



---

*Rapport de stage*

# Utilisation du contrôle optimal pour la résolution des problèmes de Turnpike et Aircraft

*Méthode directe et de collocation*

---

AKOUZ Nezha

2ème année HPC ET BIG DATA  
[nezha.akouz@etu.inp-n7.fr](mailto:nezha.akouz@etu.inp-n7.fr)



## *Tuteurs*

Gergaud Joseph

Cots Olivier

Caillau Jean-Baptiste

DÉPARTEMENT SCIENCES DU NUMÉRIQUE

2021-2022



## **Remerciements**

Tout d'abord, j'adresse mes remerciements à mes deux tuteurs de stage : **Monsieur Joseph Gergaud** et **Monsieur Olivier Cots** pour leur soutien continu, leur patience, leur motivation et de m'avoir offert l'opportunité de ce stage dans leur groupe. Leurs conseils m'ont aidé dans toutes les phases de recherche et de rédaction de ce rapport.

Outre mes tuteurs, je tiens à remercier **Monsieur Jean-Baptiste Caillau** pour ses conseils et son immense savoir.

Je remercie également tous les membres de l'INRIA de m'avoir donné l'opportunité de participer à un séminaire et de pouvoir progresser sur mon sujet de stage.

Enfin, je tiens à remercier tous les membres de la direction de l'IRIT pour leur accueil et leur gentillesse.



# Table des matières

<b>1</b>	<b>Introduction</b>	<b>6</b>
<b>2</b>	<b>Définition du problème de contrôle optimal</b>	<b>7</b>
<b>3</b>	<b>Méthode directe pour la résolution de problèmes singuliers perturbés</b>	<b>7</b>
3.1	Discrétisation . . . . .	7
3.1.1	Schéma de Trapèze . . . . .	7
3.1.2	Schéma d'Euler . . . . .	7
3.2	Résolution du problème . . . . .	8
3.2.1	Méthode directe avec un pas constant . . . . .	8
3.2.2	Méthode directe avec un pas variable . . . . .	8
<b>4</b>	<b>Problème de valeur limite associé</b>	<b>9</b>
<b>5</b>	<b>Méthode de collocation</b>	<b>10</b>
5.1	Principe de la méthode . . . . .	10
5.2	Résolution du problème . . . . .	11
5.2.1	Fonction objective . . . . .	11
5.2.2	Valeur de départ primitive . . . . .	11
5.2.3	Résolution . . . . .	11
<b>6</b>	<b>Solution avec DataFrames</b>	<b>14</b>
<b>7</b>	<b>Sauvegarde de la solution dans un fichier CSV</b>	<b>16</b>
<b>8</b>	<b>Application : Problème d'avion</b>	<b>17</b>
8.1	Valeur des paramètres . . . . .	18
8.2	Définition du problème . . . . .	18
8.3	Résolution du problème . . . . .	19
8.3.1	Méthode directe avec un pas constant . . . . .	19
8.3.2	Méthode directe avec un pas variable . . . . .	20
8.3.3	Sauvegarde de la solution dans un fichier CSV . . . . .	22

<b>9 Conclusion</b>	<b>24</b>
<b>Table des annexes</b>	<b>25</b>
<b>Annexe A Turnpike</b>	<b>25</b>
A.1 Méthode directe . . . . .	25
A.2 Méthode de collocation . . . . .	26
<b>Annexe B Aircraft</b>	<b>29</b>
B.1 Méthode directe . . . . .	29

# 1 Introduction

Récemment, de nouvelles recherches ont été menées sur les problèmes de contrôle optimal avec des propriétés de Turnpike [1] : la solution optimale reste longtemps proche de la solution en régime permanent associée au problème de contrôle optimal statique, et au début et à la fin nous avons un arc de temps court. Nous nous intéressons alors à la résolution numérique de tels problèmes.

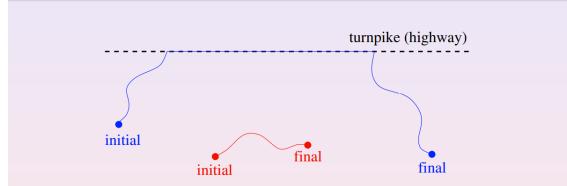


FIGURE 1 – Problème de Turnpike

Un problème de contrôle optimal avec la propriété de Turnpike est un problème de contrôle optimal singulièrement perturbé avec des variables rapides. Nous obtenons alors le problème associé de valeur limite singulièrement perturbé et nous appliquons des méthodes indirectes pour le résoudre en plus des méthodes directes.

La méthode indirecte développée est la méthode de collocation qui résout le problème issu du principe du maximum de Pontryagin. L'objectif est de résoudre le problème en tenant compte de la valeur des différents paramètres : le nombre de points de discrétisation N et l'initialisation des variables.

Ensuite, on appliquera ces méthodes pour la résolution d'un autre problème : Problème d'avion(Aircraft)[2].

Chacune des phases de vol, présentées dans la figure suivante a une dynamique différente due par exemple à la position des becs et des volets sur la voilure, la présence ou non du train d'atterrissement ou encore la variation du régime moteur.

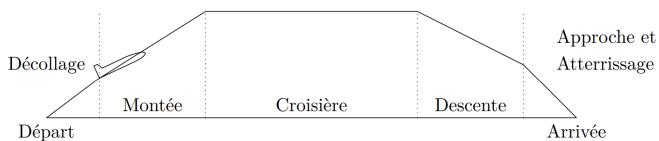


FIGURE 2 – Problème d'avion

Intéressons nous à la phase de montée, le but est de minimiser le temps durant cette phase.

## 2 Définition du problème de contrôle optimal

Nous considérons la formulation suivante du problème de contrôle optimal de Turnpike :

$$\begin{cases} \min 0.5 \int_0^1 ((y_1(t) - 1)^2 + (y_2(t) - 1)^2 + (u(t) - 2)^2) t_f dt, \\ \dot{y}(t) = t_f(y_2(t), 1 - y_1(t) + y_2^3(t) + u(t)) \\ y(0) = (1, 1) \\ y(1) = (3, 0) \end{cases}$$

Tout d'abord, nous allons résoudre ce problème en utilisant la méthode directe avec le schéma de Trapèze.

## 3 Méthode directe pour la résolution de problèmes singuliers perturbés

### 3.1 Discrétisation

Nous cherchons une solution :  $y_i = y(t_i)$  pour i dans  $[1 : N+1]$  où N est le nombre de points de discrétisation.

#### 3.1.1 Schéma de Trapèze

On a :

$$y_{i+1} = y_i + (t_{i+1} - t_i) \frac{dy_{i+1} + dy_i}{2},$$

$$y_{i+1} = y_i + h_i \frac{dy_{i+1} + dy_i}{2},$$

avec  $h_i = t_{i+1} - t_i$ .

#### 3.1.2 Schéma d'Euler

La valeur de l'intégrale est approximée par :

$$y_{i+1} = y_i + (t_{i+1} - t_i) dy_i,$$

$$y_{i+1} = y_i + h_i dy_i,$$

avec  $h_i = t_{i+1} - t_i$ .

## 3.2 Résolution du problème

### 3.2.1 Méthode directe avec un pas constant

Dans ce cas, on a  $h_i = t_{i+1} - t_i = \Delta t$

Selon la valeur de  $t_f$  et N on obtient des solutions différentes.

Pour  $t_f = 100$  et  $N=100$  :

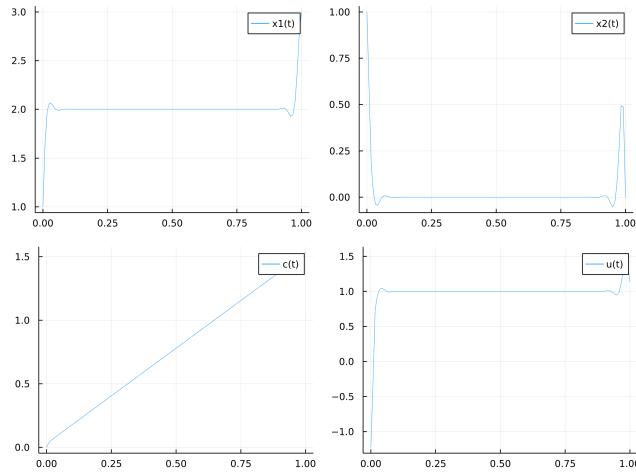


FIGURE 3 – Représentation des variables  $x_1$  et  $x_2$  et de la fonction objective et du contrôle  $u$  pour  $t_f = 100$  et  $N=100$

$x_1$  et  $x_2$  sont la solution approximative du problème et  $c$  est la valeur objective. Cette solution est obtenue en utilisant la solution statique  $(2,0,-1,-1)$  comme valeur de départ primaire.

Dans la phase statique, la solution est constante. Donc, utiliser un pas constant ne donne pas une solution très optimale. C'est pourquoi nous allons résoudre le problème en utilisant un pas variable.

### 3.2.2 Méthode directe avec un pas variable

Dans ce cas,  $h_i$  n'est pas constant. Il est considéré comme une variable. Cette solution est obtenue en utilisant la solution statique  $(2,0,-1,-1)$  comme valeur de départ primaire.

Ainsi, la fonction objective est de minimiser  $c[N+1] + \alpha \sum_{i=1}^N ((h_i - \Delta t)^2)$ .  $c[N+1]$  est la fonction de coût et  $\alpha = 10^{-3}$  et le facteur de normalisation.

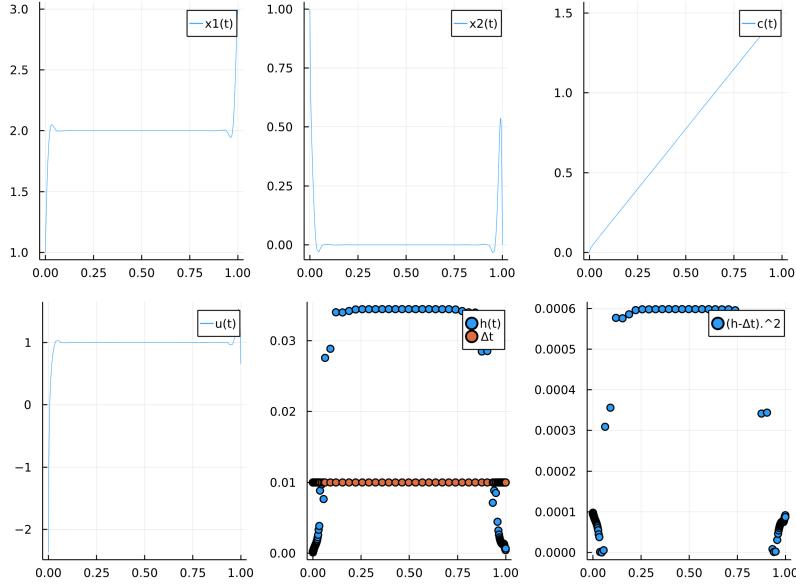


FIGURE 4 – Représentation de la solution, de la fonction objective, du contrôle et des erreurs locaux en fonction des temps  $t_i$  pour  $t_f = 100$  et  $N = 100$

On constate que pendant la phase statique, le pas  $h_i$  est constant et supérieur au pas constant  $\Delta t$ . Aussi, au début et à la fin on a des petits pas.

L'objectif dans les parties suivantes est de résoudre le problème par la méthode indirecte en tenant compte de l'influence de  $t_f$ .

## 4 Problème de valeur limite associé

Nous allons résoudre le problème en utilisant une méthode indirecte appelée méthode de collocation. Pour cela, nous allons d'abord définir le problème de valeur limite associé et utiliser les deux schémas définis précédemment (schémas de Trapèze et d'Euler). Nous verrons pour quelle valeur de  $t_f$  nous pouvons résoudre le problème.

Pour ce problème :

$$\begin{cases} \min 0.5 \int_0^1 ((y_1(t) - 1)^2 + (y_2(t) - 1)^2 + (u(t) - 2)^2) t_f dt, \\ \dot{y}(t) = t_f(y_2(t), 1 - y_1(t) + y_2^3(t) + u(t)) \\ y(0) = (1, 1) \\ y(1) = (3, 0) \end{cases}$$

L'Hamiltonien[5] associé est donné par :

$$H(y, p, p^0, u) = p^0(0.5t_f((y_1(t)-1)^2+(y_2(t)-1)^2+(u(t)-2)^2))+p_1t_fy_2(t)+p_2t_f(1-y_1(t)+y_2^3(t)+u(t))$$

Les équations adjointes[5] :

$$\begin{aligned}\dot{p}(t) &= -\frac{\partial H}{\partial y} = -t_f(p^0(y_1(t) - 1) - p_2(t), p^0(y_2(t) - 1) + p_1(t) + 3p_2(t)y_2^2(t)) \\ \dot{y}(t) &= \frac{\partial H}{\partial p} = t_f(y_2(t), 1 - y_1(t) + y_2^3(t) + u(t))\end{aligned}$$

La condition de maximisation du Hamiltonien[5] donne :

$$0 = \frac{\partial H}{\partial u} = p^0 t_f(u - 2) + p_2 t_f$$

Si  $p^0 = 0$  alors  $(p^0, p) = (0, (0, 0))$ , ce qui est impossible, ainsi  $p^0 = -1$ .

Alors, on obtient les équations suivantes :

$$\begin{aligned}\dot{p}(t) &= t_f(y_1(t) - 1 + p_2(t), y_2(t) - 1 - p_1(t) - 3p_2(t)y_2^2(t)) \\ \dot{y}(t) &= t_f(y_2(t), 1 - y_1(t) + y_2^3(t) + u(t)) \\ u(t) &= 2 + p_2(t)\end{aligned}$$

avec  $y = (y_1, y_2)$

Ces équations permettent de définir le nouveau problème :

$$\begin{cases} \varepsilon \dot{y}(t) = (y_2(t), 1 - y_1(t) + y_2^3(t) + u(t)) \\ \varepsilon \dot{p}(t) = (y_1(t) + p_2(t) - 1, y_2(t) - 3y_2^2(t)p_2(t) - p_1(t) - 1) \\ y(0) = (1, 1) \\ y(1) = (3, 0) \end{cases}$$

Avec  $\varepsilon = 1/t_f$ ,  $y = (y_1, y_2)$  et  $p = (p_1, p_2)$

## 5 Méthode de collocation

### 5.1 Principe de la méthode

Dans la plupart des cas, la solution obtenue par les deux schémas précédents n'est pas la solution exacte du problème. L'erreur de discréétisation est la principale source d'erreur dans les méthodes de différences finies. L'objectif est de minimiser l'erreur de discréétisation en tenant compte des valeurs de  $\varepsilon$  et d'autres paramètres tels que la taille du maillage et l'estimation initiale de la solution.

Cette méthode consiste à résoudre le problème limite associé. Ainsi, nous prenons également en compte  $p$  (variables duals) comme variable de notre problème. Nous utilisons les deux schémas précédents (Trapèze et Euler) et nous essayons de minimiser l'erreur entre les solutions obtenues par chaque schéma. Le but est de trouver pour quel  $\varepsilon$  le problème peut être résolu.

## 5.2 Résolution du problème

### 5.2.1 Fonction objective

Soient  $x$  ( $y$ ) la solution obtenue par le schéma de Trapèze (Euler),  $h$  le pas variable et  $\alpha = 10^{-3}$  le facteur de régularisation de la norme L2.

La fonction objectif est :  $\frac{\sum_{i=1}^4 (\sum_{j=1}^N ((x_{ij} - y_{ij})^2))}{N} + \alpha \sum_{i=1}^N ((h_i - \Delta t)^2)$

### 5.2.2 Valeur de départ primitive

Dans notre analyse nous considérons deux valeurs pour l'initialisation :  
La solution statique  $(2,0,-1,-1)$ ,  $h = \Delta t$  et une solution arbitraire  $(8,6,-4,-7)$ ,  $h = 0$ .

### 5.2.3 Résolution

Dans tous les graphiques suivants,  $x_1, x_2$  ( $y_1, y_2$ ) sont les variables primaires et  $x_3, x_4$  ( $y_3, y_4$ ) sont les variables duales.

Les graphiques suivants montrent la solution pour  $t_f = 10$  et  $N = 120$ .

La solution statique en tant que valeur de départ de la variable primaire :

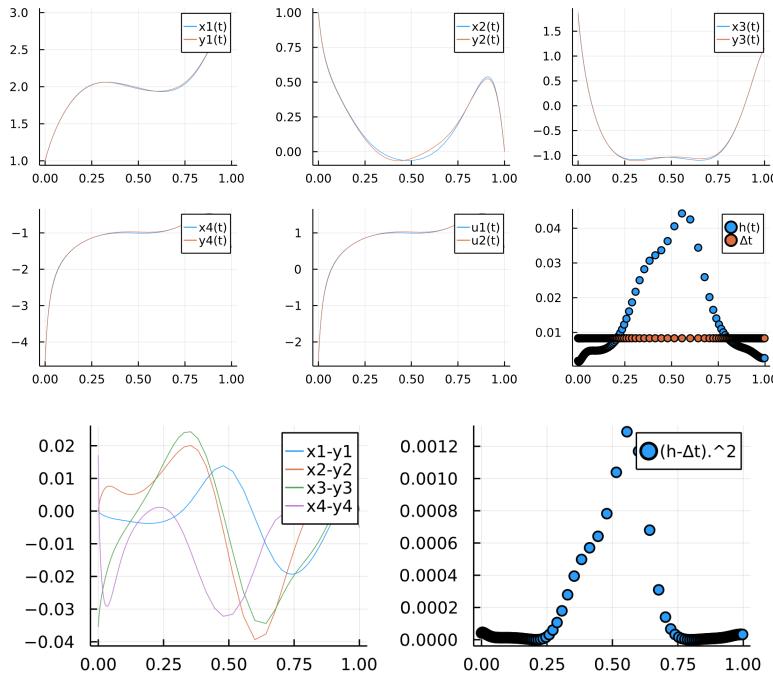


FIGURE 5 – Comparaison des solutions obtenues par le schéma d'Euler et le schéma de Trapèze ainsi que la différence entre le pas variable et le pas uniforme en fonction des temps  $t_i$  pour  $t_f = 10$  et  $N=120$  avec la solution statique comme initialisation.

Une solution arbitraire comme valeur de départ primaire :

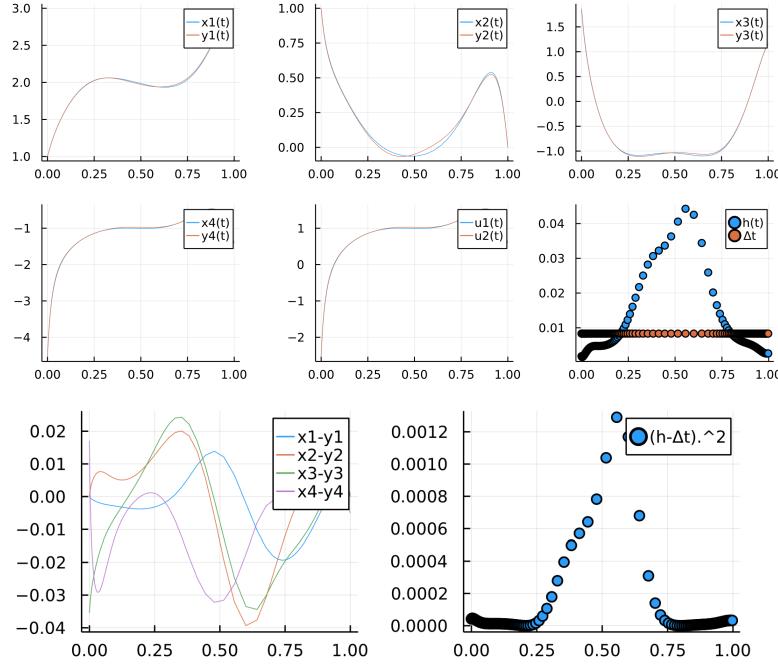


FIGURE 6 – Comparaison des solutions obtenues par le schéma d'Euler et le schéma de Trapèze ainsi que la différence entre le pas variable et le pas uniforme en fonction des temps  $t_i$  pour  $t_f = 10$  et  $N=120$  avec la solution arbitraire comme initialisation.

Pour cette valeur de  $t_f$ , nous n'observons pas de différence entre la solution obtenue avec la solution statique comme valeur de départ primaire et la solution obtenue avec une solution arbitraire comme valeur de départ primaire.

Dans les deux cas, nous observons que les solutions convergent vers la solution statique pendant la phase statique. De plus, dans cette phase, le pas est plus élevé que le pas constant, ce qui est l'un des objectifs de cette méthode.

Les graphiques suivants montrent la solution pour  $t_f = 80$  et  $N = 120$ .

La solution statique comme valeur de départ primaire :

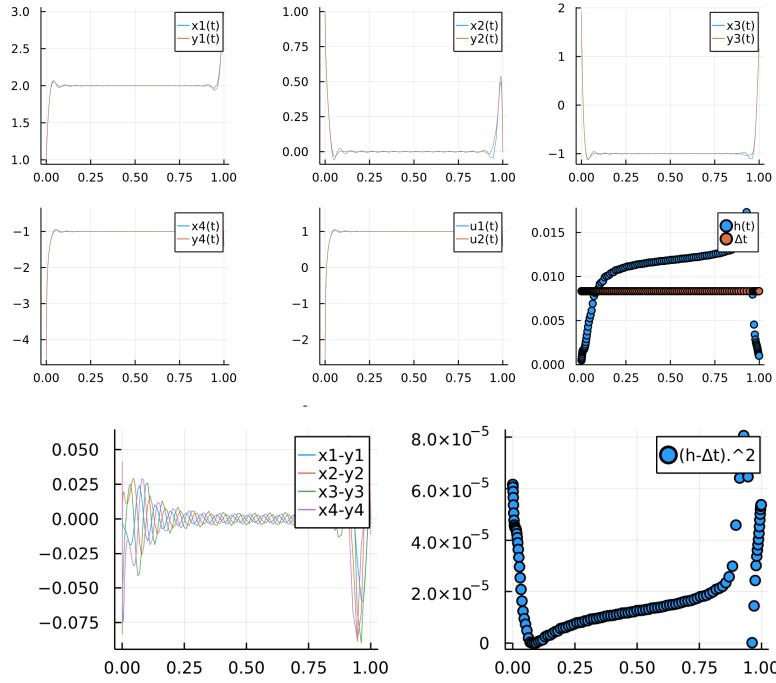


FIGURE 7 – Comparaison des solutions obtenues par le schéma d’Euler et le schéma de Trapèze ainsi que la différence entre le pas variable et le pas uniforme en fonction des temps  $t_i$  pour  $t_f = 80$  et  $N=120$  avec la solution statique comme initialisation.

Une solution arbitraire comme valeur de départ primaire :

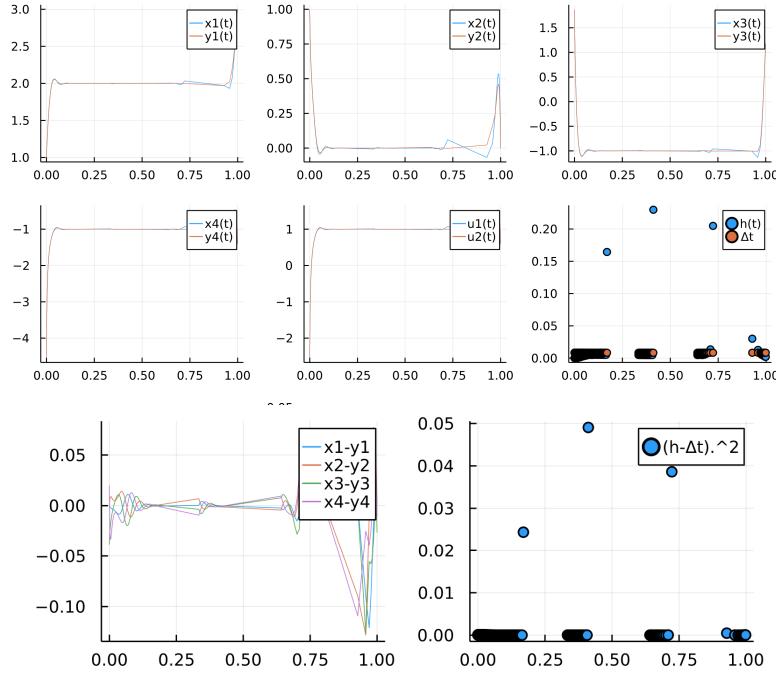


FIGURE 8 – Comparaison des solutions obtenues par le schéma d’Euler et le schéma de Trapèze ainsi que la différence entre le pas variable et le pas uniforme en fonction des temps  $t_i$  pour  $t_f = 80$  et  $N=120$  avec la solution arbitraire comme initialisation.

Contrairement au cas avec  $t_f = 10$ , dans ce cas  $t_f = 80$  nous observons une grande différence entre les solutions. Nous pouvons donc conclure que l’initialisation est un facteur très important.

Avec la solution arbitraire comme valeur de départ de la primitive, nous pouvons observer quelques impulsions pendant la phase statique. De plus, la valeur du pas n’est pas la meilleure pendant cette phase.

Alors, nous pouvons dire que la solution statique comme valeur de départ primaire est la meilleure initialisation. La partie suivante permettra de mieux l'affirmer.

## 6 Solution avec DataFrames

Afin de trouver la valeur de  $t_f$  pour laquelle nous pouvons résoudre le problème, et aussi pour comparer la solution avec la solution statique et la solution arbitraire comme valeur de départ primaire, nous utilisons un DataFrame[3].

	tf	N	Objective_Value_SS	Objective_Value_AS
	Int64	Int64	Any	Any
1	10	120	0.000592421	0.000592421
2	20	120	0.000666687	0.000666687
3	30	120	0.000734199	0.000734199
4	40	120	0.000820513	0.117101
5	50	120	0.00083261	0.000881642
6	60	120	0.000996987	0.000996987
7	70	120	0.00129785	0.00129785
8	80	120	0.00162432	0.00171839
9	90	120	0.00218751	X
10	100	120	X	X

TABLE 1 – La valeur de la fonction objective selon  $t_f$  et pour  $N = 120$ . X signifie que l'algorithme ne converge pas.

	tf	N	Objective_Value_SS	Objective_Value_AS
	Int64	Int64	Any	Any
1	10	200	0.000221724	0.000221724
2	20	200	0.000239161	0.000239161
3	30	200	0.000250396	0.000247127
4	40	200	0.000262578	X
5	50	200	0.000260702	0.000276928
6	60	200	0.000273672	0.000289233
7	70	200	0.000288138	0.000314302
8	80	200	0.000306197	0.000313787
9	90	200	0.0003674	0.000333481
10	100	200	0.000402549	0.000402549
11	110	200	X	X

TABLE 2 – La valeur de la fonction objective selon  $t_f$  et pour  $N = 200$ . X signifie que l'algorithme ne converge pas.

La colonne Objective-Value-SS respectivement la colonne Objective-Value-AS signifie la valeur de la fonction objective avec la solution statique respectivement une solution arbitraire comme initialisation.

Les résultats obtenus montrent que la solution statique est la meilleure valeur de départ primaire. Pour des grandes valeurs de  $t_f$  (plus petite valeur de  $\varepsilon$ ) nous ne pouvons pas résoudre le problème correctement dans les deux cas.

La plus grande valeur de  $t_f$  pour laquelle on peut résoudre ce problème avec solution statique comme initialisation est 90.

## 7 Sauvegarde de la solution dans un fichier CSV

Afin d'avoir accès à la solution à tout moment, nous avons décidé de l'enregistrer dans un fichier CSV[4].

Nous avons écrit quelques fonctions qui peuvent faire cela. Nous pouvons créer une liste d'expériences avec les noms de l'algorithme et du schéma, la valeur de  $t_f$  et N . Nous avons créé le fichier CSV à partir d'un DataFrame et chaque fois avant d'exécuter une expérience, nous vérifions si nous l'avons déjà enregistrée dans le fichier CSV ou non. Cette technique permet d'avoir accès à la solution après avoir fermé le notebook. Nous avons également écrit une fonction qui permet de supprimer une expérience du fichier CSV.

Nous avons créé une liste d'expériences avec les méthodes directes et la méthode de collocation. Le tableau suivant résume l'ensemble des travaux de cet article.

La colonne Algorithme correspond à l'algorithme utilisé pour résoudre le problème : bvp-solve pour la méthode de collocation, nlpsolve-V correspond à la méthode directe avec pas variable et nlpsolve-U correspond à la méthode directe avec pas constant.

Row	Algorithm String15	Scheme String15	Initialization String	tf Int64	N Int64	Convergence String3	Objective_Value Float64	Time_cpu Float64
1	bvpsolve	EulerTrapeze	[0.0083333333333333, 2.0, 0.0,...	10	120	yes	0.000592421	1.73366
2	bvpsolve	EulerTrapeze	[0.0083333333333333, 2.0, 0.0,...	20	120	yes	0.000666687	5.02504
3	bvpsolve	EulerTrapeze	[0.0083333333333333, 2.0, 0.0,...	40	120	yes	0.000820513	6.47571
4	bvpsolve	EulerTrapeze	[0.0083333333333333, 2.0, 0.0,...	50	120	yes	0.00083261	8.97828
5	bvpsolve	EulerTrapeze	[0.0083333333333333, 2.0, 0.0,...	60	120	yes	0.000996987	9.64052
6	bvpsolve	EulerTrapeze	[0.0083333333333333, 2.0, 0.0,...	70	120	yes	0.00129785	4.78975
7	bvpsolve	EulerTrapeze	[0.0083333333333333, 2.0, 0.0,...	80	120	yes	0.00162432	3.58699
8	bvpsolve	EulerTrapeze	[0.0083333333333333, 2.0, 0.0,...	90	120	yes	0.00218751	4.17051
9	bvpsolve	EulerTrapeze	[0.0083333333333333, 2.0, 0.0,...	100	120	no	0.003664	137.364
10	nlpsolve_V	Trapeze	[0.0083333333333333, 2.0, 0.0,...	10	120	yes	1.74453	0.688485
11	nlpsolve_V	Trapeze	[0.0083333333333333, 2.0, 0.0,...	20	120	yes	1.62239	1.99896
12	nlpsolve_V	Trapeze	[0.0083333333333333, 2.0, 0.0,...	30	120	yes	1.58162	1.30791
13	nlpsolve_V	Trapeze	[0.0083333333333333, 2.0, 0.0,...	40	120	yes	1.56123	1.38206
14	nlpsolve_V	Trapeze	[0.0083333333333333, 2.0, 0.0,...	50	120	yes	1.54899	1.48785
15	nlpsolve_V	Trapeze	[0.0083333333333333, 2.0, 0.0,...	60	120	yes	1.54083	1.70827
16	nlpsolve_V	Trapeze	[0.0083333333333333, 2.0, 0.0,...	70	120	yes	1.535	1.78483
17	nlpsolve_V	Trapeze	[0.0083333333333333, 2.0, 0.0,...	80	120	yes	1.53063	1.93194
18	nlpsolve_V	Trapeze	[0.0083333333333333, 2.0, 0.0,...	90	120	yes	1.52723	2.09166
19	nlpsolve_V	Trapeze	[0.0083333333333333, 2.0, 0.0,...	100	120	yes	1.52451	2.78948
20	nlpsolve_U	Trapeze	[2.0, 0.0, 1.5, 1.0]	10	120	yes	1.74595	0.0679598
21	nlpsolve_U	Trapeze	[2.0, 0.0, 1.5, 1.0]	20	120	yes	1.62509	0.0671753
22	nlpsolve_U	Trapeze	[2.0, 0.0, 1.5, 1.0]	30	120	yes	1.5853	0.069521
23	nlpsolve_U	Trapeze	[2.0, 0.0, 1.5, 1.0]	40	120	yes	1.56572	0.0715824
24	nlpsolve_U	Trapeze	[2.0, 0.0, 1.5, 1.0]	50	120	yes	1.55418	0.0649825
25	nlpsolve_U	Trapeze	[2.0, 0.0, 1.5, 1.0]	60	120	yes	1.54664	0.0589106
26	nlpsolve_U	Trapeze	[2.0, 0.0, 1.5, 1.0]	70	120	yes	1.54136	0.0637847
27	nlpsolve_U	Trapeze	[2.0, 0.0, 1.5, 1.0]	80	120	yes	1.53748	0.0699854
28	nlpsolve_U	Trapeze	[2.0, 0.0, 1.5, 1.0]	90	120	yes	1.53453	0.0675259
29	nlpsolve_U	Trapeze	[2.0, 0.0, 1.5, 1.0]	100	120	yes	1.53222	0.0716552

TABLE 3 – Les résultats obtenus par la méthode de collocation et la méthode directe pour les deux types de pas avec N=120.

On remarque que pour la méthode de collocation, la valeur maximale de  $t_f$  pour laquelle l'algorithme peut converger est 90. Pour les deux méthodes directes, la méthode à pas uniforme converge plus rapidement que la méthode avec pas variable, aussi on a pas une grande différence entre les valeurs de la fonction objective et tout dépend de l'initialisation[5].

Row	Algorithm	Scheme	Initialization	tf	N	Convergence	Objective_Value	Time_cpu
	String15	String15	String31	Int64	Int64	String3	Float64	Float64
1	bvpsole	EulerTrapeze	[0.005, 2.0, 0.0, -1.0, -1.0]	10	200	yes	0.000221724	4.29107
2	bvpsole	EulerTrapeze	[0.005, 2.0, 0.0, -1.0, -1.0]	20	200	yes	0.000239161	10.0504
3	bvpsole	EulerTrapeze	[0.005, 2.0, 0.0, -1.0, -1.0]	40	200	yes	0.000262578	15.8674
4	bvpsole	EulerTrapeze	[0.005, 2.0, 0.0, -1.0, -1.0]	50	200	yes	0.000260702	12.0264
5	bvpsole	EulerTrapeze	[0.005, 2.0, 0.0, -1.0, -1.0]	60	200	yes	0.000273672	14.0513
6	bvpsole	EulerTrapeze	[0.005, 2.0, 0.0, -1.0, -1.0]	70	200	yes	0.000288138	25.1344
7	bvpsole	EulerTrapeze	[0.005, 2.0, 0.0, -1.0, -1.0]	80	200	yes	0.000306197	38.0284
8	bvpsole	EulerTrapeze	[0.005, 2.0, 0.0, -1.0, -1.0]	90	200	yes	0.0003674	18.629
9	bvpsole	EulerTrapeze	[0.005, 2.0, 0.0, -1.0, -1.0]	100	200	yes	0.000402549	26.4097
10	nlpsole_V	Trapeze	[0.005, 2.0, 0.0, 1.5, 1.0]	10	200	yes	1.74437	0.894776
11	nlpsole_V	Trapeze	[0.005, 2.0, 0.0, 1.5, 1.0]	20	200	yes	1.62228	2.74183
12	nlpsole_V	Trapeze	[0.005, 2.0, 0.0, 1.5, 1.0]	30	200	yes	1.58153	2.29704
13	nlpsole_V	Trapeze	[0.005, 2.0, 0.0, 1.5, 1.0]	40	200	yes	1.56115	3.89168
14	nlpsole_V	Trapeze	[0.005, 2.0, 0.0, 1.5, 1.0]	50	200	yes	1.54893	4.44757
15	nlpsole_V	Trapeze	[0.005, 2.0, 0.0, 1.5, 1.0]	60	200	yes	1.54078	3.98757
16	nlpsole_V	Trapeze	[0.005, 2.0, 0.0, 1.5, 1.0]	70	200	yes	1.53495	8.29447
17	nlpsole_V	Trapeze	[0.005, 2.0, 0.0, 1.5, 1.0]	80	200	yes	1.53059	2.91827
18	nlpsole_V	Trapeze	[0.005, 2.0, 0.0, 1.5, 1.0]	90	200	yes	1.52719	4.58167
19	nlpsole_V	Trapeze	[0.005, 2.0, 0.0, 1.5, 1.0]	100	200	yes	1.52447	3.44269
20	nlpsole_U	Trapeze	[2.0, 0.0, 1.5, 1.0]	10	200	yes	1.74492	0.0707392
21	nlpsole_U	Trapeze	[2.0, 0.0, 1.5, 1.0]	20	200	yes	1.62338	0.077668
22	nlpsole_U	Trapeze	[2.0, 0.0, 1.5, 1.0]	30	200	yes	1.58307	0.0647028
23	nlpsole_U	Trapeze	[2.0, 0.0, 1.5, 1.0]	40	200	yes	1.56307	0.0808286
24	nlpsole_U	Trapeze	[2.0, 0.0, 1.5, 1.0]	50	200	yes	1.55118	0.0776901
25	nlpsole_U	Trapeze	[2.0, 0.0, 1.5, 1.0]	60	200	yes	1.54333	0.0839635
26	nlpsole_U	Trapeze	[2.0, 0.0, 1.5, 1.0]	70	200	yes	1.53777	0.0706079
27	nlpsole_U	Trapeze	[2.0, 0.0, 1.5, 1.0]	80	200	yes	1.53365	0.0704574
28	nlpsole_U	Trapeze	[2.0, 0.0, 1.5, 1.0]	90	200	yes	1.53049	0.0774732
29	nlpsole_U	Trapeze	[2.0, 0.0, 1.5, 1.0]	100	200	yes	1.52798	0.0954199

TABLE 4 – Les résultats obtenus par la méthode de collocation et la méthode directe pour les deux types de pas avec N=200.

Row	Algorithm	Scheme	Initialization	tf	N	Convergence	Objective_Value	Time_cpu
	String15	String15	String31	Int64	Int64	String3	Float64	Float64
1	bvpsole	EulerTrapeze	[0.0025, 2.0, 0.0, -1.0, -1.0]	10	400	yes	5.81597e-5	6.09093
2	bvpsole	EulerTrapeze	[0.0025, 2.0, 0.0, -1.0, -1.0]	20	400	yes	6.31969e-5	42.1168
3	bvpsole	EulerTrapeze	[0.0025, 2.0, 0.0, -1.0, -1.0]	40	400	yes	6.76809e-5	55.8793
4	bvpsole	EulerTrapeze	[0.0025, 2.0, 0.0, -1.0, -1.0]	50	400	yes	6.88693e-5	93.2476
5	bvpsole	EulerTrapeze	[0.0025, 2.0, 0.0, -1.0, -1.0]	60	400	yes	6.97974e-5	128.74
6	bvpsole	EulerTrapeze	[0.0025, 2.0, 0.0, -1.0, -1.0]	70	400	yes	7.05655e-5	156.85
7	bvpsole	EulerTrapeze	[0.0025, 2.0, 0.0, -1.0, -1.0]	80	400	yes	7.12398e-5	88.5189
8	bvpsole	EulerTrapeze	[0.0025, 2.0, 0.0, -1.0, -1.0]	90	400	yes	7.47913e-5	114.651
9	bvpsole	EulerTrapeze	[0.0025, 2.0, 0.0, -1.0, -1.0]	100	400	yes	8.09031e-5	64.1469
10	nlpsole_V	Trapeze	[0.0025, 2.0, 0.0, 1.5, 1.0]	10	400	yes	1.7443	6.07886
11	nlpsole_V	Trapeze	[0.0025, 2.0, 0.0, 1.5, 1.0]	20	400	yes	1.62223	10.0559
12	nlpsole_V	Trapeze	[0.0025, 2.0, 0.0, 1.5, 1.0]	30	400	yes	1.58149	9.71333
13	nlpsole_V	Trapeze	[0.0025, 2.0, 0.0, 1.5, 1.0]	40	400	yes	1.56112	9.8178
14	nlpsole_V	Trapeze	[0.0025, 2.0, 0.0, 1.5, 1.0]	50	400	yes	1.5489	12.6986
15	nlpsole_V	Trapeze	[0.0025, 2.0, 0.0, 1.5, 1.0]	60	400	yes	1.54075	6.96018
16	nlpsole_V	Trapeze	[0.0025, 2.0, 0.0, 1.5, 1.0]	70	400	yes	1.53493	6.07042
17	nlpsole_V	Trapeze	[0.0025, 2.0, 0.0, 1.5, 1.0]	80	400	yes	1.53056	5.93359
18	nlpsole_V	Trapeze	[0.0025, 2.0, 0.0, 1.5, 1.0]	90	400	yes	1.52717	10.1622
19	nlpsole_V	Trapeze	[0.0025, 2.0, 0.0, 1.5, 1.0]	100	400	yes	1.52445	8.19683
20	nlpsole_U	Trapeze	[2.0, 0.0, 1.5, 1.0]	10	400	yes	1.74445	0.154171
21	nlpsole_U	Trapeze	[2.0, 0.0, 1.5, 1.0]	20	400	yes	1.62254	0.16318
22	nlpsole_U	Trapeze	[2.0, 0.0, 1.5, 1.0]	30	400	yes	1.58193	0.144878
23	nlpsole_U	Trapeze	[2.0, 0.0, 1.5, 1.0]	40	400	yes	1.56169	0.257626
24	nlpsole_U	Trapeze	[2.0, 0.0, 1.5, 1.0]	50	400	yes	1.54958	0.145147
25	nlpsole_U	Trapeze	[2.0, 0.0, 1.5, 1.0]	60	400	yes	1.54154	0.141941
26	nlpsole_U	Trapeze	[2.0, 0.0, 1.5, 1.0]	70	400	yes	1.53581	0.136206
27	nlpsole_U	Trapeze	[2.0, 0.0, 1.5, 1.0]	80	400	yes	1.53154	0.163605
28	nlpsole_U	Trapeze	[2.0, 0.0, 1.5, 1.0]	90	400	yes	1.52823	0.141582
29	nlpsole_U	Trapeze	[2.0, 0.0, 1.5, 1.0]	100	400	yes	1.52559	0.268855

TABLE 5 – Les résultats obtenus par la méthode de collocation et la méthode directe pour les deux types de pas avec N=400.

## 8 Application : Problème d'avion

Nous nous intéressons au contrôle optimal en temps d'un avion pendant sa phase de montée. Cette phase est déterminée par sa propre dynamique régie par le système de dimension quatre.

## 8.1 Valeur des paramètres

On considère pour les valeurs initiales  $x_0 = (3480, 151.67, 69000)$  et pour les valeurs finales  $xf = (9144, 191, 68100)$

Data	Value	Unit	Data	Value	Unit
S	122.6	$m^2$	$C_{s_1}$	$1.055e^{-5}$	$kg.s^{-1}.N^{-1}$
g	9.81	$m.s^{-2}$	$C_{s_2}$	441.54	$m.s^{-1}$
$C_{T_1}$	141040	N	R	287.058	$J.kg^{-1}.K^{-1}$
$C_{T_2}$	14909.9	m	$\Theta_0$	288.15	K
$C_{T_3}$	$6.997e^{-10}$	$m^{-2}$	$\beta$	0.0065	$K.m^{-1}$
$C_{D_1}$	0.0242		$P_0$	101325	Pa
$C_{D_2}$	0.0469				

## 8.2 Définition du problème

La phase de montée est déterminée par sa propre dynamique régie par le système quadridimensionnel :

$$\begin{aligned} \frac{dh}{dt} &= v \sin(\gamma) \\ \frac{dv}{dt} &= \frac{T(h)}{m} - \frac{\phi(h)Sv^2}{2m}(C_{D1} + C_{D2}u^2) - g \sin(\gamma) \\ \frac{dm}{dt} &= -C_s(v)T(h) \\ \epsilon \frac{d\gamma}{dt} &= \frac{\phi(h)Sv}{2m}u - \frac{g}{v} \cos(\gamma) \end{aligned}$$

Avec :

$$\begin{aligned} T(h) &= C_{T1}\left(1 - \frac{h}{C_{T2}} + h^2 C_{T3}\right) \\ C_s(v) &= C_{s1}\left(1 + \frac{v}{C_{s2}}\right) \\ \phi(h) &= \frac{P_0}{R\theta_0^{\frac{q}{\beta R}}}(\theta_0 - \beta h)^{\frac{q}{\beta R}-1} \end{aligned}$$

Le but est de résoudre ce problème en tenant en compte les différentes valeurs de  $\epsilon$ .

## 8.3 Résolution du problème

Les figures suivantes représentent les solutions obtenues. On a les tracés des variables primaires ainsi que les variables duales.

### 8.3.1 Méthode directe avec un pas constant

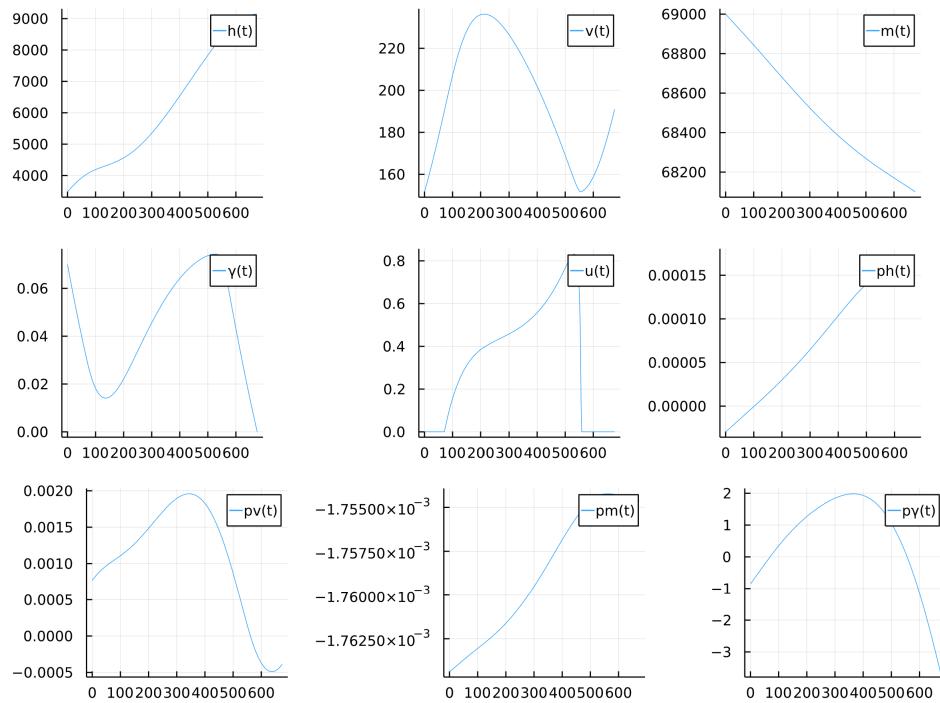


FIGURE 9 – Les variables primaires, duales et le contrôle pour  $N=300$  et  $\varepsilon = 100$  pour le pas uniforme.

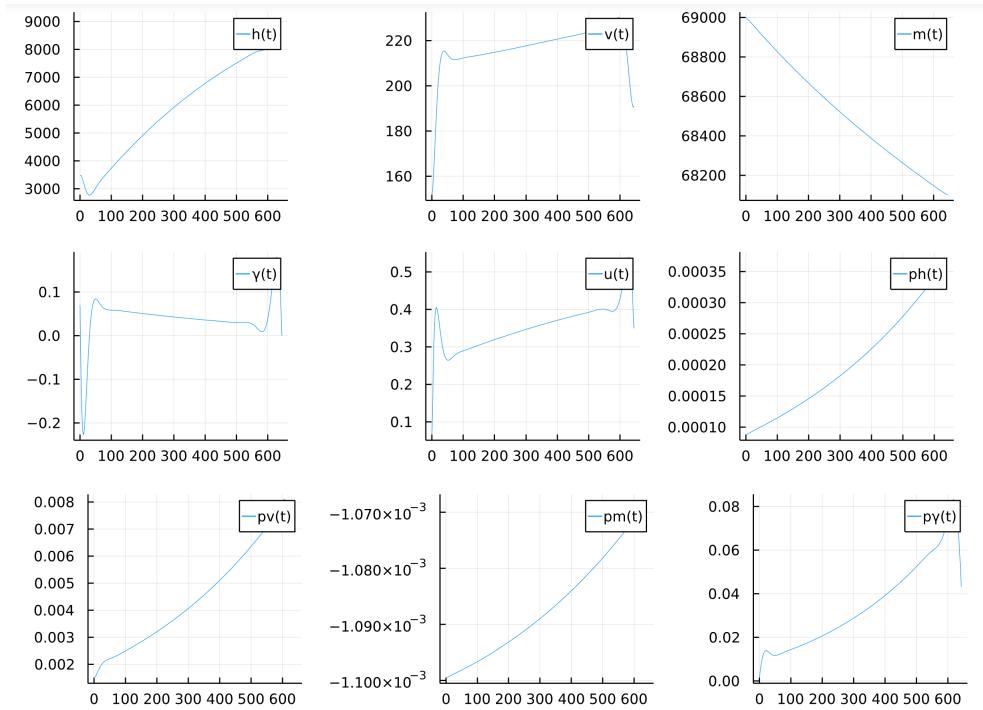


FIGURE 10 – Les variables primaires, duales et le contrôle pour  $N=300$  et  $\varepsilon = 1$  pour le pas uniforme.

D'après les figures précédentes, le temps final vaut environ 657 secondes.

### 8.3.2 Méthode directe avec un pas variable

Dans ce cas,  $h_i$  n'est pas constant. Il est considéré comme une variable. Ainsi, la fonction objective est de minimiser  $\Delta t + \alpha \sum_{i=1}^N ((h_i - \Delta t)^2)$ .  $c[N+1]$  est la fonction de coût et  $\alpha = 10^{-3}$  et le facteur de normalisation.

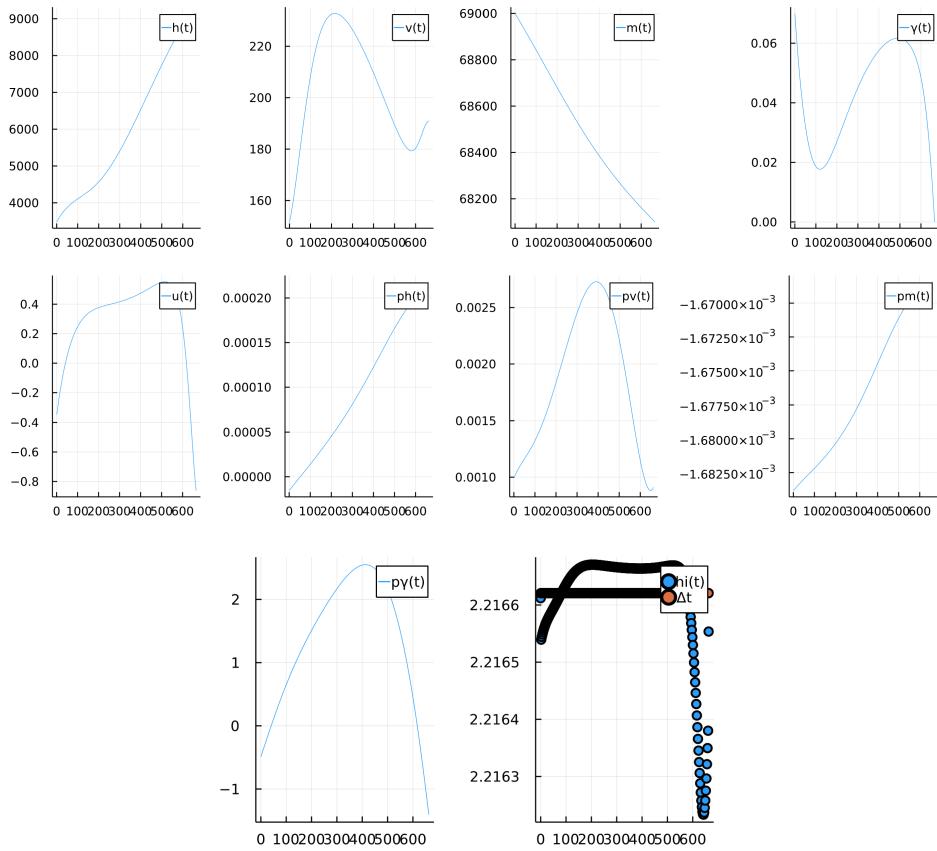


FIGURE 11 – Les variables primaires, duales et le contrôle pour  $N=300$  et  $\varepsilon = 100$  pour le pas variable.

Pour la résolution avec un pas variable, on n'observe pas une grande différence entre les allures des graphes. Pour les valeurs du pas, au début et à la fin on a des pas plus petit par rapport au pas uniforme mais avec un ordre de grandeur très petit.

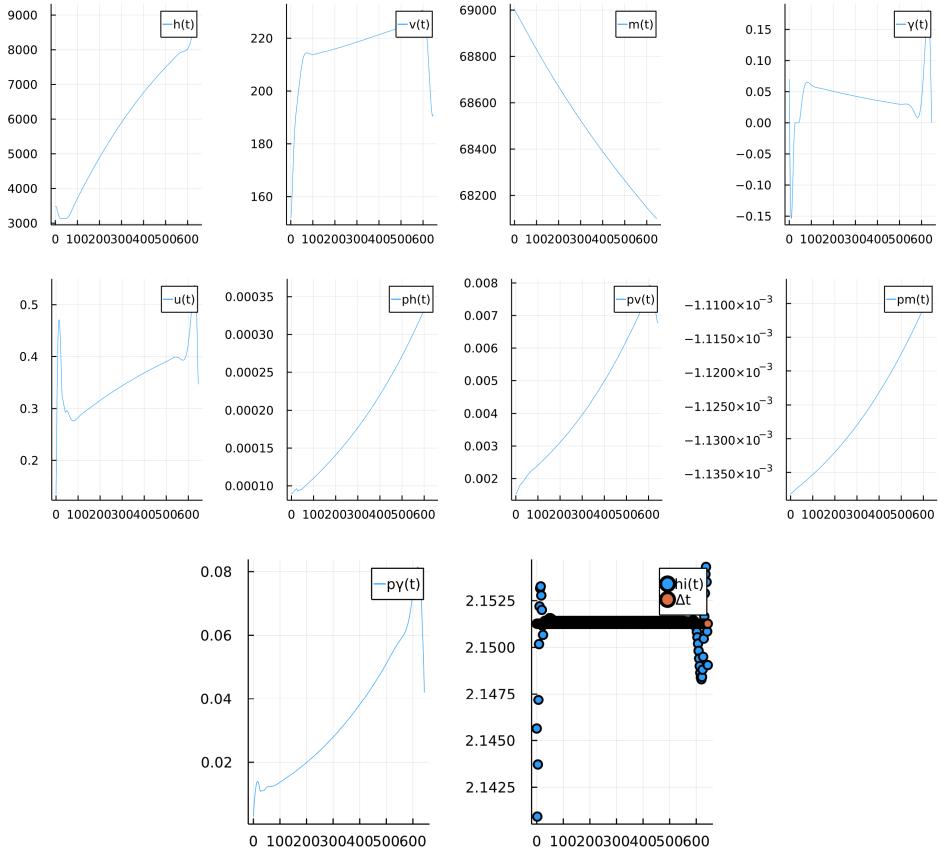


FIGURE 12 – Les variables primaires, duales et le contrôle pour  $N=300$  et  $\varepsilon = 1$  pour le pas variable.

### 8.3.3 Sauvegarde de la solution dans un fichier CSV

Pour les mêmes raisons que le cas du problème de Turnpike, on a décidé de sauvegardé les solutions dans un fichier CSV en utilisant les fonctions écrites précédemment.

Les tables suivant montrent les résultats pour des valeurs particulières de  $\varepsilon$  et pour des N ,nombres de points de discréétisation, différents.

Row	Algorithm	Scheme	Initialization	eps	N	Convergence	Objective_Value	Time_cpu
	String15	String7	String	Int64	Int64	String3	Float64	Float64
1	nlpsove_V	Trapeze	[2,16666666666666665, 6312.0, 171...	1	300	yes	2.15127	67.1524
2	nlpsove_V	Trapeze	[2,16666666666666665, 6312.0, 171...	10	300	yes	2.15785	27.0193
3	nlpsove_V	Trapeze	[2,16666666666666665, 6312.0, 171...	20	300	yes	2.15844	13.8679
4	nlpsove_V	Trapeze	[2,16666666666666665, 6312.0, 171...	40	300	yes	2.16216	8.83088
5	nlpsove_V	Trapeze	[2,16666666666666665, 6312.0, 171...	50	300	yes	2.16715	9.84058
6	nlpsove_V	Trapeze	[2,16666666666666665, 6312.0, 171...	60	300	yes	2.1747	11.9726
7	nlpsove_V	Trapeze	[2,16666666666666665, 6312.0, 171...	70	300	yes	2.18423	9.5715
8	nlpsove_V	Trapeze	[2,16666666666666665, 6312.0, 171...	80	300	yes	2.19488	23.4901
9	nlpsove_V	Trapeze	[2,16666666666666665, 6312.0, 171...	90	300	yes	2.20587	13.5742
10	nlpsove_V	Trapeze	[2,16666666666666665, 6312.0, 171...	100	300	yes	2.21662	12.5585
11	nlpsove_U	Trapeze	[2,16666666666666665, 6312.0, 171...	1	300	yes	2.15015	3.83873
12	nlpsove_U	Trapeze	[2,16666666666666665, 6312.0, 171...	10	300	yes	2.15785	7.02918
13	nlpsove_U	Trapeze	[2,16666666666666665, 6312.0, 171...	20	300	yes	2.15844	6.07573
14	nlpsove_U	Trapeze	[2,16666666666666665, 6312.0, 171...	30	300	yes	2.15959	7.47628
15	nlpsove_U	Trapeze	[2,16666666666666665, 6312.0, 171...	40	300	yes	2.16233	22.2808
16	nlpsove_U	Trapeze	[2,16666666666666665, 6312.0, 171...	50	300	yes	2.16821	81.0114
17	nlpsove_U	Trapeze	[2,16666666666666665, 6312.0, 171...	60	300	yes	2.1782	13.3793
18	nlpsove_U	Trapeze	[2,16666666666666665, 6312.0, 171...	70	300	yes	2.19247	14.3608
19	nlpsove_U	Trapeze	[2,16666666666666665, 6312.0, 171...	80	300	yes	2.21848	22.696
20	nlpsove_U	Trapeze	[2,16666666666666665, 6312.0, 171...	90	300	yes	2.23123	27.9794
21	nlpsove_U	Trapeze	[2,16666666666666665, 6312.0, 171...	100	300	yes	2.2535	21.4486

TABLE 6 – Les résultats obtenues par la méthode directe pour les deux types de pas avec N=300.

Row	Algorithm	Scheme	Initialization	eps	N	Convergence	Objective_Value	Time_cpu
	String15	String7	String	Int64	Int64	String3	Float64	Float64
1	nlpsove_V	Trapeze	[1.3, 6312.0, 171.33499999999998...	1	500	yes	1.29073	29.1903
2	nlpsove_V	Trapeze	[1.3, 6312.0, 171.33499999999998...	10	500	yes	1.29469	230.498
3	nlpsove_V	Trapeze	[1.3, 6312.0, 171.33499999999998...	20	500	yes	1.29505	15.1388
4	nlpsove_V	Trapeze	[1.3, 6312.0, 171.33499999999998...	40	500	yes	1.29728	25.4623
5	nlpsove_V	Trapeze	[1.3, 6312.0, 171.33499999999998...	50	500	yes	1.30028	14.091
6	nlpsove_V	Trapeze	[1.3, 6312.0, 171.33499999999998...	60	500	yes	1.30481	14.8256
7	nlpsove_V	Trapeze	[1.3, 6312.0, 171.33499999999998...	70	500	yes	1.31053	199.298
8	nlpsove_V	Trapeze	[1.3, 6312.0, 171.33499999999998...	80	500	yes	1.31692	17.9074
9	nlpsove_V	Trapeze	[1.3, 6312.0, 171.33499999999998...	90	500	yes	1.32351	19.8257
10	nlpsove_V	Trapeze	[1.3, 6312.0, 171.33499999999998...	100	500	yes	1.32996	12.2407
11	nlpsove_U	Trapeze	[1.3, 6312.0, 171.33499999999998...	1	500	yes	1.29065	8.46036
12	nlpsove_U	Trapeze	[1.3, 6312.0, 171.33499999999998...	10	500	yes	1.29469	12.2351
13	nlpsove_U	Trapeze	[1.3, 6312.0, 171.33499999999998...	20	500	yes	1.29505	5.98454
14	nlpsove_U	Trapeze	[1.3, 6312.0, 171.33499999999998...	30	500	yes	1.29574	7.33282
15	nlpsove_U	Trapeze	[1.3, 6312.0, 171.33499999999998...	40	500	yes	1.29739	5.39384
16	nlpsove_U	Trapeze	[1.3, 6312.0, 171.33499999999998...	50	500	yes	1.30092	11.8116
17	nlpsove_U	Trapeze	[1.3, 6312.0, 171.33499999999998...	60	500	yes	1.30691	20.155
18	nlpsove_U	Trapeze	[1.3, 6312.0, 171.33499999999998...	70	500	yes	1.31547	28.1303
19	nlpsove_U	Trapeze	[1.3, 6312.0, 171.33499999999998...	80	500	yes	1.32628	42.9449
20	nlpsove_U	Trapeze	[1.3, 6312.0, 171.33499999999998...	90	500	yes	1.33873	35.6842
21	nlpsove_U	Trapeze	[1.3, 6312.0, 171.33499999999998...	100	500	yes	1.3521	43.8109

TABLE 7 – Les résultats obtenues par la méthode directe pour les deux types de pas avec N=500.

Pour toutes les valeurs de  $\varepsilon$  on a les deux algorithmes convergent. On remarque que plus N est grande la valeur de la fonction objective est plus optimale (minimale). Par contre, on ne peut rien conclure en ce qui concerne le temps de convergence. Il varie d'une manière aléatoire. En ce qui concerne l'initialisation, on voit bien qu'elle est un facteur important vu qu'elle dépend aussi de N.

## 9 Conclusion

Dans cette article, le but était de résoudre des problèmes de contrôle optimal singulièrement perturbés.

Le problème de Turnpike consistait en l'optimisation de la solution vu qu'en début et la fin de la trajectoire on a un arc de temps très court. La méthode directe était appliquée avec deux types de pas : pas uniforme et pas variable. Il a été constaté que lors de la phase statique la solution reste très proche à la solution statique d'où l'utilité de ne pas garder le même pas lors de toute la trajectoire. Une nouvelle méthode appelée la méthode de collocation était appliquée pour résoudre ce problème. Elle consistait en la résolution de problème de valeur limite associé, c'est à dire le problème en tenant en compte aussi des variables duales comme variables. La fonction objective tient aussi en compte la minimisation de l'écart entre les solutions obtenues par deux schémas de l'approximation de la solution : schéma de Trapèze et le schéma d'Euler. Lors de la résolution du problème, la valeur de différents paramètres était prise en compte. L'initialisation (valeur de départ primaire) était un paramètre très important et significative dans la convergence de la solution. Les résultats numériques montrent que l'initialisation avec la solution statique permet d'obtenir des solutions optimales aussi on peut aller plus loin dans les valeurs de  $t_f$  en initialisant avec la solution statique. Par exemple, dans le cas avec un nombre de points de discrétisation  $N=120$  on a pu aller jusqu'à  $t_f = 90$  avec solution statique comme valeur de départ primaire et pour  $N=200$  l'algorithme peut converger jusqu'à  $t_f=110$ .

L'une des applications de ces méthodes de résolution était le problème d'avion. Le but était de minimiser le temps durant la phase de montée de l'avion. Cette phase est déterminée par sa propre dynamique régie par un système de dimension quatre. On a aussi appliquées la méthode directe avec les deux types de pas et pour des  $\varepsilon$  allant de 1 à 100.

Pour les deux problèmes on a pu conclure l'influence de plusieurs paramètres sur la convergence de la solution surtout pour la valeur de l'initialisation qui pourra être considérée comme le facteur le plus important. Pour le problème de Turnpike on a trouvé des valeurs de  $t_f$  pour lesquelles la solution ne peut pas converger et pour le problème d'avion on a trouvé que pour  $\varepsilon = 1$  la solution est plus optimale.

# Annexes

## Annexe A Turnpike

Les figures suivantes sont obtenues en exécutant le notebook Turnpike.ipynb

### A.1 Méthode directe

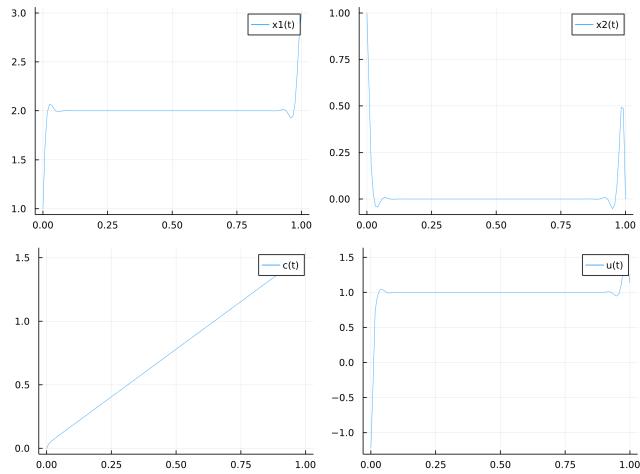


FIGURE 13 – Représentation des variables  $x_1$  et  $x_2$  et de la fonction objective et du contrôle  $u$  pour  $t_f = 100$  et  $N=100$

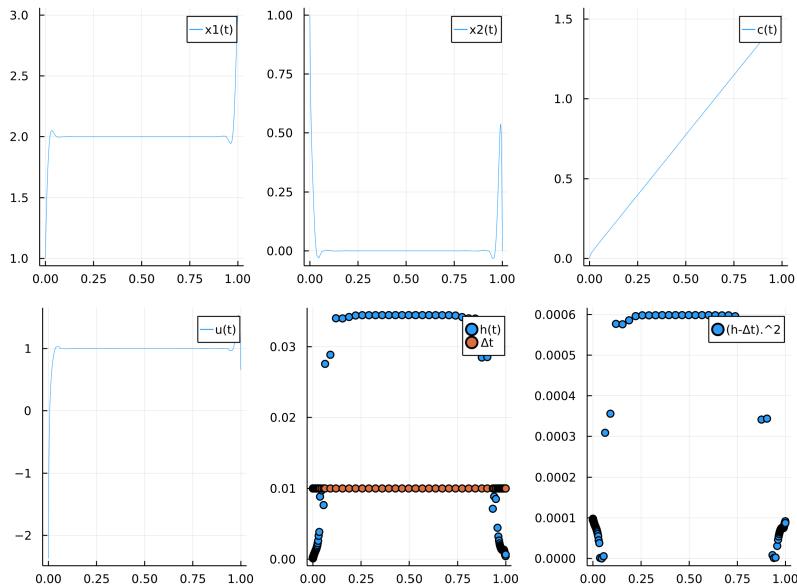


FIGURE 14 – Représentation de la solution, de la fonction objective, du contrôle et des erreurs locaux en fonction des temps  $t_i$  pour  $t_f= 100$  et  $N =100$

## A.2 Méthode de collocation

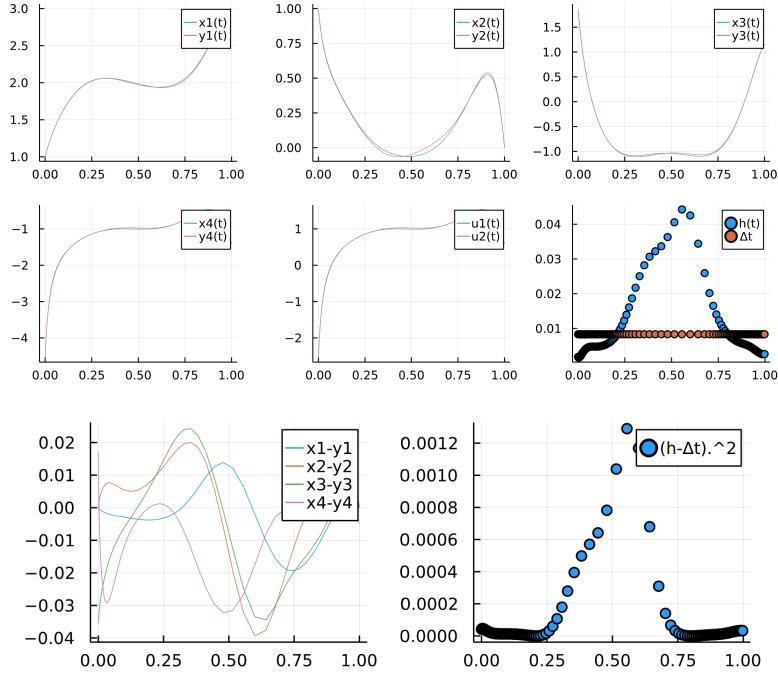


FIGURE 15 – Comparaison des solutions obtenues par le schéma d’Euler et le schéma de Trapèze ainsi que la différence entre le pas variable et le pas uniforme en fonction des temps  $t_i$  pour  $t_f = 10$  et  $N=120$  avec la solution statique comme initialisation.

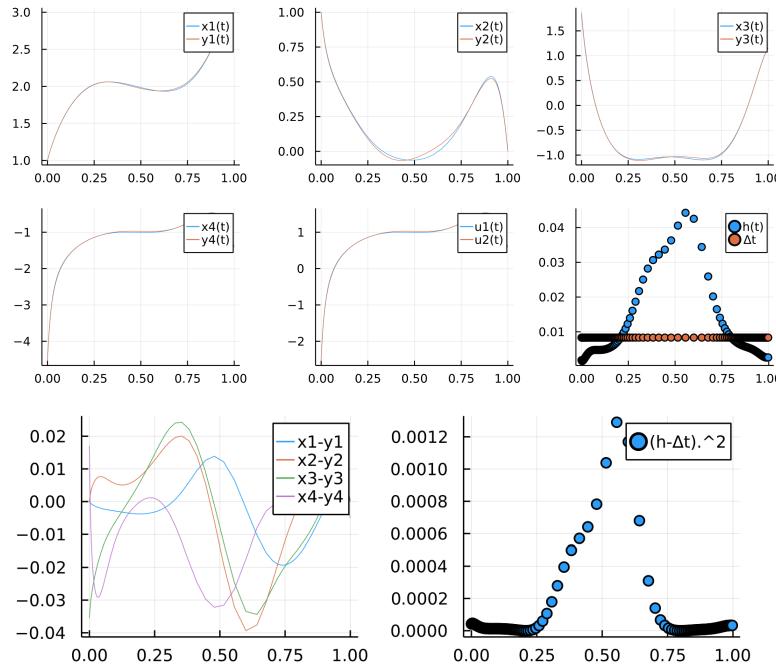


FIGURE 16 – Comparaison des solutions obtenues par le schéma d’Euler et le schéma de Trapèze ainsi que la différence entre le pas variable et le pas uniforme en fonction des temps  $t_i$  pour  $t_f = 10$  et  $N=120$  avec la solution arbitraire comme initialisation.

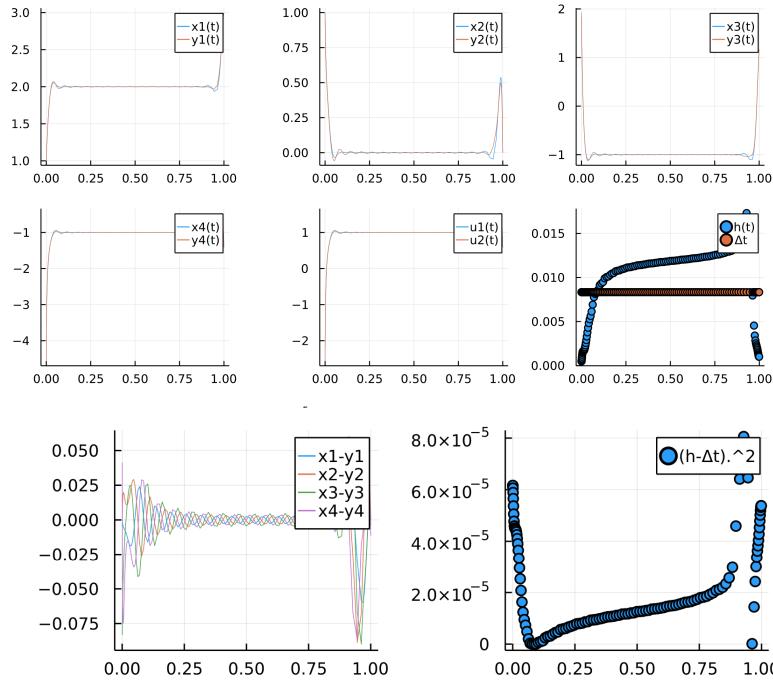


FIGURE 17 – Comparaison des solutions obtenues par le schéma d’Euler et le schéma de Trapèze ainsi que la différence entre le pas variable et le pas uniforme en fonction des temps  $t_i$  pour  $t_f = 80$  et  $N=120$  avec la solution statique comme initialisation.

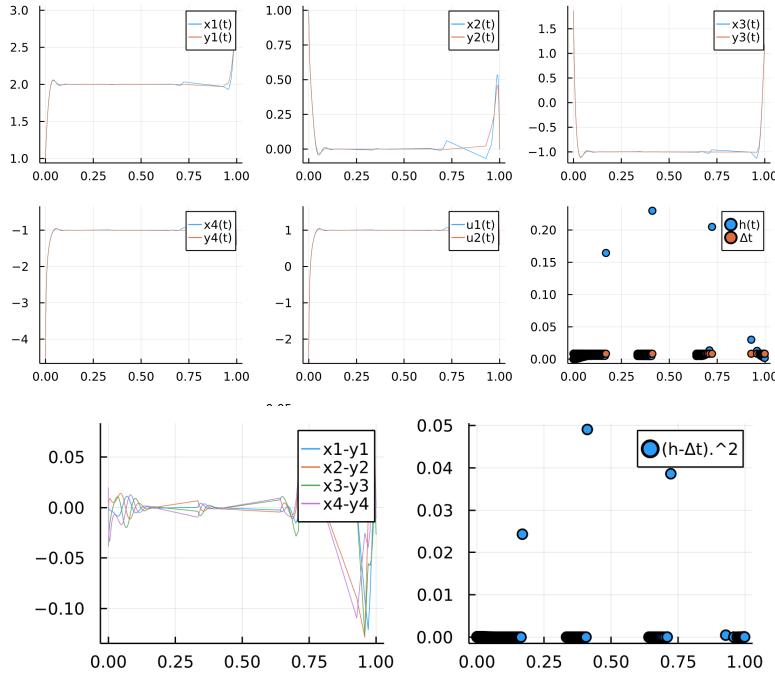


FIGURE 18 – Comparaison des solutions obtenues par le schéma d’Euler et le schéma de Trapèze ainsi que la différence entre le pas variable et le pas uniforme en fonction des temps  $t_i$  pour  $t_f = 80$  et  $N=120$  avec la solution arbitraire comme initialisation.

	tf	N	Objective_Value_SS	Objective_Value_AS
	Int64	Int64	Any	Any
1	10	120	0.000592421	0.000592421
2	20	120	0.000666687	0.000666687
3	30	120	0.000734199	0.000734199
4	40	120	0.000820513	0.117101
5	50	120	0.00083261	0.000881642
6	60	120	0.000996987	0.000996987
7	70	120	0.00129785	0.00129785
8	80	120	0.00162432	0.00171839
9	90	120	0.00218751	X
10	100	120	X	X

TABLE 8 – La valeur de la fonction objective selon  $t_f$  et pour  $N = 120$ . X signifie que l’algorithme ne converge pas.

	<b>tf</b>	<b>N</b>	<b>Objective_Value_SS</b>	<b>Objective_Value_AS</b>
	<b>Int64</b>	<b>Int64</b>	<b>Any</b>	<b>Any</b>
<b>1</b>	10	200	0.000221724	0.000221724
<b>2</b>	20	200	0.000239161	0.000239161
<b>3</b>	30	200	0.000250396	0.000247127
<b>4</b>	40	200	0.000262578	X
<b>5</b>	50	200	0.000260702	0.000276928
<b>6</b>	60	200	0.000273672	0.000289233
<b>7</b>	70	200	0.000288138	0.000314302
<b>8</b>	80	200	0.000306197	0.000313787
<b>9</b>	90	200	0.0003674	0.000333481
<b>10</b>	100	200	0.000402549	0.000402549
<b>11</b>	110	200	X	X

TABLE 9 – La valeur de la fonction objective selon  $t_f$  et pour  $N = 200$ . X signifie que l’algorithme ne converge pas.

## Annexe B Aircraft

Les figures suivantes sont obtenues en exécutant le notebook Aircraft.ipynb

### B.1 Méthode directe

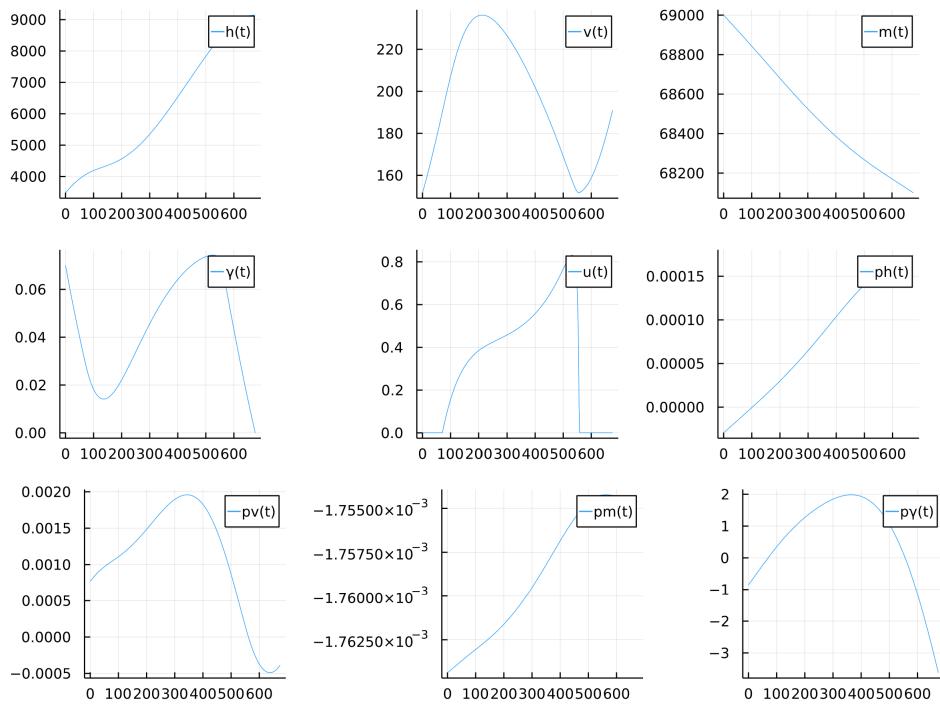


FIGURE 19 – Les variables primaires, duales et le contrôle pour  $N=300$  et  $\varepsilon = 100$  pour le pas uniforme.

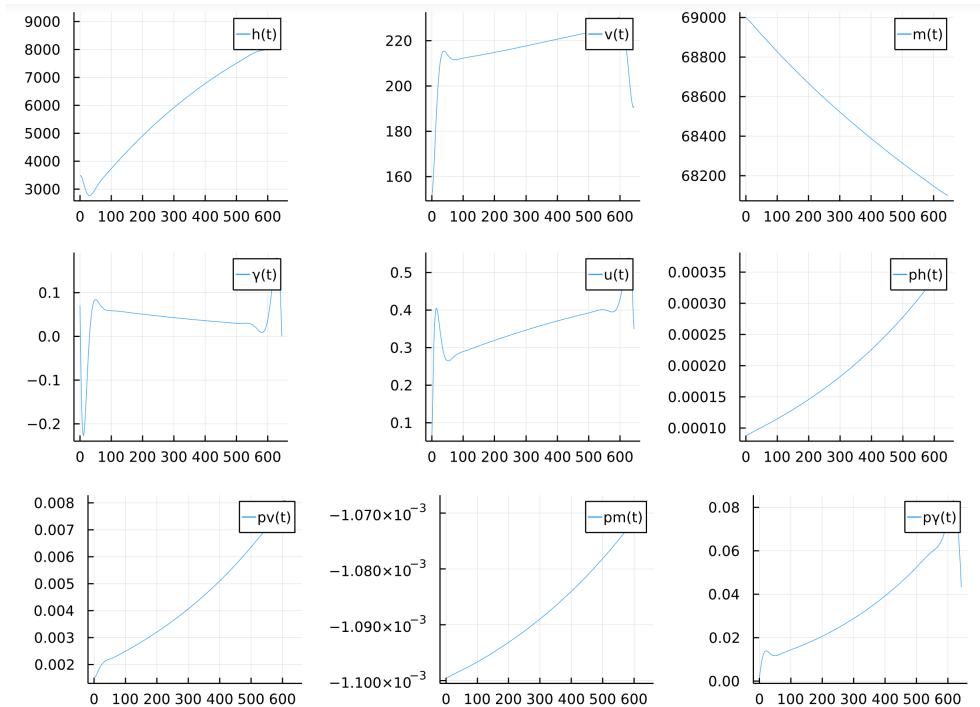


FIGURE 20 – Les variables primaires, duales et le contrôle pour  $N=300$  et  $\varepsilon = 1$  pour le pas uniforme.

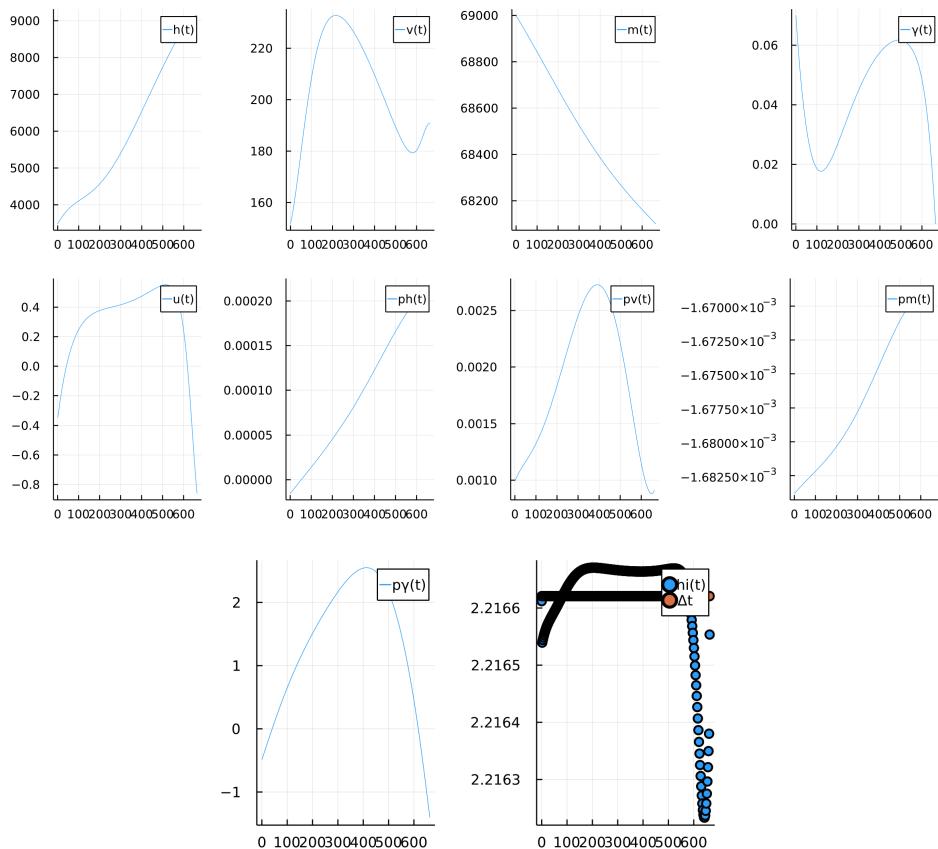


FIGURE 21 – Les variables primaires, duales et le contrôle pour  $N=300$  et  $\varepsilon = 100$  pour le pas variable.

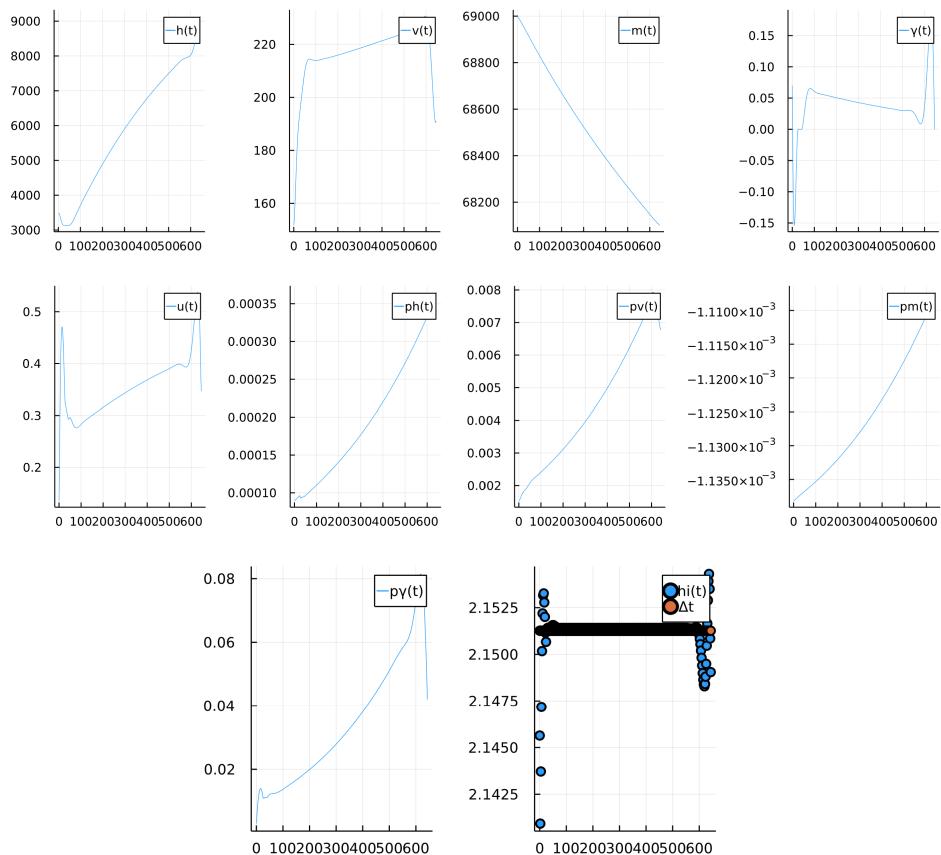


FIGURE 22 – Les variables primaires, duales et le contrôle pour  $N=300$  et  $\varepsilon = 1$  pour le pas variable.

## Références

- [1] Emmanuel Trélat. *The turnpike property in finite-dimensional nonlinear optimal control.*  
URL : <https://hal.archives-ouvertes.fr/hal-00946536/document>
- [2] DAMIEN GOUBINAT. *Contrôle géométrique et méthodes numériques : application au problème de montée d'un avion.*  
URL : <https://oatao.univ-toulouse.fr/18608/7/Goubinat.pdf>
- [3] DataFrames.jl : *Package en Julia.* URL <https://dataframes.juliadata.org/stable/>
- [4] CSV en Julia. URL : <https://csv.juliadata.org/stable/>
- [5] JuMP. URL : <https://jump.dev/JuMP.jl/stable/reference/variables/Start-values>
- [6] Olivier Cots. *Contrôle optimal.* ENSEEIHT, INP Toulouse.  
URL : <https://gitlab.irit.fr/toc/mathn7/controle-optimal/cours-etudiants>