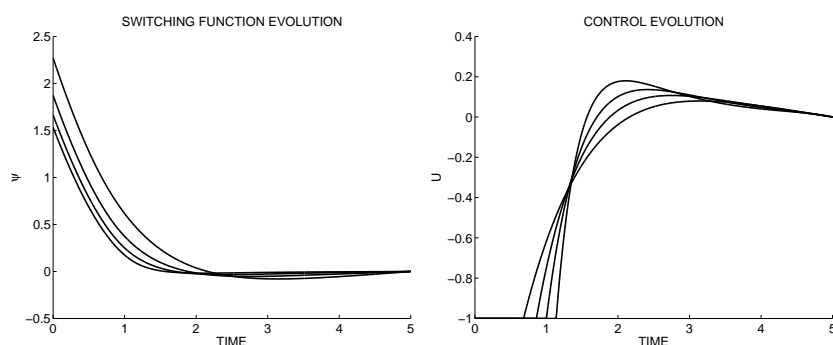
Solutions pour (P_1) et (P_2) en $\lambda = 0$

Détection de la structure singulière

Pour les deux problèmes, le suivi commence bien, et l'évolution de la fonction de commutation et du contrôle est très intéressante, comme on le voit ci-dessous.



(P_1) : fonction de commutation et contrôle pour $\lambda = 0, 0.5, 0.75, 0.9$ et 0.95



(P_2) : fonction de commutation et contrôle pour $\lambda = 0, 0.5, 0.75$ et 0.9

On voit clairement que dans les deux cas, la fonction de commutation ψ se rapproche de 0 sur certains intervalles lorsque λ augmente. Ceci suggère fortement la présence d'arcs singuliers sur ces intervalles à la solution en $\lambda = 1$, et donne déjà une idée des structures de contrôle.

Pour (P_1), la structure semble être du type *régulier-singulier-régulier*, avec l'arc singulier localisé autour de l'intervalle $[2, 7.5]$. Pour (P_2), on voit plutôt quelque chose du type *régulier-singulier*, l'arc singulier commençant vers $t = 1.5$.

Par ailleurs, en regardant l'évolution du contrôle on constate qu'en dehors des arcs singuliers suspectés, le contrôle tend pour les deux problèmes vers une structure de type bang-bang, ce qui est cohérent avec les conditions nécessaires pour $\lambda = 1$. Plus précisément, on a deux arcs $u = +1$ de part et d'autre de l'arc singulier pour (P_1), et un arc $u = -1$ avant l'arc singulier pour (P_2).

Un autre point intéressant à propos du contrôle est qu'il semble garder des valeurs intérieures sur les intervalles de temps où ψ se rapproche de 0. Comme les conditions nécessaires en $\lambda = 1$ mènent au contraire à un contrôle bang-bang, ceci est un indice supplémentaire de la présence d'arcs singuliers sur ces intervalles.

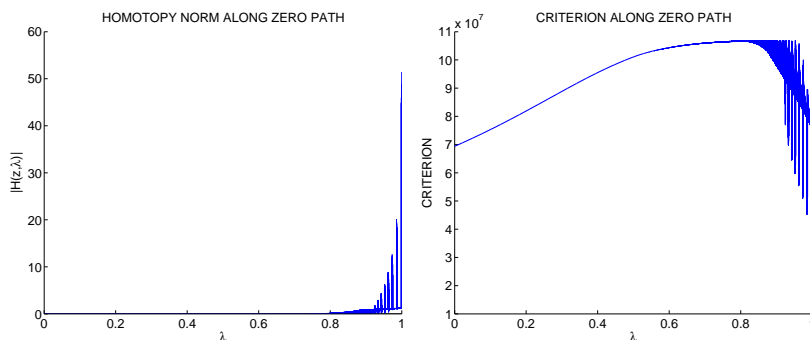
A ce moment, la continuation basée sur la perturbation quadratique de l'objectif semble donc très prometteuse. Pour les deux problèmes, on a déjà de forts indices sur la structure singulière, avec une idée de la localisation approximative des arcs singuliers suspectés. Sur le plan de la détection des arcs singuliers, cette approche semble donc plutôt efficace.

Difficultés numériques

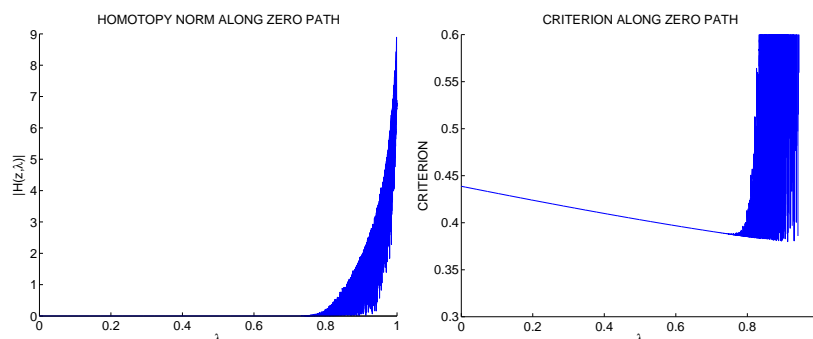
Toutefois, si l'on poursuit la continuation, le suivi de chemin rencontre des difficultés quand λ tend vers 1. Pour les deux problèmes, à partir d'un certain point, l'approximation affine par morceaux de l'homotopie se dégrade de plus en plus. Ce phénomène est illustré sur les deux graphes à gauche ci-dessous, où l'on voit la norme de l'homotopie le long du chemin de zéros augmenter brutalement, avec un comportement très oscillatoire.

De plus, on note également qu'à partir de là, la valeur de l'objectif ne s'améliore plus, et montre le même genre de dégradation brutale avec de fortes oscillations. On rappelle que sur tous les graphes, les objectifs représentés sont ceux correspondant aux problèmes originels non perturbés

$$\begin{cases} J_1 = \text{Max} \int_0^{10} \left(E - \frac{c}{x(t)} \right) u(t) U_{max} dt \\ J_2 = \text{Min} \frac{1}{2} \int_0^5 (x_1^2(t) + x_2^2(t)) dt \end{cases}$$



Norme de l'homotopie et objectif le long du chemin de zéros - Problème 1



Norme de l'homotopie et objectif le long du chemin de zéros - Problème 2

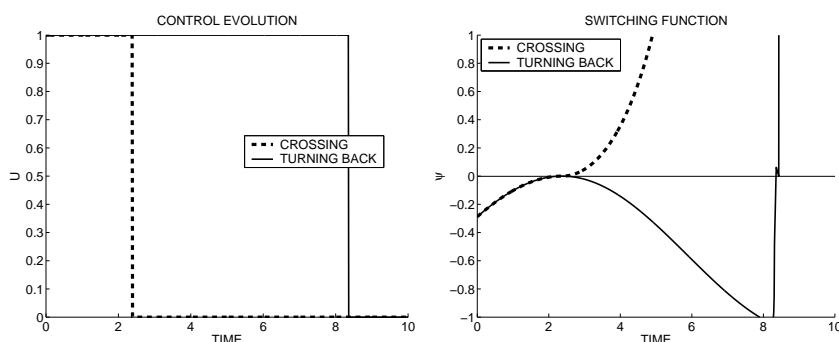
On s'attend souvent à des difficultés à la fin des approches de continuation, et d'autant plus ici en raison de la présence d'arcs singuliers. Pour les méthodes simpliciales, on peut penser à utiliser des triangulations raffinées (comme J_3 et J_4), et espérer que la taille décroissante de la triangulation permette de garder un suivi plus précis jusqu'à la convergence. Dans ce cas, cependant, l'utilisation de ces techniques ne fait que repousser la dégradation constatée, mais n'empêche pas son apparition. Accessoirement, les tests indiquent que la valeur de λ à laquelle la dégradation se manifeste est dépendante du problème. Avec les triangulations raffinées, on localise le seuil d'instabilité aux alentours de $\lambda = 0.975$ pour (P_1) et un peu plus tôt, vers $\lambda = 0.95$, pour (P_2).

4.2.3 Deux structures de contrôle différentes

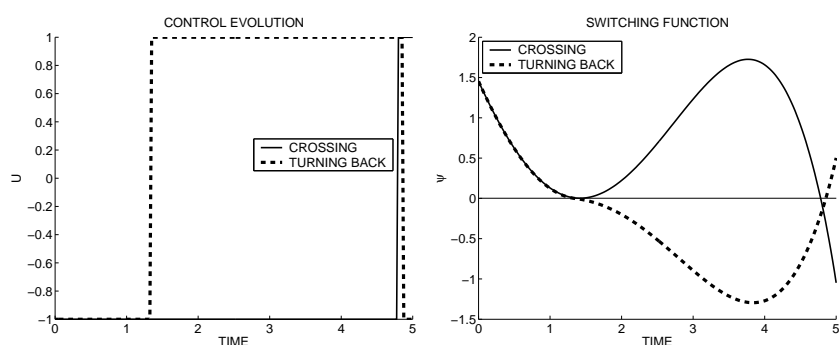
Comme l'approximation affine par morceaux de l'homotopie est obtenue par interpolation sur les étiquetages des sommets, on examine de plus près les trajectoires correspondantes. Et si l'on regarde les structures de contrôle aux sommets, on trouve en effet la raison de la dégradation de H_T . Sur la dernière face complètement étiquetée, on voit nettement deux structures de contrôle différentes suivant les sommets. Cette différence introduit naturellement la même séparation sur les étiquetages, et explique la médiocrité de l'approximation.

On montre ci-dessous pour les deux problèmes les deux types de structure de contrôle qu'on trouve parmi les sommets de la dernière face complètement étiquetée en $\lambda = 1$. Pour chaque problème, les deux structures possibles sont de type bang-bang, et il n'y a plus trace d'arc singulier.

Note : les termes "crossing" et "turning back" appliqués aux structures de contrôle sont expliqués dans la suite.



Deux structures différentes pour le contrôle aux sommets - Problème 1



Deux structures différentes pour le contrôle aux sommets - Problème 2

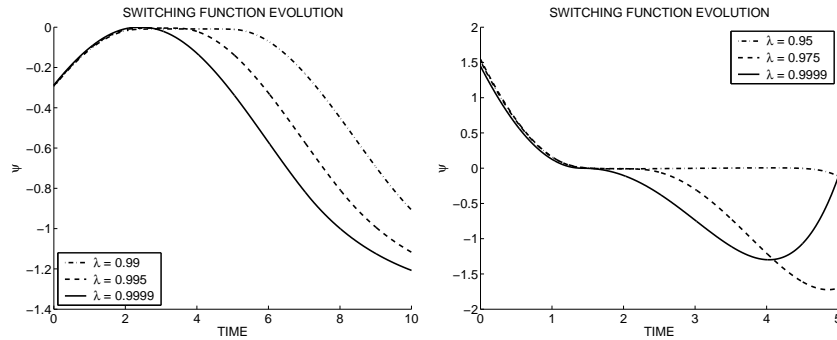
On remarque, toutefois, que les deux structures sont confondues jusqu'à un certain instant, à savoir lorsque la fonction de commutation ψ s'annule. On a alors deux cas possibles suivant le signe de ψ après ce zéro, avec les deux arcs bang opposés. Plus précisément, dans un cas la fonction de commutation reste de même signe qu'avant, et l'arc bang initial continue. Dans l'autre cas le signe de ψ change, et l'arc aussi : on a une commutation sur le contrôle. Après ce point, les deux trajectoires sont distinctes et se poursuivent chacune de leur côté. On note que ces deux structures continuent d'apparaître parmi les sommets des faces complètement étiquetées, aussi petite que soit la taille de triangulation. Ceci explique pourquoi les triangulations raffinées ne parviennent pas à empêcher la dégradation.

On peut donc raisonnablement penser que la dégradation du suivi de chemin est due à l'apparition, à un certain niveau, de deux structures de contrôle distinctes, numériquement confondues pour l'inconnue de l'homotopie z . Il est remarquable que les deux problèmes étudiés partagent ce comportement particulier, simplement avec un seuil d'apparition différent (ce qui n'est pas surprenant). Dans la suite on se réfère à ces deux structures par les termes "rebroussement (turning back)" et "traversée (crossing)", par rapport au comportement de la fonction de commutation après son annulation.

4.2.4 Instabilité du pseudo-arc près de $\lambda = 1$

On a donc vu que la dégradation du suivi est due à l'apparition de deux structures de contrôle, liées au comportement de la fonction de commutation. Comme indiqué précédemment, on remarque au début de la continuation que ψ se rapproche de 0 sur certains intervalles, que l'on appelle des "pseudo arcs singuliers". On constate que ces pseudo arcs semblent numériquement instables : ψ reste proche de 0 un certain temps, puis décroche et augmente en valeur absolue, avec un signe positif ou négatif. Suivant le signe de ψ à la sortie du pseudo arc, on obtient une structure de type soit "rebroussement" soit "traversée".

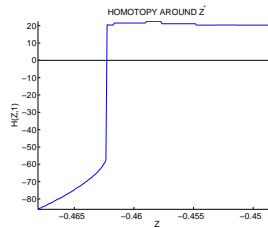
Maintenant, si l'on examine l'évolution de la fonction de commutation au-delà du seuil d'instabilité, on voit plus clairement ce qui se produit. En bref, l'instabilité numérique de la fonction de commutation près de 0 empire quand λ tend vers 1. La longueur des pseudo arcs singuliers diminue, le décrochage se produisant de plus en plus tôt.



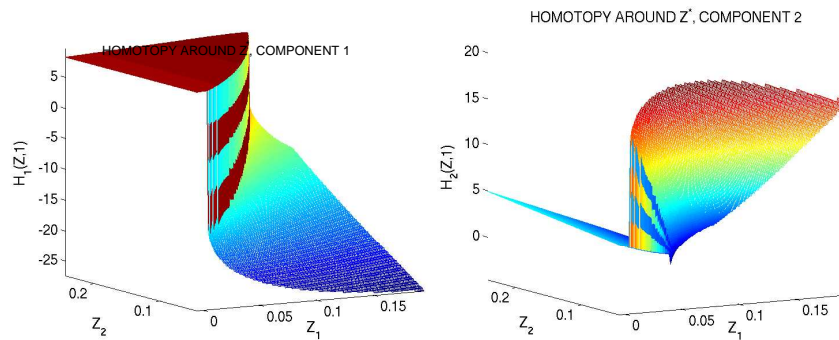
Instabilité du pseudo arc singulier près de la convergence

Problème 1 ($\lambda = 0.99, 0.995, 0.9999$) et Problème 2 ($\lambda = 0.95, 0.975, 0.9999$)

À la convergence en $\lambda = 1$, tout ce qu'il reste de l'arc singulier est un point de contact où ψ s'annule. Suivant le signe de la fonction de commutation après ce point, on a les deux structures bang-bang "rebroussement" et "traversée" correspondantes. On représente maintenant la fonction de tir au voisinage de la solution z_1^* donnée par le suivi de chemin, pour (P_1) et (P_2) .



Fonction de tir - discontinuité à l'arc singulier - Problème 1



Fonction de tir - discontinuité à l'arc singulier - Problème 2

Dans les deux cas on voit clairement la frontière qui sépare les régions correspondant aux deux structures de contrôle différentes. La présence d'arcs singuliers se manifeste par une discontinuité de la fonction de tir à la solution. Comme indiqué tout au début, ces résultats correspondent à une intégration RK4 avec 1000 pas. Toutefois, augmenter le nombre de pas à 10000, ou utiliser les trois intégrateurs à pas variable (RKF45, DOP853, ODEX) donne des résultats similaires. La seule différence que nous avons notée en changeant d'intégration est que pour (P_2) le suivi peut converger vers différents points de la "frontière" en $\lambda = 1$. On note également que la convergence est plus difficile à atteindre pour (P_2) , et l'on doit souvent utiliser en pratique une triangulation J_4 au lieu d'une J_3 plus précise. Ceci est peut-être lié au fait que le Théorème 6 s'applique pour (P_1) , mais pas pour (P_2) , comme mentionné précédemment.

En conclusion, on peut dire que cette première continuation a rencontré un succès mitigé. Bien sûr, en observant l'évolution de la fonction de commutation et du contrôle avant le seuil d'instabilité, on a de fortes indications sur la présence d'arcs singuliers. Toutefois, en raison de cette instabilité, on perd la structure singulière avant d'atteindre la convergence, où l'on obtient une solution incorrecte et inutile.

4.3 Approche BVP discrétisé

4.3.1 Formulation

On a donc vu que la continuation avec une perturbation quadratique de l'objectif est prometteuse, mais souffre d'instabilités numériques lors de l'intégration de l'(IVP) quand le problème devient presque singulier. On cherche à contourner cette difficulté, et on introduit une nouvelle approche, inspirée par le principe du tir multiple. L'idée est à la base de discrétiser les équations du problème aux deux bouts (comme par exemple dans [7], avec une méthode de Runge-Kutta symplectique). On utilise ici d'abord un schéma d'Euler pour l'état et l'état adjoint, avec un contrôle constant

par morceaux, toujours obtenu par les conditions nécessaires. Les valeurs de l'état et l'état adjoint aux noeuds de discrétisation deviennent donc une inconnue additionnelle de la fonction de tir, tandis que l'on a les conditions de raccordement (matching conditions) suivantes à vérifier en ces noeuds, d'après le schéma d'Euler :

$$Match_{cond} \begin{cases} x_i - (x_{i-1} + h \frac{\partial x}{\partial t}(t_{i-1}, x_{i-1}, p_{i-1}, u_{i-1}^*)), & \forall i \in [1..d] \\ p_i - (p_{i-1} + h \frac{\partial p}{\partial t}(t_{i-1}, x_{i-1}, p_{i-1}, u_{i-1}^*)) \end{cases}$$

où le contrôle optimal u_i^* est obtenu à partir de (x_i, p_i) par les conditions nécessaires usuelles, et donne la valeur constante du contrôle sur l'intervalle $[t_i, t_{i+1}]$. L'idée est que, même si le contrôle obtenu sur les arcs singuliers est incorrect, on espère avoir tout de même une bonne approximation de l'état et de l'état adjoint.

En termes de propriétés de convergence, grâce à ce choix d'intégration très simple, la nouvelle fonction de tir est à valeurs convexes compactes. Ceci nous permet d'espérer un bon comportement du suivi, d'après le Théorème 6 (cf 1.2.2).

La fonction de tir associée au (BVP) discrétisé est de la forme

$$S_D : \mathbf{R}^{n+(2n) \times d} \rightarrow \mathbf{R}^{n+(2n) \times d}$$

où n est la dimension de l'état (et état adjoint), et d le nombre de noeuds de discrétisation. On résume ici la composition de l'inconnue et de la valeur de la fonction de tir :

Inconnue z

- inconnue de l'IVP en t_0 (comme pour le tir simple)
- valeurs de (x^i, p^i) aux instants intérieurs t_i , $i \in [1..d]$ (on a $t_d = t_f$)

inconnue de l'IVP en t_0	(x^1, p^1)	...	(x^d, p^d)
----------------------------	--------------	-----	--------------

Valeur $S_D(z)$

- conditions de raccordement aux instants intérieurs
- conditions aux limites en t_f (comme pour le tir simple)

$Match_{cond}(t_1)$	$Match_{cond}(t_2)$...	Conditions en t_f
---------------------	---------------------	-----	---------------------

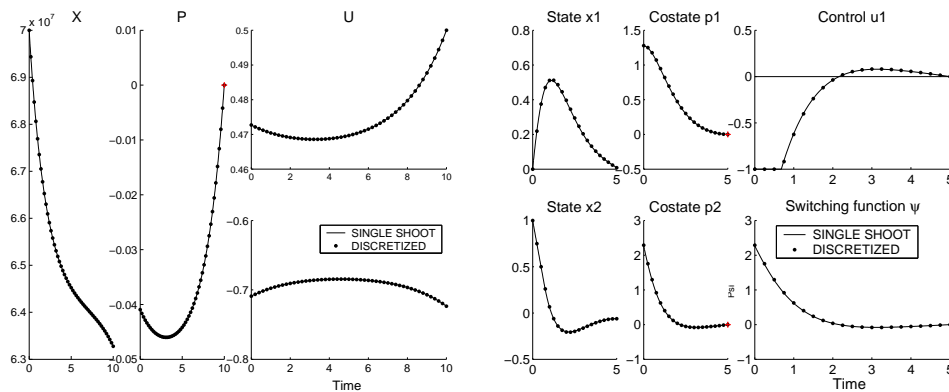
Un inconvénient majeur de cette formulation est que le couple état-état adjoint est discrétisé. Ceci limite drastiquement le nombre de noeuds de discrétisation, sinon la grande dimension de l'inconnue conduit à des temps d'exécution prohibitifs. Accessoirement, ceci impose également certaines restrictions sur l'utilisation de triangulations fines, pour les mêmes raisons de coût numérique.

Note : Dans toute la suite, on utilise 50 noeuds pour (P_1) , et 20 pour (P_2) , dont l'état est de dimension 2. Même ces discrétisations grossières mènent à des tailles totales de 101 et 82 respectivement pour le suivi de chemin...

4.3.2 Suivi de chemin

On applique avec cette formulation la même continuation qu'avant, avec la perturbation quadratique de l'objectif. Une fois encore, résoudre les problèmes en $\lambda = 0$ est immédiat (moins de 1 seconde, avec une solution en-dessous de 10^{-15}), en partant des initialisations triviales

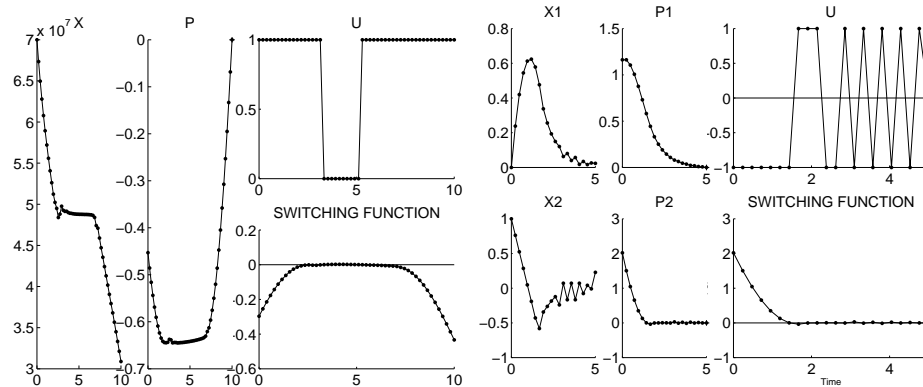
$$\begin{cases} (P_1) & p(0) = 0, & (x^i = x_0 = 7 \cdot 10^7, p^i = 0), & \forall i \in [1..d] \\ (P_2) & p(0) = (1, 1), & (x^i = (0, 0), p^i = (0, 0)), & \forall i \in [1..d] \end{cases}$$



Solutions en $\lambda = 0$ pour le BVP discrétisé (Euler) - Problème 1 et 2
(tracées par dessus les solutions du tir simple en $\lambda = 0$)

On note que les solutions du BVP discrétisé coïncident extrêmement bien avec les résultats précédents du tir simple (voir page 95), malgré le schéma d'intégration grossier utilisé.

Concernant le suivi de chemin, les choses se passent nettement mieux ici, et on atteint $\lambda = 1$ sans remarquer l'instabilité qui affecte la continuation sur le tir simple. On trace les solutions obtenues en $\lambda = 1$, avec une triangulation $K_1(10^{-2})$ pour le suivi de chemin.



Solutions en $\lambda = 1$ pour le BVP discrétisé (Euler) - Problème 1 et 2

Au premier abord, le plus intéressant est l'allure des deux fonctions de commutation. Pour les deux problèmes la présence d'arcs singuliers est flagrante, avec ψ s'annulant quasiment aux environs des intervalles $[2, 7]$ et $[1.5, 5]$ respectivement. Il est remarquable que malgré le faible nombre de pas de discrétisation, les fonctions de commutations sont numériquement bien proches de 0. Pour (P_1) en particulier, la détection de l'arc singulier est bien plus nette que ce que l'on obtenait avec la continuation sur le tir simple juste avant le seuil d'instabilité.

Comme prévu avec cette formulation, le contrôle est bang-bang, en accord avec les conditions nécessaires à chaque noeud de discrétisation. Sur les arcs singuliers, il oscille aléatoirement entre les deux valeurs extrémales possibles (0,1 et -1,1 respectivement), suivant le signe de la fonction de commutation évaluée en chaque noeud.

En fait, si l'on compare a posteriori ces trajectoires avec les solutions précises obtenues plus tard (voir 4.4), on trouve que cette approximation discrétisée de x et p n'est pas trop mauvaise. Pour (P_1) , les différences sont localisées sur l'arc singulier, où l'état et l'état adjoint discrétisés restent tout de même proches de la solution. Pour (P_2) toutefois, la solution discrétisée est moins précise : l'état et l'état adjoint ont la bonne allure générale, mais apparaissent décalés par rapport à la solution pour x_1, p_1 et p_2 , avec de fortes oscillations sur x_2 (visibles sur le graphe ci-dessus).

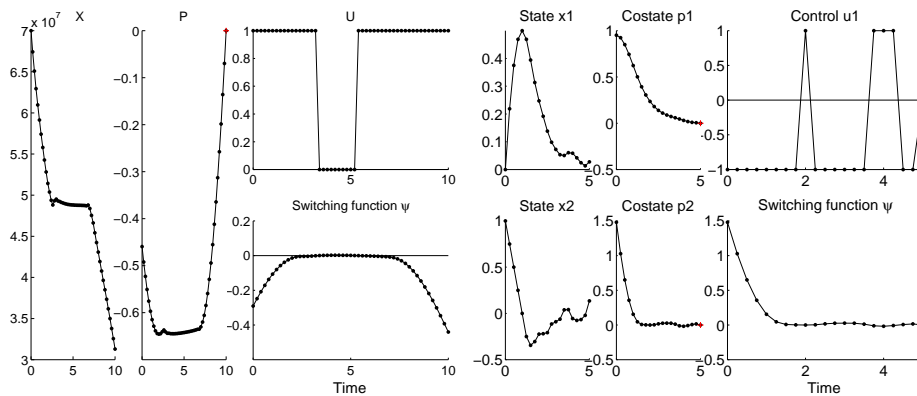
Formulation Trapèzes implicite

Néanmoins, une partie de cette formulation discrétisée qui peut être améliorée est le schéma d'intégration. Le choix de la formule d'Euler est à l'origine motivé par la volonté de rentrer dans les hypothèses du Théorème 6, mais celles-ci peuvent également s'accommoder de la formule par trapèzes (implicite). Le seul changement concerne alors les conditions de raccorde-

ment :

$$Match_{cond} \begin{cases} x_i - (x_{i-1} + \frac{h}{2} \left(\frac{\partial x}{\partial t}(t_{i-1}, x_{i-1}, p_{i-1}, u_{i-1}^*) + \frac{\partial x}{\partial t}(t_i, x_i, p_i, u_i^*) \right)) \\ p_i - (p_{i-1} + \frac{h}{2} \left(\frac{\partial p}{\partial t}(t_{i-1}, x_{i-1}, p_{i-1}, u_{i-1}^*) + \frac{\partial p}{\partial t}(t_i, x_i, p_i, u_i^*) \right)) \end{cases}$$

Comme précédemment, l'initialisation est immédiate, et le suivi de chemin se passe bien, avec les solutions ci-dessous.



Solutions en $\lambda = 1$ pour le BVP discrétisé (Trapèzes) - Problème 1 et 2

La solution pour (P_1) semble quasiment identique à celle avec Euler, par contre les oscillations sur (P_2) sont réduites. De plus, l'état, état adjoint et fonction de commutation sont a posteriori plus proches de la solution précise obtenue dans la suite. Cette différence se voit par exemple au niveau de la valeur initial de p , qui est environ $(1, 1.5)$ contre $(1.15, 2)$ pour Euler, la solution précise étant vers $(0.9422, 1.4419)$. Il semble donc que les défauts de la solution précédente pour (P_2) soient en partie dus à l'intégration par Euler, et peuvent être réduits, ce qui est rassurant. En fait, cette seconde solution peut encore être améliorée par l'utilisation de la triangulation adaptative et du raffinement de solution (voir 4.3.3), contrairement à la solution Euler.

Remarque : on peut penser à essayer des schémas d'intégration plus évolués, même s'ils ne vérifient pas les hypothèse du Théorème 6. En pratique, l'utilisation de formules de Runge Kutta d'ordre 2 et 4 donne en fait de moins bonnes solutions, avec le même genre de décalage observé sur x_2 (ceci est peut-être lié au fait que l'hypothèse de convexité n'est pas vérifiée pour (P_2)). On a également fait quelques expérimentations très préliminaires avec des schémas symplectiques, voir en Annexe B.3.2 page 149.

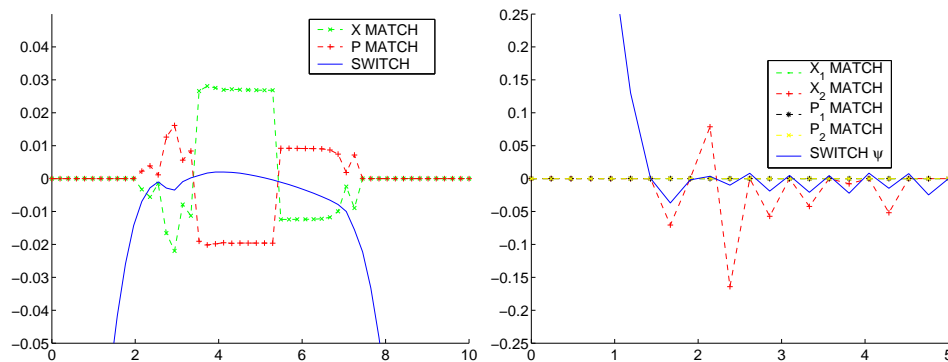
Ces solutions sont en fait déjà suffisantes pour détecter la structure singulière et initialiser correctement la résolution précise des problèmes, comme expliqué en 4.4. On va tout de même étudier plus en détails cette formulation, en particulier les oscillations observées.

Erreurs de raccordement

En dehors du remplacement de la formule d'Euler, on pourrait penser à augmenter simplement le nombre de noeuds de discrétisation pour améliorer les solutions. Toutefois, les tests réalisés dans cette direction n'ont pas été très convaincants, et cette stratégie est vite confrontée au problème de la taille de l'inconnue. De même, l'utilisation de triangulations raffinées ne donne pas de meilleurs résultats, et le tir en $\lambda = 1$ non plus (le raffinement de solution, par contre, fait un peu mieux, voir page 108).

Ces difficultés persistantes pourraient en fait provenir de l'expression du contrôle, qui est toujours donné pas les conditions nécessaires. Sur un intervalle de temps $[t_i, t_{i+1}]$ inclus dans un arc singulier, supposons que l'algorithme ait trouvé les valeurs correctes de (x^i, p^i) et (x^{i+1}, p^{i+1}) . Etant sur un arcs singulier, la valeur de la fonction de commutation $\psi(t_i)$ devrait être proche de 0, mais numériquement il s'agira d'une faible valeur positive ou négative. En fonction de ce signe les conditions nécessaires donnent un contrôle bang-bang u_i , qui est différent du contrôle singulier u_i^* . En conséquence, les conditions de raccordement pour l'état et l'état adjoint en t_{i+1} peuvent ne pas être vérifiées, bien que l'on ait les valeurs correctes. Les implications numériques de ce défaut de la formulation sont difficiles à prédire concernant la solution, mais il semble que l'algorithme trouve un équilibre entre les valeurs justes de (x, p) et les erreurs de raccordement induites.

On regarde maintenant de plus près la valeur des conditions de raccordement en $\lambda = 1$ (solution Euler). Sur les graphes suivants, on trace pour les deux problèmes la fonction de commutation ψ (trait plein), et chaque composante des conditions de raccordement sur l'état et l'état adjoint (tirets, avec différents symboles pour les noeuds). On a donc au total 3 courbes pour (P_1) ($(x, p) \in \mathbf{R}^2$), et 5 pour (P_2) ($(x, p) \in \mathbf{R}^4$).



Conditions de raccordement pour (P_1) et (P_2) en $\lambda = 1$ (Euler)

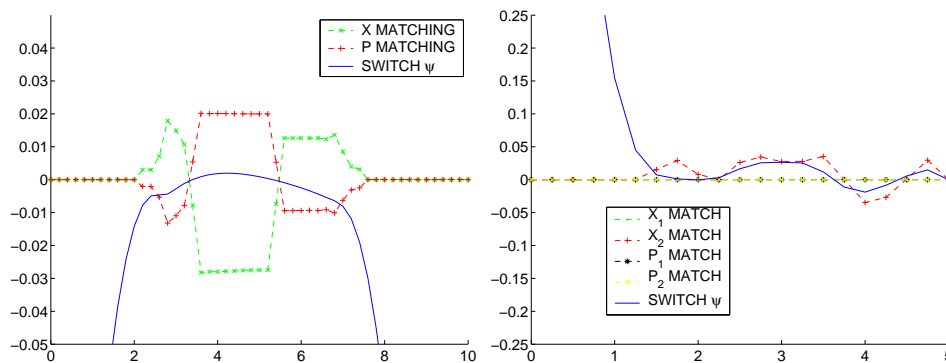
La chose la plus intéressante sur ces deux graphes est la localisation des

erreurs de raccordement. On remarque immédiatement que les composantes non nulles des conditions de raccordement ne se trouvent que sur les noeuds à l'intérieur des arcs singuliers. Sur les arcs bang, où le contrôle donné par les conditions nécessaires est correct, les erreurs de raccordement sont numériquement nulles (en dessous de 10^{-14}) pour toutes les composantes des deux problèmes.

De plus, pour (P_2) on constate également que les erreurs concernent uniquement la seconde composante de l'état (x_2). Les autres composantes (x_1, p_1, p_2) vérifient toujours leurs conditions de raccordement, même sur les arcs singuliers. Et si l'on regarde l'expression de la dynamique de l'état et état adjoint pour (P_2) (voir page 91), on s'aperçoit que x_2 est précisément la seule composante pour laquelle le contrôle u apparait. Pour (P_1) , les dérivées de x et p font toutes deux intervenir le contrôle (cf page 88), et il n'est donc pas surprenant de trouver des erreurs de raccordement pour les deux sur les arcs singuliers.

De plus, on note que le signe des erreurs de raccordement semble changer en même temps que celui de la fonction de commutation. Ceci semble cohérent avec les remarques précédentes, comme ce changement de signe correspond aussi aux commutations du contrôle bang-bang incorrect donné par les conditions nécessaires. Etant donné les valeurs du contrôle singulier, ces commutations changent typiquement le signe de "l'erreur" sur le contrôle, à savoir $u^i - u_{singular}^i$.

Voici maintenant les conditions de raccordement pour la formulation trapèzes.



Conditions de raccordement pour (P_1) et (P_2) en $\lambda = 1$ (Trapèzes)

On observe le même comportement qualitatif qu'avec Euler, simplement avec des erreurs plus faibles pour (P_2) . Ceci indiquerait que le lien entre les erreurs de raccordement et le contrôle incorrect sur les arcs singuliers n'est pas une conséquence du choix d'intégration, mais bien une caractéristique

de la formulation BVP discrétisé.

A ce stade, il semble donc raisonnable de penser que ces erreurs de raccordement persistantes à la solution sont directement liées au contrôle incorrect donné par les conditions nécessaires sur les arcs singuliers.

Utiliser l'expression algébrique du contrôle singulier ?

Mais après tout, puisque l'on connaît l'expression algébrique du contrôle singulier (voir p.91 et 92), pourquoi ne pas l'utiliser pour se débarrasser de ces erreurs ? Si à un noeud t_i on suspecte être sur un arc singulier, il semble légitime d'utiliser alors l'expression exacte du contrôle singulier $u_{singular}$ au lieu du contrôle incorrect u_i^* des conditions nécessaires. Cette idée à l'air intéressante au premier abord, mais se heurte à de grandes difficultés d'implémentation.

Le point crucial est la détection du cas singulier, et nous avons essayé plusieurs formulations portant sur la norme de la fonction de commutation ψ et de ses dérivées (qui sont censées s'annuler sur les arcs singuliers). Toutefois, les tests numériques avec les expressions suivantes

$$Sing_{cond} \begin{cases} |\psi(t_i)| < \epsilon \\ \psi^2(t_i) < \epsilon \\ \psi^2(t_i) + \dot{\psi}^2(t_i) < \epsilon \\ \psi^2(t_i) + \dot{\psi}^2(t_i) + \ddot{\psi}^2(t_i) < \epsilon \end{cases}$$

ont montré une sensibilité extrême par rapport à la valeur de la tolérance ϵ , pour ne pas dire plus.

De fait, changer la valeur de ce seuil conduit à des solutions totalement différentes, parfois meilleurs, mais parfois pires qu'avec la formulation classique. Comme nous n'avons pas trouvé de lien cohérent (pour les deux problèmes en même temps) entre la valeur de ϵ et la qualité des solutions, nous avons finalement abandonné cette idée. La racine du mal semble être que cette formulation tend à "forcer" trop arbitrairement des arcs singuliers, et peut-être qu'une façon plus nuancée d'introduire $u_{singular}$ (ou une détection plus intelligente du cas singulier) rendrait cette idée utilisable en pratique.

4.3.3 Effet de la triangulation adaptative et du raffinement de solution

Pour la formulation BVP discrétisé, on note que le tir final en $\lambda = 1$ n'améliore pas la solution, quelle que soit l'intégration choisie. Ceci est probablement dû aux erreurs de raccordement décrites plus tôt, inhérentes à

la formulation. On essaye alors de voir ce que donnent les mécanismes de triangulation adaptative et de raffinement de solution. On compare d'abord la triangulation uniforme de base avec le mode adaptatif complet (cf 2.3). Plus on ajoute le raffinement de solution (cf 2.4.2) pour le troisième test avec 10 tentatives de raffinement (autoriser 20 raffinages n'améliore pas davantage les solutions).

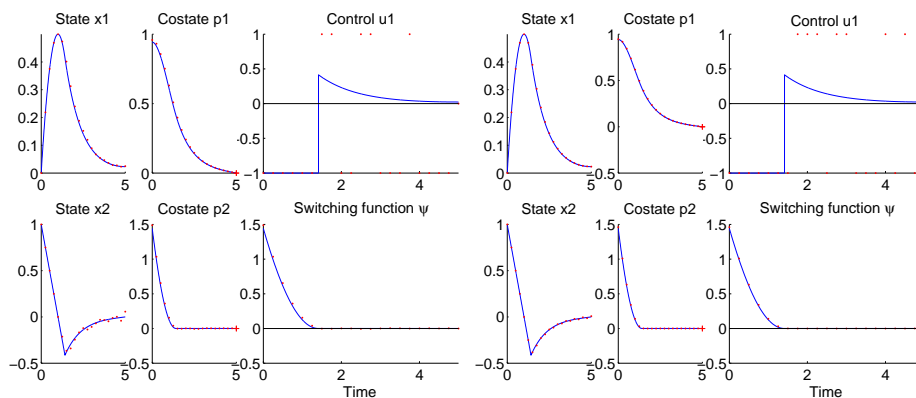
Résumé : les trois configurations de test sont :

- triangulation uniforme fixée, pas de raffinement, notée "uniforme".
- mode adaptatif complet, pas de raffinement, notée "adaptatif".
- mode adaptatif complet, 10 raffinages en $\lambda = 1$, notée "raffinage".

Tous les tests utilisent une triangulation K_1 , avec une taille initiale de 10^{-2} , et l'intégration par trapèzes (les mêmes tests avec l'intégration Euler n'ont pas été convaincants, avec persistance des oscillations observées précédemment). Comme on s'intéresse ici plutôt à la qualité des solutions qu'à la performance², on règle le contrôle de déviation plus prudemment à 10^{-2} , ce qui se révèle ici supérieur au réglage par défaut à 10^{-1} .

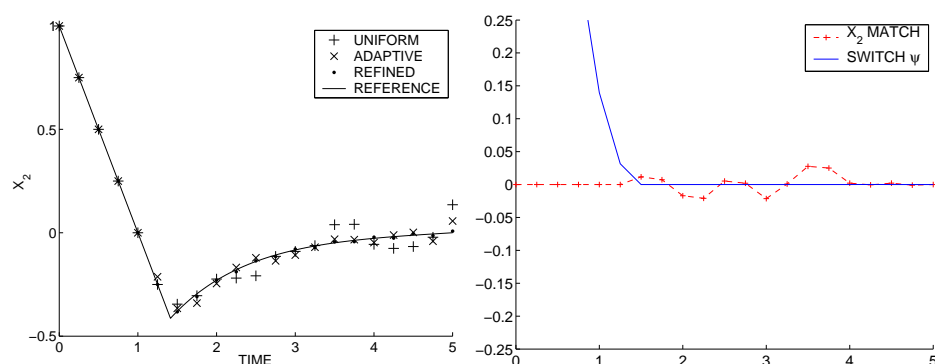
Les solutions "uniforme" sont celles déjà obtenues, et on montre ici les deux autres pour (P_2) ("adaptatif" et "raffinage"), superposées à la solution de référence obtenue par tir structuré (voir 4.4 juste après).

Remarque : on ne représente pas les solutions du problème (P_1), car elles sont très proches et visuellement impossibles à distinguer les unes des autres sur les graphes.



Solutions pour les configurations "adaptatif" et "raffinage".

²ces suivis ne prennent que quelques secondes de toute façon.



Comparaison de x_2 pour les trois configurations et condition de raccordement sur x_2 pour la solution “raffinage”.

On voit (avec de bons yeux) que les solutions “adaptatif” et “raffinage” sont plus proches de la solution de référence, ce qui est rassurant. La comparaison est plus significative pour x_2 , que l’on a représenté séparément sur le troisième graphe (pour les trois configurations de test en plus de la référence). On rappelle que x_2 est la seule composante dont la dérivée implique le contrôle, ce qui explique probablement pourquoi elle montre le plus d’irrégularités.

On remarque que la solution raffinée comporte encore quelques points égarés, qui ne semblent pas vouloir disparaître si on augmente le nombre de raffinages. On atteint sans doute ici les limites de la formulation discrétisée, comme les erreurs de raccordement décrites auparavant nous empêchent de trouver la solution exacte de toute façon. Néanmoins, comme on peut le voir sur la comparaison page 119, ces solutions raffinées sont en fait très bonnes a posteriori. On a une approximation précise de l’état et de l’état adjoint, ainsi que de la fonction de commutation, dont les valeurs sur les arcs singuliers descendent jusqu’à $10^{-5} - 10^{-6}$ pour les deux problèmes. La différence avec la solution “uniforme” est nette sur le quatrième graphe montrant les conditions de raccordement, où ψ montrait auparavant des oscillations autour de 0 (voir page 107).

Voici les détails des suivis de chemin : en plus de la norme finale, on indique le nombre de simplexes du chemin principal (**Principal**), des jonctions en incluant les raffinages (**Jonctions**), et le nombre total d’appels de l’homotopie (**Total (H)**).

Problème 1	Principal	Jonctions	Total (H)	$ S_1(z^*) $
uniforme	107 360	0	107 578	1.69
adaptatif	34 920	754	37 404	$8.81 \cdot 10^{-1}$
raffinage	34 920	1 282 374	249 154	$7.19 \cdot 10^{-1}$

Problème 2	Principal	Jonctions	Total (H)	$ S_1(z^*) $
uniforme	25161	0	25 837	$9.68 \cdot 10^{-2}$
adaptatif	9347	8 656	27 152	$8.21 \cdot 10^{-2}$
raffinage	9347	220 868	91 271	$5.51 \cdot 10^{-2}$

On note que le mode adaptatif se débrouille bien en termes de performances, avec un suivi bien plus rapide pour (P_1) . On voit ensuite que si le raffinage parvient à améliorer la solution, ceci a un coût³. Incidemment, les normes finales diminuent (légèrement) ici en accord avec la qualité des solutions. Toutefois, il est à noter qu'à cause des erreurs de raccordement résiduelles, cette norme n'est pas forcément toujours un indicateur fiable de la qualité d'une solution. Par exemple, la solution raffinée pour Euler sur (P_2) présente une norme finale faible, même si elle comporte encore de fortes oscillations comme précédemment, avec une valeur de l'objectif très médiocre.

Conclusion préliminaire

La formulation BVP discrétisé a été introduite dans l'idée d'éviter l'instabilité numérique constatée sur la continuation avec le tir simple. Les expérimentations ont montré que cette difficulté particulière disparaît effectivement, et ont également souligné certaines limitations de cette formulation.

Tout d'abord, le caractère primitif de l'intégration utilisée, avec un faible nombre de pas et un schéma extrêmement simple, qui a été légèrement amélioré en remplaçant la formule d'Euler par une règle de trapèzes implicite. Ensuite, un point plus subtil, les erreurs de raccordement causées par l'expression incorrecte du contrôle sur les arcs singuliers, que nous ne sommes pas parvenus à supprimer complètement. Néanmoins, même avec ces défauts on obtient déjà une détection fiable de la structure singulière, et une bonne initialisation pour une résolution précise, comme décrit ci-après.

De plus, grâce à l'utilisation des mécanismes de triangulation adaptative et de raffinement de solution présentés en 2.1, on parvient également à obtenir une approximation très proche de l'état, état adjoint et fonction de commutation, même si cela ne s'avère pas nécessaire dans la suite.

4.4 Tir structuré

Comme on l'a vu, la première continuation avec le tir simple donne déjà une idée de la structure singulière des deux problèmes. Ensuite, avec la formulation BVP discrétisé, on obtient des informations plus claires sur cette

³mais une fois encore, les performances ne sont pas notre objectif ici, et les configurations raffinées convergent encore en moins de 2 minutes...

structure, avec une détection très nette des arcs singuliers. On souhaite donc maintenant passer à la résolution précise grâce à cette connaissance. On utilise une variante du tir multiple classique, qu'on appelle "tir structuré". Elle partage le principe du célèbre code BNDSCO d'Oberle (cf [31]), légèrement simplifié et adapté au cas des arcs singuliers au lieu des contraintes d'état. La première différence est que les arcs singuliers, contrairement aux contraintes d'état, ne provoquent pas de discontinuités de l'état adjoint, et il n'y a donc pas de conditions de saut à traiter durant l'intégration. Ensuite, au lieu d'avoir une subdivision fixée du temps pour le tir multiple en plus des instants de commutation délimitant les changements de structure, on ne garde que ces derniers. On intègre l'IVP arc par arc, avec le contrôle optimal usuel sur les arcs bang (non singuliers), et le contrôle singulier (cf pages 91,92) sur les arcs singuliers.

Remarque : si l'on examine les expressions du contrôle singulier mentionnées ci-dessus, on constate qu'elles ne vérifient pas la contrainte $u \in U$ a priori. Il se trouve que les initialisations fournies par les deux continuations précédentes sont en fait suffisamment précises pour que l'on ne rencontre pas de difficultés en pratique. Toutefois, les expérimentations sur des initialisations perturbées montrent que des contrôles "hors limites" peuvent bel et bien apparaître si l'on est loin de la solution, voir page 118.

4.4.1 Formulation

On décrit ici deux formulations pour le tir structuré, d'abord la version complète puis une version simplifiée.

Tir structuré complet

En plus de l'inconnue classique de l'IVP (qui est pour ces deux problèmes l'état adjoint initial $p(0)$), on a donc les instants qui délimitent les arcs singuliers, et les valeurs de l'état et état adjoint à ces instants. D'autre part, on a également un ensemble de nouvelles conditions à vérifier, et la fonction de tir est de la forme

$$S_{Struct} : \mathbf{R}^{n+(2n+1) \times s} \rightarrow \mathbf{R}^{n+(2n+1) \times s}$$

avec n la dimension de l'état (et état adjoint) et s le nombre d'instantes de commutations. Les conditions sur ces instants reflètent le fait qu'ils indiquent soit le début, soit la fin d'un arc singulier, donc la fonction de commutation et ses dérivées doivent s'annuler. Les conditions sur l'état et l'état adjoint correspondent ici simplement à leur continuité (ce qui est bien plus simple qu'avec des contraintes d'état, où il faut tenir compte des sauts, comme dans BNDSCO). On note ces deux types de conditions $Switch_{cond}$ et $Match_{cond}$

dans la suite. Voici le détail de l'inconnue et de la valeur de la fonction de tir.

Inconnue : z

- inconnue de l'IVP en t_0 (comme pour le tir simple)
- valeurs de (x^i, p^i) aux instants de commutation $t_i, i = 1..s$
- longueur des arcs Δ_i , tels que $t_i = t_{i-1} + \Delta_i, \forall i \in [1..s]$

inconnue de l'IVP en t_0	(x^1, p^1)	...	(x^s, p^s)	Δ_1	...	Δ_s
----------------------------	--------------	-----	--------------	------------	-----	------------

Valeur : $S_{Struct}(z)$

- conditions de singularité et de raccordement aux instants de commutation
- conditions aux limites en t_f (comme pour le tir simple, notées $TC(t_f)$)

$Switch_{cond}(t_1)$	$Match_{cond}(t_1)$...	$Switch_{cond}(t_s)$	$Match_{cond}(t_s)$	$TC(t_f)$
----------------------	---------------------	-----	----------------------	---------------------	-----------

On détaille maintenant l'implémentation pratique des conditions de singularité et de raccordement. Comme on l'a dit, les conditions de raccordement vérifient simplement la continuité de l'état et état adjoint aux entrées et sorties d'arc singulier. D'où l'expression suivante :

$$Match_{cond}(t_i) = (x^i, p^i) - (\hat{x}^i, \hat{p}^i) \quad \forall i \in [1..s]$$

où (\hat{x}^i, \hat{p}^i) désigne l'état et l'état adjoint obtenus par intégration à la fin de l'arc $[t_{i-1}, t_i]$.

Pour la condition de singularité, on utilise

$$Switch_{cond}(t_i) = \psi^2(t_i, x^i, p^i, u_i).$$

Remarque : il est possible d'inclure les dérivées de ψ dans la condition de singularité, avec une expression de la forme

$$Switch_{cond}(t_i) = \psi^2(t_i, x^i, p^i, u_i) + \dot{\psi}^2(t_i, x^i, p^i, u_i).$$

Cependant, ceci se révèle numériquement plus fragile que la formulation ci-dessus.

Tir structuré léger

On présente maintenant une version allégée du tir structuré, que l'on appelle fort originalement "tir structuré léger". Cette modification vient du fait qu'en l'absence de contraintes d'état, l'état et l'état adjoint sont continus au début/à la fin des arcs singuliers. On pourrait donc simplement poursuivre l'intégration avec les valeurs de (x, p) obtenues à la fin de l'arc précédent (singulier ou pas). Dans ce cas, il n'est plus nécessaire d'inclure les valeurs (x^i, p^i) dans l'inconnue. Il ne reste plus alors que les longueurs

d'arcs Δ_i , en plus de l'inconnue usuelle de l'IVP. La nouvelle fonction de tir est de la forme

$$S_{LightStruct} : \mathbf{R}^{n+s} \rightarrow \mathbf{R}^{n+s}$$

avec s le nombre d'instants de commutation.

Cette formulation plus légère présente un gain en termes de dimension, mais cet avantage serait plus significatif avec un nombre plus élevé d'arcs singuliers. Pour les deux problèmes étudiés, on n'a que 2 et 1 instants de commutation, ce qui ne fait donc pas une grande différence de taille.

Par ailleurs, poursuivre l'intégration tout du long soulève une difficulté théorique. A la sortie d'un arc singulier, la fonction de commutation est censée être nulle, donc le contrôle optimal (non singulier) juste à la sortie ne peut pas être donné correctement par les conditions nécessaires. En pratique bien sûr, ψ ne s'annule que "numériquement", à une certaine précision près. A la sortie, on attend donc une très faible valeur, mais positive ou négative. Si le signe est correct, on a la bonne valeur du contrôle sur l'arc bang qui suit l'arc singulier, sinon....

Les expérimentations numériques tendent à montrer qu'avec une bonne initialisation, on est dans ce cas favorable, et la formulation allégée converge bien. Ceci souligne l'importance de bien connaître la structure du contrôle, à savoir non seulement le nombre et la localisation des arcs singuliers, mais aussi la nature des arcs bang-bang. Il est donc d'autant plus intéressant de disposer des bonnes solutions approchées des continuations précédentes.

On résume ici l'inconnue et la valeur de la fonction de tir pour la formulation allégée.

Inconnue : z

- inconnue de l'IVP en t_0 (comme pour le tir simple)
- longueur des arcs Δ_i , tels que $t_i = t_{i-1} + \Delta_i$, $\forall i \in [1..s]$

inconnue de l'IVP en t_0	Δ_1	...	Δ_s
----------------------------	------------	-----	------------

Valeur : $S_{LightStruct}(z)$

- conditions de singularité aux instants de commutation
- conditions aux limites en t_f (TC(t_f), comme pour le tir simple)

$Switch_{cond}(t_1)$...	$Switch_{cond}(t_s)$	TC(t_f)
----------------------	-----	----------------------	-------------

4.4.2 Résultats numériques

On procède maintenant à la résolution numérique des problèmes . Pour commencer, on détermine la structure singulière et l'initialisation à partir

des résultats des deux continuations précédentes, sur le tir simple et le BVP discrétisé. On résout ensuite les deux problèmes avec le tir structuré (complet et léger) présenté ci-dessus. Cette résolution ne comporte pas de continuation en elle-même, on essaye juste de résoudre les équations $S_{Struct}(z) = 0$ or $S_{LightStruct}(z) = 0$ pour les deux problèmes (P_1) et (P_2) .

Initialisations

On commence par déterminer la structure singulière que l'on attend pour chaque problème, en se basant sur les informations des continuations. Ici les fonctions de commutations indiquent clairement la même structure pour les deux continuations : *régulier-singulier-régulier* pour (P_1) , et *régulier-singulier* pour (P_2) . On extrait ensuite la longueur des arcs et les valeurs de l'état-état adjoint requises à partir des solutions des continuations, comme résumé ci-dessous. On a donc deux jeux d'initialisations (à partir des continuation tir simple et BVP discrétisé), et deux méthodes de résolution (tir structuré complet et léger).

Remarques :

- *Trouver l'initialisation correspondant à la continuation du tir simple est un peu fastidieux. Pour trouver une solution juste avant le seuil d'instabilité, on doit en pratique faire plusieurs suivis de chemin avec des triangulations raffinées.*
- *L'initialisation pour le cas BVP discrétisé correspond à la formulation la plus basique, Euler avec triangulation uniforme. Après tout, on ne cherche qu'une initialisation, et cela suffit, comme on va le voir.*

On résume ici les initialisations pour (P_1) .

Note : dans la suite on utilise les longueurs des arcs Δ_i au lieu des instants de commutation t_i , en accord avec l'implémentation. On rappelle que l'on a ici $t_1 = \Delta_1$ et $t_2 = \Delta_1 + \Delta_2$.

Problème 1 : structure de contrôle *régulier-singulier-régulier*

- **Inconnue \mathbf{z} :** $p(0), \Delta_1, \Delta_2, x(t_1), p(t_1), x(t_2), p(t_2)$

Continuation	$p(0)$	Δ_1	Δ_2	x^1	p^1	x^2	p^2
$BVP_{0.95}$	-0.429	2.5	4.5	$4.996 \cdot 10^7$	-0.600	$4.825 \cdot 10^7$	-0.587
BVP_D	-0.453	2.55	4.55	$4.839 \cdot 10^7$	-0.637	$4.741 \cdot 10^7$	-0.621

On observe que les deux initialisations sont assez proches, et que l'on a chaque fois $(x^1, p^1) \approx (x^2, p^2)$. Ceci est cohérent avec le fait que pour (P_1) , l'état et l'état adjoint sont constants sur les arcs singuliers (cf 4.1.1 page 88).

Voici les initialisations pour (P_2) .

Problème 2 : structure de contrôle *régulier-singulier*

- **Inconnue \mathbf{z} :** $p(0), \Delta_1, x(t_1), p(t_1)$

Continuation	$p(0)$	Δ_1	x^1	p^1
$BVP_{0.925}$	(0.974, 1.512)	1.5	(0.398, -0.309)	(0.401, 0.00358)
BVP_D	(1.167, 2.024)	1.429	(0.578, -0.429)	(0.586, 0.000505)

Il n'y a ici que deux arcs et un seul instant de commutation, mais l'état (et l'état adjoint) sont de dimension 2. Les initialisations sont ici plus distinctes, ce qui est en particulier dû au décalage de l'état et de l'état adjoint observé avec Euler pour (P_2) , comme mentionné précédemment. Par ailleurs, on a dans les deux cas $p_2^1 \approx 0$, ce qui est cohérent avec le fait que la fonction de commutation est censée s'annuler à l'entrée d'un arc singulier (on rappelle que $\psi = p_2$ pour (P_2) , donc on s'attend à $p_2(t_1) = 0$).

Résolution

Comme mentionné plus tôt, il n'y a ici pas de suivi de chemin, on procède directement au tir (techniquement, en $\lambda = 1$). On utilise une intégration Runge Kutta d'ordre 4, avec 1000 pas. Dans la formulation du tir structuré, dans le cas d'une intégration à pas fixe, les pas sont divisés entre les arcs proportionnellement à leur longueur (il n'y a pas de distinction entre les arcs singuliers et les autres). La convergence est immédiate (1 seconde ou moins) pour toutes les configurations de test.

Voici les résultats des tir structurés complet et léger pour (P_1) , à partir des deux jeux d'initialisations :

- **Norme initiale** est la norme de la fonction de tir au point initial
- **Norme finale** est la norme de la fonction de tir à la solution
- **objectif** est la valeur de l'objectif à la solution
- **Total (H)** indique le nombre total d'appels à l'homotopie
- **Temps** donne le temps d'exécution du tir (sur un Pentium 4 à 2.8GHz).

Initialisation	Tir	Norme initiale	Norme finale	objectif	Total (H)	Temps
$BVP_{\lambda=0.95}$	Full	$1.08 \cdot 10^{-1}$	$6.70 \cdot 10^{-12}$	$1.069 \cdot 10^8$	53	< 1s
$BVP_{\lambda=0.95}$	Light	$1.69 \cdot 10^{-1}$	$1.40 \cdot 10^{-12}$	$1.069 \cdot 10^8$	59	< 1s
BVP_D	Full	$8.44 \cdot 10^{-2}$	$1.17 \cdot 10^{-12}$	$1.069 \cdot 10^8$	111	1s
BVP_D	Light	$1.13 \cdot 10^{-1}$	$4.47 \cdot 10^{-14}$	$1.069 \cdot 10^8$	62	< 1s

*Résultats de convergence du tir structuré (complet et léger) pour (P_1)
Initialisation à partir des continuations tir simple et BVP discrétisé*

Et une comparaison plus détaillée des solutions du tir :

- p(0)** indique l'état adjoint initial à la solution
- **Δ_1, Δ_2** indique la longueur des arcs (avec $t_1 = \Delta_1, t_2 = \Delta_1 + \Delta_2$)

Initialisation	Tir	$p(0)$	Δ_1	Δ_2
$BVP_{\lambda=0.95}$	Complet	$-4.622547 \cdot 10^{-1}$	2.370416	4.618367
$BVP_{\lambda=0.95}$	Léger	$-4.622547 \cdot 10^{-1}$	2.370416	4.618367
BVP_D	Complet	$-4.622547 \cdot 10^{-1}$	2.370416	4.618367
BVP_D	Léger	$-4.622547 \cdot 10^{-1}$	2.370415	4.618366

*Comparaison des solutions du tir structuré (complet et léger) pour (P_1)
Initialisation à partir des continuations tir simple et BVP discrétisé*

On constate que la norme de la fonction de tir au point initial est légèrement meilleure par l'initialisation du BVP discrétisé. En ce qui concerne les quatre solutions obtenues, la norme finale est similaire pour les trois premières, avec la quatrième légèrement meilleure. L'objectif est en fait identique à 11 chiffres significatifs près⁴, et $p(0)$, Δ_1 , Δ_2 coïncident jusqu'à 6 chiffres (5 pour la quatrième solution). On peut donc conclure que les solutions trouvées sont numériquement identiques.

Passons maintenant au problème (P_2) .

Initialisation	Tir	Norme initiale	Norme finale	objectif	Total (H)	Temps
$BVP_{\lambda=0.925}$	Complet	1.37	$3.07 \cdot 10^{-14}$	$3.7699 \cdot 10^{-1}$	52	< 1s
$BVP_{\lambda=0.925}$	Léger	1.99	$6.85 \cdot 10^{-14}$	$3.7699 \cdot 10^{-1}$	43	< 1s
BVP_D	Complet	$3.14 \cdot 10^{-1}$	$1.01 \cdot 10^{-15}$	$3.7699 \cdot 10^{-1}$	57	< 1s
BVP_D	Léger	$1.01 \cdot 10^{-1}$	$1.56 \cdot 10^{-14}$	$3.7699 \cdot 10^{-1}$	49	< 1s

*Résultats de convergence du tir structuré (complet et léger) pour (P_2)
Initialisation à partir des continuations tir simple et BVP discrétisé*

Initialisation	Tir	$p_1(0)$	$p_2(0)$	Δ_1
$BVP_{\lambda=0.925}$	Complet	$9.421734 \cdot 10^{-1}$	1.441910	1.413764
$BVP_{\lambda=0.925}$	Léger	$9.421734 \cdot 10^{-1}$	1.441910	1.413764
BVP_D	Complet	$9.421734 \cdot 10^{-1}$	1.441910	1.413764
BVP_D	Léger	$9.421734 \cdot 10^{-1}$	1.441910	1.413764

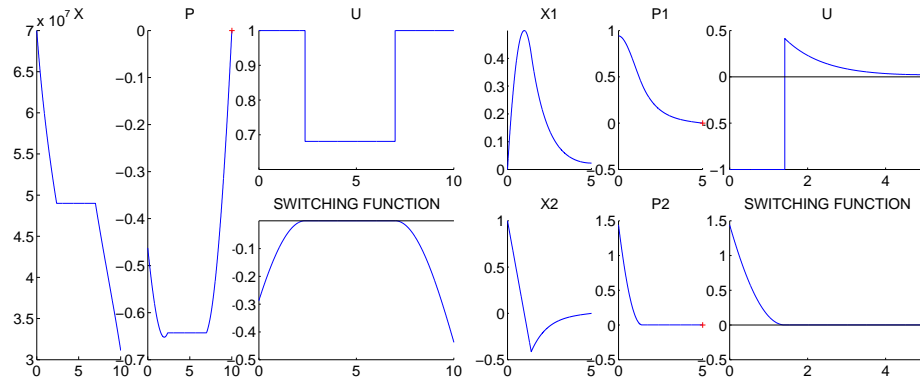
*Comparaison des solutions du tir structuré (complet et léger) pour (P_2)
Initialisation à partir des continuations tir simple et BVP discrétisé*

On voit ici plus nettement la distinction entre les deux initialisations, avec un écart sur la norme initiale d'un ordre de grandeur en faveur de l'initialisation BVP discrétisé. Comme pour (P_1) , les normes finales sont très proches, les objectifs identiques à 14 chiffres⁵, et $p(0)$, Δ_1 à plus de 6 chiffres. Une fois encore, on peut considérer que l'on converge vers la même solution.

On constate qu'utiliser les intégrateurs à pas variable RKF45, DOP853 et ODEX donne les mêmes résultats. Les normes finales sont légèrement meilleures (de 10^{-14} à 10^{-16}), ce qui n'est pas surprenant, et l'objectif reste identique à 8 chiffres près. Finalement, on représente ici les trajectoires correspondantes, sur lesquelles on voit clairement les arcs singuliers.

⁴ $1.06905997911 \cdot 10^8$ pour être tout à fait exact.

⁵0.37699193029464...



Solutions du tir structuré - Problèmes 1 et 2

On peut s'interroger sur la sensibilité des deux formulations du tir structuré. On étudie donc la stabilité de convergence en appliquant diverses perturbations à l'initialisation. Le cadre de référence est le suivant : initialisation BVP discrétisé, intégration RK4 (1000). On utilise les perturbations suivantes sur l'état adjoint initial et les instants de commutation :

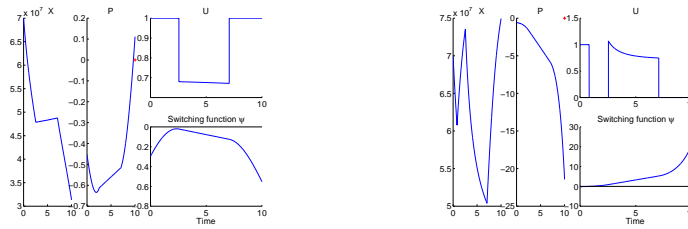
- $p(0)$: $-1, -0.5, -0.1, +0.1, +0.5, +1$ pour (P_1)
- $p(0)$: $(-2, -2), (-1, -1), (+1, +1), (+2, +2)$ pour (P_2)
- (t_1, t_2) : $(-1, +1), (+1, -1)$ pour (P_1)
- t_1 : $-1, +1$ pour (P_2)

Pour chaque tentative de tir on indique seulement la norme finale et le nombre d'appels de l'homotopie. Le * après le nombre d'appels indique que l'on a rencontré au cours du tir des contrôles "hors limites" sur les arcs singuliers (cf page 112).

Perturbation	Complet	Léger
$p(0) - 1$	$1.07 \cdot 10^{-13} / 112^*$	$6.02 / 19^*$
$p(0) - 0.5$	$0.26 / 125^*$	$4.59 / 19^*$
$p(0) - 0.1$	$0.27 / 114^*$	$9.57 \cdot 10^{-3} / 75^*$
$p(0) + 0.1$	$6.11 \cdot 10^{-14} / 87$	$1.07 \cdot 10^{-13} / 68$
$p(0) + 0.5$	$1.28 \cdot 10^{-13} / 128$	$5.68 / 17^*$
$p(0) + 1$	$4.06 / 25^*$	$2.18 / 39^*$
$(t_1, t_2) + (-1, 1)$	$9.80 \cdot 10^{-13} / 60$	$0.28 / 37$
$(t_1, t_2) + (1, -1)$	$1.83 \cdot 10^{-13} / 121$	$1.37 \cdot 10^{-13} / 92$

Stabilité de convergence - Problème 1

On constate que le tir est assez sensible, puisqu'une perturbation de -0.1 sur $p(0)$ suffit à empêcher la convergence. En fait, ceci s'explique par le fait que cette petite perturbation modifie déjà la structure du contrôle, comme on le voit ci-dessous



La perturbation modifie la structure du contrôle - Problème 1

On se retrouve ainsi dans la situation décrite au chapitre 1 : le point initial n'est pas dans la bonne région, et le tir ne converge pas.

On voit également que la formulation complète s'en sort un peu mieux que la formulation légère, avec 5 cas de convergence sur 8 tests, contre seulement 2 (et demi avec le cas à 10^{-3} ...). Il semble donc que le gain de taille de cette deuxième formulation soit contrebalancé par une stabilité réduite. De plus, on note la fréquente apparition de valeurs hors limites (ici $[0, 1]$) du contrôle singulier, indiquées par le symbole *. Celles-ci semblent coïncider remarquablement avec les cas de non convergence, avec une seule exception (premier test pour la formulation complète) où le tir parvient à converger malgré ce phénomène. Techniquement, il serait possible de tronquer ces valeurs à 0 ou 1 respectivement. On ne l'a pas introduit dans la formulation, car ce cas n'est pas censé se produire si l'on a une bonne initialisation, ce qui est tout l'objectif de la partie continuation. D'ailleurs, ce phénomène ne s'est jamais produit durant les tentatives "normales" (non perturbées) de résolution des problèmes.

Perturbation	Complet	Léger
$p(0) - (2, 2)$	$9.79 \cdot 10^{-14} / 58$	$2.27 \cdot 10^{-14} / 54$
$p(0) - (1, 1)$	$1.84 \cdot 10^{-14} / 58$	$1.26 \cdot 10^{-14} / 50$
$p(0) + (1, 1)$	$1.97 \cdot 10^{-14} / 70$	$5.37 \cdot 10^{-15} / 58$
$p(0) + (2, 2)$	$9.79 \cdot 10^{-15} / 73$	$3.24 \cdot 10^{-14} / 54$
$t_1 - 1$	$3.03 \cdot 10^{-15} / 76$	$2.88 \cdot 10^{-14} / 52$
$t_1 + 1$	$7.28 \cdot 10^{-15} / 69$	$4.50 \cdot 10^{-15} / 73$

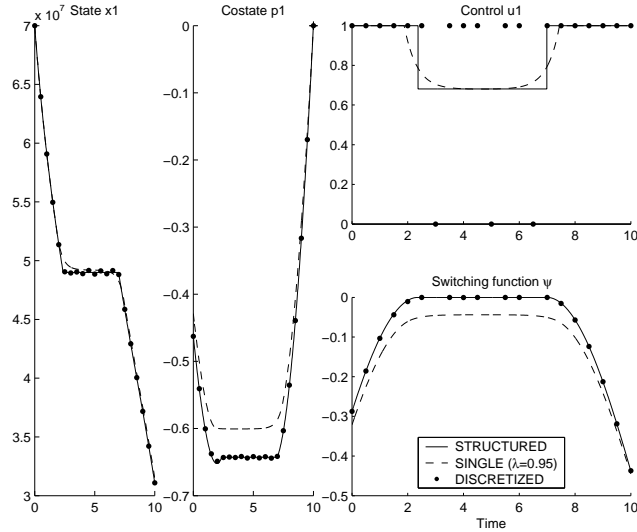
Stabilité de convergence - Problème 2

Pour (P_2) le tir est apparemment bien plus robuste, car on converge toujours malgré les perturbations. Accessoirement, on ne rencontre ici jamais de contrôles hors limites, ce qui explique peut-être cette stabilité.

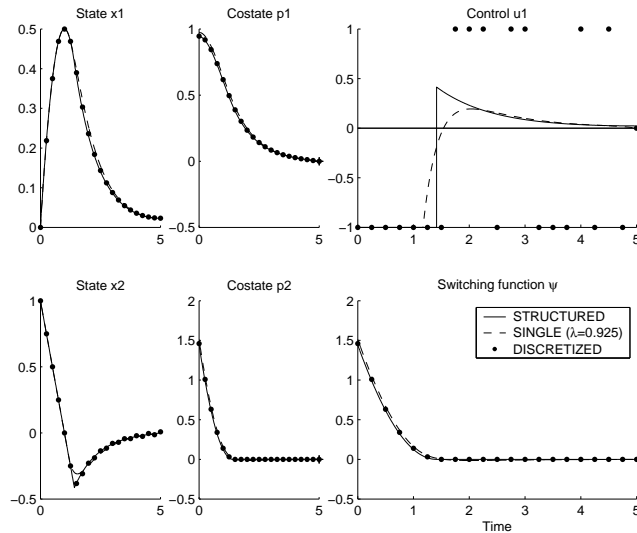
Comparaison des trois formulations

Pour finir, on effectue une comparaison entre la solution du tir structuré et les solutions approchées obtenues par les continuations tir simple et BVP discrétisé. Pour le tir simple (tirets), on prend la solution obtenue juste avant le seuil d'instabilité, c'est-à-dire en $\lambda = 0.95$ pour (P_1) et $\lambda = 0.925$ pour (P_2). Pour le BVP discrétisé (points), on prend la meilleure solution possible,

à savoir la solution raffinée avec l'intégration par trapèzes. La solution de référence du tir structuré est tracée en trait plein.



Solutions du BVP discrétisé, tir simple et structuré, pour (P_1)



Solutions du BVP discrétisé, tir simple et structuré, pour (P_2)

Comme on le voit, les trajectoires discrétisées de l'état et état adjoint pour le BVP discrétisé sont très proches de la solution de référence. Pour le tir simple, p est plus éloigné sur l'arc singulier pour (P_1) , mais on n'est qu'à $\lambda = 0.95$, donc la différence n'est pas si surprenante. En fait, on est plutôt chanceux pour (P_2) d'avoir une solution aussi proche en $\lambda = 0.925$ seulement.

Cette précision des solutions des continuations est probablement pour beaucoup dans la convergence aisée du tir structuré, car on a ainsi un très bon point initial. On note également que la fonction de commutation discrétisée est très proche de la référence, ce qui confirme que cette formulation est bien adaptée à la détection de la structure singulière (et à l'initialisation du tir structuré).

En ce qui concerne le contrôle, les valeurs bang-bang du BVP discrétisé sont inexploitable sur les arcs singuliers, et indiquent simplement le signe de la fonction de commutation quasi nulle en ces points. D'un autre côté, le contrôle du tir simple n'est pas si mauvais (surtout pour (P_1)), étant assez proche de la référence même à l'intérieur des arcs singuliers. Les différences sont en fait localisées autour des instants de commutation (au sens de la structure singulière, c'est-à-dire les débuts/fins d'arcs singuliers), ce qui n'est guère surprenant.

4.5 Formulation Contrôle discrétisé

On a vu que la formulation BVP discrétisé peut donner de très bonnes solutions approchées. Cependant, il est quelque peu frustrant que le contrôle obtenu sur les arcs singuliers soit inutilisable, et gêne en fait la méthode en introduisant des erreurs de raccordement. Aussi nous aimerions, toujours dans l'idée d'une discrétisation, obtenir également une approximation acceptable du contrôle. On s'inspire ici des méthodes semi-directes, où le contrôle discrétisé devient l'inconnue du problème (les méthodes directes discrétisent quant à elles à la fois l'état et le contrôle, cf 4.6).

4.5.1 Formulation

Dans cette formulation, le couple état-état adjoint (x, p) est intégré depuis l'instant initial t_0 , comme dans l'IVP de la méthode du tir simple classique. La différence est que cette intégration n'utilise pas directement l'expression du contrôle optimal donnée par $u^* = \text{Argmin}_{u \in U} \mathcal{H}(t, x, p, u)$. On considère à la place une approximation constante par morceaux du contrôle, dont les valeurs u^i aux instants de discrétisation $t_i, i = 1..d$ deviennent de nouvelles inconnues. Ces valeurs du contrôle doivent par ailleurs satisfaire certaines conditions, notées Control_{cond} , que l'on détaille dans la suite. Pour l'instant, il suffit de dire que ces conditions sont de même dimension que le contrôle u , d'où la forme de cette nouvelle "fonction de tir" :

$$S_{DC} : \mathbf{R}^{n+m \times d} \rightarrow \mathbf{R}^{n+m \times d}$$

avec d le nombre de noeuds de discrétisation et m la dimension du contrôle. On détaille comme d'habitude l'inconnue et la valeur de S_{DC} :

Inconnue z

- inconnue de l'IVP en t_0 (comme pour le tir simple)
- valeurs du contrôle (u^i) aux instants de discrétisation t_i , $i \in [1..d]$ (t_f n'en est pas un ici⁶)

inconnue de l'IVP en t_0	(u^1)	...	(u^d)
----------------------------	---------	-----	---------

Valeur $S_{DC}(z)$

- conditions du contrôle aux instants de discrétisation
- conditions aux limites en t_f (comme pour le tir simple)

$Control_{cond}(t_1)$	$Control_{cond}(t_2)$...	Conditions en t_f
-----------------------	-----------------------	-----	---------------------

Note : Comme le contrôle est généralement de dimension plus faible que le couple (x, p) , cette formulation peut souvent permettre davantage de points de discrétisation que pour le BVP discrétisé.

On arrive maintenant au coeur de cette formulation, à savoir l'expression des conditions sur le contrôle. A un noeud de discrétisation en t_i , la condition sur le contrôle u^i dépend de la structure du contrôle. Si l'on est sur un arc bang (non singulier), alors u^i doit coïncider avec le contrôle optimal u_i^* donné par les conditions nécessaires. Dans le cas singulier, au contraire, on ne doit pas imposer ces conditions nécessaires, sous peine d'avoir les mêmes conséquences que pour le BVP discrétisé. Cependant, la condition sur le contrôle ne devrait pas s'appuyer sur la détection du cas singulier, ce qui passe typiquement par la comparaison de la fonction de commutation ψ (et optionnellement de ses dérivées) à certaines valeurs seuil. Comme on l'a mentionné plus tôt page 108, ceci est numériquement très instable par rapport au seuil. C'est pourquoi on essaye finalement l'expression suivante, qui utilise le fait que le domaine U est dans notre cas un intervalle $[u_{low}, u_{up}]$:

$$\begin{aligned}
 &\text{Si } u^i \in U = [u_{low}, u_{up}] \\
 &\text{Alors } Control_{cond}(t_i) = (u^i - u_i^*) \times Sing_{cond}(t_i) \\
 &\text{Sinon si } u^i > u_{up}, \text{ alors } Control_{cond}(t_i) = u^i - u_{up}, \\
 &\quad \text{sinon } Control_{cond}(t_i) = u^i - u_{low}
 \end{aligned}$$

$$\text{avec } Sing_{cond} = \psi^2 + \dot{\psi}^2$$

La première partie traduit le fait que l'on impose le respect des conditions nécessaires seulement en dehors des arcs singuliers. Si $Control_{cond}(t_i) = 0$, on a soit $u^i = u_i^*$ et les conditions nécessaires sont satisfaites (ceci devrait correspondre aux arcs bang-bang), ou bien $Sing_{cond}(t_i) = 0$, ce qui indique

⁶le contrôle en t_f n'est ici pas utile, comme on n'utilise que des intégrations explicites.

un arc singulier. Le point crucial est que l'on ne choisit pas quel terme s'annule, et qu'on ne force jamais les conditions nécessaires : si le terme $Sing_{cond}$ est suffisamment proche de 0, alors u^i est autorisé à être différent de u_i^* . Concernant l'expression de $Sing_{cond}$, la formule ci-dessus avec la dérivée de ψ donne ici de meilleurs résultats que la formule plus simple $Sing_{cond} = \psi^2$ utilisée dans la formulation BVP discrétisé.

Remarque : il est en fait possible que les deux termes de la condition de contrôle s'annulent, auquel cas ce qui se passe numériquement n'est pas très clair. Ceci correspondrait au début ou à la fin d'un arc singulier (à l'intérieur de l'arc, le contrôle singulier est supposé différent du contrôle optimal incorrect u_i^ , donc $u^i - u_i^*$ ne devrait pas être nul.) Cependant, il est sans doute très peu probable qu'un tel instant coïncide en pratique avec un noeud de discrétisation.*

La seconde partie, qui correspond à la contrainte $u^i \in U$, est telle que la violation de cette contrainte peut prendre des valeurs aussi bien positives que négatives. Elle est en pratique supérieure à l'expression peut-être plus intuitive $Control_{cond}(t_i) = d(u^i, U)$, peut-être en raison de cette propriété de signe. Bien sûr, si le contrôle est de dimension plus grande que 1, cette partie doit être adaptée, mais il est difficile de prédire comment garder la stabilité de la formulation ci-dessus.

Remarque : comme pour la formulation BVP discrétisé, on a tenté d'introduire l'expression algébrique du contrôle singulier. L'idée est d'imposer une contrainte du type " $u^i = u_{singular}^i$ si t_i est dans un arc singulier" dans les conditions de contrôle. Cependant, on a rencontré ici les mêmes difficultés qu'auparavant, c'est-à-dire une sensibilité inacceptable par rapport à l'implémentation pratique de la détection du cas singulier.

Avant de passer aux résultats numériques, on tient à souligner que les propriétés de convergence pour cette formulation contrôle discrétisé restent une question ouverte.

4.5.2 Résultats numériques

On applique ici la même continuation que pour le tir simple et le BVP discrétisé. Le contrôle étant de dimension 1 dans les deux cas, on utilise 50 noeuds de discrétisation pour les deux problèmes (avec une dimension totale de 51 et 52 respectivement). On observe une convergence assez rapide⁷ et sans difficultés particulières, excepté le fait que la première jonction en $\lambda = 0$ prend curieusement un grand nombre de simplexes, surtout pour (P_2)

⁷suivant l'intégration choisie, de 15 à 30 secondes pour (P_1) et 10 à 20 secondes pour (P_2) .

(voir ci-dessous).

Le suivi de chemin est réalisé avec une triangulation $K_1(10^{-1})$ ordinaire, avec mise à l'échelle standard, et deux tirs, en $\lambda = 0$ et $\lambda = 1$. L'initialisation avant le premier tir est extrêmement grossière : on fixe $p(0) = 0$ pour (P_1) et $p(0) = (1, 1)$ pour (P_2) , avec les valeurs pour le contrôle $u^i = 0$, $i = 1..d$ dans les deux cas. Le tir initial converge néanmoins presque instantanément à une solution en dessous de 10^{-15} , à l'exception du Runge Kutta 2 pour (P_2) qui n'atteint que 10^{-2} (ce qui ne semble pas perturber le suivi).

On détaille ici le suivi de chemin :

- **Principal** indique le nombre de simplexes pour le chemin principal.
- **Jonction 1** indique le nombre de simplexes pour la première jonction.
- **Total (H)** indique le nombre global d'évaluations de l'homotopie.
- **Norme finale** indique la norme de H à la solution après le tir en $\lambda = 1$.

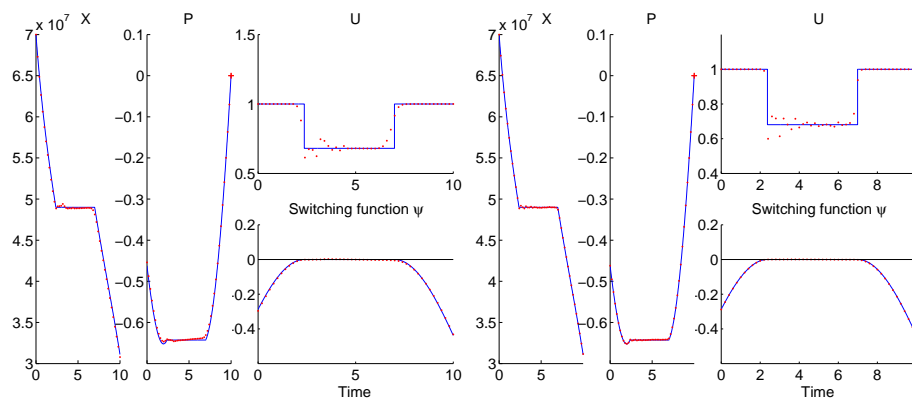
Intégration	Principal	Jonction 1	Total (H)	Norme finale
Euler	23 209	3499	26 268	$5.39 \cdot 10^{-5}$
Point milieu	24 149	6474	29 071	$6.87 \cdot 10^{-6}$
RK2	24 186	6464	29 289	$4.43 \cdot 10^{-6}$
RK4	24 173	6488	28 974	$3.88 \cdot 10^{-6}$

Problème 1 - Suivi de chemin pour la formulation Contrôle discrétisé.

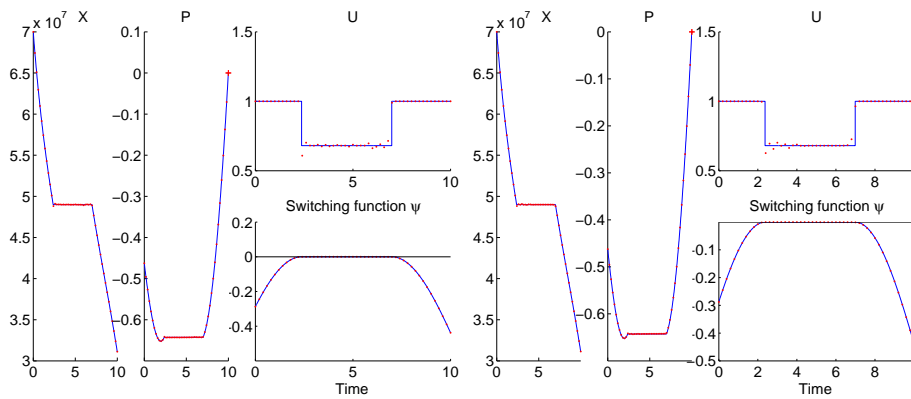
Intégration	Principal	Jonction 1	Total (H)	Norme finale
Euler	5071	27 033	23 792	$4.03 \cdot 10^{-3}$
Point milieu	5144	31 336	27 459	$1.05 \cdot 10^{-2}$
RK2	4678	42 142	33 568	$6.57 \cdot 10^{-4}$
RK4	5150	31 396	28 158	$5.58 \cdot 10^{-4}$

Problème 2 - Suivi de chemin pour la formulation Contrôle discrétisé.

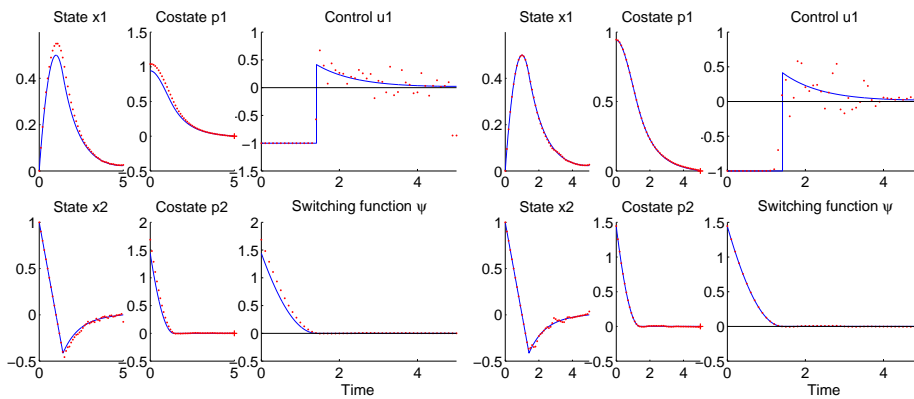
Voici les trajectoires correspondant aux différents choix d'intégration, à savoir Euler, Point milieu, Runge Kutta d'ordre 2 et 4. Une fois encore, les solutions discrétisées sont superposées à la solution de référence du tir structuré (en trait plein).



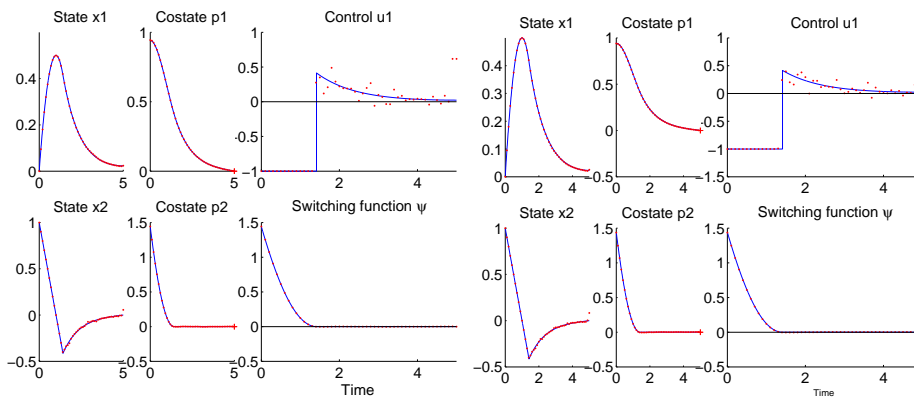
Contrôle discrétisé pour Problème 1 - Euler et Point milieu



Contrôle discrétisé pour Problème 1 - RK2 et RK4



Contrôle discrétisé pour Problème 2 - Euler et Point milieu



Contrôle discrétisé pour Problème 2 - RK2 et RK4

La première constatation est que pour les deux problèmes et tous les choix d'intégration, l'approximation de l'état et de l'état adjoint est très bonne. La seule exception est avec Euler pour (P_2) , qui présente le même phénomène de décalage par rapport à la solution que celui mentionné page 104 pour la formulation BVP discrétisé. La fonction de commutation ψ est

également très précise, et indique clairement la structure singulière pour les deux problèmes.

Mais le point le plus intéressant est bien sûr l'approximation du contrôle, qui est l'inconnue de cette formulation. Dans tous les cas on voit que le contrôle discrétisé coïncide avec la référence sur les arcs bang-bang, ce qui tend à indiquer que les conditions nécessaires sont correctement prises en compte. Quant aux arcs singuliers, on observe que le contrôle discrétisé n'y est pas si éloigné du contrôle singulier de référence, en particulier avec les deux intégrations par Runge Kutta. Cela signifie que le terme $\psi^2 + \dot{\psi}^2$ dans la condition de contrôle parvient effectivement à guider le contrôle vers la valeur singulière correcte, alors même que l'on n'utilise pas l'expression de $u_{singular}$ ⁸ ! Il s'agit là d'un point intéressant, car cela suggère que la méthode pourrait s'appliquer à des problèmes pour lesquels on ne dispose pas de l'expression algébrique du contrôle singulier.

4.6 Approche par méthode directe

On souhaite ici faire une comparaison rapide avec les méthodes directes, où l'on discrétise le problème de contrôle. Le contrôle fait alors partie des inconnues, au lieu d'être obtenu par les conditions nécessaires, et il est donc intéressant de voir comment elles se comportent sur ces problèmes d'arcs singuliers.

4.6.1 Formulation

On utilise ici pour nos expérimentations avec les méthodes directes le code KNITRO, qui traite des problèmes d'optimisation non linéaires (NLP). Des informations détaillées sur les algorithmes employés peuvent être trouvées dans [13, 14], on rappelle juste ici le principe général de la méthode. En bref, KNITRO effectue la résolution du problème NLP originel en résolvant une suite de sous-problèmes barrière. Chaque sous-problème est résolu avec une précision assez faible par des itérations de SQP (en utilisant des régions de confiance), jusqu'à ce qu'on parvienne à annuler le paramètre barrière. Il s'agit d'une méthode de point intérieur, qui a une certaine parenté avec les méthodes de continuation et de pénalisation.

La transformation du problème de contrôle optimal en un problème d'optimisation non linéaire passe par la discrétisation de l'état et du contrôle, qui deviennent les inconnues du problème. L'intégration du système discrétisé

⁸On rappelle que pour ces deux problèmes l'expression du contrôle singulier est obtenue à partir de l'équation du second ordre $\ddot{\psi} = 0$, car le contrôle disparaît de $\dot{\psi} = 0$.

introduit un ensemble de contraintes liées à la dynamique, comme la valeur de l'état à un noeud doit coïncider avec la valeur obtenue par l'intégration depuis le noeud précédent.

Note : une variante de cette approche directe consiste à ne discrétiser que le contrôle, en intégrant simplement l'état depuis l'instant initial. Cette approche "semi-directe" permet un gain important en termes de taille de problème, comme l'état discrétisé ne fait plus partie des inconnues et que les contraintes de dynamique disparaissent.

Concernant l'intégration, on se limite ici à des formules à pas fixe, car on veut pouvoir utiliser la différentiation automatique pour calculer les dérivées des contraintes de dynamique et de l'objectif.

Remarque : il n'y a pas d'état adjoint en tant que tel dans les méthodes directes, même si on peut le relier aux multiplicateurs de Lagrange du problème discrétisé.

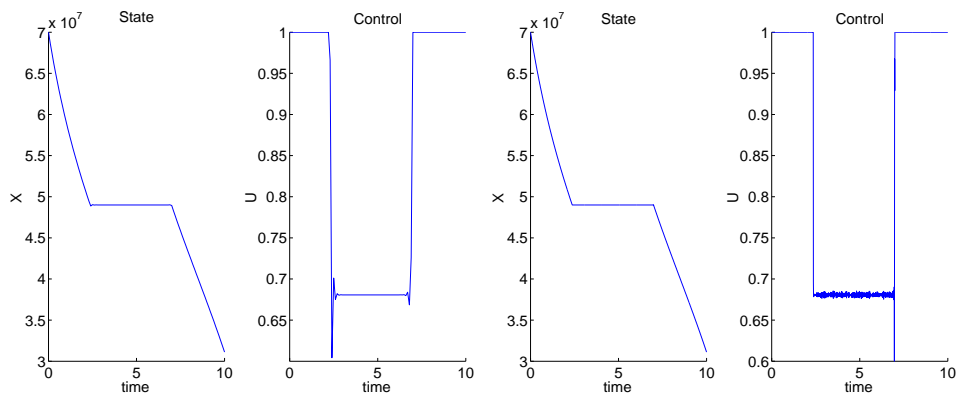
4.6.2 Résultats numériques

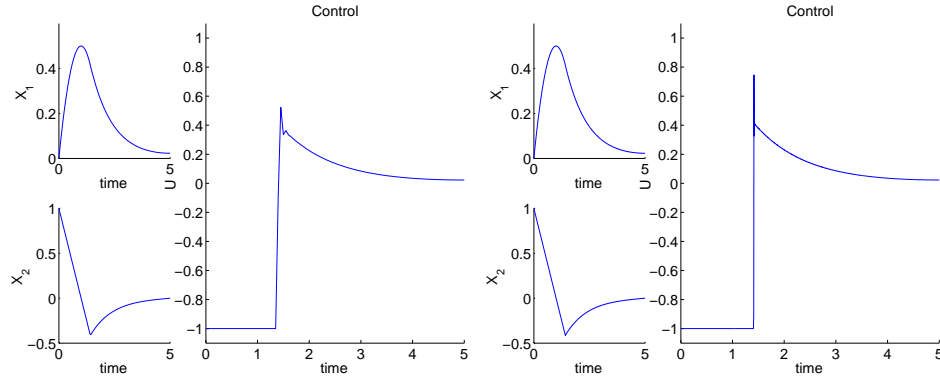
L'intégration pour les deux problèmes est un basique Runge-Kutta 4, avec 100 ou 1000 pas. Les tolérances d'admissibilité et d'optimalité sont fixées à 10^{-12} et 10^{-9} respectivement. On utilise l'initialisation triviale suivante, dans le même esprit que pour la démarche par continuation : $x^i = x_0$, et $u^i = 0$, $\forall i = 1..ns$, avec ns le nombre de pas de discrétisation.

Avec ces réglages, les temps d'exécution sont du même ordre que notre approche :

- 2 secondes (100 pas) et 81 secondes (1000 pas) pour (P_1)
- 26 secondes (100 pas) et 237 secondes (1000 pas) pour (P_2)

Voici les solutions (état et contrôle) pour 100 et 1000 pas :



Méthode directe - Solutions pour 100 et 1000 pas - Problème 1*Méthode directe - Solutions pour 100 et 1000 pas - Problème 2*

On converge visiblement vers les mêmes solutions que celles obtenues par le tir structuré. Cependant, on note la présence d'irrégularités sur les arcs singuliers, et en particulier aux commutations. Les quelques tests que nous avons réalisés tendent à indiquer que se débarrasser complètement de ces artefacts (pour obtenir une solution "lisse" comme celle du tir structuré) pourrait être délicat. Augmenter le nombre de pas est vite limité par le temps d'exécution, et n'est pas toujours convaincant. Par exemple, pour (P_1) avec 1000 pas, les oscillations sur l'arc singulier semblent en fait plus prononcées qu'avec seulement 100 pas. D'autre part, fixer des tolérances plus strictes rend rapidement la convergence bien plus difficile à atteindre.

Toutefois, nous devons insister sur le fait que nos expérimentations avec KNITRO sont très préliminaires, et que l'on peut probablement obtenir de meilleurs résultats. Une autre voie à explorer est l'approche semi-directe, où le contrôle seul est l'inconnue. Enfin, une autre limitation ici est le choix d'une intégration à pas fixe, qui provient de l'emploi de la différentiation automatique.

Conclusion

La première continuation sur le tir simple monter déjà l'apparition progressive de la structure singulière, par l'évolution du contrôle et de la fonction de commutation quand λ augmente le long du chemin. Cependant, à l'approche de la convergence les problèmes deviennent quasiment singuliers, et on perd la structure singulière à cause de l'instabilité numérique de la fonction de commutation au voisinage de 0.

Pour contourner cette difficulté, on discrétise les équations du BVP, les valeurs de l'état et de l'état adjoint aux noeuds de discrétisation devenant partie de l'inconnue de l'homotopie. On observe alors au cours des

expérimentations numériques des indications d'un défaut intrinsèque la formulation, lié à l'application erronée des conditions nécessaires sur les arcs singuliers. Ceci conduit numériquement à de persistantes erreurs de raccordement et des irrégularités sur l'état et l'état adjoint discrétisés. Malgré tout, un bon choix de l'intégration, combiné aux mécanismes de triangulation adaptative et de raffinement de solution décrits au chapitre 2, permet d'obtenir une approximation très précise de l'état, état adjoint et fonction de commutation. On a ainsi une détection fiable de la structure singulière, et une très bonne initialisation pour une résolution précise.

Il est à noter que pour ces deux continuations, l'algorithme simplicial converge en $\lambda = 1$, malgré la présence des arcs singuliers.

On passe alors à la résolution précise, avec une méthode inspirée du tir multiple et adaptée au cas singulier. Grâce à l'information fournie par les continuations précédentes, on converge instantanément vers une solution précise de deux problèmes (cohérente avec les résultats présentés dans [34, 19]).

On a donc résolu les problèmes, mais les très bons résultats de la formulation discrétisée concernant l'état et l'état adjoint nous encouragent à essayer d'obtenir également une approximation du contrôle singulier.

On introduit alors une nouvelle formulation discrétisée, dans laquelle le contrôle remplace l'état et l'état adjoint dans l'inconnue. On essaye d'imposer les conditions nécessaires sur ce contrôle, tout en respectant la structure singulière du problème pour éviter les anicroches du BVP discrétisé. Bien que très sensible à l'implémentation pratique de ces conditions sur le contrôle, cette idée donne des résultats numériques surprenants. On obtient en effet pour les deux problèmes une approximation très proche de l'état et de l'état adjoint, et plus intéressant, des valeurs satisfaisantes du contrôle sur les arcs singuliers, et ce sans utiliser l'expression algébrique du contrôle singulier. Toutefois, on manque de résultats théoriques concernant cette formulation, et le traitement d'un contrôle de dimension supérieure à 1 demanderait vraisemblablement une implémentation soignée. Par ailleurs, les conditions de contrôle peuvent probablement être améliorées pour une meilleure stabilité, peut-être en utilisant une barrière.

Pour finir, il est à noter que toutes les approches décrites dans ce chapitre sont plutôt rapides, avec un temps de convergence n'excédant pas 2 minutes (pour la continuation sur BVP discrétisé avec raffinement). En particulier, la résolution complète des problèmes (BVP discrétisé de base plus tir structuré) prend moins de 15 secondes.

Conclusions et Perspectives

Le thème central de ce travail est lié aux difficultés, en particulier le choix critique du point initial, que rencontrent les méthodes de tir sur des problèmes peu réguliers. Pour y remédier, on introduit une démarche de continuation, et on choisit en pratique les méthodes simpliciales en raison de leur robustesse. Dans ce contexte, on introduit certaines améliorations, comme la triangulation adaptative ou le raffinement de solutions.

On considère tout d'abord une famille de problèmes de transfert orbital à poussée faible, avec une structure de contrôle de type bang-bang et plusieurs centaines de commutations. La méthode simpliciale retrouve bien les résultats ayant déjà obtenus par la continuation différentielle (mais est plus lente, comme il fallait s'y attendre), et la comparaison de trois intégrateurs à pas variable souligne une fois encore l'extrême régularité de cette famille de problèmes par rapport à la poussée maximale. Les expérimentations numériques permettent également de valider le comportement qualitatif de la triangulation adaptative, avec à la fois un gain sur les simplexes parcourus et le tir final. Toutefois, le bilan quantitatif est ici mitigé, à cause du coût numérique important des jonctions adaptatives.

Ensuite, la seconde partie des expérimentations est consacrée à l'étude en parallèle de deux problèmes d'arc singulier. Les démarches de continuation, sur les formulations tir simple et problème aux deux bouts discrétisé, permettent d'obtenir une bonne connaissance de la structure singulière, tant pour la détection des arcs singuliers que pour la nature des arcs bang-bang. Elles fournissent également de bonnes solutions approchées, ce qui permet d'initialiser efficacement la résolution précise des problèmes, réalisée par des méthodes dérivées du tir multiple. Par ailleurs, on introduit également une nouvelle formulation, d'inspiration proche des méthodes directes, dans laquelle le contrôle discrétisé fait partie de l'inconnue. Les premiers résultats numériques obtenus sont intéressants, même si l'implémentation pratique est assez délicate et que le travail théorique reste à faire.

Pour finir, les perspectives futures de ce travail incluent l'étude des contraintes d'état, qui sont également difficiles à traiter par les méthodes

indirectes sans faire d'hypothèses a priori sur la structure. On note en particulier que certains résultats intéressants ont déjà été obtenus par J. Laurent-Varin avec une méthode de point intérieur appliquée à une discrétisation de Runge-Kutta symplectique des conditions d'optimalité, voir [6].

Concernant l'algorithme simplicial, un premier objectif serait d'améliorer le coût numérique de la triangulation adaptative, soit avec une meilleure formulation des jonctions, soit en trouvant une autre façon robuste (et plus rapide) d'effectuer les changements de taille de triangulation. D'autres directions à explorer seraient l'introduction de nouvelles triangulations, intégrateurs (symplectique par exemple), et solveurs non linéaires. Aussi, on souhaiterait établir une comparaison plus approfondie avec les méthodes directes sur les problèmes étudiés.

Annexe A

Présentation du package Simplicial

Ce package a été utilisé pour toutes les expérimentations numériques de ce travail, à l'exception des tests préliminaires pour les méthodes directes (effectués avec Knitro¹). Il implémente une méthode simpliciale adaptée à la résolution de problèmes de contrôle optimal, comme décrit au chapitre 2, et inclut les diverses formulations présentées au chapitre 4 pour les problèmes d'arcs singulier. Le package complet (sources, scripts Matlab, problèmes de démonstration et guide utilisateur) est disponible en ligne.

Page Web : <http://www.enseeiht.fr/lima/apo/simplicial>

Contact : pierre.martinon@enseeiht.fr

Le code lui-même est écrit en Fortran 9x (environ 7000 lignes, essentiellement du F90 standard), et utilise plusieurs codes existants en Fortran77 : le solveur non linéaire utilisé pour le tir, les trois intégrateurs à pas variable, et l'inversion de matrice. Le package Simplicial devrait contenir les fichiers suivants :

Guide utilisateur

Fichiers du noyau Simplicial

- *Simplicial.f90* : programme principal
- *Levels.f90* : jonctions et adaptatif
- *Cell.f90* : manipulation des simplexes
- *Homotopy.f90* : interface avec l'homotopie
- *Shoot.f90* : fonctions de tir
- *Integrators.f90* : interface avec les intégrateurs

¹voir www.ziena.com/knitro.html

- *Defs.f90* : variables globales
- *Makefile*

Fichiers spécifiques au problème

- *ProblemFuns.f90* : sous-routines à fournir par l'utilisateur
- *Problem.cfg* : exemple de fichier de configuration
- *Problem.in* : exemple de fichier d'initialisation

scripts Matlab

- *init.m* : génération de fichier d'initialisation
- *config.m* : génération de fichier de configuration
- *sol.m* : visualisation des solutions
- *path.m* : visualisation des chemins de zéros
- *simp.m* : visualisation des simplexes du chemin
- *valley.m* : visualisation des vallées du chemin
- *plateau.m* : visualisation du plateau de l'homotopie
- *control.m* : évolution du contrôle et fonction de commutation pour plusieurs solutions

Codes externes

- *auxsubs.f* : sous-routines auxiliaires diverses
- *inverse.f* : inversion de matrice (factorisation LU)
- *rkf45.f* : intégrateur Runge Kutta Fehlberg 4-5
- *dop853.f* : intégrateur Dormand Prince 8-5-3
- *odex.f* : intégrateur extrapolation de Gragg Bulirsch Stoer
 - Nonlinear solver : Quasi Newton (Powell method) solver (MINPACK)
Source : *hybrd.f*, by B.S. Garbow, K.E. Hillstrom and J.J More
<http://netlib.bell-labs.com/netlib/minpack/hybrd.f.gz>
 - Integrator : Runge Kutta Fehlberg (4,5) method
Source : *rkf45.f*, by H.A. Watts and L.F. Shampine
<http://netlib.bell-labs.com/netlib/ode/rkf45.f.gz>
 - Integrator : Dormand Prince (8,5-3) method
Source : *dop853.f*, by E. Hairer and G. Wanner
<http://elib.zib.de/pub/elib/hairer-wanner/nonstiff/dop853.f>
 - Integrator : Gragg Bulirsch Stoer extrapolation method
Source : *odex.f*, by E. Hairer and G. Wanner
<http://elib.zib.de/pub/elib/hairer-wanner/nonstiff/odex.f>

- Matrix inversion : via LU factorization (LAPACK-BLAS)
Source : dgetri.f, dgetrf.f
<http://netlib.bell-labs.com/netlib/lapack/double/dgetr{i,f}.f.gz>

A.1 Utiliser le package Simplicial

Pour résoudre un problème de contrôle optimal avec le package Simplicial, l'utilisateur doit compléter les sous-routines suivantes, situées dans *ProblemFuns.f90* :

- **InitPar** : initialisations spécifiques, s'il y en a.
- **Control** : contrôle optimal et fonction de commutation.
- **Dynamics** : dynamique de l'état, état adjoint, et objectif.

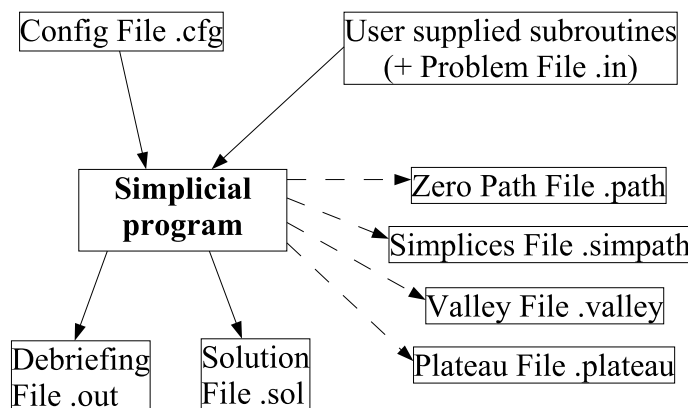
Puis pour chaque problème, deux fichiers d'entrée sont utilisés :

- **Fichier d'initialisation .in** : contient des données relatives au problème.
- **Fichier de configuration .cfg** : réglages pour l'algorithme simplicial.

Après l'exécution, le code Simplicial produit plusieurs fichiers de sortie :

- **Fichier de debriefing .out** : détails sur l'exécution.
- **Fichier de solution .sol** : état, état adjoint, contrôle (et fonction de commutation) générés à la solution (visualisation possible avec le script *sol.m*).
- Fichiers de sortie optionnels : **.path**, **.simpath**, **.valley**, **.plateau** (voir page 143 pour plus de détails).

Voici un schéma récapitulatif des fichiers d'entrée/sortie du code Simplicial, qui sont détaillés dans la suite.



Note : pour chaque problème, un seul préfixe est associé à tous les fichiers d'entrée/sortie. Par exemple si les fichiers d'entrée sont nommés *Demo1.in* et *Demo1.cfg* par exemple, les fichiers de sortie correspondants

seront *Demo1.out* et *Demo1.sol* (et idem pour les fichiers optionnels). Ceci permet de retrouver plus facilement à quel problème sont associés les fichiers.

A.1.1 Subroutines à compléter

Ces trois subroutines sont situées dans le fichier *ProblemFuns.f90*, qui est spécifique à chaque problème.

InitPar

Cette subroutine effectue les diverses initialisations du problème. Elle est appelée au début du programme, après la lecture du fichier *.in*, puis avant chaque évaluation de l'homotopie.

Variables d'entrée

- *mode* : 0 au premier appel, 1 ensuite
- *lambda* : paramètre homotopique utilisé pour la continuation

Interface

```
Subroutine InitPar(mode, lambda)
  implicit none

  integer, intent(in) :: mode
  real(kind=8), intent(in) :: lambda
```

Control

Cette subroutine donne la valeur du contrôle optimal, conformément aux conditions nécessaires. La fonction de commutation est également typiquement évaluée, mais ceci n'est pas obligatoire.

Variables d'entrée

- *lambda* : paramètre homotopique utilisé pour la continuation
- *t* : temps
- *x* : état (dimension *ns*)
- *p* : état adjoint (dimension *nc*)

Variables de sortie

- *u* : contrôle optimal (dimension *m*)
- *psi* : fonction de commutation et ses dérivées (optionnel, dimension totale *dimpsi*)

Interface

```
Subroutine Control(lambda,t,x,p,u,psi)
  implicit none
```

```

real(kind=8), intent(in) :: lambda, t
real(kind=8), intent(in), dimension(ns) :: x
real(kind=8), intent(in), dimension(nc) :: p
real(kind=8), intent(out), dimension(m) :: u
real(kind=8), intent(out), dimension(dimpsi) :: psi

```

Dynamics

Cette subroutine donne la dynamique de l'état, état adjoint, et objectif.

Variables d'entrée

- *dimphi* : dimension totale de la dynamique (> ns+nc avec l'objectif)
- *lambda* : paramètre homotopique utilisé pour la continuation
- *t* : temps
- *x* : état (dimension *ns*)
- *p* : état adjoint (dimension *nc*)
- *u* : contrôle optimal (dimension *m*)

Variables de sortie

- *phi* : dynamique (état, puis état adjoint, et objectif)

Interface

```

Subroutine Dynamics(dimphi,lambda,t,x,p,u,phi)
  implicit none

  integer, intent(in) :: dimphi
  real(kind=8), intent(in) :: lambda, t
  real(kind=8), intent(in), dimension(ns) :: x
  real(kind=8), intent(in), dimension(nc) :: p
  real(kind=8), intent(in), dimension(m) :: u
  real(kind=8), intent(out), dimension(dimphi) :: phi

```

A.1.2 Fichiers d'entrée

A.1.3 Fichier d'initialisation .in

Ce fichier contient les initialisations du problème, comme les dimensions, conditions initiales et finales, point initial, et choix d'intégration. Il peut être édité manuellement, ou généré automatiquement par le script *init.m*. Voici un bref résumé des paramètres de ce fichier.

- *Homotopy choice and problem class*
Entier [1+] : indique le choix de l'homotopie à utiliser.
 Mis à 1 par défaut, il permet de traiter plusieurs homotopies pour un seul fichier *ProblemFuns.f90*.
Entier [0..3] : fixe le type de méthode de tir (voir page 147).

- 0 : BVP discrétisé
- 1 : Tir simple
- 2 : Tir Structuré
- 3 : Tir Structuré léger
- 4 : Contrôle discrétisé

Note : Le choix de l'homotopie et le type de tir dans les fichiers .cfg et .in doivent coïncider.

- *Unknown, State, Costate and Control dimensions*
Entier Entier Entier Entier : les dimensions de l'inconnue de la fonction de tir, et de l'état, état adjoint et contrôle (les dimensions de l'état et état adjoint sont généralement les mêmes, mais peuvent différer si certaines composantes sont constantes et n'ont donc pas besoin d'être intégrées).
- *Objective and Switch dimension*
Entier [1+] : en général 1, l'objectif étant scalaire, mais des valeurs additionnelles peuvent également être calculées si leurs dérivées sont fournies dans la subroutine **Dynamics**.
Entier [0+] : comme pour l'objectif, la fonction de commutation est à valeurs scalaires, mais ses dérivées peuvent aussi être calculée dans la subroutine **Control** (observer la fonction de commutation est particulièrement utile dans le cas des arcs singuliers).
- *Number of IVP unknown values*
Entier : dimension de l'inconnue de l'IVP.
IVP unknown indices
Entier() : indices de l'inconnue de l'IVP dans le vecteur $y = (x, p)$.
- *Number of initial values*
Entier : nombre de valeurs connues en t_0 (données par les conditions aux limites).
Initial values indices
Entier() : indices de ces valeurs dans $(x(t_0), p(t_0))$.
Initial values
Réel() : valeurs initiales.
- *Number of terminal values*
Entier : nombre de valeurs connues en t_f (données par les conditions aux limites).
Terminal values indices
Entier() : indices de ces valeurs dans $(x(t_f), p(t_f))$.
Terminal values
Réel() : valeurs finales.

- *Initial and final time*
Réel Réel : t_0 et t_f
- *Starting point (z_0, λ_0) for zeropath*
Réel(n+1) : point initial z_0 pour l'inconnue de la fonction de tir, et valeur initiale λ_0 pour λ (λ_0 est typiquement 0).
- *Scaling mode*
Entier [0..1] : choix du scaling.
0 : pas de scaling
1 : scaling normal
2 : soft scaling (à [0.01, 10] au lieu de [0.1, 1])
- *Path and solution integrator choice*
Entier [0..7] : choix de l'intégrateur pour le chemin et la solution (cf page 148)
0 : Pas Fixe - Euler
1 : Pas Fixe - Midpoint
2 : Pas Fixe - Runge Kutta 2
3 : Pas Fixe - Runge Kutta 3
4 : Pas Fixe - Runge Kutta 4
5 : Pas Variable - **rkf45**
6 : Pas Variable - **dop853**
7 : Pas Variable - **odex**
- *Steps for fixed step integrators*
Entier Entier : nombre de pas pour les intégrations à pas fixe (chemin et solution).
Variable step integrator abserr and relerr
Réel Réel Réel Réel : tolérances absolues et relatives pour les intégrations à pas variable (idem).
Fixed point minimal progress and maximal iterations
Réel Entier : critère d'arrêt pour les itérations de point fixe.
- *Number of parameters*
Entier [0+] : nombre de paramètres spécifiques du problème.
Parameters
Réel() : paramètres, s'il y en a.

A.1.4 Fichier de configuration .cfg

On trouve ici une liste de réglages pour l'algorithme simplicial. Il peut être édité manuellement, ou généré automatiquement par le script *config.m*. Voici un bref résumé des paramètres de ce fichier.

– *Homotopy choice and problem class*

Entier [1+] : indique le choix de l'homotopie à utiliser.

Mis à 1 par défaut, il permet de traiter plusieurs homotopies pour un seul fichier *ProblemFuns.f90*.

Entier [0..3] : fixe le type de méthode de tir (voir page 147).

0 : BVP discrétisé

1 : Tir simple

2 : Tir Structuré

3 : Tir Structuré léger

4 : Contrôle discrétisé

Note : Le choix de l'homotopie et le type de tir dans les fichiers .cfg et .in doivent coïncider.

– *Triangulation choice*

Entier [1..5] : choix de la triangulation (cf 2.1).

1 : K_1 ¹

2 : J_1 ¹

3 : J_3

4 : J_4

5 : D_1 (expérimental)

– *Labeling choice*

Entier [0..1] : choix de l'étiquetage.

0 : Etiquetage vectoriel traditionnel par l'homotopie

1 : Etiquetage entier (expérimental, cf page 147)

– *Zero Path follow mode*

Entier [-1..3] : choix du mode adaptatif (cf 2.3).

-1 : aucun test de progression des niveaux

0 : tests de progression uniquement, pas d'adaptatif

1 : mode adaptatif, contrôle de déviation seulement

2 : mode adaptatif, déformation anisotrope seulement

3 : mode adaptatif complet

– *Initial and final solver calls*

Entier [0..1] Entier [0..1] : tir au début et à la fin du suivi.

0 : pas de tir

1 : tir

¹Note : on utilise pour K_1 et J_1 les règles de pivotage par permutation au lieu des règles par réflexion (voir [2], 13.3), car elles sont légèrement plus précises.

- *Refinement mode and maximum refinement attempts*
Entier [0..3] : raffinage de solution (cf 2.4).
 0 : pas de raffinage
 1 : raffinage J_3 entre borne de raffinage et borne supérieure
 1 : raffinage J_4 entre borne de raffinage et borne supérieure
 3 : jonctions de raffinage à la borne supérieure
Entier [0+] : nombre maximal de raffinages (pour le mode 3).

- *Deviation tolerance*
Réel : tolérance pour le contrôle de déviation (cf 2.3).
Anisotropic thresholds
Réel Réel Réel : seuils pour la déformation anisotrope (cf 2.3).

- *Maximum simplices for main and junction homotopies, check frequency*
Entier Entier : nombre maximal de simplexes pour l'homotopie principale et les jonctions. *Les homotopies de jonction (cf 2.2 pour plus de détails) ont lieu à l'initialisation du suivi et aux changements de taille de triangulation, et ne devraient pas demander plus de 1000 simplexes (les jonctions rapides prennent souvent moins de 100 simplexes). En cas de difficultés au point initial, ou pour le raffinage de solution, cependant, les jonctions peuvent devenir plus longues, et cette valeur doit être fixée suffisamment haute.*

Entier : fréquence (en simplexes) de vérification des erreurs accumulées sur les simplexes et les étiquetages. Fixer à 0 pour désactiver les vérifications (valeur usuelle 10000 ou 100000).

- *Initial triangulation meshsize for (z,lambda)*
Réel Réel : taille de triangulation initiale δ , donnée par les deux valeurs $\delta(i), i = 1, n$ et $\delta(n + 1)$. (Note : la seconde valeur est sans effet pour la J_3 , le pas suivant λ étant ici prédéterminé.)

- *Lower, refine and upper bounds for homotopic parameter lambda*
Réel Réel Réel : bornes inférieures, de raffinage (entre inférieure et supérieure), et supérieures pour le paramètre homotopique lambda. Les bornes inférieures et supérieures sont typiquement 0 et 1, mais ceci n'est pas obligatoire. En fait, l'algorithme travaille en interne avec $\lambda \in [0, 1]$, qui est re-paramétré sur [lower bound, upper bound] pour l'évaluation de l'homotopie. Ces valeurs peuvent servir à modifier le début et/ou la fin du suivi de chemin, par exemple pour décomposer l'homotopie en plusieurs passes.

- *Zero Path, Simplices, Valley and Plateau output files*

Entier[0..1] Entier[0..1] Entier[0..1] Entier[0..1] : active les fichiers de sortie optionnels `.path`, `.simpath`, `.valley` et `.plateau` (voir page 143 pour plus de détails).

0 : pas de génération

1 : génération

– *Ratio for saved paths*

Entier Entier : ratio des points (ou simplexes) sauvegardés dans les fichiers `.path` (`.simpath`), pour l’homotopie principale et le raffinement. Fixer à 1 pour sauvegarder les chemins complets (*attention : ceci peut produire des fichiers très volumineux si le nombre de simplexes suivi est grand...*).

– *Homotopy Norm and Criterion output in Path*

Entier[0-1] Entier[0-1] : voir page 143 pour la description.

0 : ne pas inclure la norme de l’homotopie / l’objectif dans la sauvegarde du chemin de zéros

1 : inclure la norme de l’homotopie / l’objectif dans la sauvegarde du chemin de zéros

(ces deux options requièrent que “Zero Path output file” soit fixé à 1.)

– *Valley range and step*

Entier Réel : voir page 143 pour la description.

– *Plateau range and step*

Entier Réel : voir page 143 pour la description.

– *Solution generation at final face vertices, barycentric control*

Entier[0-1] :

0 : ne pas générer les solutions pour les sommets de la dernière face

1 : générer les fichiers de solution `.vertxx.sol` pour les sommets de la dernière face

Entier[0-1] : (note : le contrôle barycentrique est expérimental, et en cours d’abandon...)

0 : ne pas utiliser le contrôle barycentrique

1 : réintégrer la solution avec le contrôle barycentrique

Avertissement : certaines options ne sont pas compatibles, et le code effectue certaines vérifications. Si des options conflictuelles sont détectées, un avertissement est affiché et des changements sont faits pour résoudre les conflits. Néanmoins, il est préférable dans ce cas de corriger manuellement les fichiers, car ces modifications ne correspondent pas forcément aux souhaits de l’utilisateur...

A.1.5 Fichiers de sortie

Voici un résumé des fichiers de sortie, qui peuvent être visualisés par les scripts Matlab fournis.

Exécution et fichier de debriefing

Une fois complétées les sous-routines utilisateur et les fichiers d'entrée, il suffit de lancer l'exécutable **Simplicial**. Si les fichiers par défaut *Problem.cfg* et *Problem.in* ne sont pas trouvés, le programme demande le nom (sans extension) des fichiers d'entrée, par exemple "Orbital" pour *Orbital.cfg* et *Orbital.in*. On rappelle que tous les fichiers associés à un problème partagent le même préfixe, ce qui rend aisé le suivi de différents problèmes en parallèle.

A la fin de l'exécution, un fichier de debriefing **.out** est généré, qui contient certaines informations sur la continuation, ainsi qu'une copie des fichiers d'entrée utilisés pour l'exécution. Un fichier solution **.sol** est également généré (voir plus bas), ainsi que certains fichiers optionnels si spécifié dans le fichier **.cfg**.

Fichier **.sol** : solution

Le fichier **.sol** contient la solution générée à la fin de la continuation. Le script *sol.m* trace l'état, état adjoint, contrôle, et fonction de commutation en fonction du temps.

Fichier **.path** : chemin de zéros

Le fichier **.path** contient l'approximation affine par morceaux du chemin de zéros suivi. Le ratio (taille totale / zéros sauvegardés) est fixé dans le fichier **.cfg**, et devrait être supérieur à 1 si l'on s'attend à un grand nombre de simplexes. Ce chemin peut être tracé par le script *path.m*, qui montre l'évolution de chaque composante de z en fonction de λ . Optionnellement, la norme de l'homotopie et l'objectif le long du chemin peuvent également être indiqués, si les options correspondantes sont fixées à 1 (ou davantage pour l'objectif).

Fichier **.simpath** : simplexes du chemin

Le fichier **.simpath** contient les simplexes suivis, pouvant être tracés par le script *simpath.m* (en dimension 2 ou 3).

Note : ce fichier peut devenir très volumineux si le chemin comporte beaucoup de simplexes, et le ratio devrait être alors fixé en conséquence.

Fichier .valley : vallées du chemin

Ce fichier peut servir à visualiser la norme de l'homotopie au voisinage du chemin de zéros, avec une certaine déviation pour chaque composante du chemin. Pour chaque dimension $j \in [1..n]$, pour chaque point z du chemin de zéros, on évalue l'homotopie sur *range* points de part et d'autre de z , avec un pas de *step* (les paramètres *step* et *range* sont dans le .cfg) :

Note : même remarque que précédemment.

Fichier .plateau : plateau de l'homotopie

En dimension 1 ou 2, le fichier "plateau" calcule la valeur de l'homotopie au voisinage de la solution à la convergence. Plus précisément, il contient la valeur et norme de l'homotopie aux points autour de la solution z^* . La largeur totale de la grille est de $1 + 2 \times \text{range}$, avec un pas de *step* (idem).

A.2 Structure général du code Simplicial

Le code Simplicial est organisé en couches, depuis l'algorithme simplicial jusqu'aux sous-routines utilisateur pour l'évaluation de l'homotopie.

Couche 1 : module Simplicial

La première couche contient le programme principal, et implémente le squelette de l'algorithme simplicial, avec la gestion des fichiers d'entrée/sortie.

Fichier : **Simplicial.f90**
 Programme principal : **Simplicial**
 Module : **Simplicialmod**, utilise **Levelsmod** (Couche 2)

Subroutines principales

- *Simplicial* : programme principal
- *Start* : premier simplexe et face étiquetée (première jonction)
- *Follow* : suivi de chemin de zéros
- *Refine* : raffinement de solution
- *Solve* : tir

Couche 2 : module Niveaux

La seconde couche contient les sous-routines liées aux homotopies de jonction et à la triangulation adaptative.

Fichier : **Levels.f90**
 Module : **Levelsmod**, utilise **Cellmod** (Couche 3)

Subroutines principales

- *Junction* : effectue l'homotopie de jonction
- *NewMesh* : calcule la nouvelle taille de triangulation
- *DataScale* : scaling

Couche 3 : module Cellule

La troisième couche implémente les manipulations des simplexes, faces et étiquetages.

Fichier : **Cell.f90**
 Module : **Cellmod**, utilise **Hmod** (Couche 4)

Subroutines principales

- *FirstSimplex* : construction du premier simplexe
 - *FirstLabel* : calcul du premier étiquetage
 - *Lex* : test lexicographique, détermine la face de sortie du simplexe courant
 - *Piv* : règles de pivotages pour le simplexe suivant
- Code externe : subroutines d'inversion **dgetri,dgetrf** (dans *inverse.f*).

Couche 4 : module Homotopie

La quatrième et dernière couche s'occupe de l'évaluation de l'homotopie. Dans ce contexte, l'homotopie est en fait une fonction de tir associée à un problème de contrôle optimal, et ce module est essentiellement une interface avec le module de tir décrit ci-dessous.

Fichier : **Homotopy.f90**
 Module : **Hmod**, utilise **GenFuns**

Subroutines principales

- *Homotopy* : évalue l'homotopie H au point (z, λ)
- *InitJunction* : initialisation de l'homotopie de jonction

Couche 4bis : module Tir

L'implémentation des fonctions de tir est scindée en deux fichiers séparés, pour que toutes les subroutines utilisateur soient au même endroit.

Fonctions génériques : communes à tous les problèmes

Fichier : **Shoot.f90**
 Module : **GenFuns**, utilise **SpecFuns**

Subroutines principales

- *Hom* : interface pour l’homotopie
- *IVP* : évaluation de la fonction de tir
- *Single, Structured, LightStructured, Discrete, Discrete2* :
les diverses formulations des méthodes de tir (voir B.1)
- *Hyb* : interface pour le solveur non linéaire (**hybrd**)
- *Sol* : génération du fichier solution (.sol)

Fonctions spécifiques aux problèmes, fournies par l’utilisateur

Fichier : [Problem]Funs.f90

Module : SpecFuns

Subroutines principales

- *InitPar* : initialisations
- *Control* : contrôle optimal
- *Dynamics* : dynamique de l’état, état adjoint et objectif

Annexe B

Options principales

On rappelle ici les options principales du package `Simplicial`, sans entrer dans les détails (on renvoie le lecteur intéressé au Guide Utilisateur¹ pour une description plus complète des options disponibles.)

B.1 Classe de problème

Ce réglage indique la formulation générale du problème, parmi les cinq actuellement implémentées :

- 0** : BVP discrétisé : introduit au chapitre 4 pour les arcs singuliers. Peut fournir une solution approchée pour initialiser le tir structuré ci-dessous.
- 1** : Tir simple : le grand classique.
- 2** : Tir structuré : introduit au chapitre 4 pour la résolution précise des problèmes d’arcs singuliers.
- 3** : Tir structuré léger : version simplifiée du précédent.
- 4** : Contrôle discrétisé : introduit au chapitre 4, encore expérimental.

Le choix de la formulation est transparent pour toute la partie simpliciale du code, et n’apparaît que dans `Shoot.f90`, où la fonction de tir est évaluée.

B.2 Etiquetages

Le code `Simplicial` comporte en fait un étiquetage entier, en plus de l’étiquetage usuel, activable via le paramètre “labeling choice” :

- 0** : étiquetage standard.
- 1** : étiquetage entier.

On a implémenté l’étiquetage entier décrit dans [2], page 168, défini par :

$$L : \mathbf{R}^{n+1} \rightarrow [1..n + 1]$$

¹ou à l’auteur...

$$v^i = (x^i, \lambda^i) \mapsto 1 + m$$

avec m le nombre de composantes strictement positives de $H(x^i, \lambda^i)$.

Une face est complètement étiquetée si et seulement si les étiquettes de ses sommets coïncident exactement avec l'ensemble $[1..n+1]$

$$f = \{v^1, \dots, v^{n+1}\} \text{ complètement étiquetée} \Leftrightarrow \{L(v^1), \dots, L(v^{n+1})\} = \{1, \dots, n+1\}$$

Trouver la face de sortie est donc extrêmement simple : le sommet qui doit être pivoté est celui ayant la même étiquette que le sommet restant du simplexe (la nouvelle face a donc le même étiquetage que l'ancienne et est bien complètement étiquetée aussi).

La principale difficulté avec cet étiquetage entier est l'obtention de la première face étiquetée. Utiliser directement une homotopie de jonction comme auparavant ne fonctionne pas, car on n'a pas alors de face triviale complètement étiquetée (voir page 31).

Le code est structuré de façon à rendre l'ajout d'un étiquetage aussi simple que possible. Seules trois sous-routines (dans *Cell.f90*) font effectivement intervenir la formulation de l'étiquetage (données ici pour l'étiquetage classique et entier) :

- Trouver la face de sortie du simplexe courant (**Lex / PseudoLex**)
- Etiquetage de la première face (**FirstHLabel / FirstIntLabel**)
- Test d'étiquetage complet (**LabCheck / IntLabCheck**)

B.3 Intégrateurs

B.3.1 Intégrateurs à pas variable

Les intégrateurs à pas variable disponibles pour le moment sont :

- RKF45 (code par Shampine et Watts)

Formules de Runge Kutta imbriquées, méthode de Fehlberg, avec extrapolation locale (ie 5ème ordre utilisé pour l'intégration au lieu du 4ème)

- DOP853 (code par Hairer et Wanner)

Formules de Runge Kutta imbriquées, version améliorée du Dormand-Prince 8(6) (avec l'estimateur d'erreur d'ordre 6 remplacé par un d'ordre 5-3)

- ODEX (code par Hairer et Wanner)

Méthode d'extrapolation Gragg-Bulirsch-Stoer (formule d'Aitken-Neville, cas symétrique, avec le schéma de Gragg comme formule de base d'ordre

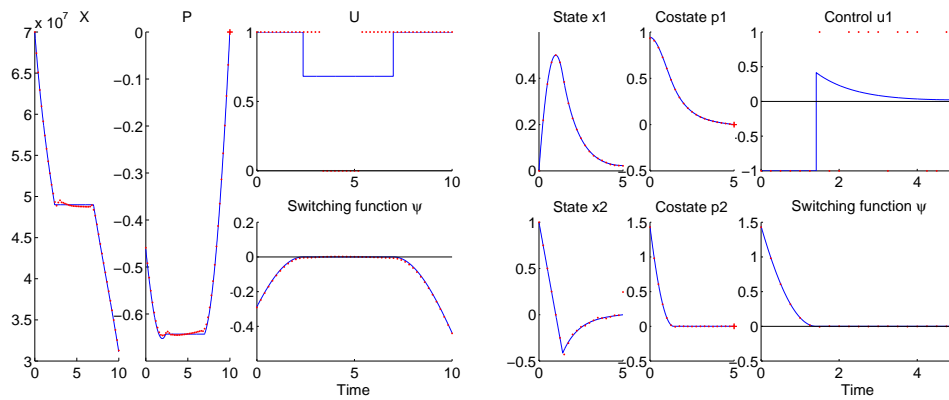
2), à la fois ordre et pas variables.

Les liens vers les sources de ces intégrateurs sont indiqués page 134. Une description complète des algorithmes peut être trouvée dans [27].

B.3.2 Intégrateurs symplectiques

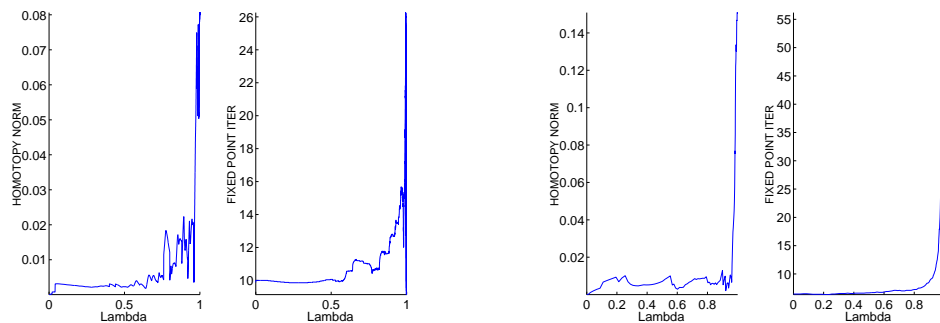
Comme on intègre un système Hamiltonien, nous avons fait quelques essais avec des intégrateurs symplectiques, à savoir les méthodes de Stormer-Verlet et Gauss décrites dans [27]. Il est à noter que notre implémentation de ces méthodes est extrêmement basique, en particulier pour la partie point fixe (les deux méthodes sont implicites). Quoi qu'il en soit, certains des résultats préliminaires sont intéressants.

Pour le premier exemple, on représente la méthode de Stormer-Verlet appliquée à la formulation BVP discrétisé du chapitre 4.



BVP discrétisé BVP pour (P_1) et (P_2) - Stormer-Verlet (symplectique)

Les solutions sont très satisfaisantes, surtout pour (P_2) . On peut regarder l'évolution des itérations moyennes de point fixe le long du chemin.

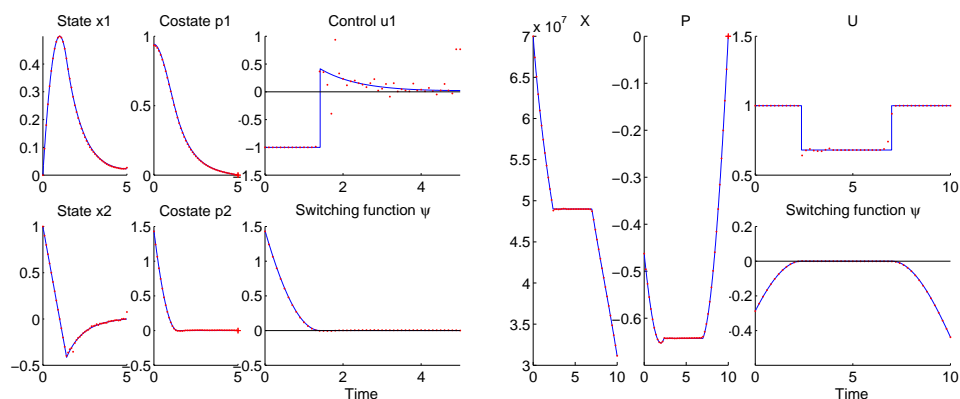


Evolution de $|H|$ et des itérations de point fixe pour (P_1) et (P_2) (S-V)

Il est intéressant de voir que l'augmentation brutale du nombre d'itérations de point fixe correspond à la dégradation de la norme de l'homotopie. Les

deux sont sans doute liés à la perte de régularité lorsqu'on s'approche des arcs singuliers.

Le second exemple est avec la méthode de Gauss sur la formulation contrôle discrétisé.



Contrôle discrétisé pour (P_1) et (P_2) - Gauss (symplectique)

Les solutions sont là aussi très bonnes, du même niveau que les meilleures obtenues par RK4, mais avec un temps d'exécution plus long. De plus, les réglages numériques du point fixe sont ici délicats pour (P_2) . Quoiqu'il en soit, ces difficultés sont probablement dues à notre implémentation grossière des méthodes, qui devra être améliorée avant de poursuivre plus avant les expérimentations.

Annexe C

Exemples de fichiers de problèmes

On reproduit ici certains des fichiers d'entrée utilisés pour les tests, pour montrer qu'il est facile de modifier les réglages pour l'algorithme simplicial ou le problème lui-même. On tient à souligner que tous les tests ont été conduits avec le même code `Simplicial` (avec seulement un exécutable séparé pour chaque problème étudié, indépendamment de la formulation en particulier).

C.1 Problème de démonstration

Il s'agit du premier exemple simple décrit aux chapitres 1 et 2.

C.1.1 `InitPar`

On commence avec l'initialisation des paramètres. Pas grand chose à dire, si ce n'est que l'on considère ici deux continuations : la continuation sur l'objectif introduite page 11, ainsi que celle sur les conditions finales mentionnée page 28. La variable "homotop" correspond au paramètre "Homotopy choice" dans les fichiers `.cfg` et `.in`, et indique la continuation utilisée.

```
Subroutine InitPar(mode, lambda)
  implicit none

  integer, intent(in) :: mode
  real(kind=8), intent(in) :: lambda

  !mode = 0: first call
  !mode = 1: homotopy call

  if (mode == 0) then

    tf0 = tf
    cf0 = cf
```

```

        ci0 = ci

    elseif (homotop == 2) then

        cf(1) = 0.5d0 - lambda

    end if

end subroutine InitPar

```

C.1.2 Control

Voici la subroutine donnant le contrôle optimal, qui correspond aux expressions données aux chapitres 1 et 2.

```

Subroutine Control(lambda,t,x,p,u,psi)
    implicit none

    real(kind=8), intent(in) :: lambda, t
    real(kind=8), intent(in), dimension(ns) :: x
    real(kind=8), intent(in), dimension(nc) :: p
    real(kind=8), intent(out), dimension(m) :: u
    real(kind=8), intent(out), dimension(dimpsi) :: psi

    select case (homotop)

    case (1)

        if (lambda < 1d0) then
            if (abs(p(2)) <= lambda .or. p(2)==0) then
                u(1) = 0d0
            elseif (abs(p(2)) >= 2d0 - lambda) then
                u(1) = -p(2) / abs(p(2))
            else
                u(1) = -p(2)*(abs(p(2))-lambda) / (2d0*(1d0-lambda)*abs(p(2)))
            end if
        else
            if (abs(p(2)) < 1d0 .or. p(2)==0) then
                u(1) = 0d0
            else
                u(1) = -p(2) / abs(p(2))
            end if
        end if

    case (2)

        if (abs(p(2)) < 1d0 .or. p(2)==0) then
            u(1) = 0d0
        else
            u(1) = -p(2) / abs(p(2))
        end if

    case default
        write(outputfid,*) 'ERROR : Control >>> Unknown homotop...',homotop
    end select

```

```

    stop
end select

!switch value and successive derivatives
psi(1) = 1d0 - abs(p(2))

end subroutine Control

```

C.1.3 Dynamics

Voici maintenant la dynamique de l'état, état adjoint et objectif (on peut reconnaître la continuation pour le premier objectif).

```

Subroutine Dynamics(dimphi,lambda,t,x,p,u,phi)
  implicit none

  integer, intent(in) :: dimphi
  real(kind=8), intent(in) :: lambda, t
  real(kind=8), intent(in), dimension(ns) :: x
  real(kind=8), intent(in), dimension(nc) :: p
  real(kind=8), intent(in), dimension(m) :: u
  real(kind=8), intent(out), dimension(dimphi) :: phi

  select case (homotop)

  case (1)

    phi(1) = x(2)
    phi(2) = u(1)
    phi(3) = 0d0
    phi(4) = -p(1)

    !objective dynamic
    if (dimphi > ns+nc) phi(ns+nc+1) = lambda * abs(u(1)) + (1d0-lambda)*u(1)**2

  case (2)

    phi(1) = x(2)
    phi(2) = u(1)
    phi(3) = 0d0
    phi(4) = -p(1)

    !objective dynamic
    if (dimphi > ns+nc) phi(ns+nc+1) = abs(u(1))

  case default
    write(outputfid,*) 'ERROR : Dynamics >>> Unknown homotop...',homotop
    stop
  end select

end subroutine Dynamics

```

C.1.4 Fichiers d'entrée

Voici les fichiers d'entrée correspondant au graphe page 51, avec une triangulation K_1 jusqu'à $\lambda = 0.9$ et un raffinement J_3 jusqu'à $\lambda = 1$ ("refinement mode" fixé à 1 et "refine bound" pour λ à 0.9). On remarque que "path" et "simplices output files" sont fixés à 1, pour générer les fichiers utilisés par les scripts de visualisation.

Configuration file

```

Simplicial algorithm parameters
Homotopy choice and problem class
1 1
Triangulation choice
1
Labeling choice
0
Zero path follow mode
-1
Initial and final solver call
1 0
Refinement mode and Max refinement attempts
1 1
Deviation tolerance
1d-1
Anisotropic thresholds
2.0 0.2 0.1

Maximum simplices for main and junction homotopy, check frequency
500000 10000 100000
Starting triangulation size
1d-1 1d-1
Lower, refine and upper bounds for lambda
0d0 0.9d0 1d0

Path, simplices, valley and plateau output files
1 1 0 0
Ratio for saved paths
1 1
Homotopy Norm and Criterion output in Path
1 0
Valley range and step
50 0.1
Plateau range and steps
50 0.1

Solution generation at final face vertices, Barycentric control
0 0

```

Problem file

```

Homotopy choice and problem class

```

```

1 1
Unknown, State, Costate and Control dimensions
2 2 2 1
Objective and Switch dimension
1 1

Number of IVP unknown values
2
IVP unknown indices
3 4

Number of initial values
2
Initial values indices
1 2
Initial values
0 0

Number of terminal values
2
Terminal values indices
1 2
Terminal Values
0.5 0

Initial and final time
0.0 2.0

Starting point (x0,lambda0) for zeropath
0
0
0

Scaling mode
1
Path and solution integrator choice
4 4
steps for fixed steps integrators
1000 1000
Variable step integrator abserr and relerr
1d-16 1d-14 1d-16 1d-14
Fixed Point minimal progress and maximal iterations
1d-2 25

Number of parameters
0
Parameters

```

C.2 Transfert orbital

Concernant les transferts orbitaux étudiés au chapitre 3, les expressions du contrôle optimal et de la dynamique sont nettement plus compliquées

(pour l'état adjoint en particulier...), et plutôt que de copier/coller quatre pages de code, on renvoie le lecteur intéressé à la thèse de Thomas Haberkorn ([26]), qui a écrit les formules originelles. Voici les fichiers pour le transfert à $T_{max} = 0.1N$, avec un suivi basique uniforme (“follow mode” à 0).

Fichier de configuration

```

Simplicial algorithm parameters
Homotopy choice and problem class
1 1
Triangulation choice
1
Labeling choice
0
Zero path follow mode
0
Initial and final solver call
0 1
Refinement mode and Max refinement attempts
0 0
Deviation tolerance
1d-1
Anisotropic thresholds
2 0.2 0.1

Maximum simplices for main and junction homotopy, check frequency
10000 500 5000
Starting triangulation size
1d-1 1d-1
Lower, refine and upper bounds for lambda
0 0.9 1

Path, simplices, valley and plateau output files
0 0 0 0
Ratio for saved paths
1 1
Homotopy Norm and Criterion output in Path
0 0
Valley range and step
10 0.1
Plateau range and step
10 0.1

Solution generation at final face vertices, barycentric control
0 0

```

Fichier d'initialisation

Celui-ci correspond au tir avec DOP853, comme indiqué par le second “integrator choice” à 6 (le 5 précédent choisit RKF45 pour l'intégration le long du chemin). On note également l'utilisation du “soft scaling” (“scaling mode” à 2).


```
Homotopy choice and problem class
1 1
Unknown, State, Costate and Control dimensions
8 8 8 3
Objective and Switch dimension
1 2

Number of IVP unknown values
8
IVP unknown indices
8 9 10 11 12 13 14 15

Number of initial values
8
Initial values indices
1 2 3 4 5 6 7 16
Initial values
11.625 0.75 0 0.0612 0 0 1500 0

Number of terminal values
8
Terminal values indices
1 2 3 4 5 6 15 16
Terminal Values
42.165 0 0 0 0 1 0 0

Initial and final time
3.14159 2678.52159

Starting point (x0,lambda0) for zeropath
1.53157896392868E+04
-7.82647827592651E+02
-2.13866401262451E+04
-3.97023354861161E+00
3.56466373170132E+03
-1.22948602187706E+00
4.17593485715192E+00
6.08318784116258E+00
0.00000000000000E+00

Scaling mode
2
Integrator choice
5 6
Fixed step integrator steps
1500 2000
Variable step integrator abserr and relerr
1e-8 1e-6 1e-12 1e-10
Fixed point min prog and max iter
1d-2 25

Number of parameters
5
Parameters
```

0.1 1.77 0.0142 5165.86 12.96

Note : T_{max} est le premier paramètre sur la dernière ligne, les autres étant divers coefficients utilisés dans la formulation du transfert.

C.3 Problème de pêche

On a ici le premier des deux problèmes d'arcs singuliers étudiés au chapitre 4. On donne les fichiers pour la continuation BVP discrétisé, comme indiqué par "problem class" à 0. Plus spécifiquement, il s'agit de la formulation trapèzes ("integrator choice" à 1 dans le .in), avec mode adaptatif complet ("follow mode" à 3), et 10 raffinages à la solution ("refinement mode" à 3 et "Max refinement attempts" à 10).

Fichier de Configuration

```

Simplicial algorithm parameters
Homotopy choice and problem class
1 0
Triangulation choice
1
Labeling choice
0
Zero path follow mode
3
Initial and final solver call
1 0
Refinement mode and Max refinement attempts
3 10
Deviation tolerance
1d-2
Thresholds for Anisotropic
2.0 0.2 0.1

Maximum simplices for main and junction homotopy, check frequency
1000000 100000 100000
Starting triangulation size
1d-2 1d-2
Lower, refine and upper bounds for lambda
0 0.9 1.0

Path, simplices, valley and plateau output files
0 0 0 0
Ratio for saved paths
1 1
Homotopy Norm and Criterion output in Path
0 0
Valley range and step
10 0.1
Plateau range and step
10 0.1

```

Solution generation at final face vertices, Barycentric control
0 0

Fichier d'Initialisation

Homotopy choice
1 0
Unknown, State, Costate and Control dimensions
101 1 1 1
Objective and Switch dimension
1 1

Number of IVP unknown values
1
IVP unknown indices
2

Number of initial values
1
Initial values indices
1
Initial values
7e+07

Number of terminal values
1
Terminal values indices
2
Terminal Values
0

Initial and final time
0 10

Starting point (x0,lambda0) for zeropath
0
7e7 0 7e7 0 7e7 0 7e7 0 7e7 0
7e7 0 7e7 0 7e7 0 7e7 0 7e7 0
7e7 0 7e7 0 7e7 0 7e7 0 7e7 0
7e7 0 7e7 0 7e7 0 7e7 0 7e7 0
7e7 0 7e7 0 7e7 0 7e7 0 7e7 0
7e7 0 7e7 0 7e7 0 7e7 0 7e7 0
7e7 0 7e7 0 7e7 0 7e7 0 7e7 0
7e7 0 7e7 0 7e7 0 7e7 0 7e7 0
7e7 0 7e7 0 7e7 0 7e7 0 7e7 0
7e7 0 7e7 0 7e7 0 7e7 0 7e7 0
0.0

Scaling mode
1
Path integrator choice
1 1

```

Steps for fixed step integrator
1 1
Variable step integrator abserr and relerr
1d-16 1d-14 1d-16 1d-14
Fixed point minimal progression and maximal iterations
1d-4 100

Number of parameters
7
Parameters
1 1.75d+07 0.71d0 8.05d+07 2d+07 0d0 1d0

```

C.4 Régulateur quadratique

Pour finir, voici les fichiers pour la continuation contrôle discrétisé (“problem class” à 4), pour le second problème d’arcs singuliers.

Fichier de Configuration

```

Simplicial algorithm parameters
Homotopy choice and problem class
1 4
Triangulation choice
1
Labeling choice
0
Zero path follow mode
0
Initial and final solver call
1 1
Refinement mode and Max refinement attempts
0 0
Deviation tolerance
1d-1
Anisotropic thresholds
2.0 0.2 0.1

Maximum simplices for main and junction homotopy, check frequency
5000000 500000 100000
Starting triangulation size
1d-1 1d-1
Lower, refine and upper bounds for lambda
0d0 0.9d0 1.0d0

Path, simplices, valley and plateau output files
0 0 0 0
Ratio for saved paths
1 1
Homotopy Norm and Criterion output in Path
0 1
Valley range and step
10 0.1

```

Plateau range and step
10 0.1

Solution generation at final face vertices, barycentric control
0 0

Fichier d'Initialisation

Homotopy choice

1 4

Unknown, State, Costate and Control dimensions

52 2 2 1

Objective and Switch dimension

1 2

Number of IVP unknown values

2

IVP unknown indices

3 4

Number of initial values

2

Initial values indices

1 2

Initial values

0 1

Number of terminal values

2

Terminal values indices

3 4

Terminal Values

0 0

t0 tf

0d0 5d0

Starting point (x0,lambda0) for zeropath

1 1

0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0

0.0

Scaling mode

1

Path integrator choice

4 4

fixed step integrators

1 1

Variable step integrator abserr and relerr

```
1d-16 1d-14 1d-16 1d-14  
fpmprog and fpmxiter  
1d-6 100
```

```
Number of parameters  
3  
Parameters  
1d0 -1d0 1d0
```

Bibliographie

- [1] E. ALLGOWER and K. GEORG. Simplicial and continuation methods for approximating fixed points and solutions to systems of equations. *Siam review*, 22(1) :28–85, 1980.
- [2] E. ALLGOWER and K. GEORG. *Numerical Continuation Methods*. Springer-Verlag, Berlin-Heidelberg-New York, 1990.
- [3] E. ALLGOWER and K. GEORG. Piecewise linear methods for nonlinear equations and optimization. *Journal of Computational and Applied Mathematics*, 124 :245–261, 2000. Special Issue on Numerical Analysis 2000 : Vol. IV : Optimization and Nonlinear Equations.
- [4] U.M. ASCHER, R.M. MATTHEIJ, and R.D. RUSSEL. *Numerical Solution of Boundary Value Problems for Ordinary Differential Equations*. Prentice Hall, 1988.
- [5] J.P. AUBIN and A. CELLINA. *Differential Inclusion*. Springer-Verlag, 1984.
- [6] N. BEREND, F. BONNANS, M. HADDOU, J. LAURENT-VARIN, and C. TALBOT. A preliminary interior point algorithm for solving optimal control problems. November 2003. 5th International Conference on Launcher Technology (Madrid).
- [7] N. BEREND, F. BONNANS, M. HADDOU, J. LAURENT-VARIN, and C. TALBOT. On the refinement of discretization for optimal control problems. June 2004. 16th IFAC Symposium on Automatic Control in Aerospace (St Petersburg).
- [8] C. BERGE. *Espaces topologiques*. Dunod, Paris, 1959.
- [9] H. BOCK. Numerical treatment of inverse problems in chemical reaction kinetics. In W. Jiger K. Ebert, P. Deuffhard, editor, *Modelling of Chemical Reaction Systems*, pages 102–125. Springer, Berlin, 1981.
- [10] J.F. BONNANS. The shooting algorithm for optimal control problems : a review of some theoretical and numerical aspects. Technical report, Université d’El Manar (Tunis), 2002. Lectures notes, DEA de Mathématiques Appliquées de l’ENIT.
- [11] BREZIS. *Analyse fonctionnelle*. Masson, 1983.

- [12] A.E. BRYSON and Y.C. HO. *Applied Optimal Control - Optimization, Estimation, and Control*. Blaisdell publishing company, 1969.
- [13] R.H. BYRD, J.C. GILBERT, and J. NOCEDAL. *A trust region method based on interior point techniques for nonlinear programming*. 1998.
- [14] R.H. BYRD, M.E. HRIBAR, and J. NOCEDAL. *An interior point algorithm for large scale nonlinear programming*. 1997.
- [15] JB. CAILLAU, R. DUJOL, J. GERGAUD, T. HABERKORN, P. MARTINON, J. NOAILLES, and D. PREDA. Mise au point d'une méthode de résolution efficace pour les problèmes de contrôle optimal à solution "bang-bang" - application au calcul de trajectoires à poussée faible. Technical report, ENSEEIHT-IRIT, UMR CNRS 5505, 2 rue Camichel, F-31071 Toulouse, January 2004. Rapport de fin de phase 2 - Contrat 02/CNES/0257/00 - DPI 500.
- [16] JB. CAILLAU, R. DUJOL, J. GERGAUD, T. HABERKORN, P. MARTINON, J. NOAILLES, and D. PREDA. Mise au point d'une méthode de résolution efficace pour les problèmes de contrôle optimal à solution "bang-bang" - application au calcul de trajectoires à poussée faible. Technical report, ENSEEIHT-IRIT, UMR CNRS 5505, 2 rue Camichel, F-31071 Toulouse, January 2005. Rapport de contrat - Contrat 02/CNES/0257/00 - DPI 500.
- [17] JB. CAILLAU, J. GERGAUD, and J. NOAILLES. 3D Geosynchronous Transfer of a Satellite : Continuation on the Thrust. *Journal of Optimization Theory and Applications*, 118(3) :541–565, 2003.
- [18] L. CESARI. *Optimization theory and application. Problems with ordinary differential equations*. Springer-Verlag, New York, 1983.
- [19] Y. CHEN and J. HUANG. A numerical algorithm for singular optimal control synthesis using continuation methods. *Optimal Control Applications & Methods*, 15 :223–236, 1994.
- [20] C.W. CLARK. *Mathematical Bioeconomics*. John Wiley & Sons, 1976.
- [21] C. DANG. The d_1 triangulation of \mathbf{R}^n for simplicial algorithms for computing solutions of nonlinear equations. *Mathematics of Operations Research*, 16(1) :148–161, 1991.
- [22] A.F. FILIPPOV. *Differential Equations with Discontinuous Righthand Sides*. Kluwer Academic Publishers, Dordrecht-Boston-London, 1988.
- [23] W.H. FLEMMING and R.W. RISHEL. *Deterministic and Stochastic Optimal Control*. Springer-Verlag, 1975.
- [24] J. GERGAUD. *Résolution numérique de problèmes de commande optimale à solution Bang-Bang par des méthodes homotopiques simpliciales*. PhD thesis, Institut National Polytechnique de Toulouse, 1989.
- [25] J. GERGAUD, T. HABERKORN, and P. MARTINON. Low thrust minimum-fuel orbital transfer : an homotopic approach. *Journal of Guidance, Control and Dynamics*, 27(6) :1046–1060, 2004.

- [26] T. HABERKORN. *Transfert orbital à poussée faible avec minimisation de la consommation : résolution par homotopie différentielle*. PhD thesis, Institut National Polytechnique de Toulouse, 2004.
- [27] E. HAIRER, S.P. NØRSETT, and G. WANNER. *Solving Ordinary Differential Equations I : Nonstiff Problems*, volume 8 of *Springer Series in Comput. Mathematics*. Springer-Verlag, Berlin-Heidelberg-New York, 1993. Second Revised Edition.
- [28] R. HARTL, S. SETHI, and R. VICKSON. A survey of the maximum principles for optimal control problems with state constraints. *Siam review*, 37(2) :181–218, 1995.
- [29] P. MARTINON and J. GERGAUD. An application of pl continuation methods to singular arcs problems. In A. Seeger, editor, *Recent Advances in Optimization*, Lectures Notes in Economics and Mathematical Systems. Springer-Verlag, 2005 (second semester).
- [30] H. NIKAIKO. *Convex structures and economic theory*. Academic Press, 1968.
- [31] H.J. OBERLE and W. GRIMM. BNDSCO - A Program for the Numerical Solution of Optimal Control Problems. Technical Report 515, Institut for Flight System Dynamics, Oberpfaffenhofen, German Aerospace Research Establishment DLR, 1989.
- [32] L. PONTRIAGUINE, V. BOLTIANSKI, R. GAMKRELIDZE, and E. MICHENCHENKO. *Théorie Mathématique des Processus Optimaux*. Editions Mir, Moscou, 1974.
- [33] R. ROBERT. *Contributions à l'analyse non linéaire*. PhD thesis, Université Scientifique et Médicale de Grenoble et Institut National Polytechnique de Grenoble, 1976.
- [34] K. SCHILLING. An algorithm to solve boundary value problems for differential inclusions and applications in optimal control. *Numerical Functional Analysis and Optimization*, 10(7-8) :733–764, 1989.
- [35] J. STOER and R. BULIRSCH. *Introduction to Numerical Analysis*. Springer-Verlag, Berlin-Heidelberg-New York, 1983.
- [36] M.J. TODD. The computation of fixed points and applications. In *Springer Lectures Notes in Economics and Mathematical Systems*, volume 124 (VII). Springer-Verlag, Heidelberg-New York, 1976.
- [37] M.J. TODD. Union jack triangulations. In *Fixed Points : Algorithms and Applications*, pages 315–336. Karamadian, Academic Press, New York, 1977.
- [38] L.T. WATSON, M. SOSONKINA, R.C. MELVILLE, A.P. MORGAN, and H.F. WALKER. Algorithm 777 : HOMPAC90 : A suite of fortran90 codes for globally convergent algorithms. *ACM Transactions on Mathematical Software*, 23 :514–549, 1997.

“Résolution numérique de problèmes de contrôle optimal par une méthode homotopique simpliciale.”

Résumé : On s’intéresse ici à la résolution numérique de problèmes de contrôle optimal peu réguliers. On utilise à la base les méthodes dites indirectes, à la fois précises et rapides, mais en pratique parfois très sensibles à l’initialisation. Cette difficulté nous amène à utiliser une démarche homotopique, dans laquelle on part d’un problème apparenté plus facile à résoudre. Le “suivi de chemin” de l’homotopie connectant les deux problèmes, est ici réalisé par un algorithme de type simplicial. On s’intéresse en premier lieu à un problème de transfert orbital avec maximisation de la masse utile, puis à deux problèmes présentant des arcs singuliers. Les perspectives futures liées à ces travaux comprennent en particulier l’étude de problèmes à contraintes d’état, également délicats à résoudre par les méthodes indirectes. Par ailleurs, on souhaite comparer cette approche avec les méthodes directes, qui impliquent la discrétisation totale ou partielle du problème.

Mots-clefs : contrôle optimal, méthodes de tir, principe du Maximum, contrôle bang-bang, arcs singuliers, homotopie, méthode simpliciale.

Classification MSC2000 : 49J30, 49M05, 65H20

“Numerical resolution of optimal control problems by a simplicial method.”

Abstract : This study deals with the numerical resolution of optimal control problems with a low regularity. We primarily use indirect methods, which are both fast and accurate, but suffer from a great sensitiveness to the initialization. This difficulty leads us to introduce a continuation approach, in which we start from a related, but easier to solve problem. The “path following” between the two problems is here implemented with a simplicial method. We first study an orbital transfer problem with payload maximization, then two singular arcs problems. The future perspectives related to this work include in particular the study of state constraints problems, which are difficult to solve with indirect methods. Also, we would like to compare this approach with direct methods, which imply total or partial discretization of the problem.

Keywords : optimal control, shooting methods, Maximum principle, bang-bang control, singular arcs, continuation, simplicial method.

MSC2000 Classification : 49J30, 49M05, 65H20