

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE  
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE  
SCIENTIFIQUE  
UNIVERSITÉ MOULOU D MAMMERI DE TIZI-OUZOU



FACULTÉ DE GÉNIE ÉLECTRIQUE ET D'INFORMATIQUE  
DÉPARTEMENT D'AUTOMATIQUE

**Mémoire De Fin D'études**

AUTOMATIQUE ET INFORMATIQUE INDUSTRIELLE

Présenté par

**Mohellebi Nabil**

**Sadoudi Toufik**

Mémoire proposé et dirigé par M. **Ahmed MAIDI**

---

**Application des techniques de commande pour l'amélioration de la  
précision des méthodes d'intégration numérique des équations  
différentielles ordinaires**

---

Mémoire soutenu publiquement le 25/06/2024 devant le jury composé de:

M. Ahmed MAIDI	Encadreur	Professeur	UMMTO
M. TRIKI Ahcène	Président	MCB	UMMTO
M. HAMMOUCHE Sofiane	Examineur	MAA	UMMTO
M. ARKOUB Nouara	Examinatrice	MAB	UMMTO

---

Ce projet de fin d'études est réalisé au sein du Laboratoire L2CSP (UMMTO)

# *Remerciements*

Nous souhaitons exprimer notre gratitude au Tout-Puissant pour la force, la détermination, la patience et la santé qui nous ont permis de surmonter les défis de ce parcours.

Nous adressons nos plus sincères remerciements au Professeur MAIDI Ahmed pour son encadrement, son suivi rigoureux et ses précieux conseils tout au long de la réalisation de ce mémoire. Sa confiance et son soutien ont été déterminants pour la réussite de ce projet.

Nos remerciements vont également aux membres du jury pour le temps consacré à l'examen et à l'évaluation de notre travail. Leur expertise et leurs commentaires constructifs ont été grandement appréciés.

Nous tenons à exprimer notre reconnaissance aux membres du laboratoire L2CSP et du département d'Automatique pour leur accueil chaleureux et leur soutien continu.

Enfin, nous adressons nos remerciements les plus chaleureux à toutes les personnes qui nous ont soutenus, de près ou de loin, tout au long de nos années d'études. Leur encouragement, leur aide et leur sympathie ont été des sources inestimables de motivation. Puissent-ils trouver ici l'expression de notre profonde gratitude et de notre respect sincère.

## *Dédicaces*

Je dédie ce modeste travail à ma famille, dont le soutien indéfectible, les encouragements constants et l'amour inconditionnel m'ont offert la force et la détermination nécessaires pour surmonter chaque défi et poursuivre mes rêves. Leur patience infinie et leur dévouement sans faille m'ont montré la voie de la persévérance et du travail acharné. Que ce travail rende hommage à leur importance primordiale dans ma vie.

À tous ceux qui m'ont offert une parole d'encouragement et un moment de soutien, vous avez été des pierres angulaires dans la construction de ce mémoire. Votre présence a été une source inépuisable de motivation et de réconfort. Je vous dédie ce travail avec une profonde reconnaissance pour tout ce que vous avez fait pour moi.

Merci à vous, du fond du cœur.

Nabil

## *Dédicaces*

Je dédie ce modeste travail à,

Mes parents, surtout ma chère mère ; il est difficile de trouver les mots pour exprimer toute ma gratitude et mon amour envers elle. Son soutien indéfectible, ses encouragements constants et son amour inconditionnel ainsi que sa patience infinie et son dévouement sans faille m'ont donné la force et la détermination nécessaires pour surmonter les défis et poursuivre mes rêves. À travers ses sacrifices, elle m'a montré la voie de la persévérance et du travail acharné. Que ce travail lui rende hommage et témoigne de l'importance qu'elle a dans ma vie.

Je tiens à dédier ce travail également à mon petit frère, ma grand-mère, mon grand-père, mes tantes, mes oncles, mes cousins, mes cousines pour chaque parole d'encouragement et chaque moment de soutien. Ils ont joué un rôle crucial dans l'accomplissement de ce mémoire, et je suis profondément reconnaissant pour tout ce qu'ils ont fait pour moi.

Merci à vous.

Toufik

# Table des matières

<b>Introduction générale</b>	<b>1</b>
<b>1 Modèles mathématiques des systèmes dynamiques</b>	<b>4</b>
1.1 Introduction . . . . .	4
1.2 Notion de système dynamique . . . . .	4
1.3 Notion de modèle mathématique . . . . .	5
1.4 Obtention du modèle mathématique . . . . .	6
1.4.1 Modélisation mathématique . . . . .	6
1.4.2 Modélisation expérimentale . . . . .	7
1.5 Types de modèles mathématiques . . . . .	8
1.5.1 Modèle sous forme d'équations différentielles ordinaires . . . . .	8
1.5.2 Modèle d'état d'un système dynamique . . . . .	9
1.6 Exemple d'application . . . . .	10
1.7 Utilisation d'un modèle mathématique . . . . .	13
1.8 Conclusion . . . . .	15
<b>2 Résolution numérique des équations différentielles ordinaires</b>	<b>16</b>
2.1 Introduction . . . . .	16
2.2 Position du problème . . . . .	16
2.3 Méthodes numériques à pas multiples . . . . .	17
2.4 Méthode d'Euler explicite . . . . .	18
2.5 Méthodes d'Adams-Bashforth . . . . .	20
2.6 Conclusion . . . . .	24
<b>3 Commande linéaire quadratique (LQ)</b>	<b>25</b>
3.1 Introduction . . . . .	25
3.2 Problème de commande linéaire quadratique . . . . .	25
3.3 Programmation dynamique . . . . .	26
3.3.1 Principe d'optimalité de Bellman . . . . .	27
3.3.2 Discrétisation du problème de commande optimale . . . . .	27
3.3.3 Détermination de la commande optimale . . . . .	28

3.4	Exemple d'application . . . . .	32
3.5	Conclusion . . . . .	33
<b>4</b>	<b>Correction des méthodes d'intégration numérique des équations différentielles</b>	<b>34</b>
4.1	Introduction . . . . .	34
4.2	Correction des méthodes d'intégration numérique . . . . .	34
4.3	Exemples d'applications . . . . .	37
4.3.1	Méthode d'Euler . . . . .	37
4.3.2	Méthode d'Adams-Bashforth . . . . .	40
4.4	Conclusion . . . . .	48
	<b>Conclusion générale</b>	<b>50</b>

# Table des figures

1.1	Diagramme d'un système dynamique. . . . .	5
1.2	Système mécanique (deux pendules). . . . .	10
1.3	Évolution des positions angulaires des masses. . . . .	13
2.1	Méthode d' <b>Euler</b> : évolution de l'erreur pour différentes valeurs de $h$ . . . . .	20
2.2	Méthode d' <b>Adams-Bashforth</b> : évolution de l'erreur pour différentes valeurs de $h$ . . . . .	23
3.1	Principe d'optimalité de <b>Bellman</b> . . . . .	27
3.2	Commande optimale de l'exemple d'application. . . . .	33
4.1	Évolution de l'erreur dans le cas de la méthode d' <b>Euler</b> pour $t_f = 5$ . . . . .	39
4.2	Évolution de l'erreur dans le cas de la méthode d' <b>Euler</b> pour $t_f = 10$ . . . . .	39
4.3	Évolution de l'erreur dans le cas de la méthode d' <b>Euler</b> pour $t_f = 20$ . . . . .	40
4.4	Évolution de l'erreur dans le cas de la méthode d' <b>Adams-Bashforth</b> pour $t_f = 5$ . . . . .	47
4.5	Évolution de l'erreur dans le cas de la méthode d' <b>Adams-Bashforth</b> pour $t_f = 10$ . . . . .	47
4.6	Évolution de l'erreur dans le cas de la méthode d' <b>Adams-Bashforth</b> pour $t_f = 20$ . . . . .	47

# Liste des tableaux

2.1	Solution numérique avec la méthode d' <b>Euler</b> . . . . .	20
2.2	Solution numérique avec la méthode d' <b>Adams-Bashforth</b> . . . . .	23
3.1	Commande optimale de l'exemple d'application. . . . .	32

# Liste des symboles

$a$	Accélération
$A$	Matrice d'état
$\bar{A}$	Matrice d'état augmentée
$B$	Matrice de commande
$\bar{B}$	Matrice de commande augmentée
$C$	Matrice d'observation
$\bar{C}$	Matrice d'observation augmentée
$d$	Perturbation
$e$	Erreur d'approximation
$F$	Force
$g$	Accélération due à la gravité
$h$	Pas d'intégration
$I_2$	Matrice identité de taille $2 \times 2$
$J$	Fonction de coût (critère)
$J^*$	Fonction de coût optimale
$K$	Matrice de gain de retour d'état
$k$	Instant échantillonnage
$l$	Longueur
$m$	Masse
$N$	Nombre total de pas de temps
$n$	Ordre du système
$P$	Matrice de pondération
$Q$	Matrice de pondération ( $Q \geq 0$ )
$\Re$	Ensemble des nombres réels
$R$	Matrice de pondération ( $R > 0$ )
$r$	Rayon

$s$	Nombre de pas
$T$	Énergie cinétique
$t$	Temps
$u$	Variable de commande
$u^*$	Commande optimale
$V$	Énergie potentielle
$x$	Vecteur d'état
$x_a$	Solution analytique
$x_n$	Solution numérique
$y$	Variable de sortie du système
$\Delta t$	Pas d'intégration
$\alpha$	Ordre de l'erreur de troncature locale
$\varepsilon$	Erreur de troncature locale
$\eta$	degré de polynôme
$\theta$	Angle
$\lambda$	Coefficient de proportionnalité
$\mu$	Ordre de convergence de la méthode
$\tau$	Couple
$\psi$	Fonction non linéaire
$()^T$	Transposée d'une matrice
$()^{-1}$	Inverse d'une matrice

# Introduction générale

Les méthodes d'intégration numérique sont largement utilisées dans divers domaines tels que l'ingénierie, la physique, la chimie, et même les sciences sociales [2, 5]. Leur importance réside dans leur capacité à transformer des problèmes continus en problèmes discrets, rendant ainsi possible l'utilisation des ordinateurs pour simuler et analyser des systèmes complexes [2, 5]. Par exemple, en automatique, les méthodes numériques sont souvent employées pour simuler le comportement dynamique des systèmes, pour implémenter des techniques de commande basée sur l'optimisation comme la commande optimale, la commande prédictive, la commande à modèle interne, et la commande adaptative à modèle de référence [1].

En effet, les méthodes d'intégration numérique jouent un rôle crucial dans la résolution des équations différentielles ordinaires, c'est-à-dire les modèles, particulièrement lorsque les solutions analytiques ne sont pas disponibles ou sont difficiles à obtenir [2, 5]. Ces méthodes numériques permettent de déterminer des approximations des solutions de manière simple, efficace et fiable en utilisant des formules de récurrence généralement simples à établir. Cependant, le recours à des approximations numériques des solutions engendre ce qu'on appelle une erreur d'approximation, qu'on souhaite avoir faible. Néanmoins, en pratique comme cette erreur d'approximation s'accumule au fil du temps, et peut devenir importante au point de fausser la solution. Par conséquent, le contrôle de cette erreur est essentiel pour garantir une solution avec la précision souhaitée [3, 4].

Les méthodes d'intégration numériques explicites, à un seul pas ou à pas multiples, sont parmi les méthodes les plus populaires, car la solution est obtenue en utilisant des formules récursives simples à implémenter sur une machine [2, 5]. Les méthodes à un seul pas, comme la méthode d'**Euler** sont simples et intuitives, faciles à mettre en œuvre mais peuvent introduire des erreurs d'approximation significatives pour des pas d'intégration importants ; par conséquent elles peuvent conduire à une instabilité numérique [5]. La méthode d'**Adams-Bashforth**, qui est une méthode à pas multiples, est plus complexe mais offre une meilleure précision pour les mêmes conditions [5]. Néanmoins, sur le plan pratique, aucune de ces méthodes n'est exempte d'erreurs, et ces erreurs doivent être contrôlées efficacement pour réduire leur accumulation afin d'avoir des solutions (approximatives).

mations) très fiables [2, 5].

L'utilisation des techniques de commande des systèmes dynamiques pour contrôler l'erreur d'approximation d'une méthode d'intégration numérique constitue une idée très intéressante lorsque la solution analytique est disponible. Le principe consiste à déterminer le modèle dynamique régissant la dynamique de l'erreur d'approximation, et de chercher par la suite un correcteur en utilisant les outils de la théorie de commande des systèmes [2, 3] qui permet de générer une action correctrice afin de minimiser cette erreur d'approximation.

La commande linéaire quadratique (LQ) est une méthode de commande optimale qui réalise, via un retour d'état ou de sortie, un placement de pôles optimal en boucle fermée [2, 3, 6, 9]. Cette technique de commande permet de trouver le meilleur compromis entre la poursuite de trajectoire (précision en sens automatique) et l'effort de commande mis en œuvre, en minimisant un critère de performance qui intègre à la fois l'erreur de poursuite et le coût associé à l'application des commandes [9]. Ainsi, dans le contexte des méthodes d'intégration numérique, l'utilisation de la commande LQ, en se basant sur le modèle de la dynamique de l'erreur d'approximation, est une solution intéressante pour contrôler l'erreur d'approximation, et réduire davantage son accumulation au fil du temps ; ce qui conduit à des solutions fiables dont la précision est avantageusement améliorée[9].

L'objectif de ce présent travail consiste à utiliser la technique de commande LQ pour corriger en ligne les méthodes d'intégration numérique. Pour concevoir la commande LQ, nous allons utiliser la programmation dynamique [6] basée sur le principe d'optimalité de **Bellman** sous sa forme discrète [3, 9], c'est-à-dire en utilisant l'équation fonctionnelle de **Bellman** qui est une équation de récurrence. Ainsi, une démarche de synthèse est présentée et illustrée par deux exemples d'application (méthode d'**Euler** et méthode d'**Adams-Bashforth**).

La présent mémoire est structuré comme suit :

Le **premier** chapitre est consacré à la modélisation des systèmes dynamiques par des équations différentielles ordinaires. Nous y introduisons les concepts fondamentaux et les différentes représentations mathématiques des systèmes, ainsi que les principales techniques de modélisation.

Dans le **deuxième** chapitre, on s'intéresse à la résolution numérique des équations différentielles ordinaires du premier ordre. Nous nous focalisons sur les méthodes explicites, c'est-à-dire sur les méthodes utilisant des formules de récurrences pour déterminer la solution de l'équation différentielle [2, 5]. Nous présentons en détail les méthodes d'**Euler**

et d'**Adams-Bashforth** tout en mettant en évidence leurs avantages et leurs limites.

Le **troisième** chapitre est dédié à la commande LQ des systèmes dynamiques. Nous allons d'abord formuler le problème de commande optimale, puis nous utilisons la programmation dynamique [6], basée sur le principe d'optimalité de **Bellman** sous la forme discrète [3, 9], pour déterminer la commande optimale sous forme d'un retour d'état ou de sortie. Un exemple d'application est présenté à la fin du chapitre.

Dans le **quatrième** chapitre, nous allons utiliser la technique de commande LQ pour minimiser l'erreur d'approximation des méthodes d'intégration numérique afin d'améliorer la précision de la solution. Nous commençons par la formulation du problème de commande en précisant le modèle régissant la dynamique de l'erreur d'approximation et le critère de performances caractérisant la précision de la solution. Deux applications concernant les méthodes d'**Euler** et d'**Adams-Bashforth** sont présentées avec des résultats de simulation démontrant l'apport de la commande LQ dans l'amélioration de la qualité de la solution.

Le mémoire se termine par une conclusion générale sur l'ensemble de l'étude présentée tout en indiquant quelques perspectives futures.

# Chapitre 1

## Modèles mathématiques des systèmes dynamiques

### 1.1 Introduction

La modélisation mathématique est une méthode essentielle pour traduire des phénomènes observables en équations mathématiques. Elle joue un rôle crucial dans de nombreux domaines scientifiques et techniques en permettant de formaliser et de comprendre les comportements complexes des systèmes réels. En effet, la complexité du modèle mathématique nous permet d'approcher au mieux les caractéristiques effectives des phénomènes physiques.

Dans ce chapitre, nous allons présenter des généralités sur la modélisation des systèmes dynamiques par des équations différentielles ordinaires.

### 1.2 Notion de système dynamique

Un système dynamique, de manière simple, peut être défini comme un ensemble d'éléments interconnectés de manière à former une entité destinée à accomplir une certaine tâche ou de réaliser un objectif spécifique. Cette définition englobe une grande variété de structures et de processus, allant des machines complexes aux phénomènes naturels.

Prenons l'exemple d'un moteur électrique, qui est un système composé de divers éléments, tels que le stator, le rotor et d'autres éléments, agencés de manière à convertir l'énergie électrique en mouvement afin de démarrer par exemple un véhicule électrique. Dans ce cas, le stator et le rotor interagissent de manière coordonnée pour produire le mouvement nécessaire au fonctionnement du moteur et, par extension, à la mise en marche du véhicule [7].

Les systèmes peuvent être de nature physique, comme les machines, les dispositifs électroniques, ou encore les systèmes biologiques, mais ils peuvent aussi être abstraits, comme les systèmes d'information ou les systèmes de gestion. Dans tous les cas, la notion de système implique une organisation spécifique des éléments en interaction, visant à atteindre un objectif bien déterminé.

Un système dynamique est caractérisé par un certain nombre de variables (Figure 1.1) qu'on peut classer comme suit [9] :

1. Les états : ce sont les variables internes qui décrivent l'état du système dynamique.
2. Les sorties : ce sont les variables mesurables qui représentent le résultat ou le comportement du système. Généralement, ce sont les variables qu'on doit commander pour leur imposer un comportement désiré. On peut distinguer les variables à mesurer et les variables à commander.
3. Les commandes : ce sont les variables qu'on peut manipuler pour modifier le comportement des sorties du système.

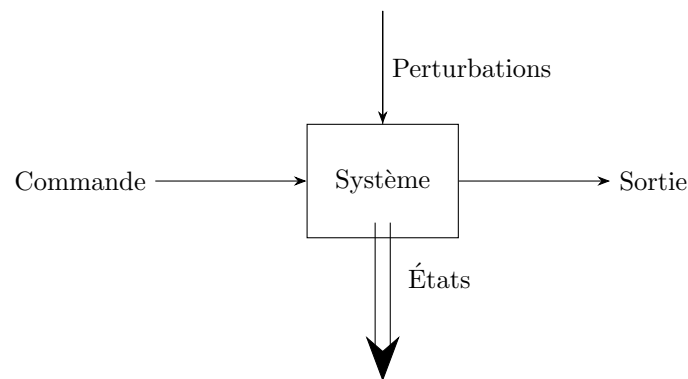


FIGURE 1.1 – Diagramme d'un système dynamique.

### 1.3 Notion de modèle mathématique

Le modèle d'un système dynamique est une structure mathématique qui, soumise aux mêmes signaux d'action que le système réel, reproduit le même comportement d'une expérience [7, 8]. C'est-à-dire que les variables calculées à partir du modèle sont voisines des valeurs mesurées sur le système. Il faut toujours rester prudent sur l'aptitude d'un modèle à reproduire la réalité [7, 8]. En effet, si le modèle a été validé à partir d'une expérience, il se peut que, pour d'autres conditions expérimentales, le modèle ne fournisse plus des résultats voisins des mesures. Il est d'usage de parler d'un modèle établi pour un domaine de fonctionnement donné et de préciser les hypothèses simplificatrices

considérées [1].

Un modèle, quelle que soit la valeur du signal appliqué à l'entrée, donnera un résultat ; par contre pour le système réel, le signal appliqué peut le détruire (par exemple, un moteur électrique peut supporter une certaine tension d'alimentation maximale). La réalité physique d'un phénomène est généralement très complexe, et, bien souvent, on est amené à faire des hypothèses simplificatrices pour traduire cette réalité en équations [1]. L'aptitude du modèle à décrire une expérience sera sa qualité essentielle, mais un modèle exact n'existe pas. Le modèle doit satisfaire deux exigences opposées :

1. Reproduire la réalité le plus précisément possible et,
2. Avoir une structure mathématique la plus simple possible (pour une utilisation aisée) [1].

Ainsi, un modèle mathématique reste un compromis dans un ensemble de contraintes.

## 1.4 Obtention du modèle mathématique

Dans cette section, nous explorerons les différentes méthodes utilisées pour obtenir un modèle mathématique représentatif d'un système dynamique. Nous aborderons la modélisation mathématique [1, 8], la modélisation expérimentale (identification) [1], et présenterons quelques exemples illustratifs de modèles mathématiques. La modélisation mathématique conduit à un modèle appelé de connaissance, et la modélisation expérimentale conduit à un modèle de comportement [1].

### 1.4.1 Modélisation mathématique

Un modèle mathématique est représenté par des équations dont les termes sont des collections ou des arrangements de symboles combinés avec des opérations mathématiques telles que l'addition, la soustraction, la multiplication, etc. Les mathématiques permettent de rendre les modèles aussi abstraits et rigides que possible en utilisant des symboles uniques et une logique qui définit les relations entre les symboles et donc entre les composants réels du système représentés par ces symboles. C'est un avantage majeur dans la modélisation, car les modèles mathématiques peuvent être manipulés, simplifiés, résolus ou transformés à l'aide de théorèmes et d'autres résultats mathématiques établis.

Le modèle mathématique est obtenu en considérant les points suivants [1] :

1. les limites du système physique,
2. les variables caractéristiques,

3. les phénomènes physiques à prendre en compte,
4. les hypothèses simplificatrices.

En général, le modèle est obtenu en écrivant les lois de la physique [7] (lois de **Kirchhoff**, lois de **Newton**, formalise de **Lagrange**, lois de **Bernoulli**, bilan de masse, bilan énergétique, ...). Lorsque le modèle mathématique est élaboré à partir des lois de la physique et ses paramètres sont connus, le modèle obtenu est appelé modèle de connaissance ou boîte blanche [1].

### 1.4.2 Modélisation expérimentale

Lors de la construction d'un modèle mathématique, il est souvent nécessaire d'introduire certains coefficients appelés paramètres du modèle ou du système. Ces paramètres peuvent avoir une signification physique, comme dans les modèles de connaissance, ou n'en avoir aucune, comme c'est le cas pour les modèles de comportement. Dans les deux cas, leur valeur numérique doit souvent être déterminée à partir de données expérimentales.

Cependant, la détermination des paramètres a des objectifs différents qu'il s'agisse d'un modèle de connaissance ou d'un modèle boîte noire. Les termes utilisés pour décrire cette étape diffèrent également : on parle plutôt d'estimation des paramètres dans le premier cas et d'identification dans le second. En effet, l'objectif de l'identification est de calculer les paramètres d'un modèle de processus à partir de données expérimentales de manière à ce que le comportement du processus et celui du modèle soient identiques, et ce pour toutes les séquences de variables d'entrée habituellement utilisées [1].

Le modèle expérimentale est obtenu en suivant les étapes suivantes [1] :

1. Mise au point du protocole expérimental,
2. Réalisation de la campagne de mesure,
3. Prétraitement des données (filtrage pour éliminer le bruit),
4. Choix de la structure du modèle (linéaire ou non linéaire),
5. Calcul des paramètres du modèle (identification ou estimation),
6. Validation des résultats.

Le modèle obtenu par modélisation expérimentale représente le comportement du système en se basant sur des mesures d'entrées et de sorties collectées. Il est souvent linéaire et sa validité est limitée à de petites variations autour du point de fonctionnement considéré. Ce modèle est désigné sous le vocable modèle de comportement ou boîte noire [1].

Dans ce mémoire, on s'intéresse aux modèles obtenus par la modélisation mathématique, c'est-à-dire les modèles de connaissance (boîte blanche).

**Remarque I.1.** Il existe un autre type de modélisation qui combine les modélisations mathématique et expérimentale. Dans ce cas, la structure du modèle est obtenue par modélisation mathématique (écriture des lois de la physique), et les paramètres sont obtenus par modélisation expérimentale (en utilisant les mesures des variables). Ce type de modèle s'appelle boîte grise [1].

## 1.5 Types de modèles mathématiques

Dans le domaine de la modélisation des systèmes dynamiques, il existe une grande variété de types de modèles, chacun ayant ses propres caractéristiques, ses avantages, ses inconvénients, et les domaines d'application. Comprendre ces différents types de modèles est essentiel pour choisir l'approche la plus appropriée en fonction des besoins spécifiques de modélisation. Dans ce qui suit, on s'intéresse aux représentations mathématiques dans le domaine temporel d'un système monovisible (une seule entrée et une seule sortie).

### 1.5.1 Modèle sous forme d'équations différentielles ordinaires

En général, la modélisation mathématique conduit à un ensemble d'équations différentielles ordinaires. Une équation différentielle ordinaire établit la relation entre la variable de sortie du système  $y$ , de ses dérivées  $\dot{y}, \ddot{y}, \dots, y^{(n)}$ , de la variable de commande  $u$ , et de ses dérivées  $\dot{u}, \ddot{u}, \dots, u^{(m)}$ . Ainsi, le modèle mathématique est décrit par l'équation différentielle suivante :

$$F(t, y(t), \dot{y}(t), \ddot{y}(t), \dots, y^{(n)}(t), u(t), \dot{u}(t), \ddot{u}(t), \dots, u^{(m)}(t)) = 0 \quad (1.1)$$

Les variables dépendantes  $u$  et  $y$  dépendent de la variable indépendante du temps notée  $t$ . Selon la nature de la fonction  $F$ , le modèle peut être linéaire ou non linéaire. Le nombre  $n$  représente l'ordre du système. Pour compléter le modèle, des conditions initiales doivent être spécifiées, généralement, elles sont données par les valeurs de la sortie et de ses dérivées à l'instant initial  $t_0$ , c'est-à-dire

$$y(t_0) = y_0, \dot{y}(t_0) = y_1, \dots, y^{(n)}(t_0) = y_n \quad (1.2)$$

Souvent on prend  $t_0 = 0$ , et  $y_0, y_1, \dots, y_n$  sont des constantes connues [4].

### 1.5.2 Modèle d'état d'un système dynamique

Les modèles d'état sont utilisés dans divers domaines pour représenter le comportement dynamique des systèmes, en particulier en automatique. Le modèle d'état d'un système dynamique est donné sous forme d'un ensemble d'équations d'état du premier ordre. Il est obtenu directement à partir des équations différentielles ordinaires du système en introduisant de nouvelles variables, appelées variables d'états, qui représentent des variables internes du système. Ces dernières sont regroupées dans un vecteur  $x$  désigné sous le vocable vecteur d'état. Ce vecteur est de dimension  $n$ .

Un modèle d'état comprend deux équations [3, 8, 9] :

1. Équation d'état : Il représente la relation entre le vecteur d'état  $x$  et de la commande  $u$ . Cette relation est donné par un ensemble d'équations différentielles ordinaires d'ordre un comme suit :

$$\begin{aligned} \dot{x}_1(t) &= f_1(x_1(t), x_2(t), \dots, x_n(t), u(t)), & x_1(t_0) &= x_{10} \\ \dot{x}_2(t) &= f_2(x_1(t), x_2(t), \dots, x_n(t), u(t)), & x_2(t_0) &= x_{20} \\ & \dots & & \\ \dot{x}_n(t) &= f_n(x_1(t), x_2(t), \dots, x_n(t), u(t)), & x_n(t_0) &= x_{n0} \end{aligned} \quad (1.3)$$

qu'on peut écrire sous la forme générale suivante

$$\dot{x}(t) = f(x(t), u(t)) \quad (1.4)$$

$$x(t_0) = x_0 \quad (1.5)$$

où  $t_0$  représente l'instant initial,  $x_0$  et  $x$  sont respectivement les vecteurs de conditions initiales et d'état donnés comme suit :

$$x_0 = \begin{bmatrix} x_{10} \\ x_{20} \\ \vdots \\ x_{n0} \end{bmatrix}, \quad x(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{bmatrix}$$

2. Équation de sortie : Il représente la relation entre le vecteur d'état  $x$  et de la sortie  $u$ . Cette relation est donné par une équation algébrique de la forme (dans le cas d'une seule sortie)

$$y(t) = h(x_1(t), x_2(t), \dots, x_n(t), u(t)) \quad (1.6)$$

où encore

$$y(t) = h(x(t), u(t)) \quad (1.7)$$

En résumé, un modèle d'état est de la forme

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t)), & x(t_0) &= x_0 \\ y(t) &= h(x(t), u(t)) \end{aligned} \tag{1.8}$$

## 1.6 Exemple d'application

Dans cette section, nous allons déterminer le modèle d'état du système mécanique de la Figure 1.2 en utilisant le formalisme de Lagrange. On désigne les déplacements angulaires des deux masses  $m_1$  et  $m_2$  respectivement par  $\theta_1(t)$  et  $\theta_2(t)$ . Les grandeurs  $\tau_1$  et  $\tau_2$  sont des couples qu'on applique au système pour réaliser le déplacement des masses.

Calculons les énergies cinétique et potentielle du système. Le système mécanique comprend deux masses, alors l'énergie cinétique totale du système est :

$$T = T_{m_1} + T_{m_2} = \frac{1}{2} m_1 l^2 \dot{\theta}_1^2(t) + \frac{1}{2} m_2 l^2 \dot{\theta}_2^2(t) \tag{1.9}$$

Concernant l'énergie potentielle du système, on a un seul ressort, alors :

$$V = \frac{1}{2} k \Delta x^2 = \frac{1}{2} k (X_1 - X_2)^2 \tag{1.10}$$

et comme les déplacements rectilignes des extrémités du ressort sont  $X_1(t) = r \sin \theta_1(t)$  et  $X_2(t) = r \sin \theta_2(t)$ , alors

$$V = \frac{1}{2} k r^2 (\sin \theta_1(t) - \sin \theta_2(t))^2 \tag{1.11}$$

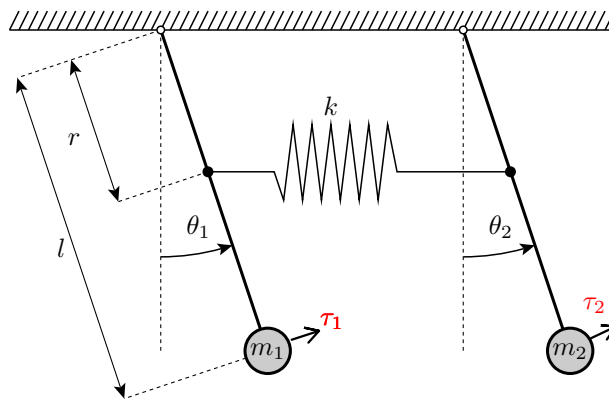


FIGURE 1.2 – Système mécanique (deux pendules).

Chaque masse est soumise aux efforts (couples) appliqués et celui engendré par leurs

poinds. Ainsi, la masse  $m_1$  est soumise au couple appliqué  $\tau_1$  et au couple engendré par son poids  $c_1 = -m_1 g l \sin \theta_1(t)$ . De même, la masse  $m_2$  est soumise aux couples  $\tau_2$  et  $c_2 = -m_2 g l \sin \theta_2(t)$ .

En utilisant le formalisme de **Lagrange**, on obtient pour la masse  $m_1$  :

$$\frac{d}{dt} \left( \frac{\partial T^*}{\partial \dot{\theta}_1} \right) - \frac{\partial T^*}{\partial \theta_1} + \frac{\partial V}{\partial \theta_1} = e_1 \quad (1.12)$$

$$\frac{d}{dt} \left( m_1 l^2 \dot{\theta}_1(t) \right) + k r^3 \left( \sin \theta_1(t) - \sin \theta_2(t) \right)^2 \cos \theta_1(t) = W_1 + \tau_1 \quad (1.13)$$

$$\frac{d}{dt} \left( m_1 l^2 \dot{\theta}_1(t) \right) + k r^3 \left( \sin \theta_1(t) - \sin \theta_2(t) \right)^2 \cos \theta_1(t) - W_1 = \tau_1 \quad (1.14)$$

$$m_1 l^2 \ddot{\theta}_1(t) + k r^3 \left( \sin \theta_1(t) - \sin \theta_2(t) \right)^2 \cos \theta_1(t) - W_1 = \tau_1 \quad (1.15)$$

avec  $W_1 = -m_1 g l \sin \theta_1(t)$ . Pour la masse  $m_2$  :

$$\frac{d}{dt} \left( \frac{\partial T^*}{\partial \dot{\theta}_2} \right) - \frac{\partial T^*}{\partial \theta_2} + \frac{\partial V}{\partial \theta_2} = e_2 \quad (1.16)$$

$$\frac{d}{dt} \left( m_2 l^2 \dot{\theta}_2(t) \right) - k r^3 \left( \sin \theta_1(t) - \sin \theta_2(t) \right)^2 \cos \theta_2(t) = W_2 + \tau_2 \quad (1.17)$$

$$\frac{d}{dt} \left( m_2 l^2 \dot{\theta}_2(t) \right) - k r^3 \left( \sin \theta_1(t) - \sin \theta_2(t) \right)^2 \cos \theta_2(t) - W_2 = \tau_2 \quad (1.18)$$

$$m_2 l^2 \ddot{\theta}_2(t) - k r^3 \left( \sin \theta_1(t) - \sin \theta_2(t) \right)^2 \cos \theta_2(t) - W_2 = \tau_2 \quad (1.19)$$

avec  $W_2 = -m_2 g l \sin \theta_2(t)$ .

En résumé, le modèle du système est donné par les deux équations différentielles ordinaires de deuxième ordre suivantes :

$$m_1 l^2 \ddot{\theta}_1(t) + k r^3 \left( \sin \theta_1(t) - \sin \theta_2(t) \right)^2 \cos \theta_1(t) + m_1 g l \sin \theta_1(t) = \tau_1 \quad (1.20)$$

$$m_2 l^2 \ddot{\theta}_2(t) - k r^3 \left( \sin \theta_1(t) - \sin \theta_2(t) \right)^2 \cos \theta_2(t) + m_2 g l \sin \theta_2(t) = \tau_2 \quad (1.21)$$

Ainsi, en introduisant les variables d'état suivantes :

$$x_1 = \theta_1, \quad x_2 = \dot{\theta}_1, \quad x_3 = \theta_2, \quad x_4 = \dot{\theta}_2$$

et en considérant comme sorties  $y_1 = \theta_1$  et  $y_2 = \theta_2$ , on peut mettre le modèle (1.20) et (1.21) sous la forme du modèle d'état (1.8) comme suit :

$$\begin{aligned}
 \dot{x}_1 &= x_2 \\
 \dot{x}_2 &= -\frac{k r^3}{m_1 l^2} (\sin x_1 - \sin x_3)^2 \cos x_1 - \frac{g}{l} \sin x_1 + \frac{1}{m_1 l^2} \tau_1 \\
 \dot{x}_3 &= x_4 \\
 \dot{x}_4 &= -\frac{k r^3}{m_2 l^2} (\sin x_1 - \sin x_3)^2 \cos x_3 - \frac{g}{l} \sin x_3 + \frac{1}{m_2 l^2} \tau_2 \\
 y_1 &= x_1 \\
 y_2 &= x_3
 \end{aligned} \tag{1.22}$$

Le modèle obtenu est fortement non linéaire. Par exemple, dans le cas où on considère que les déplacements angulaires  $\theta_1$  et  $\theta_2$  sont faibles (hypothèse simplificatrice), alors on a :

$$\cos x_1 \approx \cos x_3 = 1$$

et

$$\sin x_1 \approx x_1, \quad \sin x_3 \approx x_3$$

Le modèle d'état fortement non linéaire **(1.22)** prend la forme suivante :

$$\begin{aligned}
 \dot{x}_1 &= x_2 \\
 \dot{x}_2 &= -\frac{k r^3}{m_1 l^2} (x_1 - x_3)^2 - \frac{g}{l} x_1 + \frac{1}{m_1 l^2} \tau_1 \\
 \dot{x}_3 &= x_4 \\
 \dot{x}_4 &= -\frac{k r^3}{m_2 l^2} (x_1 - x_3)^2 - \frac{g}{l} x_3 + \frac{1}{m_2 l^2} \tau_2 \\
 y_1 &= x_1 \\
 y_2 &= x_3
 \end{aligned} \tag{1.23}$$

Les évolutions des positions angulaires  $\theta_1$  et  $\theta_2$  obtenues avec les paramètres  $m_1 = 1$  kg,  $m_2 = 4$  kg,  $k = 1$  N/m,  $l = 1$  m et  $r = l/2$  m, et les conditions initiales  $\theta_1(0) = 30^\circ$  et  $\theta_2(0) = 45^\circ$  sont données par la Figure **1.3**.

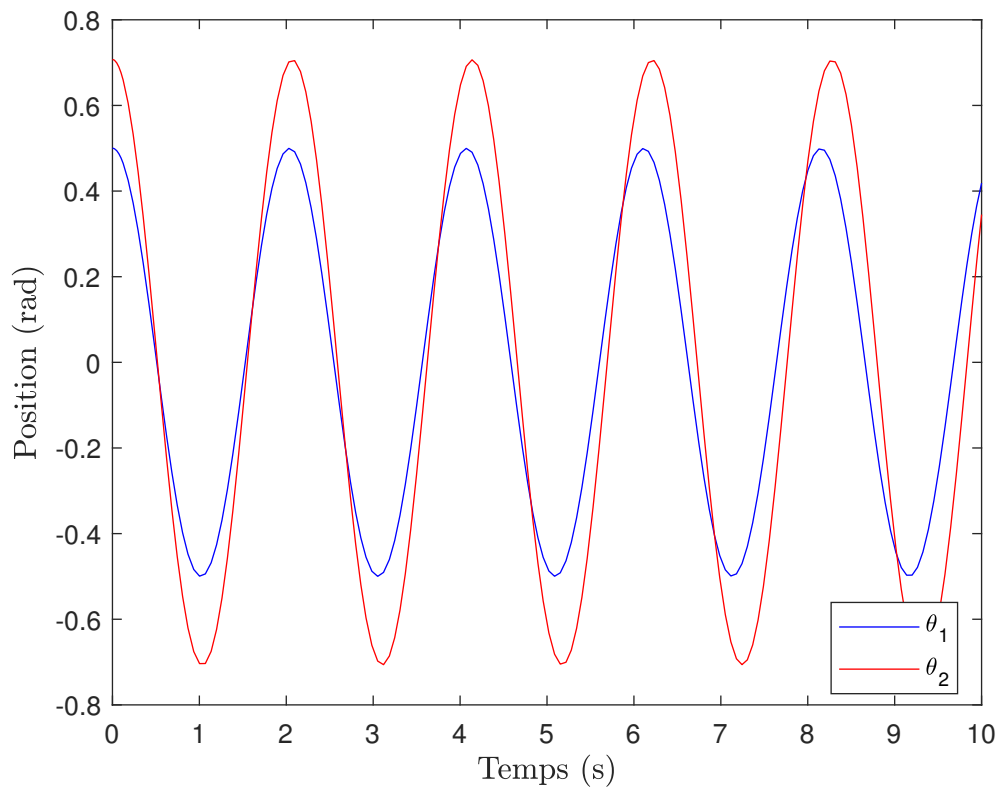


FIGURE 1.3 – Évolution des positions angulaires des masses.

## 1.7 Utilisation d'un modèle mathématique

Le modèle mathématique est une représentation abstraite du système physique. Ce modèle mathématique sera utilisé à la place du système physique. Cette approche présente plusieurs avantages. En effet, il est intéressant d'effectuer des tests sur le modèle que sur le système physique pour les raisons suivantes :

- Sécurité : certains tests peuvent être dangereux pour le système et l'environnement. Par exemple, appliquer des commandes importantes pour un moteur électrique ou un réacteur nucléaire.
- Économique : certains tests physiques sont très coûteux. Par exemple, étudier l'influence de la concentration d'un produit chimique sur le produit final nécessite d'utiliser des produits avec différentes concentrations.
- Gain en temps : certaines expériences pratiques peuvent nécessiter un temps important pour observer le résultat, par contre avec une simulation directe sur le modèle, le résultat est immédiat.

L'utilisation d'un modèle mathématique repose essentiellement sur la résolution des équations différentielles ordinaires. Ainsi, l'exploitation d'un modèle mathématique nécessite la résolution des équations du modèle. Cette dernière peut se faire analytiquement ou numériquement selon la complexité du modèle [2, 5, 8]. Souvent la résolution se fait

numériquement en utilisant des méthodes d'intégration des équations différentielles ordinaires [2, 5]. Ainsi, le choix d'une méthode numérique appropriée et sa paramétrisation jouent un rôle important pour la validation du modèle [2, 5].

Les différentes applications du modèle mathématique basées sur la résolution des équations du modèle sont :

- Simulation du comportement du modèle mathématique : la résolution numérique d'un modèle mathématique permet de reproduire le comportement dynamique du système physique pour certaines conditions de fonctionnement, par exemple pour étudier l'influence des conditions initiales ou d'un paramètre physique sur la stabilité du système, et des perturbations externes.
- Validation d'un modèle mathématique : un modèle mathématique issu de la modélisation est généralement entaché des erreurs due principalement aux hypothèses simplificatrices considérées [1]. En effet, il est impossible de considérer tout les phénomènes physiques lors de la modélisation [8]. Par conséquent, avant toute utilisation du modèle l'étape de validation est primordiale. Cette dernière consiste à confronter les prédictions, obtenues par résolution numérique du modèle mathématique, avec des mesures expérimentales collectées pour d'autres signaux de commandes [1]. Sur la base de cette confrontation, on peut juger la validité du modèle élaboré.
- Analyse des propriétés fondamentales du système : on peut utiliser la simulation numérique pour étudier les propriétés fondamentales d'un système dynamique, par exemple la commandabilité et la stabilité.
- Validation des lois de commande : avant d'implémenter les lois de commande sur le système physique, il est important de les valider sur le modèle mathématique en évaluant leurs performances de poursuite et de régulation.
- Utilisation en ligne du modèle mathématique : pour l'implémentation de certaines lois de commandes ou techniques de diagnostic, l'utilisation d'un observateur d'état est indispensable. Ce dernier est construit sur la base d'un modèle mathématique, par conséquent la résolution en temps réel du modèle du système est inéluctable. Aussi, dans certains types de commande, la résolution du modèle mathématique joue un rôle du premier plan, par exemple dans le cas de la commande prédictive [1], à chaque instant d'échantillonnage, un problème d'optimisation faisant intervenir le modèle mathématique doit être résolu. Dans le cas de la commande adaptative

à modèle de référence, le modèle du système doit être résolu en ligne pour définir l'erreur de poursuite. Dans le cas de la commande à modèle interne, le modèle est utilisé pour assurer une certaine robustesse vis-à-vis des incertitudes [1], et aussi pour le rejet de perturbation. En effet, une erreur entre la sortie commande et celle du modèle interne est définie. Cette dernière est utilisée pour calculer la commande permettant de minimiser cette erreur.

## 1.8 Conclusion

Dans ce chapitre, nous avons présenté des généralités sur la représentation mathématique des systèmes dynamiques dans le le domaine temporel. Nous avons présenté les deux approches utilisées pour l'obtention du modèle mathématique basée respectivement sur l'écriture des lois de la physique et sur une campagne de mêmes entrée-sortie. Puis, nous avons axé l'exposé sur les modèles mathématiques donnés sous formes d'équations différentielles ordinaires dont le modèle d'état est cas particulier. Ensuite, nous avons présenté un exemple d'application. À la fin du chapitre, nous avons présenté les différentes utilisations du modèle mathématique où on a souligné l'importance de la résolution en ligne des équations du modèle mathématique.

La résolution numérique des équations des modèles fera l'objet du chapitre suivant.

# Chapitre 2

## Résolution numérique des équations différentielles ordinaires

### 2.1 Introduction

L'utilisation d'un modèle mathématique passe inéluctablement par la résolution de ses équations. Ainsi, la solution peut être obtenue analytiquement ou numériquement selon la complexité de ces équations. Si la solution analytique peut être obtenue dans le cas des équations linéaires et particulièrement pour certaines équations non linéaires, la solution numérique des équations est toujours possible.

Dans ce chapitre, on s'intéresse à la résolution numérique des équations différentielles ordinaires du premier ordre. Rappelons qu'une équation différentielle ordinaire ou un système d'équations différentielles ordinaires d'ordre supérieur peuvent être écrits sous forme d'un système d'équations différentielles ordinaires d'ordre un en utilisant les variables d'état.

### 2.2 Position du problème

Un système d'équations ordinaires d'ordre un est de la forme :

$$\frac{dx(t)}{dt} = f(t, x(t)) \quad (2.1)$$

où  $t$  est la variable indépendante,  $x(t)$  est la fonction vectorielle inconnue qu'on cherche à déterminer, et  $f$  est une fonction vectorielle donnant le taux de changement de  $x$  par rapport à  $t$ . La solution de cette équation est la fonction  $x(t)$  qui satisfait l'équation différentielle ordinaire (2.1) et la condition initiale  $x(t_0) = x_0$ .

Avec les méthodes numériques de résolution d'équations différentielles, il n'est plus possible d'obtenir une solution pour toutes les valeurs de la variable indépendante  $t$ . On obtient plutôt une approximation de la solution analytique seulement à certaines valeurs de  $t$  notée  $t_i$  et distancées d'une valeur constante  $h = t_{i+1} - t_i$  ( $h$  est appelé le pas de  $t$ ). Ainsi, on désigne par  $x(t_i)$  la solution analytique de l'équation différentielle (2.1) en  $t = t_i$ , et  $x_i$  la solution approximative (numérique) en  $t = t_i$  obtenue à l'aide d'une méthode numérique.

L'utilisation des méthodes numériques permettent de déterminer la suite suivante :

$$x_1 = x(t_1), x_2 = x(t_2), x_3 = x(t_3), \dots, x_i = x(t_i), \dots \quad (2.2)$$

qui représente la solution numérique de l'équation différentielle ordinaire (2.1). Notons qu'avec une méthode numérique, il n'est pas possible d'obtenir une solution pour toutes les valeurs de la variable indépendante  $t$ . Néanmoins, les valeurs de la solution entre  $t_i$  et  $t_{i+1}$  peut être obtenue par des techniques d'interpolation.

Dans ce qui suit, on va présenter quelques méthodes qui permettent de construire la suite de valeurs (2.2), c'est-à-dire la solution numérique de l'équation différentielle ordinaire (2.1). L'étude sera limitée à une équation différentielle dont  $f$  est un scalaire, mais le développement peut être généralisé facilement pour le cas où  $f$  est une fonction vectorielle.

## 2.3 Méthodes numériques à pas multiples

On appelle méthode numérique à  $s + 1$  pas, toutes méthodes qui permettent de calculer la solution  $x_i$  de l'équation différentielle ordinaire (2.1) en utilisant l'équation de récurrence suivante

$$x_{i+1} = \psi(t_i, x_i, t_{i-1}, x_{i-1}, \dots, t_{i-s}, x_{i-s}, h) \quad (2.3)$$

où  $\psi$  est une fonction non linéaire. Le nombre de pas  $s + 1$  représente le nombre de valeurs de solutions précédentes nécessaires pour calculer une nouvelle solution [2].

L'intérêt des méthodes à  $s + 1$  pas, vient du fait qu'on peut obtenir un ordre élevé pour une complexité de calcul nettement inférieure à celle des méthodes de Runge-Kutta par exemple [2, 5]. L'un des problèmes essentiels, néanmoins, est de s'assurer que la stabilité numérique reste suffisamment bonne.

Par exemple, une méthode à un pas ( $s = 0$ ) permet de déterminer la solution en

utilisant la formule suivante

$$x_{i+1} = \psi(t_i, x_i, h) \quad (2.4)$$

et une méthode à deux pas ( $s = 1$ ) utilise la formule suivante

$$x_{i+1} = \psi(t_i, x_i, t_{i-1}, x_{i-1}, h) \quad (2.5)$$

## 2.4 Méthode d'Euler explicite

La méthode d'**Euler** (explicite) est de loin la méthode à un pas la plus simple de résolution numérique d'équation différentielle ordinaire. Bien que son emploi soit facile, toutefois, elle est relativement peu utilisée en raison de sa faible précision et d'instabilité numérique si le pas  $h$  est mal choisi [5].

L'idée fondamentale de la méthode d'**Euler** est de discrétiser le temps avec un pas  $h$  constant et d'utiliser la pente de la solution à chaque étape pour estimer la valeur de la solution à l'itération suivante. Dans ce cas, la fonction non-linéaire  $\psi$  est donnée comme suit

$$\psi(t_i, x_i, h) = x_i + h f(t_i, x_i) \quad (2.6)$$

c'est-à-dire la suite des valeurs de la solution sont obtenues en utilisant l'équation de récurrence suivante

$$x_{i+1} = x_i + h f(t_i, x_i) \quad (2.7)$$

d'après la formule (2.7), il est clair que pour calculer la solution  $x_{i+1}$ , on a besoin de la solution  $x_i$ , par conséquent la méthode d'**Euler** est une méthode à un pas ( $s = 0$ ).

Une méthode à un pas converge à l'ordre  $\mu$  si [5]

$$\varepsilon_i(h) = \max_{1 \leq i \leq N} |x(t_i) - x_i| = \mathcal{O}(h^\mu) \quad (2.8)$$

où  $N$  est le nombre total de pas de temps. L'ordre de convergence d'une méthode à un pas dépend de l'erreur commise à chaque pas de temps caractérisée par l'erreur de troncature locale  $\varepsilon_{i+1}$ . Ainsi, pour  $t = t_i$ , l'erreur  $\varepsilon_{i+1}$  est définie comme suit [5]

$$\varepsilon_{i+1}(h) = \frac{x_{i+1} - x_i}{h} - f(t, x(t_i)) \quad (2.9)$$

L'erreur de troncature locale mesure la précision avec laquelle la solution analytique vérifie l'équation de récurrence (2.7). Par conséquent, on peut conclure que cette précision

dépend du pas  $h$ .

Par exemple, pour la méthode d'**Euler**, en effectuant un développement de **Taylor** de la solution  $x_{i+1}$  autour du point  $t = t_i$ , on obtient

$$\begin{aligned} x_{i+1} &= x(t_i + h) \\ &= x(t_i) + h \dot{x}(t_i) + \frac{h^2}{2} \ddot{x}(t_i) + \mathcal{O}(h^3) \end{aligned} \quad (2.10)$$

En substituant  $x_{i+1}$  par son expression donnée par (2.10), on obtient l'erreur de troncature locale suivante

$$\varepsilon_{i+1}(h) = \dot{x}(t_i) + \frac{h}{2} \ddot{x}(t_i) + \mathcal{O}(h^2) - f(t, x(t_i)) \quad (2.11)$$

et comme  $\dot{x}(t_i) = f(t, x(t_i))$ , alors

$$\varepsilon_{i+1}(h) = \frac{h}{2} \ddot{x}(t_i) + \mathcal{O}(h^2) \quad (2.12)$$

où simplement

$$\varepsilon_{i+1}(h) = \mathcal{O}(h) \quad (2.13)$$

En utilisant la relation (2.8), on déduit que la méthode d'**Euler** explicite converge à l'ordre 1 ( $\mu = 1$ ).

Pour illustrer la méthode d'**Euler** explicite, on considère l'équation différentielle ordinaire suivante

$$\dot{x}(t) = -x(t) + t + 1 \quad (2.14)$$

$$x(0) = 1 \quad (2.15)$$

dont la solution exacte est

$$x(t) = e^{-t} + t \quad (2.16)$$

En considérant un pas  $h = 0,1$ ; l'équation de récurrence est donnée comme suit

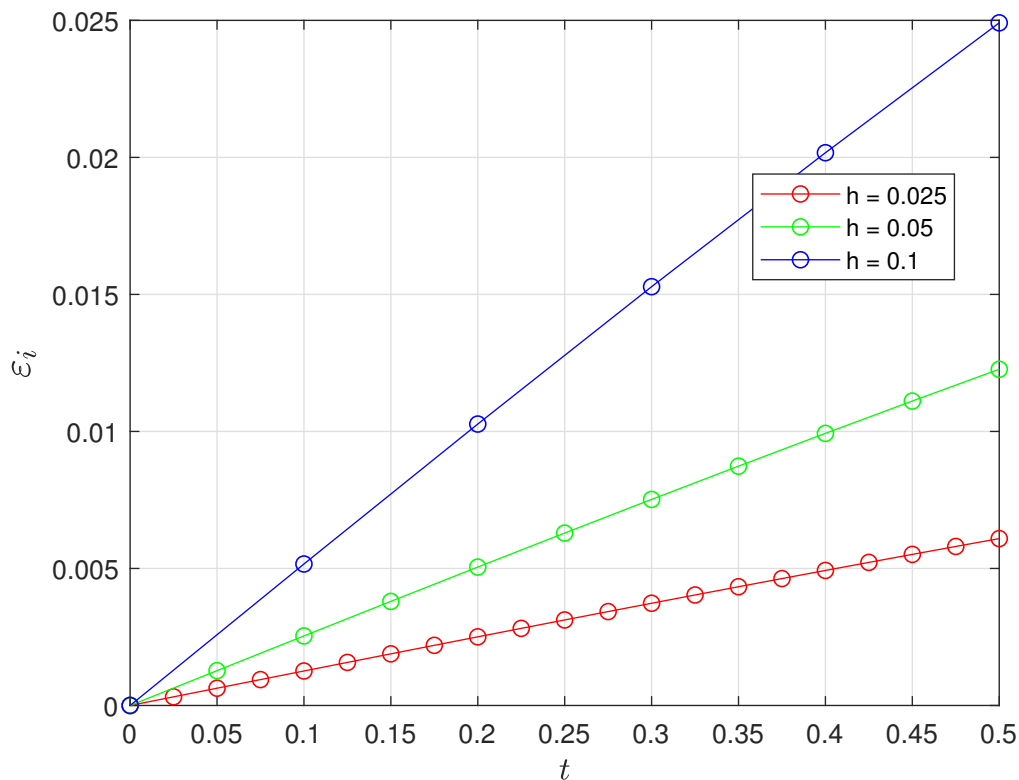
$$x_{i+1} = x_i + h(-x_i + t_i + 1) \quad (2.17)$$

Le Tableau 2.1 donne les résultats des cinq premiers pas de temps.

$i$	$t_i$	$x(t_i)$	$x_i$	$\varepsilon_i =  x(t_i) - x_i $
0	0	1.00000000	1.00000000	0.00000000
1	0,1	1.00483742	1.01000000	0.00516258
2	0,2	1.01873075	1.02900000	0.01026925
3	0,3	1.04081822	1.05610000	0.01528178
4	0,4	1.07032005	1.09049000	0.02016995
5	0,5	1.10653066	1.13144100	0.02491034

TABLE 2.1 – Solution numérique avec la méthode d'**Euler**.

Les résultats du tableau montrent une différence entre la solution numérique et la solution analytique. On remarque aussi que cette erreur augmente lorsque  $t$  augmente. Pour montrer l'influence du pas  $h$  sur la précision de la solution numérique, on représente l'erreur  $\varepsilon_i$  pour  $h = 0,025$ ,  $h = 0,05$  et  $h = 0,1$ . Les résultats obtenus sont donnés par la Figure 2.1. On remarque que l'erreur devient importante avec l'augmentation de  $h$ .

FIGURE 2.1 – Méthode d'**Euler** : évolution de l'erreur pour différentes valeurs de  $h$ .

## 2.5 Méthodes d'Adams-Bashforth

L'idée de base des méthodes à pas multiples consiste à intégrer l'équation différentielle (2.1) dans l'intervalle  $[t_i, t_{i+1}]$  comme suit

$$\int_{t_i}^{t_{i+1}} \dot{x}(t) dt = \int_{t_i}^{t_{i+1}} f(t, x(t)) dt \quad (2.18)$$

ce qui donne

$$x(t_{i+1}) - x(t_i) = \int_{t_i}^{t_{i+1}} f(t, x(t)) dt \quad (2.19)$$

ou encore

$$x(t_{i+1}) = x(t_i) + \int_{t_i}^{t_{i+1}} f(t, x(t)) dt \quad (2.20)$$

$$x_{i+1} = x_i + \int_{t_i}^{t_{i+1}} f(t, x(t)) dt \quad (2.21)$$

Il reste à trouver une approximation de l'intégrale présente dans l'équation (2.21). Pour y arriver, on utilise une interpolation de la fonction  $f(t, x(t))$  à partir des valeurs de  $x(t)$  calculées aux itérations précédentes. On note  $f_i = f(t_i, x_i)$  l'approximation de  $f(t_i, x(t_i))$ . On peut évaluer la fonction  $f(t, x(t))$  au moyen d'un polynôme  $p_\eta(t)$  dans l'intervalle  $[t_i, t_{i+1}]$ . L'évaluation de  $p_i(t)$  ne requiert que des valeurs connues provenant des itérations précédentes. En augmentant successivement le degré du polynôme, on obtient des approximations de plus en plus précises [5].

Par exemple, si l'on utilise le polynôme de degré  $\eta = 0$  (une constante), on a l'approximation

$$f(t, x(t)) \approx p_0(t) = f_i \quad (2.22)$$

et en utilisant l'équation (2.21), on obtient

$$\begin{aligned} x_{i+1} &= x_i + \int_{t_i}^{t_{i+1}} f(t, x(t)) dt \\ &= x_i + \int_{t_i}^{t_{i+1}} f_i dt \\ &= x_i + f_i (t_{i+1} - t_i) \\ &= x_i + h f(t_i, x_i) \end{aligned} \quad (2.23)$$

Dans ce cas, on obtient la formule de la méthode d'**Euler** explicite.

Si on utilise, un polynôme d'ordre  $\eta = 1$ , on a l'approximation suivante

$$f(t, x(t)) \approx p_1(t) = f_i + \frac{f_{i+1} - f_i}{t_{i+1} - t_i} (t - t_i) \quad (2.24)$$

Ainsi, en utilisant l'équation (2.21), il vient

$$\begin{aligned}
 x_{i+1} &= x_i + \int_{t_i}^{t_{i+1}} f(t, x(t)) dt \\
 &= x_i + \int_{t_i}^{t_{i+1}} \left[ f_i + \frac{f_{i+1} - f_i}{t_{i+1} - t_i} (t - t_i) \right] dt \\
 &= x_i + \frac{h}{2} (3f_i - f_{i-1}) \\
 &= x_i + \frac{h}{2} [3f(t_i, x_i) - f(t_{i-1}, x_{i-1})]
 \end{aligned} \tag{2.25}$$

Dans ce cas, on obtient la méthode d'**Adams-Bashforth** à deux pas. En effet pour calculer la solution  $x_{i+1}$ , on a besoin des deux solutions précédentes  $x_{i-1}$  et  $x_i$  (deux pas).

Ainsi, en considérant des polynômes  $p_\eta(t)$  d'ordre  $\eta = 2$  et  $\eta = 3$ , on obtient les méthodes d'**Adams-Bashforth** à trois pas et quatre pas dont les formules sont respectivement

$$x_{i+1} = x_i + \frac{h}{12} [23f(t_i, x_i) - 16f(t_{i-1}, x_{i-1}) - 5f(t_{i-2}, x_{i-2})] \tag{2.26}$$

$$x_{i+1} = x_i + \frac{h}{24} [55f(t_i, x_i) - 59f(t_{i-1}, x_{i-1}) + 37f(t_{i-2}, x_{i-2}) - 9f(t_{i-3}, x_{i-3})] \tag{2.27}$$

On constate qu'en utilisant un polynôme de degré  $\eta$ , on obtient une méthode à  $s + 1$  pas dont l'erreur de troncature locale est d'ordre  $\alpha + 1$ . On définit l'erreur de troncature locale liée aux méthodes à pas multiples d'une manière semblable à ce que l'on fait dans le cas des méthodes à un pas.

Pour illustrer la méthode d'**Adams-Bashforth** à deux pas, on considère le même exemple étudié précédemment avec la méthode d'**Euler** explicite

$$\dot{x}(t) = -x(t) + t + 1 \tag{2.28}$$

$$x(0) = 1 \tag{2.29}$$

dont la solution exacte est

$$x(t) = e^{-t} + t \tag{2.30}$$

En considérant un pas  $h = 0,1$ ; l'équation de récurrence obtenue avec la méthode

d'Adams-Bashforth est donnée comme suit

$$x_{i+1} = x_i + \frac{h}{2} [3(-x_i + t_i + 1) - (-x_{i-1} + t_{i-1} + 1)] \quad (2.31)$$

Le Tableau 2.2 rassemble les résultats des cinq premiers pas de temps.

$i$	$t_i$	$x(t_i)$	$x_i$	$\varepsilon_i =  x(t_i) - x_i $
0	0	1.00000000	1.00000000	0.00000000
1	0,1	1.00483742	1.00000000	0.00483742
2	0,2	1.01873075	1.02500000	0.00626925
3	0,3	1.04081822	1.05625000	0.01543178
4	0,4	1.07032005	1.09406250	0.02374245
5	0,5	1.10653066	1.13776562	0.03123497

TABLE 2.2 – Solution numérique avec la méthode d'Adams-Bashforth.

Les résultats du tableau montrent une différence entre la solution numérique et la solution analytique. On remarque aussi que cette erreur augmente lorsque  $t$  augmente.

Pour montrer l'influence du pas  $h$  sur la précision de la solution numérique, on représente l'erreur  $\varepsilon_i$  pour  $h = 0.025$ ,  $h = 0.05$ ,  $h = 0.1$  et  $h = 0.15$ . Les résultats obtenus sont donnés par la Figure 2.2. Ces résultats démontrent l'impact du choix du pas de temps sur la précision de la solution numérique.

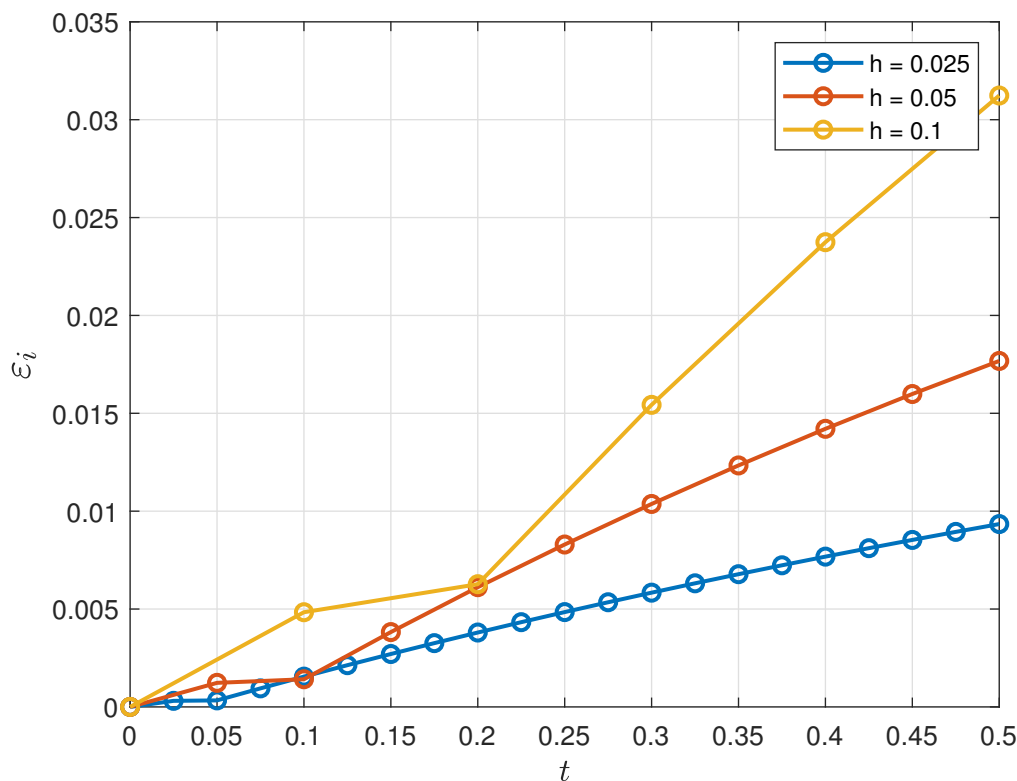


FIGURE 2.2 – Méthode d'Adams-Bashforth : évolution de l'erreur pour différentes valeurs de  $h$ .

## 2.6 Conclusion

Dans ce chapitre, nous avons présenté le principe général des méthodes d'intégration numérique des équations différentielles ordinaires à un pas et à pas multiples. Nous avons démontré les formules des méthodes d'**Euler** et d'**Adams-Bashforth** à deux pas. Ces deux méthodes ont été illustrées par un exemple d'application où on montre l'influence du pas d'intégration sur la précision et la stabilité numérique de la solution.

Le pas d'intégration numérique joue un rôle important pour assurer la stabilité de la méthode et obtenir une solution avec la précision souhaitée. Néanmoins, en pratique, le choix de ce pas est difficile surtout dans le cas des équations différentielles non linéaires.

Ainsi, pour assurer la stabilité et la précision d'une méthode d'intégration numérique, il est intéressant, pour un pas donné, de contrôler l'erreur de troncature donnée, c'est-à-dire forcer cette erreur à être faible. Cet objectif peut être achevé en utilisant des techniques de commande automatique, par exemple la commande LQ. Le chapitre suivant sera consacré à la commande LQ.

# Chapitre 3

## Commande linéaire quadratique (LQ)

### 3.1 Introduction

La commande optimale permet de transférer l'état d'un système dynamique d'un état initial vers un état final, imposé ou libre, tout en minimisant un critère de performances et en respectant un certain nombre de contraintes physiques [3, 6, 9], par exemple la saturation des actionneurs.

Pour formuler un problème de commande optimale, on doit préciser le modèle du système sous forme d'état, les conditions terminales (état initial et final), le critère de performances, et les contraintes physiques. Lorsque le modèle est linéaire et le critère de performances est quadratique, on parle d'une commande linéaire quadratique (LQ) [6].

Une commande LQ permet de réaliser un placement de pôles optimal. Elle vise à minimiser deux objectifs : l'erreur et l'énergie mise en œuvre. Dans ce chapitre, nous nous intéressons à la conception d'une commande LQ par la programmation dynamique basée le principe d'optimalité de **Bellman**.

### 3.2 Problème de commande linéaire quadratique

Un problème de commande LQ à horizon de commande fini prend la forme suivante [3, 6]

$$\begin{aligned} \min_{u(t)} J(u(t), t) &= \int_{t_0}^{t_f} x^T(t) Q x^T(t) + u^T(t) R u(t) dt \\ \text{sujet à :} & \\ \dot{x}(t) &= A x(t) + B u(t) \\ x(t_0) &= x_0 \end{aligned} \tag{3.1}$$

où  $x$  est l'état du système,  $u$  est le vecteur de commande,  $A \in \mathfrak{R}^{n \times n}$  et  $B \in \mathfrak{R}^{n \times m}$  sont les matrices d'état et de commande du système,  $Q = Q^T \geq 0$  et  $R = R^T > 0$  sont des matrices de pondération,  $t_0$  et  $t_f$  sont les instants initial et final,  $x_0$  est l'état initial du système. Dans ce qui suit, on suppose que l'état final est libre.

L'objectif consiste à chercher la commande  $u$  qui permet de minimiser le critère  $J$ . En utilisant le principe du minimum de **Pontryagin**, on peut démontrer que la commande optimale est un retour d'état de la forme [3]

$$u(t) = K x(t) \tag{3.2}$$

En substituant l'expression de la commande dans le modèle du système, on obtient en boucle fermée le système suivant

$$\dot{x}(t) = (A + B K) x(t) \tag{3.3}$$

On remarque que le choix de la matrice  $K$  permet de fixer ou réaliser un placement de pôles pour le système en boucle fermée. Comme la matrice  $K$  est déterminée en minimisant le critère de performances  $J$ , alors le placement de pôles est optimal.

Pour déterminer la matrice  $K$ , on va utiliser la programmation dynamique basée sur le principe d'optimalité de **Bellman**.

### 3.3 Programmation dynamique

La programmation dynamique [3, 9] est l'une des méthodes utilisée pour la résolution des problèmes d'optimisation dynamique (les solutions recherchées sont des fonctions), comme le cas de la commande optimale où on cherche à déterminer la fonction  $u$  comme une fonction de la variable  $t$  qui minimise le coût  $J$ . La programmation dynamique est basée sur le principe d'optimalité de **Bellman** [3].

### 3.3.1 Principe d'optimalité de Bellman

Le principe d'optimalité de **Bellman** s'énonce comme suit [3] :

Si  $\mathcal{B}$  est un point intermédiaire de la trajectoire optimale allant de l'état initial  $\mathcal{A}$  à l'état final  $\mathcal{C}$  (Figure 3.1), la portion  $\mathcal{BC}$  de cette trajectoire constitue la trajectoire optimale reliant l'état intermédiaire  $\mathcal{B}$  à l'état final  $\mathcal{C}$ .

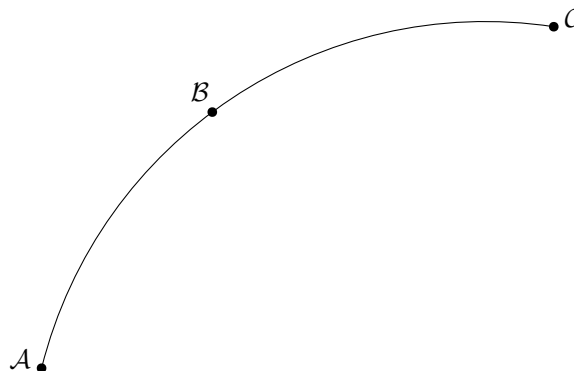


FIGURE 3.1 – Principe d'optimalité de **Bellman**.

### 3.3.2 Discrétisation du problème de commande optimale

L'application de la programmation dynamique pour concevoir une commande optimale peut se faire de deux manières. Ainsi, en considérant le problème de commande optimale sous sa forme continue, le principe d'optimalité conduit à l'équation de **Hamilton-Jacobi-Bellman** qui est une équation aux dérivées partielles dont la résolution est souvent difficile [3, 9]. En considérant le problème de commande optimale sous sa forme discrète, le principe d'optimalité conduit à l'équation fonctionnelle de **Bellman** qui est une équation de récurrence simple à manipuler [3]. Cette deuxième forme est la plus utilisée en commande optimale.

Pour appliquer l'équation fonctionnelle de **Bellman**, on doit alors mettre le problème de commande optimale (3.1) sous la forme discrète. Pour discrétiser le système, on utilise la méthode d'**Euler**, ce qui donne[3]

$$x(k+1) = A_d x(k) + B_d u(k) \quad (3.4)$$

avec

$$A_d = I + \Delta t A, \quad B_d = \Delta t B$$

où  $\Delta t$  est la période échantillonnage,  $I$  est une matrice identité de même dimension que

la matrice  $A$ , et  $t = k \Delta t$ .

La discrétisation du critère en utilisant la méthode des rectangles [2, 5], utilisée pour le calcul approché des intégrales, donne :

$$J_d(u(k), k) = \sum_{i=1}^N x^T(k) Q x(k) + u^T(k-1) R u(k-1) \quad (3.5)$$

### 3.3.3 Détermination de la commande optimale

Dans cette sous-section, on va appliquer le principe d'optimalité de **Bellman** pour résoudre le problème de commande LQ. Ainsi, de manière générale, en prenant  $\mathcal{A} \equiv x(k)$ ,  $\mathcal{B} \equiv x(N)$  et  $\mathcal{C} \equiv x(k+1)$ , et en utilisant le principe d'optimalité de **Bellman**, le coût total pour transférer le système de l'état  $x(k)$  à l'état final  $x(N)$  (qui est libre dans notre cas) est donné par la somme du coût partiel  $x^T(k+1) Q x(k+1) + u^T(k) R u(k)$  et du coût optimal  $J_d^*(x(k+1), k+1)$ , c'est-à-dire

$$J_d(x(k), k) = x^T(k+1) Q x(k+1) + u^T(k) R u(k) + J_d^*(x(k+1), k+1) \quad (3.6)$$

Comme le coût  $J_d^*(x(k+1), k+1)$  est supposé connu, alors le problème revient à déterminer la commande  $u(k)$  qui minimise le coût total  $J_d(x(k), k)$ , c'est-à-dire à résoudre le problème de commande d'optimisation suivant

$$\min_{u(k)} J_d(x(k), k) = x^T(k+1) Q x(k+1) + u^T(k) R u(k) + J_d^*(x(k+1), k+1) \quad (3.7)$$

Notons que  $x(k+1)$  est la conséquence de l'application de la commande  $u(k)$ . Ainsi, en utilisant le modèle discret du système (3.4), le problème d'optimisation (3.7) prend la forme suivante

$$\begin{aligned} \min_{u(k)} J_d(x(k), k) &= [A_d x(k) + B_d u(k)]^T Q [A_d x(k) + B_d u(k)] + u^T(k) R u(k) \\ &+ J_d^*(x(k+1), k+1) \end{aligned} \quad (3.8)$$

ou encore sous la forme simplifiée

$$\begin{aligned} \min_{u(k)} J_d(x(k), k) &= u^T(k) [B_d^T Q B_d + R] u(k) + 2 x^T(k) A_d^T Q B_d u(k) \\ &+ x^T(k) A_d^T Q A_d x(k) + J_d^*(x(k+1), k+1) \end{aligned} \quad (3.9)$$

On peut démontrer en utilisant le principe du minimum de **Pontryagin** que le coût optimal est de la forme [9]

$$J_d^*(x(k), k) = x^T(k) P(k) x(k) \quad (3.10)$$

où  $P$  est une matrice à déterminer.

Dans ce cas, le problème d'optimisation (3.9) prend la forme suivante

$$\begin{aligned} \min_{u(k)} J_d(x(k), k) &= u^T(k) [B_d^T Q B_d + R] u(k) + 2 x^T(k) A_d^T Q B_d u(k) \\ &+ x^T(k) A_d^T Q A_d x(k) + x^T(k+1) P(k+1) x(k+1) \end{aligned} \quad (3.11)$$

et en utilisant l'équation du modèle discrétisé (3.4), on obtient

$$\begin{aligned} \min_{u(k)} J_d(x(k), k) &= u^T(k) [B_d^T Q B_d + R] u(k) + 2 x^T(k) A_d^T Q B_d u(k) \\ &+ x^T(k) A_d^T Q A_d x(k) + [A_d x(k) + B_d u(k)]^T P(k+1) \\ &[A_d x(k) + B_d u(k)] \end{aligned} \quad (3.12)$$

qu'on peut simplifier comme suit :

$$\begin{aligned} \min_{u(k)} J_d(x(k), k) &= u^T(k) [B_d^T Q B_d + R] u(k) + 2 x^T(k) A_d^T Q B_d u(k) \\ &+ x^T(k) A_d^T Q A_d x(k) + x^T(k) A_d^T P(k+1) A_d x(k) \\ &+ 2 x^T(k) A_d^T P(k+1) B_d u(k) + u^T(k) B_d^T P(k+1) B_d u(k) \end{aligned} \quad (3.13)$$

Ainsi, en calculant le gradient de  $J_d(x(k), k)$  par rapport à  $u(k)$ , on obtient :

$$\begin{aligned} \frac{\partial J_d(x(k), k)}{\partial u(k)} &= 2 [B_d^T Q B_d + R] u(k) + 2 [x^T(k) A_d^T Q B_d]^T \\ &+ 2 [x^T(k) A_d^T P(k+1) B_d]^T + 2 B_d^T P(k+1) B_d u(k) \\ &= 2 [B_d^T (Q + P(k+1)) B_d + R] u(k) \\ &+ 2 [B_d Q A_d^T + B_d^T P(k+1) A_d] x(k) \end{aligned} \quad (3.14)$$

et en annulant ce gradient, on obtient l'équation suivante

$$[B_d^T (Q + P(k+1)) B_d + R] u(t) + [B_d Q A_d^T + B_d^T P(k+1) A_d] x(k) = 0 \quad (3.15)$$

la résolution de l'équation (3.15) par rapport à  $u(k)$  permet de déterminer l'expression de la commande optimale suivante

$$u^*(k) = - [B_d^T (Q + P(k+1)) B_d + R]^{-1} [B_d Q A_d^T + B_d^T P(k+1) A_d] x(k) \quad (3.16)$$

qu'on peut écrire sous la forme

$$u^*(k) = K(k) x(k) \quad (3.17)$$

avec

$$K(k) = - [B_d^T (Q + P(k+1)) B_d + R]^{-1} [B_d Q A_d^T + B_d^T P(k+1) A_d] \quad (3.18)$$

On remarque que la commande optimale (3.16) dépend de la matrice  $P(k)$  qui est à déterminer. Ainsi, d'après l'équation (3.13), on peut écrire

$$\begin{aligned} J_d^*(x(k), k) = & u^{*T}(k) [B_d^T Q B_d + R] u^*(k) + 2x^T(k) B_d^T Q A_d u^*(k) \\ & + x^T(k) A_d^T Q A_d x(k) + x^T(k) A_d^T P(k+1) A_d x(k) \\ & + 2x^T(k) A_d^T P(k+1) B_d u^*(k) + u^{*T}(k) B_d^T P(k+1) B_d u^*(k) \end{aligned} \quad (3.19)$$

et en utilisant les expressions du coût optimal (3.10) et de la commande optimale (3.17), l'équation (3.19) prend la forme suivante

$$\begin{aligned} x^T(k) P(k) x(k) = & x^T(k) K^T(k) [B_d^T Q B_d + R] K(k) x(k) \\ & + x^T(k) A_d^T Q A_d K(k) x(k) + x^T(k) A_d^T P(k+1) A_d x(k) \\ & 2x^T(k) A_d^T P(k+1) B_d K(k) x(k) + x^T(k) K^T(k) B_d^T \\ & P(k+1) B_d K(k) x(k) \end{aligned} \quad (3.20)$$

L'équation (3.20) peut être réécrite sous la forme

$$x^T(k) P(k) x(k) = x^T(k) M(k) x(k) \quad (3.21)$$

avec

$$\begin{aligned}
 M(k) = & K^T(k) [B_d^T Q B_d + R] K(k) + 2 A_d^T Q A_d K(k) + A_d^T P(k+1) A_d \\
 & + 2 A_d^T P(k+1) B_d K(k) + K^T(k) B_d^T P(k+1) B_d K(k)
 \end{aligned} \tag{3.22}$$

Ainsi, par identification, on obtient :

$$\begin{aligned}
 P(k) = & M(k) \\
 = & K^T(k) [B_d^T Q B_d + R] K(k) + 2 B_d^T Q A_d K(k) + A_d^T P(k+1) A_d \\
 & + 2 A_d^T P(k+1) B_d K(k) + K^T(k) B_d^T P(k+1) B_d K(k)
 \end{aligned} \tag{3.23}$$

En substituant  $K(k)$  par son expression (3.18) dans (3.23), et après simplification on obtient

$$\begin{aligned}
 P(k) = & A_d^T \left[ Q + P(k+1) - (Q + P(k+1)) B_d [B_d^T (Q + P(k+1)) B_d + R]^{-1} \right. \\
 & \left. B_d^T (Q + P(k+1)) \right] A_d
 \end{aligned} \tag{3.24}$$

Par conséquent, les valeurs de la matrice  $P(k)$  sont déterminées en utilisant l'équation de récurrence (3.24), d'après l'expression du critère (3.5), il vient pour  $k = N$

$$J_d^*(x(N), N) = 0_{n \times n} \tag{3.25}$$

et en utilisant l'équation (3.6), on a

$$J_d^*(x(N), N) = x^T(N) P(N) x(N) \tag{3.26}$$

Ainsi en utilisant les deux équations (3.25) et (3.26), on déduit que

$$P(N) = 0_{n \times n} \tag{3.27}$$

En résumé, l'implémentation d'un correcteur LQ sous sa forme discrète passe par les étapes suivantes :

1. Détermination, en temps inversé, de la matrice  $P(k)$  en utilisant l'équation de récurrence (3.24) en partant de la condition (3.27),
2. A chaque instant  $k$ , on calcule la commande optimale  $u^*(k)$  en utilisant l'équation (3.16).
3. Appliquer la commande optimale  $u^*(k)$  obtenue au système.
4. Répéter les étapes 2 et 3 jusqu'à ce que  $k = N$ .

### 3.4 Exemple d'application

Considérons le problème de commande optimale définie par

$$\min_{u(t)} J_d(x(k), k) = \sum_{k=0}^3 2x^2(k) + u^2(k-1) \quad (3.28)$$

sujet à :

$$x(k+1) = x(k) + u(k) \quad (3.29)$$

$$x(0) = 5 \quad (3.30)$$

L'identification des différentes matrices donne

$$A_d = 1, B_d = 1, Q = 2, R = 1$$

En utilisant l'équation (3.24), on obtient :

$$P(k) = \frac{P^3(k+1) + 6P^2(k+1) + 12P(k+1) + 8}{P(k+1) + 3} \quad (3.31)$$

avec  $P(3) = 0$ . (Équation (3.27)).

L'expression de la commande optimale, obtenue en utilisant la relation (3.16), est

$$u^*(k) = K(k)x(k) \quad (3.32)$$

avec

$$K(k) = -\frac{P(k+1) + 2}{P(k+1) + 3} \quad (3.33)$$

Les résultats de calcul sont donnés par le Tableau 3.1. On remarque que la commande optimale force l'état du système à converger vers zéro, puisque notre objectif consiste à minimiser l'énergie de l'état, c'est-à-dire son évolution.

$k$	$P(k)$	$K(k)$	$u^*(k)$	$x(k)$
0	378.3825	-0.9522	-4.7610	5
1	17.9340	-0.8235	-0.1968	0.239
2	2.6666	-0.6666	$-2.8130 \times 10^{-2}$	$4.22 \times 10^{-2}$
3	0	-0.6666	$-9.3990 \times 10^{-3}$	$1.41 \times 10^{-2}$

TABLE 3.1 – Commande optimale de l'exemple d'application.

Les résultats des simulations sont présentés graphiquement ci-dessous, illustrant la variation de  $P(k)$ ,  $K(k)$ ,  $u^*(k)$ , et  $x(k)$ , en fonction de  $k$ .

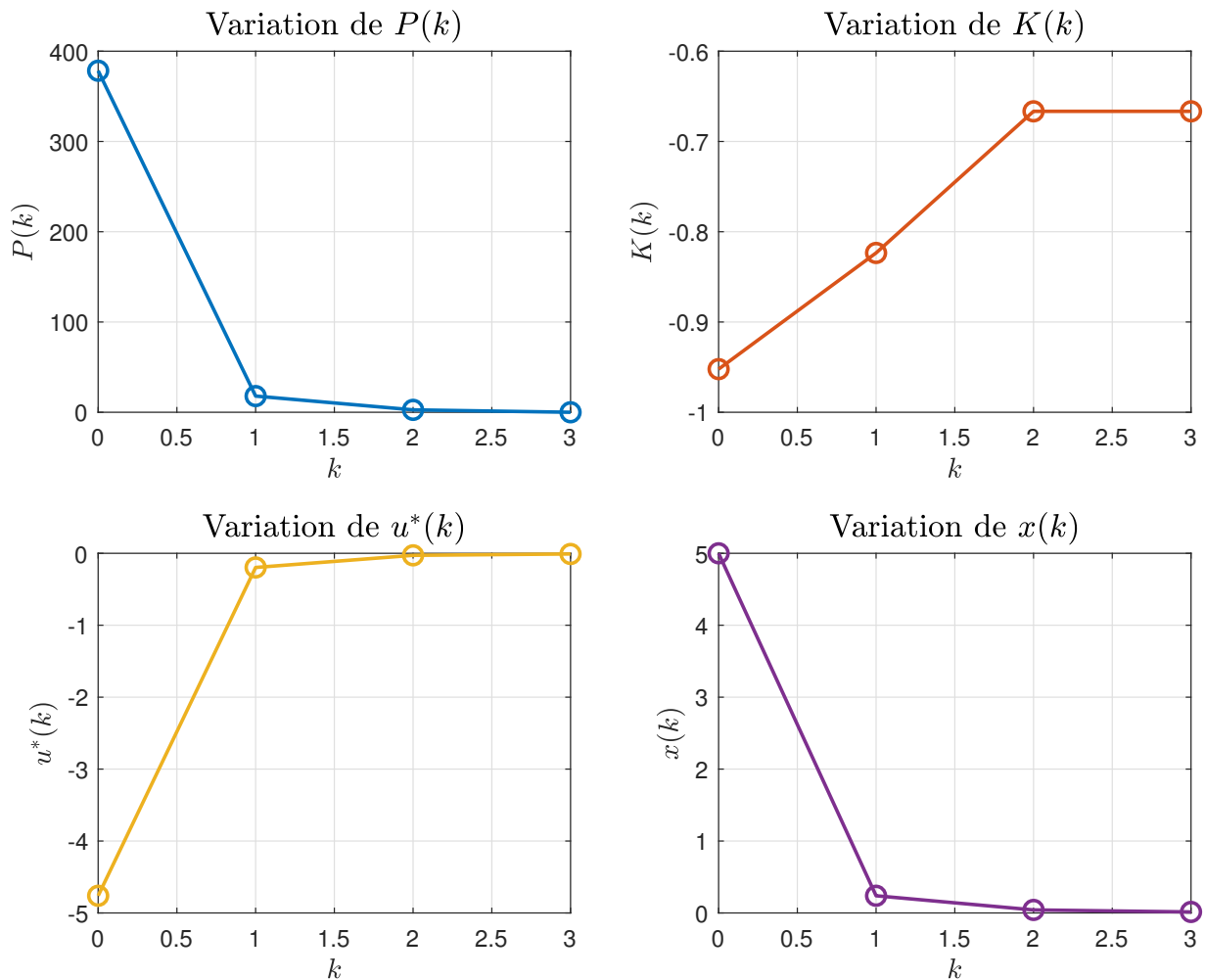


FIGURE 3.2 – Commande optimale de l'exemple d'application.

### 3.5 Conclusion

Dans ce chapitre, on s'est intéressé à la commande LQ des systèmes dynamiques. Cette commande permet de réaliser un placement de pôles optimal pour un système en boucle fermée de sorte à minimiser l'évolution de l'état et l'énergie mise en œuvre.

Pour déterminer l'expression de la loi de commande optimale, nous avons utilisé la programmation dynamique basée sur la forme discrète du principe d'optimalité de **Bellman**. Ainsi, nous avons déterminé l'expression de la commande optimal sous forme d'état dont le gain de retour est variable en fonction du temps. Nous avons aussi, présenté un exemple d'application pour montrer les performances de la commande LQ.

Dans le chapitre suivant, nous allons utiliser la commande LQ pour minimiser l'erreur lors de l'intégration numérique en ligne des équations différentielles ordinaires.

# Chapitre 4

## Correction des méthodes d'intégration numérique des équations différentielles

### 4.1 Introduction

Dans certaines techniques de commande, par exemple la commande prédictive, la commande adaptative à modèle de référence, et les techniques de commande basées sur l'optimisation de paramètres ou de fonctions, l'intégration des équations du modèle sur un horizon de commande généralement fini est indispensable. Nous avons montré au chapitre **3**, que les méthodes d'intégration numérique introduisent une erreur d'approximation qui s'accumule au cours de l'intégration pour devenir importante si l'horizon d'intégration est grand. Par conséquent, les performances des techniques de commande utilisant cette solution numérique peuvent se détériorer et devenir médiocres.

Lorsqu'une solution analytique du modèle est disponible, par exemple pour les systèmes linéaires et une classe de systèmes non linéaires, alors on peut penser à contrôler l'erreur d'approximation en utilisant une technique de commande. Dans ce chapitre, nous allons présenter le principe de correction des méthodes d'intégration numérique en utilisant une commande LQ. L'approche sera illustrée en considérant deux exemples de méthodes à savoir la méthode d'**Euler** (un seul pas) et la méthode d'**Adams-Bashforth** (à deux pas) présentées aux chapitre **2**.

### 4.2 Correction des méthodes d'intégration numérique

Dans ce chapitre, on s'intéresse à la correction en ligne des méthodes d'intégration numérique des équations différentielles ordinaires de la forme

$$\dot{x}(t) = f(x(t)) \quad (4.1)$$

$$x(0) = x_0 \quad (4.2)$$

en utilisant une commande LQ présenté dans le chapitre **3**. On suppose que l'équation différentielle ordinaire (4.1) admet une solution analytique notée  $x_a(t)$ , c'est-à-dire que la fonction  $f$  est Lipschitzienne sur l'intervalle de temps sur lequel on veut résoudre cette équation.

Pour illustrer, en général, le principe de la correction basée sur le correcteur LQ, on considère le cas d'une méthode numérique à un seul pas. Le principe reste valide même pour les méthodes à pas multiples. Dans le cas d'une méthode numérique (explicite) à un seul pas, la solution numérique, noté  $x_n(k)$  est obtenue en utilisant la formule de récurrence suivante

$$x_n(\Delta t(k+1)) = x_n(\Delta t k) + F(\Delta t, x_n(\Delta t k)) \quad (4.3)$$

où  $F(\Delta t, x_n(\Delta t k))$  est une fonction (incrément) qui dépend du pas de discrétisation  $\Delta t$  et de la solution  $x_n(\Delta t k)$  à l'instant  $\Delta t k$ . La forme de cette fonction dépend de la méthode numérique utilisée.

On définit l'erreur d'approximation comme la différence entre la solution numérique et la solution analytique, à l'instant  $\Delta t k$ , donnée comme suit

$$e(\Delta t k) = x_n(\Delta t k) - x_a(\Delta t k) \quad (4.4)$$

Dans ce qui suit, pour simplifier la présentation, on va omettre le pas d'intégration  $\Delta t$ . Ainsi, pour l'instant  $k+1$ , on a l'erreur suivante

$$e(k+1) = x_n(k+1) - x_a(k+1) \quad (4.5)$$

En substituant la solution numérique  $x_n(k+1)$  obtenue en utilisant l'équation **4.3** dans l'équation (4.5), on obtient

$$e(k+1) = x_n(k) + F(\Delta t, x_n(k)) - x_a(k+1) \quad (4.6)$$

qu'on peut manipuler mathématiquement dans le cas linéaire pour la mettre sous la forme suivante

$$e(k+1) = A e(k) + d(k+1) \quad (4.7)$$

où  $d(k+1)$  représente la perturbation correspondant à l'accumulation de l'erreur d'approximation. L'objectif de la commande LQ revient à réduire l'effet de la perturbation  $d$  dont la forme générale est

$$d(k+1) = G(x_n(k), x_a(k), \Delta t) \quad (4.8)$$

Ainsi, pour compenser l'effet de la perturbation  $d$ , on propose d'introduire une commande  $u(k)$  pour corriger le système d'erreur donnée par l'équation (4.7) comme suit

$$e(k+1) = A e(k) + B u(k) + d(k+1) \quad (4.9)$$

Par conséquent, le problème de la correction de l'erreur d'approximation  $e(k)$  revient à déterminer une commande  $u(k)$  qui permet de compenser l'effet de la perturbation  $d$ . Du point de vue automatique, on va concevoir une commande  $u(k)$  qui force l'erreur  $e(k+1)$  à être proche de zéro le long de l'intervalle de résolution de l'équation différentielle ordinaire (4.1).

Pour réaliser cet objectif, on propose d'utiliser la commande LQ, pour concevoir un retour d'état de la forme

$$u(k) = K(k) e(k) \quad (4.10)$$

ce qui donne en boucle fermée le système d'erreur suivant :

$$e(k+1) = (A + B K(k)) e(k) + d(k+1) \quad (4.11)$$

Ainsi, en choisissant un  $K(k)$  qui donne une matrice d'état  $A + B K(k)$  stable permet de garantir que le système est stable en boucle fermée, c'est-à-dire lorsque  $k$  tend vers l'infini, l'erreur  $e(k)$  soit borné et proche de zéro. Si et comme la perturbation  $d(k)$  dépend de l'erreur  $e(k)$ , alors son effet devient insignifiant quand  $k$  tend vers l'infini.

Pour réaliser cet objectif, on propose de concevoir un retour d'état de la forme (4.10) qui permet de minimiser le critère suivant

$$J(u(k), k) = \sum_{i=1}^N e^T(k) Q e(k) + u^T(k-1) R u(k-1) \quad (4.12)$$

Ainsi, le problème revient à concevoir une commande LQ dont le principe et la synthèse sont présentés dans le chapitre 3.

Dans la section suivante, on va appliquer l'approche présentée pour corriger deux

méthodes de résolution des équations différentielles ordinaires, en l'occurrence la méthode d'**Euler** (à un seul pas) et la méthode d'**Adams-Bashforth** (à deux pas).

## 4.3 Exemples d'applications

Dans cette section, nous allons appliquer la commande LQ pour corriger les méthodes d'**Euler** et d'**Adams-Bashforth** présentées au chapitre 2. La méthode d'**Euler** est une méthode à un seul pas, et la méthode d'**Adams-Bashforth** est une méthode à deux pas.

### 4.3.1 Méthode d'Euler

Pour illustrer l'approche de correction présentée dans la section précédente, on considère l'équation différentielle ordinaire du premier ordre suivante

$$\dot{x}(t) = \lambda x(t) \quad (4.13)$$

$$x(0) = 1 \quad (4.14)$$

La solution analytique de cette équation est

$$x_a(t) = x(0) e^{\lambda t} \quad (4.15)$$

La discrétisation (en mettant  $t = \Delta t k$ ) de la solution analytique (4.15) donne

$$x_a(k) = x(0) e^{\lambda \Delta t k} \quad (4.16)$$

La solution numérique de l'équation différentielle ordinaire (4.13), avec la condition initiale (4.14), en utilisant la méthode d'**Euler**, est obtenue en utilisant l'équation de récurrence suivante :

$$x_n(k) = (1 + \lambda \Delta t) x_n(k-1) \quad (4.17)$$

L'erreur d'approximation définie comme la différence entre la solution analytique (4.16) et la solution numérique (4.17) est donnée comme suit

$$\begin{aligned} e(k) &= x_n(k) - x_a(k) \\ &= (1 + \lambda \Delta t) x_n(k-1) - e^{\lambda \Delta t k} \end{aligned} \quad (4.18)$$

Pour l'instant  $k + 1$ , on a

$$e(k+1) = (1 + \lambda \Delta t) x_n(k) - e^{\lambda \Delta t(k+1)} \quad (4.19)$$

Pour exprimer  $e(k+1)$  en fonction de  $e(k)$ , on propose d'ajouter et de soustraire le terme  $e^{\lambda \Delta t k}$  au terme à gauche de l'équation (4.19) comme suit :

$$e(k+1) = x_n(k) - e^{\lambda \Delta t k} + e^{\lambda \Delta t k} + \lambda \Delta t x(k) - e^{\lambda \Delta t(k+1)} \quad (4.20)$$

et comme  $e(k) = x_n(k) - e^{\lambda \Delta t k}$ , alors

$$e(k+1) = e(k) + e^{\lambda \Delta t k} + \lambda \Delta t x_n(k) - e^{\lambda \Delta t(k+1)} \quad (4.21)$$

En posant

$$d(k) = e^{\lambda \Delta t k} + \lambda \Delta t x_n(k) - e^{\lambda \Delta t(k+1)} \quad (4.22)$$

L'équation (4.21) prend la forme suivante

$$e(k+1) = e(k) + d(k+1) \quad (4.23)$$

Ainsi, pour corriger cette erreur d'approximation, on doit utiliser la commande  $u(k)$  comme suit :

$$e(k+1) = e(k) + u(k) + d(k+1) \quad (4.24)$$

En identifiant l'équation d'état (4.24) avec l'équation (4.9), on obtient les matrices d'état et de commande suivantes

$$\begin{aligned} A &= 1 \\ B &= 1 \end{aligned}$$

Pour déterminer le correcteur LQ, on considère la minimisation du critère quadratique (4.12) avec les matrices de pondérations suivantes

$$\begin{aligned} Q &= 1 \\ R &= 1 \end{aligned}$$

Ainsi, en utilisant les résultats du chapitre précédent, le correcteur LQ est donné comme suit :

$$u(k) = K(k) e(k) \tag{4.25}$$

$$K(k) = -\frac{1 + P(k+1)}{2 + P(k+1)} \tag{4.26}$$

$$P(k) = \frac{(1 + P(k+1))^3}{2 + P(k+1)} \tag{4.27}$$

$$P(N) = 0 \tag{4.28}$$

Pour montrer l'apport du correcteur LQ dans l'amélioration de la précision de la solution (ou de l'approximation de la solution), on va comparer l'erreur d'approximation  $e(k)$  avec et sans correction pour un pas de temps  $h = 0.1$  avec plusieurs temps de simulation en prenant  $\lambda = 1$ . Les résultats obtenus sont donnés par les Figures 4.1, 4.2 et 4.3.

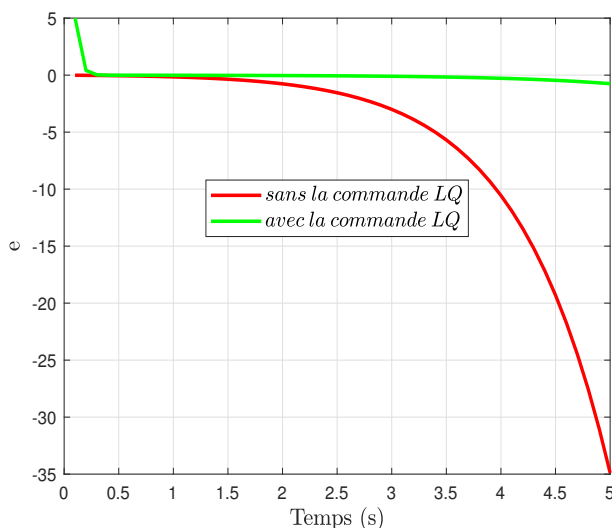


FIGURE 4.1 – Évolution de l'erreur dans le cas de la méthode d'**Euler** pour  $t_f = 5$

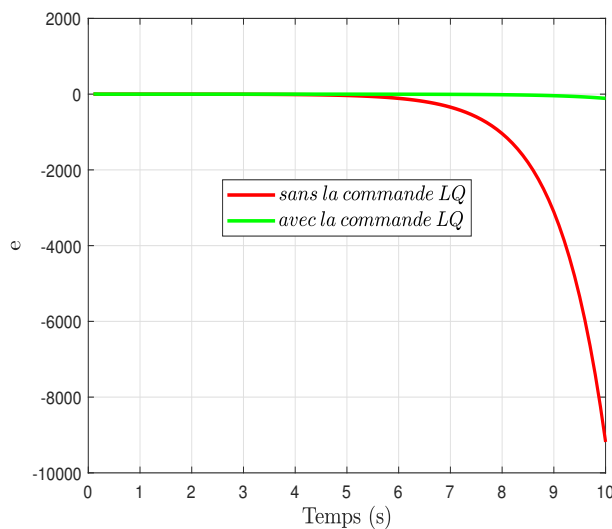


FIGURE 4.2 – Évolution de l'erreur dans le cas de la méthode d'**Euler** pour  $t_f = 10$

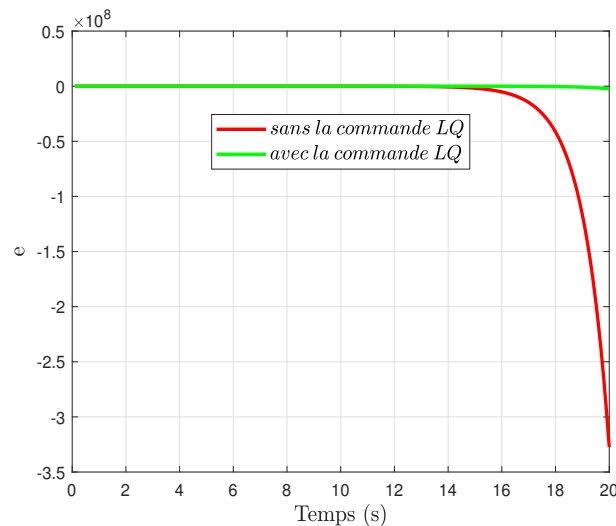


FIGURE 4.3 – Évolution de l'erreur dans le cas de la méthode d'**Euler** pour  $t_f = 20$

On remarque clairement que plus le temps de simulation est grand, plus l'erreur d'approximation sans correction devient importante. Contrairement à l'erreur d'approximation avec correction qui est borné et proche de zéro. En conclusion, la commande LQ permet de contrôler effectivement l'erreur d'approximation en la forçant à être très proche de zéro.

### 4.3.2 Méthode d'Adams-Bashforth

Pour illustrer la correction de la méthode d'**Adams-bashforth**, on reprend l'équation différentielle ordinaire (4.1) dont la solution analytique  $x_a(t)$  discrétisée est donnée par l'équation (4.16). En utilisant la méthode d'**Adams-Bashforth**, la solution numérique  $x_n(t)$  est obtenue par la formule de récurrence (4.16), c'est-à-dire

$$x_n(k+2) = x_n(k+1) + \frac{\Delta t}{2} [3\lambda x_n(k+1) - \lambda x_n(k)] \quad (4.29)$$

Pour corriger l'erreur d'approximation, on considère le système dynamique discret équivalent suivant :

$$x_n(k+2) - x_n(k+1) = \frac{3}{2}(u(k+1) - u(k)) \quad (4.30)$$

et en appliquant la transformée en  $Z$ , on obtient

$$(z^2 - z) X_n(z) = \left(\frac{3}{2}z - \frac{1}{2}\right) U(z) \quad (4.31)$$

Par conséquent la fonction de transfert discrète entre la solution analytique  $X_n(k)$  et  $U(z)$  est

$$\frac{X_n(z)}{U(z)} = \frac{\frac{3}{2}z - \frac{1}{2}}{z(z-1)} \quad (4.32)$$

dont la réalisation minimale (commandable et observable) est donnée comme suit

$$w(k+1) = Aw(k) + Bu(k) \quad (4.33)$$

$$x_n(k) = Cw(k) \quad (4.34)$$

avec

$$A = \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 1 \end{bmatrix}, \quad C = \begin{bmatrix} -\frac{1}{2} \\ \frac{3}{2} \end{bmatrix}$$

Pour corriger le système (4.33) et (4.34), on propose d'utiliser la loi de commande basée sur un retour de sortie sous forme linéaire suivante

$$u(k) = \lambda \Delta t x_n(k) + v(k) \quad (4.35)$$

et en utilisant l'équation de sortie (4.34), on a

$$u(k) = \lambda \Delta t C w(k) + v(k) \quad (4.36)$$

Le problème revient à chercher la commande  $v(k)$  qui permet de réduire l'erreur d'approximation et d'améliorer la précision de la méthode d'intégration numérique d'**Adams-Bashforth**.

Le système en boucle fermée (système(4.12) (4.24) commandé par (4.36)) est donnée comme suit :

$$w(k+1) = \bar{A}w(k) + Bv(k) \quad (4.37)$$

$$x_n(k) = Cw(k) \quad (4.38)$$

avec

$$\bar{A} = A + \lambda \Delta t BC \quad (4.39)$$

Introduisons maintenant l'erreur d'approximation  $e(k)$  qui est définie comme la différence entre la solution numérique  $x_n(k)$  et la solution analytique (4.15), c'est-à-dire

$$e(k) = x_n(k) - e^{\lambda \Delta t k} \quad (4.40)$$

et en utilisant l'équation (4.38), il vient

$$e(k) = C w(k) - e^{\lambda \Delta t k} \quad (4.41)$$

Ainsi, pour les instants  $k + 1$  et  $k + 2$ , on a

$$e(k + 1) = C w(k + 1) - e^{\lambda \Delta t (k+1)} = C \bar{A} w(k) + C B v(k) - e^{\lambda \Delta t (k+1)} \quad (4.42)$$

$$e(k + 2) = C w(k + 2) - e^{\lambda \Delta t (k+2)} = C \bar{A}^2 w(k) + C \bar{A} w(k) + C B v(k) - e^{\lambda \Delta t (k+2)} \quad (4.43)$$

avec

$$\bar{A} = \begin{bmatrix} 0 & -a_0 \\ 1 & -a_1 \end{bmatrix} \quad (4.44)$$

Comme la matrice  $\bar{A}$  est carrée d'ordre 2, et son polynôme caractéristique est

$$P(Z) = |Z I_2 - \bar{A}| = Z^2 + a_1 Z + a_0 \quad (4.45)$$

alors d'après le théorème de **Cayley-Hamilton** [2], en remplaçant formellement  $Z$  par la matrice  $\bar{A}$  dans le polynôme  $P(z)$ , le résultat est une matrice nulle  $0_2$ , c'est-à-dire

$$\bar{A}^2 + a_1 \bar{A} + a_0 I_2 = 0_2 \quad (4.46)$$

Ainsi,

$$\begin{aligned} a_0 e(k) + a_1 e(k + 1) + e(k + 2) &= C (\bar{A}^2 + a_1 \bar{A} + a_0 I_2) w(k) + (a_1 C B + C \bar{A} B) v(k) \\ &\quad + C B v(k + 1) - (a_0 + a_1 e^{\lambda \Delta t} + e^{2\lambda \Delta t}) e^{\lambda \Delta t k} \end{aligned} \quad (4.47)$$

et en tenant compte de l'équation (4.46), il vient

$$\begin{aligned} a_0 e(k) + a_1 e(k + 1) + e(k + 2) &= (a_1 C B + C \bar{A} B) v(k) + C B v(k + 1) \\ &\quad - (a_0 + a_1 e^{\lambda \Delta t} + e^{2\lambda \Delta t}) e^{\lambda \Delta t k} \end{aligned} \quad (4.48)$$

Ainsi, en utilisant la transformée en  $Z$ , on obtient

$$E(z) = \frac{(a_1 C B + C \bar{A} B) + C B z}{a_0 + a_1 z + z^2} V(z) - \frac{1}{a_0 + a_1 z + z^2} \text{TZ}[(a_0 + a_1 e^{\lambda \Delta t} + e^{2\lambda \Delta t}) e^{\lambda \Delta t k}] \quad (4.49)$$

ou sous forme d'état

$$X(k+1) = A_e X(k) + \bar{B} v(k) + d(k) \quad (4.50)$$

$$e(k) = C X(k) \quad (4.51)$$

avec

$$\bar{A} = \begin{bmatrix} 0 & -a_0 \\ 1 & -a_1 \end{bmatrix}, \quad \bar{B} = \begin{bmatrix} a_1 C B + C \bar{A} B \\ C B \end{bmatrix}, \quad \bar{C} = \begin{bmatrix} 0 & 1 \end{bmatrix}$$

et la perturbation

$$d(k) = (a_0 + a_1 e^{\lambda \Delta t} + e^{2\lambda \Delta t}) e^{\lambda \Delta t k} \quad (4.52)$$

Pour déterminer la commande  $v(k)$  qui permet de réduire l'effet de la perturbation et améliorer la précision de la méthode d'intégration numérique, on propose d'utiliser la commande LQ qui permet de minimiser le critère suivant

$$J(u(k), k) = \sum_{i=1}^N e^T(k) Q e(k) + v^T(k-1) R v(k-1) \quad (4.53)$$

avec

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad R = 1$$

Ainsi, en utilisant les résultats du chapitre précédent, le correcteur LQ est donné comme suit :

$$v(k) = K(k) e(k) \quad (4.54)$$

$$K(k) = \begin{bmatrix} k_1 & k_2 \end{bmatrix} \quad (4.55)$$

$$P(k) = \begin{bmatrix} p_1(k) & p_2(k) \\ p_2(k) & p_3(k) \end{bmatrix} \quad (4.56)$$

$$P(N) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (4.57)$$

avec

$$k_1(k) = \frac{2(k_{01} + k_{02} + k_{03} - k_{04} + k_{05} - k_{06} + k_{07} + k_{08} - k_{09})}{k_{10} + k_{11} + k_{12} + k_{13} - k_{14} + k_{15} - k_{16} + k_{17} + k_{18} + k_{19} - k_{20}} \quad (4.58)$$

et

$$k_{01} = 3a_0 + 5a_1 + 3p_1(k+1) - 18p_2(k+1) + 27p_3(k+1) \quad (4.59)$$

$$k_{02} = 18a_0a_1 + 3a_0p_1(k+1) + 4a_0p_2(k+1) + a_1p_1(k+1) \quad (4.60)$$

$$k_{03} = 3a_0p_3(k+1) + 5a_1p_3(k+1) + 39a_0^2p_1(k+1) \quad (4.61)$$

$$k_{04} = 3a_0p_2^2(k+1) + 3a_1^2p_1(k+1) - a_1p_2^2(k+1) \quad (4.62)$$

$$k_{05} = 27a_0^2p_3(k+1) + 15a_1^2p_3(k+1) + 39a_0^2 + 15a_1^2 \quad (4.63)$$

$$k_{06} = 27a_0^2p_2^2(k+1) - 3a_1^2p_2^2(k+1) + 27a_0^2p_1(k+1)p_3(k+1) \quad (4.64)$$

$$k_{07} = 3a_1^2p_1(k+1)p_3(k+1) + 18a_0a_1p_1(k+1) + 24a_0a_1(k+1)p_2(k+1) \quad (4.65)$$

$$k_{08} = 18a_0a_1p_3(k+1) + 3a_0p_1(k+1)p_3(k+1) + a_1p_1(k+1)p_3(k+1) \quad (4.66)$$

$$k_{09} = 18a_0a_1p_2^2(k+1) + 18a_0a_1p_1(k+1)p_3(k+1) + 42 \quad (4.67)$$

$$k_{10} = 18a_0 + 30a_1 + 15p_1(k+1) - 84p_2(k+1) + 131p_3(k+1) \quad (4.68)$$

$$k_{11} = 54a_0a_1 + 18a_0p_1(k+1) + 24a_0p_2(k+1) + 6a_1p_1(k+1) \quad (4.69)$$

$$k_{12} = 18a_0p_3(k+1) + 30a_1p_3(k+1) + p_1(k+1)p_3(k+1) \quad (4.70)$$

$$k_{13} = 117a_0^2p_1(k+1) - 18a_0p_2^2(k+1) + 9a_1^2p_1(k+1) \quad (4.71)$$

$$k_{14} = 6a_1p_2^2(k+1) + 81a_0^2p_3(k+1) + 45a_1^2p_3(k+1) \quad (4.72)$$

$$k_{15} = 117a_0^2 + 45a_1^2 - p_2^2(k+1) - 81a_0^2p_2^2(k+1) \quad (4.73)$$

$$k_{16} = 9a_1^2p_2^2(k+1) + 81a_0^2p_1(k+1)p_3(k+1) \quad (4.74)$$

$$k_{17} = 9a_1^2p_1(k+1)p_3(k+1) + 54a_0a_1p_1(k+1) \quad (4.75)$$

$$k_{18} = 72a_0a_1p_2(k+1) + 54a_0a_1p_3(k+1) \quad (4.76)$$

$$k_{19} = 18a_0p_1(k+1)p_3(k+1) + 6a_1p_1(k+1)p_3(k+1) \quad (4.77)$$

$$k_{20} = 54a_0a_1p_2^2(k+1) + 54a_0a_1p_1(k+1)p_3(k+1) + 201 \quad (4.78)$$

De même,  $k_2(k)$  est défini par :

$$k_2(k) = \frac{2N_{k_2}}{k_{35} + k_{36} + k_{37} - k_{38} + k_{39} + k_{40} + k_{41} + k_{42} + k_{43} - k_{44}} \quad (4.79)$$

$$N_{k2} = k_{21} + k_{22} + k_{23} - k_{24} + k_{25} + k_{26} + k_{27} - k_{28} + k_{29} - k_{30} - k_{31} + k_{32} + k_{33} + k_{34} \quad (4.80)$$

avec

$$k_{21} = 42 a_1 - 19 a_0 - 12 a_0 a_1 - 2 a_0 p_1 (k + 1) + 6 a_0 p_2 (k + 1) \quad (4.81)$$

$$k_{22} = 3 a_1 p_1 (k + 1) - 14 a_0 p_3 (k + 1) - 18 a_1 p_2 (k + 1) \quad (4.82)$$

$$k_{23} = 27 a_1 p_3 (k + 1) + 18 a_0 a_1^2 + 39 a_0^2 a_1 \quad (4.83)$$

$$k_{24} = 9 a_0^2 p_1 (k + 1) + a_0 p_2^2 (k + 1) - 12 a_0^2 p_2 (k + 1) \quad (4.84)$$

$$k_{25} = a_1^2 p_1 (k + 1) - 9 a_0^2 p_3 (k + 1) + 3 a_1^3 p_1 (k + 1) \quad (4.85)$$

$$k_{26} = 5 a_1^2 p_3 (k + 1) + 15 a_1^3 p_3 (k + 1) - 9 a_0^2 + 5 a_1^2 \quad (4.86)$$

$$k_{27} = 15 a_1^3 + 9 a_0^2 p_2^2 - a_1^2 p_2^2 (k + 1) - 3 a_1^3 p_2^2 (k + 1) \quad (4.87)$$

$$k_{28} = 9 a_0^2 p_1 (k + 1) p_3 (k + 1) + a_1^2 p_1 (k + 1) p_3 (k + 1) \quad (4.88)$$

$$k_{29} = 3 a_1^3 p_1 (k + 1) p_3 (k + 1) - 18 a_0 a_1^2 p_2^2 (k + 1) \quad (4.89)$$

$$k_{30} = 27 a_0^2 a_1 p_2^2 (k + 1) + 4 a_0 a_1 p_2 (k + 1) - 12 a_0 a_1 p_3 (k + 1) \quad (4.90)$$

$$k_{31} = a_0 p_1 (k + 1) p_3 (k + 1) + 18 a_0 a_1^2 p_1 (k + 1) \quad (4.91)$$

$$k_{32} = 39 a_0^2 a_1 p_1 (k + 1) + 24 a_0 a_1^2 p_2 (k + 1) \quad (4.92)$$

$$k_{33} = 18 a_0 a_1^2 p_3 (k + 1) + 27 a_0^2 a_1 p_3 (k + 1) \quad (4.93)$$

$$k_{34} = 18 a_0 a_1^2 p_1 (k + 1) p_3 (k + 1) + 27 a_0^2 a_1 p_1 (k + 1) p_3 (k + 1) \quad (4.94)$$

$$k_{35} = 18 a_0 + 30 a_1 + 15 p_1 (k + 1) - 84 p_2 (k + 1) + 131 p_3 (k + 1) + 54 a_0 a_1 \quad (4.95)$$

$$k_{36} = 18 a_0 p_1 (k + 1) + 24 a_0 p_2 (k + 1) + 6 a_1 p_1 (k + 1) + 18 a_0 p_3 (k + 1) \quad (4.96)$$

$$k_{37} = 30 a_1 p_3 (k + 1) + p_1 (k + 1) p_3 (k + 1) + 117 a_0^2 p_1 (k + 1) \quad (4.97)$$

$$k_{38} = 18 a_0 p_2^2 (k + 1) + 9 a_1^2 p_1 (k + 1) - 6 a_1 p_2^2 (k + 1) \quad (4.98)$$

$$k_{39} = 81 a_0^2 p_3 (k + 1) + 45 a_1^2 p_3 (k + 1) + 117 a_0^2 \quad (4.99)$$

$$k_{40} = 45 a_1^2 - p_2^2 (k + 1) - 81 a_0^2 p_2^2 (k + 1) - 9 a_1^2 p_2^2 (k + 1) \quad (4.100)$$

$$k_{41} = 81 a_0^2 p_1 (k + 1) p_3 (k + 1) + 9 a_1^2 p_1 (k + 1) p_3 (k + 1) \quad (4.101)$$

$$k_{42} = 54 a_0 a_1 p_1 (k + 1) + 72 a_0 a_1 p_2 (k + 1) + 54 a_0 a_1 p_3 (k + 1) \quad (4.102)$$

$$k_{43} = 18 a_0 p_1 (k + 1) p_3 (k + 1) + 6 a_1 p_1 (k + 1) p_3 (k + 1) \quad (4.103)$$

$$k_{44} = 54 a_0 a_1 p_2^2 (k + 1) + 54 a_0 a_1 p_1 (k + 1) p_3 (k + 1) + 201 \quad (4.104)$$

Les termes  $p_1(k)$ ,  $p_2(k)$  et  $p_3(k)$  sont calculés comme suit :

$$p_1(k) = \frac{N_{p1}}{D_{p1}} \quad (4.105)$$

avec

$$N_{p1} = -p_2^2(k+1) + p_1(k+1) + 5p_3(k+1) + p_1(k+1)p_3(k+1) + 5 \quad (4.106)$$

$$D_{p1} = p_1(k+1) - 6p_2(k+1) + 9p_3(k+1) + 14 \quad (4.107)$$

$$p_2(k) = -\frac{p_{11} + p_{12} - p_{13} + p_{14}}{p_{15}} \quad (4.108)$$

$$p_{11} = 3a_0 + 5a_1 + 3a_0p_1(k+1) + 4a_0p_2(k+1) \quad (4.109)$$

$$p_{12} = a_1p_1(k+1) + 3a_0p_3(k+1) + 5a_1p_3(k+1) \quad (4.110)$$

$$p_{13} = 3a_0p_2^2(k+1) - a_1p_2^2(k+1) \quad (4.111)$$

$$p_{14} = 3a_0p_1(k+1)p_3(k+1) + a_1p_1(k+1)p_3(k+1) \quad (4.112)$$

$$p_{15} = p_1(k+1) - 6p_2(k+1) + 9p_3(k+1) + 14 \quad (4.113)$$

$$p_3(k) = -\frac{p_{16} + p_{17} - p_{18} + p_{19} + p_{20} + p_{21}}{p_{22}} \quad (4.114)$$

$$p_{16} = 6a_0a_1 + 13a_0^2p_1(k+1) + a_1^2p_1(k+1) + 9a_0^2p_3(k+1) \quad (4.115)$$

$$p_{17} = 5a_1^2p_3(k+1) + 13a_0^2 + 5a_1^2 - 9a_0^2p_2^2(k+1) \quad (4.116)$$

$$p_{18} = a_1^2p_2^2(k+1) + 9a_0^2p_1(k+1)p_3(k+1) \quad (4.117)$$

$$p_{19} = a_1^2p_1(k+1)p_3(k+1) + 6a_0a_1p_1(k+1) \quad (4.118)$$

$$p_{20} = 8a_0a_1p_2(k+1) + 6a_0a_1p_3(k+1) - 6a_0a_1p_2^2(k+1) \quad (4.119)$$

$$p_{21} = 6a_0a_1p_1(k+1)p_3(k+1) \quad (4.120)$$

$$p_{22} = p_1(k+1) - 6p_2(k+1) + 9p_3(k+1) + 14 \quad (4.121)$$

Pour montrer l'apport du correcteur LQ dans l'amélioration de la précision de la solution (ou de l'approximation de la solution), on va comparer l'erreur d'approximation  $e(k)$  avec et sans correction pour un pas de temps  $h = 0.1$  avec plusieurs temps de simulation en prenant  $\lambda = 1$ . Les résultats obtenus sont donnés par les Figures **cp4.4**, **4.5** et **4.6**.

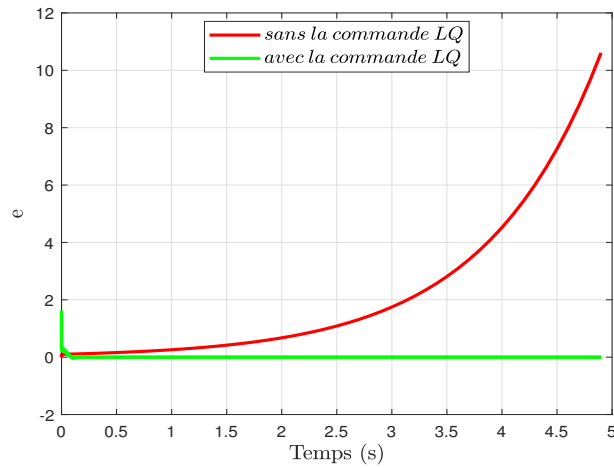


FIGURE 4.4 – Évolution de l'erreur dans le cas de la méthode d'Adams-Bashforth pour  $t_f = 5$

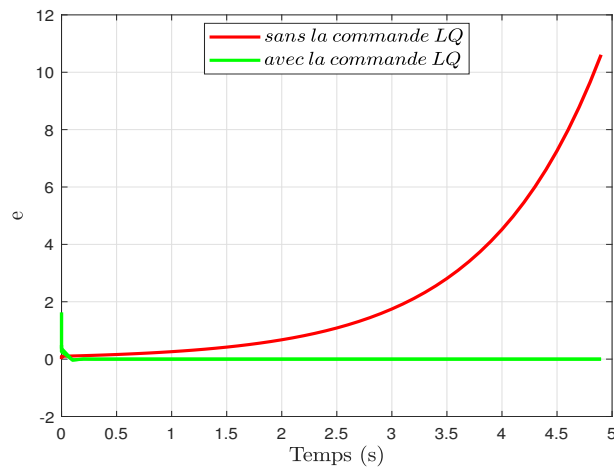


FIGURE 4.5 – Évolution de l'erreur dans le cas de la méthode d'Adams-Bashforth pour  $t_f = 10$

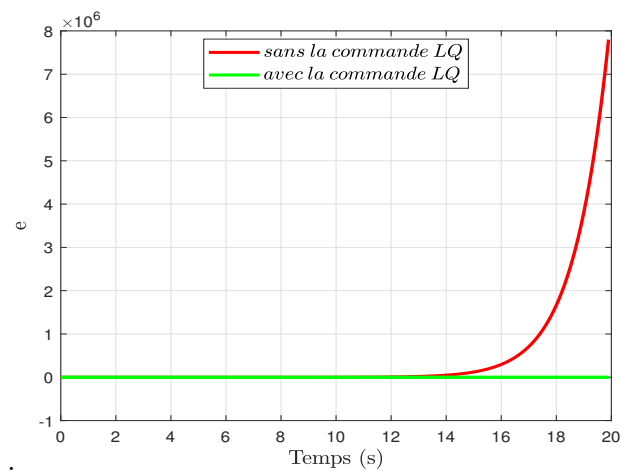


FIGURE 4.6 – Évolution de l'erreur dans le cas de la méthode d'Adams-Bashforth pour  $t_f = 20$

Les résultats obtenus montrent clairement que l'erreur d'approximation sans correction diverge. Contrairement à l'erreur d'approximation avec correction qui converge rapidement vers le zéro. On peut conclure que la commande LQ synthétisée permet de contrôler l'erreur d'approximation en la forçant à converger rapidement vers zéro.

## 4.4 Conclusion

Dans ce chapitre, nous avons implémenté la commande LQ pour améliorer la précision des méthodes d'intégration numérique des équations différentielles ordinaires. Nous avons formulé le problème sous forme d'un problème de commande LQ dont le modèle est l'équation d'évolution de l'erreur entre la solution analytique et la solution numérique. Puis pour contrôler, l'erreur d'approximation et de la forcer à être très proches de zéro, nous avons considéré un critère de performances dont l'objectif consiste à minimiser l'intégrale du carré de cette erreur tout en réduisant l'amplitude de l'action correctrice (terme qui permet de contrôler l'erreur).

Le problème de commande LQ a été résolu en utilisant la programmation dynamique basée sur le principe fonctionnelle de **Bellman**. La commande obtenue est un retour d'état linéaire. Deux exemples d'application ont été présentés, ils concernent la méthode d'**Euler** (un seul pas) et la méthode d'**Adams-Bashforth**. Les résultats de simulation obtenus ont montré que la correction des méthodes numériques en utilisant le correcteur LQ permet d'améliorer davantage la précision comparativement à l'implémentation de la méthode sans correction.

# Conclusion générale

Le travail présenté dans ce mémoire s'inscrit dans le cadre de l'amélioration de la précision des méthodes d'intégration numérique des équations différentielles ordinaires. Il est axé principalement sur l'application de la commande LQ pour minimiser les erreurs d'approximation des méthodes explicites à un seul pas et à pas multiples dans la résolution en ligne des équations différentielles ordinaires.

Pour atteindre cet objectif, nous avons structuré notre travail en plusieurs parties. Nous avons tout d'abord rappelé les concepts fondamentaux liés à la modélisation des systèmes dynamiques par des équations différentielles ordinaires. Ensuite, nous avons présenté le principe des méthodes d'intégration numérique explicites tout en mettant en évidence leurs avantages et leurs limites. Après, nous avons introduit le problème de la commande LQ, et sa résolution en utilisant la programmation dynamique basée sur le principe d'optimalité de **Bellman** [3, 9]. L'intérêt de la commande LQ réside dans le fait que le correcteur est conçu de sorte à minimiser l'écart entre la sortie réelle (observée) et celle désirée. Par la suite, nous avons exploité cette technique de commande pour concevoir un correcteur capable de réduire l'écart entre la solution numérique et la solution analytique d'une équation différentielle ordinaire ; ce qui permet d'améliorer davantage la précision et réduire l'accumulation des erreurs d'approximation. À la fin, nous avons illustré l'apport de la commande LQ à travers des exemples d'applications, en occurrence la correction des méthodes d'**Euler** et d'**Adams-Bashforth**.

Les résultats de simulation obtenus montrent clairement que l'utilisation de la théorie de commande pour contrôler l'erreur d'approximation d'une méthode d'intégration numérique est une idée très intéressante et prometteuse. En effet l'utilisation de la commande LQ a permis de réduire considérablement l'erreur d'approximation, même en utilisant un pas d'intégration relativement grand, et d'améliorer la précision de la solution numérique. La qualité de la solution dépend de la méthode d'intégration numérique utilisée. En effet, les méthodes à pas multiples (cas de la méthode d'**Adams-Bashforth**) sont plus précises par rapport aux méthodes à un seul pas (cas de la méthode d'**Euler**) pas mais nécessitent un développement mathématique plus poussé pour formuler le problème de commande optimale.

L'application de la commande LQ pour corriger les méthodes d'intégration numérique des équations différentielles ordinaires ouvre plusieurs perspectives qui méritent d'être explorées. Ainsi, l'approche de synthèse d'une commande LQ présentée dans ce mémoire peut être étendue pour développer des lois de commande pour d'autres méthodes d'intégration numérique. Aussi, il est intéressant de penser à explorer le cas des méthodes d'intégration numérique implicites.

# Bibliographie

- [1] **J. M. Flaus.** *La régulation industrielle : régulateurs PID, prédictifs et flous.* Édition Hermes Science Publications, Paris, 2000.
- [2] **J.-P. Corriou.** *Méthodes numériques et optimisation : Théorie et pratique pour l'ingénieur.* Édition Ellipses, Paris, 2022.
- [3] **P. Naslin.** *Théorie de la commande et conduite optimale.* Édition Dunod, Paris, 1969.
- [4] **N. Piskounov.** *Calcul différentiel et intégral.* Édition MIR, Moscou, 1980.
- [5] **A. Fortin.** *Analyse numérique pour ingénieurs.* Presses Internationales Polytechnique, Montreal, 1995.
- [6] **R. M. Faye et F. Mora-Camino.** *La commande optimale : Approche variationnelle.* Édition Harmattan, Paris, 2017.
- [7] **M. Vergé et D. Jaume.** *Modélisation structure des systèmes avec les Bond Graphs.* Édition Technip, Paris, 2003.
- [8] **J. J. DiStefano.** *Dynamic Systems Biology Modeling and Simulation.* Academic Press Edition, San Diego, 2015.
- [9] **B. D. O. Anderson and J. B. Moore.** *Linear Optimal Control.* Prentice Hall Edition, New Jersey, 1971.