



HAL
open science

Développement de modèles réduits adaptatifs pour le contrôle optimal des écoulements

Mourad Oulghelou

► **To cite this version:**

Mourad Oulghelou. Développement de modèles réduits adaptatifs pour le contrôle optimal des écoulements. Mécanique des fluides [physics.class-ph]. Université de La Rochelle, 2018. Français. NNT : 2018LAROS014 . tel-02009844

HAL Id: tel-02009844

<https://theses.hal.science/tel-02009844>

Submitted on 6 Feb 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNIVERSITÉ DE LA ROCHELLE

ÉCOLE DOCTORALE EUCLIDE

LABORATOIRE DES SCIENCES DE L'INGÉNIEUR POUR L'ENVIRONNEMENT (LaSIE)

THÈSE

présentée par

MOURAD OULGHELOU

pour obtenir le titre de : DOCTEUR DE L'UNIVERSITÉ DE LA ROCHELLE

Discipline : « MÉCANIQUE DES FLUIDES »

Développement de modèles réduits adaptatifs pour le contrôle optimal des écoulements

Thèse dirigée par CYRILLE ALLERY

soutenue le 26 Juin 2018

Jury :

MEJDI AZAIEZ	Professeur	Institut Polytechnique de Bordeaux	(Rapporteur)
LAURENT CORDIER	Chargé de recherche CNRS - HDR	Insitut PPRIME (Poitiers)	(Rapporteur)
CHRISTOPHE AIRIAU	Professeur	Université de Toulouse	(Examineur)
AMINE AMMAR	Professeur	ENSAM Angers	(Examineur)
AZIZ HAMDOUNI	Professeur	Université de La Rochelle	(Examineur)
ANTOINE PLACZEK	Ingénieur de Recherche	ONERA	(Examineur)
TOMÁS CHACÓN	Professeur	Université de Seville	(Examineur)
CYRILLE ALLERY	Maître de Conférences - HDR	Université de La Rochelle	(Directeur de Thèse)

Développement de de modèles réduits adaptatifs pour le contrôle optimal des écoulements

Résumé

La résolution des problèmes de contrôle optimal nécessite des temps de calcul et des capacités de stockage très élevés. Pour s'affranchir de ces contraintes, il est possible d'utiliser les méthodes de réduction de modèles comme la POD (Proper Orthogonal Decomposition). L'inconvénient de cette approche est que la base POD n'est valable que pour des paramètres situés dans un voisinage proche des paramètres pour lesquels elle a été construite. Par conséquent, en contrôle optimal, cette base peut ne pas être représentative de tous les paramètres qui seront proposés par l'algorithme de contrôle. Pour s'affranchir de cet handicap, une méthodologie de contrôle optimal utilisant des modèles réduits adaptatifs a été proposée dans ce manuscrit. Les bases réduites adaptées sont obtenues à l'aide de la méthode d'interpolation ITSGM (Interpolation on Tangent Subspace of Grassman Manifold) ou de la méthode d'enrichissement PGD (Proper Generalized Decomposition). La robustesse de cette approche en termes de précision et de temps de calcul a été démontrée pour le contrôle optimal (basé sur les équations adjointes) des équations 2D de réaction-diffusion et de Burgers. L'approche basée sur l'interpolation ITSGM a également été appliquée avec succès pour contrôler l'écoulement autour d'un cylindre 2D. Deux méthodes de réduction non intrusives, ne nécessitant pas la connaissance des équations du modèle étudié, ont également été proposées. Ces méthodes appelées NIMR (Non Intrusive Model Reduction) et HNIMR (Hyper Non Intrusive Model Reduction) ont été couplées à un algorithme génétique pour résoudre rapidement un problème de contrôle optimal. Le problème du contrôle optimal de l'écoulement autour d'un cylindre 2D a été étudié et les résultats ont montré l'efficacité de cette approche. En effet, l'algorithme génétique couplé avec la méthode HNIMR a permis d'obtenir les solutions avec une bonne précision en moins de 40 secondes.

Mots clefs : Modèles d'ordre réduit, Décomposition Orthogonale aux valeurs propres (POD), Proper Generalized Decomposition (PGD), contrôle optimal, interpolation de bases, Algorithme génétique.

Development of adaptive reduced order models for optimal flow control

Abstract

The numerical resolution of adjoint based optimal control problems requires high computational time and storage capacities. In order to get over these high requirement, it is possible to use model reduction techniques such as POD (Proper Orthogonal Decomposition). The disadvantage of this approach is that the POD basis is valid only for parameters located in a small neighborhood to the parameters for which it was built. Therefore, this basis may not be representative for all parameters in the optimizer's path eventually suggested by the optimal control loop. To overcome this issue, a reduced optimal control methodology using adaptive reduced order models obtained by the ITSGM (Interpolation on a Tangent Subspace of the Grassman Manifold) method or by the PGD (Proper Generalized Decomposition) method, has been proposed in this work. The robustness of this approach in terms of precision and computation time has been demonstrated for the optimal control (based on adjoint equations) of the 2D reaction-diffusion and Burgers equations. The interpolation method ITSGM has also been validated in the control of flow around a 2D cylinder. In the context of non intrusive model reduction, two non-intrusive reduction methods, which do not require knowledge of the equations of the studied model, have also been proposed. These methods called NIMR (Non-Intrusive Model Reduction) and HNIMR (Hyper Non-Intrusive Model Reduction) were developed and then coupled to a genetic algorithm in order to solve an optimal control problem in quasi-real time. The problem of optimal control of the flow around a 2D cylinder has been studied and the results have shown the effectiveness of this approach. Indeed, the genetic algorithm coupled with the HNIMR method allowed to obtain the solutions with a good accuracy in less than 40 seconds.

Keywords : Reduced order models, Proper Orthogonal Decomposition (POD), Proper Generalized Decomposition (PGD), Optimal control, Interpolation of reduced bases, Genetic algorithms.

Remerciements

Il me sera très difficile de remercier tout le monde car c'est grâce à l'aide de nombreuses personnes que j'ai pu mener cette thèse à son terme. Je voudrais tout d'abord remercier grandement mon directeur de thèse, Cyrille Allery, pour toute son aide. Je suis ravi d'avoir travaillé en sa compagnie car outre son appui scientifique, il a toujours été là pour me soutenir et me conseiller au cours de l'élaboration de cette thèse.

Je remercie également Laurent CORDIER et Mejdî AZAIEZ qui m'ont fait l'honneur d'être rapporteurs de ma thèse, ils ont pris le temps de lire mon manuscrit, de m'écouter et de discuter avec moi. Leurs remarques m'ont permis d'envisager mon travail sous un autre angle. Pour tout cela je les remercie.

Je remercie également Christophe AIRIAU, Amine AMMAR et Antoine PLACZEK pour l'honneur qu'ils me font d'être dans mon jury de thèse. Je tiens à remercier spécialement Aziz HAMDOUNI de sa participation à mon jury de thèse ainsi que pour toutes nos discussions et ses conseils qui m'ont accompagné tout au long de mon cursus de recherche.

Il m'est impossible d'oublier mes collègues du laboratoire et mes amis pour leur soutien direct ou indirect, ainsi que toutes les personnes avec qui j'ai partagé de bons moments pendant ces années de thèse.

Mes derniers remerciements vont à ma famille, mes frères et sœurs, et plus particulièrement mes parents qui ont fait de moi la personne que je suis aujourd'hui.

À vous tous, Merci ...

TABLE DES MATIÈRES

TABLE DES MATIÈRES	vii
1 OPTIMISATION SOUS CONTRAINTES	5
1.1 INTRODUCTION	6
1.2 OPTIMISATION SUR UN ENSEMBLE K D'UN ESPACE DE BANACH Z	6
1.2.1 Existence d'un minimum	7
1.2.2 Condition d'optimalité	7
1.2.3 Algorithme de descente lorsque $K = Z$	8
1.2.4 Algorithme de descente lorsque $K \subset Z$ est convexe fermé	12
1.2.5 Cas particulier de l'optimisation quadratique	12
1.3 OPTIMISATION SOUS CONTRAINTES : CONTRÔLE OPTIMAL	13
1.3.1 Position du problème	13
1.3.2 Calcul des directions de descente	14
1.3.3 Algorithme de contrôle optimal par l'approche des multiplicateurs de Lagrange	18
1.4 CONTRÔLE OPTIMAL DES ÉCOULEMENTS	19
1.4.1 Problème de Navier-Stokes	19
1.4.2 Formulation variationnelle des équations de Navier-Stokes :	20
1.4.3 Formulation du problème de contrôle optimal	21
1.5 CONCLUSION DU CHAPITRE	24
2 RÉDUCTION DE MODÈLES ET OPTIMISATION	25
2.1 INTRODUCTION	26
2.2 CONSTRUCTION DES BASES RÉDUITES	27
2.2.1 Méthode BT (Balanced Truncation)	28
2.2.2 Méthode POD (Proper Orthogonal Decomposition)	30
2.2.3 Méthode POD récursive	39
2.2.4 Méthode DMD (Dynamic Modes Decomposition)	40
2.3 CONSTRUCTION DU MODÈLE D'ORDRE RÉDUIT (ROM)	42
2.3.1 Cas général	42
2.3.2 Construction du ROM pour les équations de Navier-Stokes	43
2.4 CONTRÔLE OPTIMAL RÉDUIT	48
2.4.1 Position du problème de contrôle optimal réduit	48

2.4.2	Optimisation réduite "Differentiate Then Reduce"	48
2.4.3	Optimisation réduite "Reduce Then Differentiate"	50
2.4.4	Exemple d'illustration	52
2.5	CONCLUSION DU CHAPITRE	59
3	ADAPTATION DES MODÈLES D'ORDRE RÉDUIT	61
3.1	INTRODUCTION	62
3.2	MÉTHODES CLASSIQUES D'INTERPOLATION	63
3.3	MÉTHODE ITSGM POUR L'ADAPTATION DE BASES RÉDUITES	66
3.3.1	Quelques rappels de géométrie différentielle	66
3.3.2	Interpolation des bases POD	70
3.3.3	Coût asymptotique de la méthode ITSGM	73
3.4	ADAPTATION DE BASES RÉDUITES PAR PGD	73
3.4.1	PGD basée sur le critère d'orthogonalité de Galerkin	74
3.4.2	Mise en forme de la base PGD	76
3.5	CONCLUSION DU CHAPITRE	79
4	CONSTRUCTION D'UN MODÈLE RÉDUIT NON INTRUSIF	81
4.1	INTRODUCTION	82
4.2	MÉTHODE NIMR	83
4.2.1	Position du problème	83
4.2.2	Description de la méthode	83
4.3	MÉTHODE HNIMR	88
4.4	COÛT ASYMPTOTIQUE DES MÉTHODES NIMR ET HNIMR	90
4.5	APPLICATION SUR LE PROBLÈME DE L'ÉCOULEMENT AUTOUR D'UN CYLINDRE	91
4.5.1	Position du problème	91
4.5.2	Construction des bases réduites d'apprentissage	93
4.5.3	Mise en œuvre des méthodes NIMR et HNIMR	94
4.5.4	Évaluation des performances des méthodes NIMR et HNIMR	97
4.6	CONCLUSION DU CHAPITRE	105
5	CONTRÔLE OPTIMAL BASÉ SUR DES MODÈLES RÉDUITS ADAPTATIFS	109
5.1	INTRODUCTION	110
5.2	CONTRÔLE OPTIMAL RÉDUIT AVEC ADAPTATION DE LA BASE RÉDUITE	110
5.2.1	Algorithme de contrôle optimal réduit avec phase d'adaptation	110
5.2.2	Application au contrôle de l'équation de réaction-diffusion	113
5.2.3	Application au contrôle de l'équation de Burgers	122
5.2.4	Application au contrôle de l'équation de Navier-Stokes	128
5.3	CONTRÔLE OPTIMAL RÉDUIT EN UTILISANT LES ALGORITHMES GÉNÉTIQUES	134
5.3.1	Principe des algorithmes génétiques	134
5.3.2	Représentation génétique	135
5.3.3	Opérateurs génétiques	136

5.3.4	Application au contrôle de l'écoulement d'un fluide autour d'un cylindre	138
5.4	CONCLUSION DU CHAPITRE	144
	BIBLIOGRAPHIE	149

INTRODUCTION GÉNÉRALE

Les problèmes d'optimisation sous contraintes, ou problèmes de contrôle optimal, interviennent dans de nombreuses applications industrielles. Pour pouvoir les résoudre, ces problèmes sont généralement transformés en des problèmes d'optimisation sans contraintes par l'introduction de multiplicateurs de Lagrange. Ces problèmes d'optimisation sans contraintes sont résolus par une méthode de descente classique de type gradient nécessitant plusieurs fois la résolution des équations d'état et des équations adjointes, ce qui est coûteux en temps de calcul. De plus, comme la résolution du problème adjoint, qui dépend de la variable d'état, se fait en remontant le temps, la variable d'état préalablement calculée doit être disponible donc stockée : ce qui nécessite des capacités de stockages importantes. Ainsi, les ressources de calcul et les capacités de stockage très élevées représentent un inconvénient majeur à la mise en œuvre de cette approche. Pour s'affranchir de ces difficultés, il est possible d'utiliser les méthodes de réduction de modèles. Ces méthodes facilitent, d'une part, le stockage et le traitement de données de très grande taille et permettent, d'autre part, d'obtenir rapidement et à faible coût la dynamique temporelle de phénomènes complexes. Elles consistent à approximer la solution $h(t, x, \theta)$ d'un problème physique sur un sous espace vectoriel de faible dimension (engendré par une base réduite $\varphi(x, \theta)$), capable de capturer les caractéristiques dominantes et les propriétés les plus élémentaires de ce problème. Ainsi, la solution s'écrit

$$h(t, x, \theta) \approx \sum_{k=1}^q \alpha^k(t, \theta) \varphi^k(x, \theta)$$

où q est la dimension de la base censée être significativement petite, θ un paramètre de contrôle et $\alpha^1, \dots, \alpha^q$ représentent les coefficients temporels à déterminer. Une fois la base réduite construite, les coefficients temporels sont obtenus par la résolution d'un système d'équations différentielles de taille q résultant de la projection de Galerkin des équations gouvernant le problème étudié sur chaque élément de la base réduite.

De nombreuses techniques de réduction existent, ce qui les différencie, c'est le principe de construction de la base φ . Parmi les différentes méthodes de réduction de modèle, il est possible de distinguer deux grandes familles, celle dite "a posteriori" qui nécessite une connaissance préalable des solutions du problème afin de construire la base réduite, et celle dite "a priori" qui ne nécessite aucune information préalable du problème. Dans ce manuscrit, deux méthodes de réduction de modèles sont considérées. La première est la POD (Proper Orthogonal Decomposition) qui est la méthode a posteriori, la plus utilisée en raison de son optimalité à produire une base de rang minimal contenant la

quasi totalité de l'information présente dans l'ensemble des snapshots nécessaires à sa construction. La seconde est l'approche "a priori" appelée PGD (Proper Generalized Decomposition) qui est une méthode itérative dont le principe est d'enrichir la base réduite à chaque itération par un nouveau mode.

Le contrôle optimal couplé à des modèles réduits obtenus par POD, a été utilisé dans de nombreuses applications. Il consiste à construire une base POD pour un ou plusieurs paramètres de contrôle d'apprentissage, et à formuler le problème de contrôle optimal réduit associé, pour lequel les coefficients temporels représentent les variables d'état. Ainsi, la dimension du problème de contrôle est drastiquement réduite et sa résolution devient très rapide. L'inconvénient de cette approche est que la base POD n'est valable que pour des paramètres situés dans un voisinage proche des paramètres pour lesquels elle a été construite. Par conséquent, en contrôle optimal, cette base peut ne pas être représentative de tous les paramètres qui seront proposés par l'algorithme de contrôle.

Dans le but d'améliorer cette approche classique, l'objectif de cette thèse est de développer des méthodes d'interpolation ou d'adaptation de bases réduites rapides, permettant de mettre à jour les modèles réduits à chaque itération de l'algorithme de contrôle. Dans un premier temps, un algorithme de contrôle optimal réduit, basé sur les équations adjointes, avec une phase d'adaptation de bases est proposé. Dans ce cas, l'adaptation de bases se fait par deux approches. La première, notée ITSGM (Interpolation on Tangent Subspace of Grassman Manifold), consiste à construire un échantillonnage de bases POD pour différents paramètres de contrôle, puis à interpoler ces bases, en passant par l'espace tangent de la variété de Grassman, pour obtenir la base correspondant à un nouveau paramètre. La seconde consiste à adapter, en utilisant la méthode PGD, la base associée au paramètre de l'itération précédente pour qu'elle soit valide pour le paramètre de l'itération courante.

La seconde partie de cette thèse est consacrée au développement d'une nouvelle méthode de réduction de modèle non intrusive, appelée NIMR (Non Intrusive Model Reduction). Cette méthode consiste à interpoler à la fois les bases spatiales et temporelles obtenues par POD, en utilisant la méthode ITSGM; puis à appliquer un processus de calibration aux bases interpolées afin d'obtenir des bases ayant la structure des bases POD. Le problème de calibration de ces bases revient à résoudre un problème d'optimisation sous contraintes dont la solution est analytiquement déterminée. Une variante de cette méthode notée sous l'acronyme HNIMR, qui permet de réduire drastiquement les temps de calcul de méthode NIMR, est également proposée. Ces approches sont ensuite couplées à un algorithme génétique pour le contrôle optimal des écoulements.

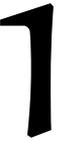
Ce manuscrit est organisé de la façon suivante :

- Le Chapitre 1 est consacré à la formulation d'un problème de contrôle optimal. La première partie de ce chapitre concerne une brève description théorique de l'optimisation sur un ensemble convexe fermé. La deuxième est dédiée à une description détaillée de la méthode des multiplicateurs de Lagrange pour un problème géné-

ral d'optimisation où les contraintes sont décrites par des équations aux dérivées partielles. Enfin, le chapitre se termine par la formulation du problème de contrôle optimal associé aux écoulements fluides.

- Le Chapitre 2 est dédié à la réduction de modèles et à la formulation du problème de contrôle optimal réduit. Un aperçu de quelques méthodes de réduction de modèles existantes est tout d'abord donné, avec un focus sur la méthode POD dans sa version discrète. La construction du modèle réduit par la projection de Galerkin et la dérivation du modèle réduit associé aux équations de Navier-Stokes seront détaillées dans la deuxième section. Enfin, l'utilisation des modèles d'ordre réduit dans le contexte du contrôle optimal est examinée dans la dernière partie.
- Le Chapitre 3 est voué à la description des méthodes d'adaptation de bases réduites. Ce chapitre commence par un bref rappel des méthodes d'interpolation usuelles qui seront utilisées dans ce manuscrit. La deuxième partie est consacrée à la description de la méthode d'interpolation des bases réduites ITSGM (Interpolation on a Tangent Space of the Grassmann Manifold) et la troisième à la description de la méthode d'adaptation de bases fondée sur la PGD (Proper Generalized Decomposition). Ces deux méthodes seront couplées à l'algorithme de contrôle optimal réduit au chapitre 5.
- Le Chapitre 4 est consacré à la description détaillée des méthodes de réduction de modèles non intrusives NIMR et HNIMR développées dans ce travail de thèse. La robustesse et la précision de ces méthodes sont testées sur le problème d'écoulement incompressible d'un fluide Newtonien autour d'un cylindre.
- Le Chapitre 5 concerne la mise en œuvre du contrôle optimal réduit adaptatif basé sur l'équation adjointe ou sur un algorithme génétique. Dans la première partie de ce chapitre, les modifications algorithmiques engendrées par la phase d'adaptation de bases dans l'algorithme de contrôle optimal réduit (basé sur l'équation adjointe) sont décrites. Le potentiel des méthodes d'adaptation de bases ITSGM et PGD dans cet algorithme est ensuite illustré sur le problème de contrôle optimal des équations de réaction-diffusion, de Burgers et des équations de Navier-Stokes. La seconde partie concerne le contrôle optimal réduit en utilisant un algorithme génétique couplé avec les méthodes de réduction de modèles non intrusives NIMR et HNIMR. La pertinence de l'approche en termes de précision et de temps de calcul est évaluée pour le problème de contrôle optimal de l'écoulement d'un fluide autour d'un cylindre.

OPTIMISATION SOUS CONTRAINTES



SOMMAIRE

1.1	INTRODUCTION	6
1.2	OPTIMISATION SUR UN ENSEMBLE K D'UN ESPACE DE BANACH Z	6
1.2.1	Existence d'un minimum	7
1.2.2	Condition d'optimalité	7
1.2.3	Algorithme de descente lorsque $K = Z$	8
1.2.4	Algorithme de descente lorsque $K \subset Z$ est convexe fermé	12
1.2.5	Cas particulier de l'optimisation quadratique	12
1.3	OPTIMISATION SOUS CONTRAINTES : CONTRÔLE OPTIMAL	13
1.3.1	Position du problème	13
1.3.2	Calcul des directions de descente	14
1.3.3	Algorithme de contrôle optimal par l'approche des multiplicateurs de Lagrange	18
1.4	CONTRÔLE OPTIMAL DES ÉCOULEMENTS	19
1.4.1	Problème de Navier-Stokes	19
1.4.2	Formulation variationnelle des équations de Navier-Stokes :	20
1.4.3	Formulation du problème de contrôle optimal	21
1.5	CONCLUSION DU CHAPITRE	24

1.1 INTRODUCTION

Les problèmes d'optimisation sous contraintes sont les problèmes pour lesquels une fonction objectif $\mathcal{J}(x)$ est minimisée ou maximisée sous une contrainte booléenne $\mathcal{N}(x)$. Ce type de problèmes apparaît au quotidien dans de nombreuses applications, par exemple, pour la régulation du trafic routier ou du trafic aérien, la conception de forme, le pilotage des satellites, le contrôle d'écoulements, l'optimisation des coûts en économie ... Dans ce travail, les problèmes abordés concernent la minimisation d'une fonctionnelle coût sous des contraintes décrites par des équations aux dérivés partielles (EDP) non linéaires, ou autrement dit, des problèmes de contrôle optimal. La résolution numérique de ces problèmes s'avère complexe à réaliser. En effet, vu la dépendance implicite entre les variables d'état et les variables de contrôle, une forme explicite des conditions d'optimalité est généralement difficile à obtenir. Il est cependant possible de s'affranchir de cette difficulté à l'aide de la théorie des multiplicateurs de Lagrange introduite par Hestenes [1] et Powell [2]. En introduisant ces multiplicateurs, le problème d'optimisation avec contraintes, se transforme en un problème d'optimisation sans contrainte pour lequel les conditions d'optimalités sont explicites. Des algorithmes d'optimisation classiques basés sur une direction de descente peuvent être utilisés pour résoudre ce problème d'optimisation sans contrainte. Cette approche a été intensément étudiée et appliquée dans de nombreux domaines et en particulier en mécanique des fluides [3][4][5][6]...etc

Ainsi, ce chapitre est dédié à la description théorique et méthodologique de la résolution des problèmes de contrôle optimal. La première partie du chapitre est consacrée à une brève description théorique de l'optimisation sur un convexe fermé et des conditions d'optimalités associées. La deuxième partie de ce chapitre est dédiée à la description de la méthode des multiplicateurs de Lagrange pour un problème général d'optimisation où les contraintes sont décrites par des équations aux dérivés partielles. Il se terminera par l'écriture du problème associé au contrôle des écoulements fluides pour lequel les contraintes sont régies par les équations de Navier-Stokes.

1.2 OPTIMISATION SUR UN ENSEMBLE K D'UN ESPACE DE BANACH Z

Cette section est dédiée à une brève description des résultats théoriques et des méthodes numériques pour chercher la solution d'un problème de minimisation sur un ensemble convexe fermé. Dans ce cas, le problème de minimisation s'écrit donc :

$$\min_{x \in KCZ} \mathcal{J}(x) \quad (1.1)$$

où \mathcal{J} est la fonctionnelle objectif à minimiser (supposée continûment différentiable au sens de Fréchet) et x la variable d'optimisation. Z est un espace de Banach¹ et $K \subset Z$ est un ensemble convexe fermé de Z . Ce qui suit est un rappel, sans démonstration, des

1. V est appelé espace de Banach, si il est un espace vectoriel muni d'une norme $\|\cdot\|_V$, et qu'il est complet pour la topologie induite par cette norme (toute suite de Cauchy est convergente).

résultats classiques de l'optimisation. Pour plus de détails, le lecteur peut se référer entre autres aux références suivantes [7, 8, 9, 10, 11].

1.2.1 Existence d'un minimum

Si Z est de dimension finie, l'existence du minimum est donnée par le théorème suivant :

Théorème 1.1 *Soit Z un espace vectoriel normé de dimension finie. Soit K un ensemble fermé non vide de Z , et $\mathcal{J} : K \rightarrow \mathbb{R}$ une fonction continue sur K , et coercive "infinie à l'infini", c'est à dire :*

$$\forall (x^n)_n \subset K, \lim_{n \rightarrow +\infty} \|x^n\| = +\infty \implies \lim_{n \rightarrow +\infty} \mathcal{J}(x^n) = +\infty \quad (1.2)$$

Alors il existe au moins un point minimum de \mathcal{J} sur K . De plus, de toute suite minimisante de \mathcal{J} dans K , on peut extraire une sous suite qui converge vers un point minimum sur K .

L'astuce principale pour prouver le théorème 1.1 réside dans le fait qu'en dimension finie, les fermés bornés sont compacts. Il suffit alors de montrer que toute suite minimisante de \mathcal{J} sur K est bornée. Cette propriété est assurée par (1.2). En outre, elle est automatiquement vérifiée si K est borné.

Toutefois, en dimension infinie, les fermés bornés ne sont pas forcément des compacts [12]. Dans ce cas, le théorème 1.1 se généralise de la façon suivante :

Théorème 1.2 *Soit K un convexe fermé non vide d'un espace de Banach réflexif Z , et \mathcal{J} une fonction convexe continue et coercive sur K (i.e., qui vérifie la condition (1.2)), alors il existe un minimum de \mathcal{J} sur K . Si de plus \mathcal{J} est strictement convexe sur K , ce minimum est unique.*

Un espace de Banach Z est réflexif lorsqu'il coïncide avec le dual de son dual (i.e., $Z = Z^{**}$). C'est bien le cas pour une large classe d'espaces tels que les espaces de Hilbert.

1.2.2 Condition d'optimalité

Définissons maintenant la notion de différentiabilité au sens de Fréchet.

Définition 1.1 *Soit Z et Y deux espaces de Banach et \mathcal{J} une fonction définie sur un voisinage de $x \in Z$ à valeurs dans Y . On dit que \mathcal{J} est différentiable au sens de Fréchet en x s'il existe une application linéaire continue L de Z dans Y telle que*

$$\mathcal{J}(x + \omega) = \mathcal{J}(x) + L(\omega) + o(\omega), \quad \text{avec} \quad \lim_{\omega \rightarrow 0} \frac{\|o(\omega)\|_Y}{\|\omega\|_Z} = 0.$$

On note $L = \nabla \mathcal{J}(x)$. $\nabla \mathcal{J}(x)$ est appelée différentielle, dérivée ou gradient de \mathcal{J} en x .

Pour le cas particulier où $Y = \mathbb{R}$, la différentielle $L = \nabla \mathcal{J}(x)$ est une forme linéaire continue sur Z , c'est à dire que $L(\omega) = \langle \nabla \mathcal{J}(x), \omega \rangle_{Z^*, Z}$ où $\langle \cdot, \cdot \rangle_{Z^*, Z}$ désigne le crochet de dualité. En pratique, il n'est pas toujours évident de déterminer la différentielle au sens de Fréchet. Il apparait donc utile d'introduire une notion plus faible de dérivée dite dérivée

directionnelle. La dérivée directionnelle d'une fonction \mathcal{J} en un point $x \in Z$ suivant une direction $z \in Z$ est définie par

$$\langle \nabla \mathcal{J}(x), z \rangle_{Z^*, Z} = \lim_{\delta \rightarrow 0} \frac{\mathcal{J}(x + \delta z) - \mathcal{J}(x)}{\delta} \quad (1.3)$$

Si de plus, pour toute direction $z \in Z$ la dérivée directionnelle en un point x existe et que $z \mapsto \langle \nabla \mathcal{J}(x), z \rangle_{Z^*, Z}$ est une application linéaire continue de Z dans Y , alors on dit que \mathcal{J} est différentiable au sens de Gâteaux en x . On note que la différentiabilité d'une fonction au sens de Fréchet implique sa différentiabilité au sens de Gâteaux. La réciproque est en général fautive, même en dimension finie.

La condition d'optimalité pour le problème (1.1) est donnée par le théorème suivant :

Théorème 1.3 *Soit $x \in K$. On suppose que \mathcal{J} est différentiable en x . Si x est un minimum local de \mathcal{J} sur K , alors*

$$\langle \nabla \mathcal{J}(x), z - x \rangle_{Z^*, Z} \geq 0 \quad \forall z \in K. \quad (1.4)$$

Notons que la condition nécessaire (1.4), dite inéquation d'Euler, devient nécessaire et suffisante si en plus \mathcal{J} est convexe. Il faut aussi remarquer que si $K = Z$, l'équation (1.4) se réduit simplement à l'équation d'Euler $\nabla \mathcal{J}(x) = 0$.

1.2.3 Algorithme de descente lorsque $K = Z$

a) Construction d'une suite minimisante

On se place dans le cas $K = Z$, le problème d'optimisation devient donc

$$\min_{x \in Z} \mathcal{J}(x) \quad (1.5)$$

La résolution de ce problème s'effectue généralement à l'aide des algorithmes d'optimisation dits de descente. L'idée des algorithmes de descente est de chercher pour le candidat $x^n \in Z$, une direction $d^n \in Z$ telle que $\chi_n(t) \stackrel{\text{def}}{=} \mathcal{J}(x^n + td^n)$ est décroissante à $t = 0$, c'est à dire

$$\nabla \chi_n(0) = \langle \nabla \mathcal{J}(x^n), d^n \rangle_{Z^*, Z} < 0.$$

Cette descente peut toutefois être très petite. Afin de vérifier la qualité de la direction de descente, pour $\eta > 0$ fixé, il est naturel de vérifier la condition (condition sur l'angle)

$$\langle -\nabla \mathcal{J}(x^n), d^n \rangle_{Z^*, Z} \geq \eta \|\nabla \mathcal{J}(x^n)\|_{Z^*} \|d^n\|_Z \quad (1.6)$$

Ces approches nécessitent également le choix d'un pas de descente σ^n de telle sorte que

$$\chi_n(\sigma^n) < \chi_n(0).$$

Les algorithmes de descente consistent donc à construire une suite minimisante $(x_n)_{n \in \mathbb{N}}$ telle que $|x_n - x^*| \xrightarrow[n \rightarrow \infty]{} 0$. Le nouveau itéré de la suite est calculé à l'aide de la relation de récurrence suivante

$$x^{n+1} = x^n + \sigma^n d^n$$

Les différentes étapes des algorithmes de descentes sont répertoriés dans l'algorithme 1.

Algorithme 1 : Algorithme de descente

```

1 choisir un vecteur initial  $x^0$ ;
2 for  $n = 1, 2, \dots$  do
3   Si  $\|\nabla \mathcal{J}(x^n)\| < \varepsilon$ , STOP;
4   Choisir une direction de descente  $d^n \in Z$  telle que  $\langle \nabla \mathcal{J}(x^n), d^n \rangle_{Z^*, Z} < 0$ ;
5   Choisir un pas de descente  $\sigma^n > 0$  tel que  $\mathcal{J}(x^n + \sigma^n d^n) < \mathcal{J}(x^n)$ ;
6   Déterminer le nouveau itéré  $x^{n+1} = x^n + \sigma^n d^n$ .
7 end

```

b) Conditions d'admissibilité

Dans le cadre général, il n'est pas encore possible de prouver une convergence globale de l'algorithme 1. Pour y parvenir, des conditions supplémentaires sur la qualité de la direction de descente ainsi que sur les pas de descente seront nécessaires. Ces conditions sont dites conditions d'admissibilité et sont données comme suit :

— Admissibilité des directions de descente d^n

$$\frac{\langle \nabla \mathcal{J}(x^n), d^n \rangle_{Z^*, Z}}{\|d^n\|_Z} \xrightarrow{n \rightarrow \infty} 0 \implies \|\nabla \mathcal{J}(x^n)\|_{Z^*} \xrightarrow{n \rightarrow \infty} 0$$

— Admissibilité des pas de descente σ^n

$$\begin{cases} \mathcal{J}(x^n + \sigma^n d^n) < \mathcal{J}(x^n) \quad \forall n \text{ et} \\ |\mathcal{J}(x^n + \sigma^n d^n) - \mathcal{J}(x^n)| \xrightarrow{n \rightarrow \infty} 0 \end{cases} \implies \frac{\langle \nabla \mathcal{J}(x^n), d^n \rangle_{Z^*, Z}}{\|d^n\|_Z} \xrightarrow{n \rightarrow \infty} 0$$

L'expression $\frac{\langle \nabla \mathcal{J}(x^n), d^n \rangle_{Z^*, Z}}{\|d^n\|_Z}$ n'est autre que la pente de \mathcal{J} au point x^n dans la direction d^n , définie par

$$\left. \frac{d}{dt} \mathcal{J} \left(x^n + t \frac{d^n}{\|d^n\|_Z} \right) \right|_{t=0} = \frac{\langle \nabla \mathcal{J}(x^n), d^n \rangle_{Z^*, Z}}{\|d^n\|_Z}$$

Par conséquent, la propriété d'admissibilité des pas de descente signifie que si la décroissance de \mathcal{J} devient de plus en plus petite, la pente suivant les directions d^n devient également de plus en plus petite. La propriété d'admissibilité des directions de descente quant à elle, signifie que si la pente le long de d^n devient de plus en plus petite, alors les fortes pentes possibles deviennent de plus en plus petite. Ces deux conditions permettent de prouver la convergence globale de l'algorithme de descente énoncée dans le théorème suivant :

Théorème 1.4 Soit \mathcal{J} continûment Fréchet différentiable et x^n, d^n, σ^n généré par l'algorithme de descente 1. On suppose que d_k, σ^n sont admissibles et que $\mathcal{J}(x^n)$ est bornée inférieurement. Alors

$$\lim_{n \rightarrow \infty} \nabla \mathcal{J}(x^n) = 0. \quad (1.7)$$

En particulier, tout point d'accumulation de x^n est un point stationnaire.

Démonstration. Soit $\mathcal{J}^* = \inf_{n \geq 0} \mathcal{J}(x^n) > -\infty$. Alors, d'après $\mathcal{J}(x^n + \sigma^n d^n) - \mathcal{J}(x^n) < 0$ on a $\mathcal{J}(x^n) \longrightarrow \mathcal{J}^*$ et

$$\mathcal{J}(x^0) - \mathcal{J}^* = \sum_{n=0}^{\infty} \mathcal{J}(x^n) - \mathcal{J}(x^{n+1}) = \sum_{n=0}^{\infty} \mathcal{J}(x^n) - \mathcal{J}(x^n + \sigma^n d^n).$$

ceci implique que $\mathcal{J}(x^n + \sigma^n d^n) - \mathcal{J}(x^n) \xrightarrow{n \rightarrow \infty} 0$. Ainsi, d'après l'admissibilité de σ^n on obtient

$$\frac{\langle \nabla \mathcal{J}(x^n), d^n \rangle_{Z^*, Z}}{\|d^n\|_Z} \xrightarrow{n \rightarrow \infty} 0$$

Il résulte de l'admissibilité de d^n que

$$\|\nabla \mathcal{J}(x^n)\|_{Z^*} \xrightarrow{n \rightarrow \infty} 0$$

Ensuite, considérons le cas où x^* est un point d'accumulation de x^n . Alors il existe une sous suite toujours noté x^n qui converge dans Z vers x^* . D'après la monotonie de la suite $\mathcal{J}(x^n)$ on conclut que $\mathcal{J}(x^*) \leq \mathcal{J}(x^n)$ pour tout n . En appliquant (1.7) on a par continuité

$$\nabla \mathcal{J}(x^*) = \lim_{n \rightarrow \infty} \nabla \mathcal{J}(x^n) = 0.$$

□

Dans la pratique, deux questions assez fréquentes se manifestent :

1. Comment vérifier si une direction de descente est admissible ?
2. Comment vérifier si les pas de descente sont admissibles ?

Une réponse à la question 1 est donnée par le lemme suivant :

Lemme 1.1 *Les directions de descente d^n sont dites admissibles si elles satisfont la condition (1.6).*

Démonstration. Découle directement de la condition (1.6)

$$\|\nabla \mathcal{J}(x^n)\|_{Z^*} \leq -\frac{1}{\eta} \frac{\langle \nabla \mathcal{J}(x^n), d^n \rangle_{Z^*, Z}}{\|d^n\|_Z}$$

□

Dans le cadre des espaces de Banach, le calcul des directions de descente d^n n'est pas facile. Il est à noter que par le fait que $Z^* \ni \nabla \mathcal{J}(x^n) \notin Z$ la dérivée négative de \mathcal{J} ne convient pas comme direction de descente. Cependant dans le cas particulier des espaces de Hilbert, en utilisant le théorème de représentation de Riesz [13], on peut choisir $Z^* = Z$ et $\langle \cdot, \cdot \rangle_{Z^*, Z} = (\cdot, \cdot)_Z$. Par conséquent $\nabla \mathcal{J}(x^n) \in Z$ et $-\nabla \mathcal{J}(x^n)$ sont des directions de descente. Les méthodes de descente pour lesquelles $d^n = -\nabla \mathcal{J}(x^n)$ sont appelées méthodes de descente de type gradient.

c) Exemples de choix de pas de descente

Pour répondre à la question 2, de nombreuses règles pour le choix du pas de descente existent dans la littérature. Nous décrivons uniquement celles qui sont les plus utilisées :

1. Règle de minimisation

Il semble naturel de déterminer le pas σ^n par la règle de Cauchy comme solution du problème de minimisation

$$\min_{\sigma \geq 0} \chi_n(\sigma) \quad (1.8)$$

Ce pas est appelé pas de Cauchy ou pas optimal [11][14]. Cette méthode est qualifiée de recherche linéaire "exacte", mais n'est utilisée que pour des cas simple, notamment quand la fonction χ_n peut être explicitement exprimée en fonction de σ .

2. Règle de minimisation bornée

Un nombre $s > 0$ fixé est choisi et σ^n est calculé comme solution du problème

$$\min_{\sigma \in [0, s]} \chi_n(\sigma) \quad (1.9)$$

3. Règle d'Armijo

Cette méthode dite inexacte a été développée pour les cas où le calcul d'un pas "exact" devient impossible. Soit d^n une direction de descente de \mathcal{J} au point x^n . La condition d'Armijo [9] consiste à imposer que \mathcal{J} décroisse autant qu'une portion $\delta \in (0, 1)$, et s'exprime comme suit

$$\mathcal{J}(x^n + \sigma^n d^n) - \mathcal{J}(x^n) \leq \delta \sigma^n \langle \nabla \mathcal{J}(x^n), d^n \rangle_{Z^*, Z} \quad (1.10)$$

Le pas de descente σ^n correspondant est donc choisi comme étant le maximum de l'ensemble $\{1, 1/2, 1/4, \dots\}$ qui vérifie la condition (1.10)

d) Exemples de choix de direction de descente

Dans un algorithme de descente, les directions de descente peuvent être obtenues en calculant l'opposé du gradient de la fonctionnelle \mathcal{J} au point courant, c'est à dire $d^n = -\nabla \mathcal{J}(x^n)$. L'algorithme de descente de type gradient est donc défini par la suite récurrente suivante :

$$x^{n+1} = x^n - \sigma_n \nabla \mathcal{J}(x^n), \quad \sigma_n > 0.$$

où le pas σ_n peut être déterminé par l'une des stratégies présentées dans le paragraphe précédent. Les directions de descente peuvent également être calculées à l'aide de l'algorithme de Newton, qui consiste à chercher les points critiques de la fonction \mathcal{J} en résolvant l'équation :

$$\nabla \mathcal{J}(x) = 0$$

Supposons que la fonction \mathcal{J} est de classe \mathcal{C}^2 et que la matrice hessienne $\mathbf{H}[f](x^n)$ est inversible, l'algorithme de descente de type Newton est défini par la suite récurrente suivante

$$x^{n+1} = x^n - \sigma_n [\mathbf{H}[f](x^n)]^{-1} \nabla \mathcal{J}(x^n), \quad \sigma_n > 0,$$

où $d^n = -[H[f](x^n)]^{-1} \nabla \mathcal{J}(x^n)$ est appelée direction de Newton. L'algorithme de Newton présente l'avantage d'avoir une convergence quadratique. Toutefois, il présente quelques inconvénients :

- Un coût de calcul important dû au calcul de la hessienne $H[f](x^n)$ et à la résolution du système linéaire $H[f](x^n)d^n = \nabla \mathcal{J}(x^n)$. De plus, l'expression analytique des dérivées secondes est rarement disponible dans les applications.
- L'absence de convergence si le premier itéré est trop loin de la solution, ou si la hessienne est singulière.
- Il n'y a pas de distinction entre minima, maxima et points stationnaires.

En ce qui concerne les applications numériques traitées dans ce manuscrit, seule la méthode de descente de type gradient sera utilisée.

1.2.4 Algorithme de descente lorsque $K \subset Z$ est convexe fermé

Revenons maintenant au cas où $K \subset Z$ est convexe fermé. Un scénario fréquemment rencontré en pratique, et d'ailleurs celui qui nous intéresse le plus, est lorsque K est défini comme

$$K = \{x \in Z; \quad x_L \leq x \leq x_R \text{ presque partout}\}$$

avec x_L et x_R qui sont bien définis. Il est maintenant devenu très facile de calculer la projection P_K sur K [15] de tout élément $\omega \in Z$ en utilisant

$$P_K(\omega) = \max(x_L, \min(\omega, x_R)) \quad (1.11)$$

L'algorithme pour le cas où $K \subset Z$ est alors obtenu par une légère modification de l'algorithme 1. Les étapes d'optimisation sur un convexe fermé sont répertoriées dans l'algorithme 2.

Algorithme 2 : Algorithme de descente projetée

- 1 choisir un vecteur initial $x^0 \in K$;
 - 2 **for** $n = 1, 2, \dots$ **do**
 - 3 Si $\|\nabla \mathcal{J}(x^n)\| < \varepsilon$, STOP;
 - 4 Choisir une direction de descente $d^n \in Z$ telle que $\langle \nabla \mathcal{J}(x^n), d^n \rangle_{Z^*, Z} < 0$;
 - 5 Choisir un pas de descente $\sigma_n > 0$ tel que $\mathcal{J}(P_K(x^n + \sigma_n d^n)) < \mathcal{J}(x^n)$;
 - 6 Déterminer le nouveau itéré $x^{n+1} = P_K(x^n + \sigma_n d^n)$.
 - 7 **end**
-

1.2.5 Cas particulier de l'optimisation quadratique

Soit la fonctionnelle quadratique $\mathcal{J} : \mathbb{R}^N \rightarrow \mathbb{R}$ définie par

$$\mathcal{J}(x) = \frac{1}{2}(Ax, x) - (b, x), \quad (1.12)$$

où (\cdot, \cdot) désigne le produit scalaire Euclidien canonique de \mathbb{R}^N , A est une matrice $N \times N$ symétrique définie positive et $b \in \mathbb{R}^N$. Soit B une application linéaire de \mathbb{R}^N dans \mathbb{R}^M

identifiable à une matrice de taille $M \times N$.

Le problème d'optimisation sous contraintes consiste à minimiser la fonctionnelle $\mathcal{J}(x)$ tel que x appartient au noyau de la matrice \mathbf{B} . C'est à dire :

$$\begin{cases} \mathcal{J}(x) \leq \mathcal{J}(z), \forall z \in \text{Ker}\mathbf{B} = \{z \in \mathbb{R}^N, \mathbf{B}z = 0\}, \\ x \in \text{Ker}\mathbf{B} \end{cases} \quad (1.13)$$

Ce problème d'optimisation sous contraintes est un problème de point selle, dont la solution numérique peut être calculée en utilisant l'algorithme d'Uzawa. Il existe plusieurs variantes de l'algorithme d'Uzawa [16]. Les plus utilisés sont l'algorithme du résidu minimal (Algorithme 3) et l'algorithme du gradient conjugué (Algorithme 4).

Algorithme 3 : Algorithme d'Uzawa (résidu minimal)

- 1 Pour $n = 0$ choisir un vecteur initial $\zeta^0 \in \mathbb{R}^M$ et calculer $x^0 = \mathbf{A}^{-1}(b - \mathbf{B}^T \zeta^0)$;
- 2 ζ^n, x^n connus, on détermine $z^n, \rho_n, \zeta^{n+1}, x^{n+1}$ par

$$z^n = -\mathbf{A}^{-1} \mathbf{B}^T \mathbf{B} x^n, \quad (1.14)$$

$$\rho_n = -\frac{(\mathbf{B} x^n, \mathbf{B} z^n)}{|\mathbf{B} z^n|^2} \quad (1.15)$$

$$\zeta^{n+1} = \zeta^n + \rho_n \mathbf{B} x^n, \quad (1.16)$$

$$x^{n+1} = x^n + \rho_n z^n. \quad (1.17)$$

1.3 OPTIMISATION SOUS CONTRAINTES : CONTRÔLE OPTIMAL

1.3.1 Position du problème

L'objectif de cette section est de traiter la classe des problèmes variationnels de la forme

$$\min \mathcal{J}(y, \theta) \text{ sur } (y, \theta) \in V \times K \quad (1.25)$$

sous la contrainte

$$\mathcal{N}(y, \theta) = 0 \quad (1.26)$$

où $\mathcal{J} : V \times Z \rightarrow \mathbb{R}$ désigne la fonctionnelle coût, et $\mathcal{N} : V \times Z \rightarrow W$ décrit l'équation des contraintes appelée aussi équation d'état. De plus, nous supposons que $K \subset Z$ est un ensemble convexe fermé de Z et V, W et Z sont des espaces de Hilbert. Les problèmes qui s'inscrivent dans le cadre des équations (1.25) et (1.26) sont assez généraux et se retrouvent dans de nombreux domaines d'application. Ils comprennent notamment les problèmes de contrôle optimal, l'optimisation de forme, les problèmes inverses pour l'estimation de paramètres, les problèmes de contact et de friction, les problèmes de reconstruction

Algorithme 4 : Algorithme d'Uzawa (gradient conjugué)

- 1 Pour $n = 0$ choisir un vecteur initial $\zeta^0 \in \mathbb{R}^M$ et calculer $x^0 = A^{-1}(b - B^T \zeta^0)$;
- 2 On calcule la direction de descente w_n par

$$w_0 = g_0 = -Bx^0 \text{ si } n = 0, \quad (1.18)$$

$$w_n = -Bx^n + \lambda_n w_{n-1}, \text{ si } n \geq 1 \quad (1.19)$$

$$\lambda_n = \frac{|Bx^n|^2}{|Bx^{n-1}|^2} \text{ si } n \geq 1, \quad (1.20)$$

- 3 ζ^n, x^n connus, on détermine $z^n, \rho_n, \zeta^{n+1}, x^{n+1}$ par

$$z^n = A^{-1}B^T w_n, \quad (1.21)$$

$$\rho_n = -\frac{|Bx^n|^2}{(Bx^n, Bz^n)} \quad (1.22)$$

$$\zeta^{n+1} = \zeta^n - \rho_n w_n, \quad (1.23)$$

$$x^{n+1} = x^n + \rho_n z^n. \quad (1.24)$$

d'image etc. Dans le contexte du contrôle optimal, la variable y est appelée variable d'état, θ le paramètre de contrôle, et \mathcal{J} la fonctionnelle coût choisie de manière appropriée pour atteindre des performances spécifiques. Si \mathcal{N} permet d'exprimer la variable y en fonction de θ , alors le problème décrit par les equations (1.25) et (1.26) peut être réduit à

$$\min_{\theta \in K} \mathcal{J}(y(\theta), \theta) \text{ sous la contrainte } \mathcal{N}(y(\theta), \theta) = 0. \quad (1.27)$$

La relation entre les variables y et θ est assurée par l'équation (1.26). Cette relation est typiquement décrite par des equations aux dérivées partielles définies sur Ω et évoluant en temps sur $]0, T]$. En particulier, dans ce manuscrit, on s'intéresse à des problèmes d'optimisation dont les contraintes sont des équations aux dérivées partielles et les espaces fonctionnelles sont des espaces de Hilbert tels que

$$V = L^2(0, T; H^1(\Omega)) \quad W = L^2(0, T; L^2(\Omega)) \quad Z = \mathbb{R}^{d_p}$$

1.3.2 Calcul des directions de descente

D'un point de vue numérique, l'implémentation d'un algorithme de descente permettant de fournir une solution pour un problème de type (1.27) nécessite le calcul des directions de descente. Celles ci peuvent être évaluées en calculant le gradient de la fonctionnelle par rapport à la variable d'optimisation. Toutefois, l'évaluation du gradient de la fonctionnelle est généralement difficile. Cette difficulté découle de la dépendance implicite entre la variable d'état et la variable d'optimisation, via l'équation des contraintes

(1.26). Une approche possible pour évaluer $\nabla_{\theta}\mathcal{J}$ consiste simplement à utiliser l'approximation de Taylor de premier ordre. Une autre approche plus précise consiste à utiliser la méthode des sensibilités. Enfin, une troisième approche consiste à traiter y et θ comme des variables indépendantes et à utiliser la méthode des multiplicateurs de Lagrange [17][18]. Comme déjà mentionné précédemment, cette dernière approche permet de transformer le problème d'optimisation sous contraintes (1.27) en un problème d'optimisation sans contrainte facile à résoudre. L'évaluation de $\nabla_{\theta}\mathcal{J}$ par les trois approches mentionnées est décrite ci après.

L'expression du gradient de la fonctionnelle objectif par rapport à la variable d'optimisation $\theta \in Z$ s'exprime comme suit

$$\nabla_{\theta}\mathcal{J}(y, \theta) = \partial_y\mathcal{J}(y, \theta)y'(\theta) + \partial_{\theta}\mathcal{J}(y, \theta) \quad (1.28)$$

Dans cette expression, on note bien que les termes $\partial_y\mathcal{J}(y, \theta)$ et $\partial_{\theta}\mathcal{J}(y, \theta)$ sont faciles à évaluer. En revanche, le terme de sensibilité $y'(\theta)$ est difficile à évaluer.

a) Approximation de Taylor d'ordre 1

Afin d'évaluer le terme de sensibilité $y'(\theta)$, il est possible d'approximer la ligne tangente de $y(\theta)$. Cette approximation est appelée approximation polynomiale de Taylor d'ordre 1 et est définie pour ε suffisamment petit comme suit

$$y(\theta) \approx y(\theta - \varepsilon\delta\theta) + \varepsilon\langle y'(\theta), \delta\theta \rangle$$

ceci implique que

$$\langle y'(\theta), \delta\theta \rangle \approx \frac{y(\theta) - y(\theta - \varepsilon\delta\theta)}{\varepsilon}$$

L'utilisation d'une approximation par différences finies du gradient peut sembler attractive en raison de sa facilité de mise en œuvre. Toutefois, cette approche n'est pas recommandée à cause de son manque en précision. De plus, son coût numérique augmente énormément lorsque la dimension des variables d'optimisation θ est élevée.

b) Équations de sensibilités

Il est également possible d'évaluer le terme de sensibilité $y'(\theta)$ par la résolution du système linéaire

$$\partial_y\mathcal{N}(y, \theta)y'(\theta) = -\partial_{\theta}\mathcal{N}(y, \theta) \quad (1.29)$$

L'avantage de cette approche par rapport à l'approximation de Taylor d'ordre 1 est que le calcul de $y'(\theta)$ est plus précis [19] et se fait en une seule étape (résolution de systèmes linéaires). Toutefois, il est nécessaire de résoudre autant de systèmes linéaires qu'il y a de directions possibles. Bien que cette approche soit plus performante que l'approximation de Taylor d'ordre 1, elle reste quand même coûteuse à implémenter.

c) Méthode des multiplicateurs de Lagrange

Dans ce paragraphe, on introduit la méthode des multiplicateurs de Lagrange pour le calcul du gradient de la fonctionnelle objectif. Cette approche est plus performante et permet le calcul du gradient de la fonctionnelle avec un faible coût par rapport aux deux approches précédentes.

Supposons que $\mathcal{J} : V \times Z \rightarrow \mathbb{R}$ et $\mathcal{N} : V \times Z \rightarrow W$ soient continûment Fréchet différentiables, et que la dérivée partielle de \mathcal{N} par rapport à y au point $(\bar{y}, \bar{\theta})$ satisfait la condition suivante

$$\partial_y \mathcal{N}(\bar{y}, \bar{\theta}) \in \mathcal{L}(V, W) \text{ est une bijection} \quad (1.30)$$

où \bar{y} est donné par $\bar{y} = y(\bar{\theta})$. Si $\bar{\theta}$ est un minimum local pour (1.27), alors

$$(\theta - \bar{\theta})^T \nabla_{\theta} \mathcal{J}(\bar{y}, \bar{\theta}) \geq 0 \quad \forall \theta \in K. \quad (1.31)$$

On introduit maintenant une définition qui permettra de transformer cette condition d'optimalité implicite en une forme explicite et exploitable.

Définition 1.2 *Un élément $\xi \in W$ est appelé l'état adjoint associée à la variable d'état \bar{y} si il est solution du problème adjoint suivant :*

$$\partial_y \mathcal{J}(\bar{y}, \bar{\theta}) + \partial_y \mathcal{N}(\bar{y}, \bar{\theta})^* \xi = 0 \quad (1.32)$$

où $\partial_y \mathcal{N}(\bar{y}, \bar{\theta})^*$ désigne l'opérateur adjoint de $\partial_y \mathcal{N}(\bar{y}, \bar{\theta})$.

Les conditions d'optimalité pour le problème d'optimisation sous contraintes (1.27) sont données par le théorème suivant :

Théorème 1.5 *Soit $\bar{\theta}$ un minimum local pour (1.27) et \bar{y} l'état associé. Si la condition (1.30) est vérifiée, alors il existe un état adjoint $\xi \in W$ tel que le système d'optimalité suivant est satisfait :*

$$\mathcal{N}(\bar{y}, \bar{\theta}) = 0, \quad (1.33)$$

$$\partial_y \mathcal{J}(\bar{y}, \bar{\theta}) + \partial_y \mathcal{N}(\bar{y}, \bar{\theta})^* \xi = 0, \quad (1.34)$$

$$(\theta - \bar{\theta})^T [\partial_{\theta} \mathcal{J}(\bar{y}, \bar{\theta}) + \partial_{\theta} \mathcal{N}(\bar{y}, \bar{\theta})^* \xi] \geq 0 \quad \forall \theta \in K. \quad (1.35)$$

Démonstration. L'opérateur adjoint $\partial_y \mathcal{N}(\bar{y}, \bar{\theta})^*$ étant bijectif (voir [20] pour plus de détails), il existe un état adjoint $\xi \in W$ qui satisfait l'équation (1.34). Pour obtenir les équations (1.35), on calcule $\nabla_{\theta} \mathcal{J}$ au point $(\bar{y}, \bar{\theta})$ dans la direction $\delta\theta \in Z$ comme suit

$$\langle \nabla_{\theta} \mathcal{J}(\bar{y}, \bar{\theta}), \delta\theta \rangle_Z = \langle \partial_y \mathcal{J}(\bar{y}, \bar{\theta}), \langle y'(\bar{\theta}), \delta\theta \rangle_V \rangle + \langle \partial_{\theta} \mathcal{J}(\bar{y}, \bar{\theta}), \delta\theta \rangle_Z$$

En utilisant l'équation adjointe (1.32) on peut écrire

$$\begin{aligned} \langle \nabla_{\theta} \mathcal{J}(\bar{y}, \bar{\theta}), \delta\theta \rangle_Z &= - \langle \partial_y \mathcal{N}(\bar{y}, \bar{\theta})^* \xi, \langle y'(\bar{\theta}), \delta\theta \rangle_V \rangle + \langle \partial_{\theta} \mathcal{J}(\bar{y}, \bar{\theta}), \delta\theta \rangle_Z \\ &= - \langle \xi, \partial_y \mathcal{N}(\bar{y}, \bar{\theta}) \langle y'(\bar{\theta}), \delta\theta \rangle_W \rangle + \langle \partial_{\theta} \mathcal{J}(\bar{y}, \bar{\theta}), \delta\theta \rangle_Z \end{aligned}$$

Grâce à l'équation (1.29) on a

$$\begin{aligned}\langle \nabla_{\theta} \mathcal{J}(\bar{y}, \bar{\theta}), \delta\theta \rangle_Z &= \langle \xi, \langle \partial_{\theta} \mathcal{N}(\bar{y}, \bar{\theta}), \delta\theta \rangle_W \rangle + \langle \partial_{\theta} \mathcal{J}(\bar{y}, \bar{\theta}), \delta\theta \rangle_Z \\ &= \delta\theta^T [\partial_{\theta} \mathcal{N}(\bar{y}, \bar{\theta})^* \xi + \partial_{\theta} \mathcal{J}(\bar{y}, \bar{\theta})]\end{aligned}$$

Finalement, en choisissant $\delta\theta = \theta - \bar{\theta}$ pour tout $\theta \in K$, puis en utilisant (1.31), on obtient

$$(\theta - \bar{\theta})^T [\partial_{\theta} \mathcal{J}(\bar{y}, \bar{\theta}) + \partial_{\theta} \mathcal{N}(\bar{y}, \bar{\theta})^* \xi] \geq 0.$$

□

Le système d'optimalité formé par les équations (1.33), (1.34) et (1.35) peut facilement être obtenu par l'introduction de la fonction de Lagrange, dans laquelle la variable d'état y et d'optimisation θ sont supposés indépendantes. La fonction de Lagrange associée au problème (1.27) est donnée par

$$\begin{aligned}\mathcal{L} : V \times Z \times W &\longrightarrow \mathbb{R} \\ (y, \theta, \xi) &\longrightarrow \mathcal{L}(y, \theta, \xi) = \mathcal{J}(y, \theta) + \langle \xi, \mathcal{N}(y, \theta) \rangle_W\end{aligned}$$

où ξ désigne le multiplicateur de Lagrange. La différentielle du Lagrangien par rapport à la variable d'état y , suivant la direction $\delta y \in V$, s'exprime comme suit

$$\begin{aligned}\langle \partial_y \mathcal{L}(y, \theta, \xi), \delta y \rangle_V &= \langle \partial_y \mathcal{J}(y, \theta), \delta y \rangle_V + \langle \xi, \partial_y \mathcal{N}(y, \theta) \delta y \rangle_W \\ &= \langle \partial_y \mathcal{J}(y, \theta), \delta y \rangle_V + \langle \partial_y \mathcal{N}(y, \theta)^* \xi, \delta y \rangle_V.\end{aligned}\quad (1.36)$$

Par conséquent, l'annulation de l'équation (1.36) permet d'obtenir l'équation (1.34) donnée par

$$\partial_y \mathcal{J}(y(\bar{\theta}), \bar{\theta}) + \partial_y \mathcal{N}(y(\bar{\theta}), \bar{\theta})^* \xi = 0.$$

De manière identique, en prenant la différentielle du Lagrangien par rapport θ suivant la direction $\delta\theta \in Z$, on obtient

$$\begin{aligned}\langle \partial_{\theta} \mathcal{L}(y, \theta, \xi), \delta\theta \rangle_Z &= \langle \partial_{\theta} \mathcal{J}(y, \theta), \delta\theta \rangle_Z + \langle \xi, \partial_{\theta} \mathcal{N}(y, \theta) \delta\theta \rangle_W \\ &= \langle \partial_{\theta} \mathcal{J}(y, \theta), \delta\theta \rangle_Z + \langle \partial_{\theta} \mathcal{N}(y, \theta)^* \xi, \delta\theta \rangle_Z.\end{aligned}$$

d'où

$$\partial_{\theta} \mathcal{L}(y, \theta, \xi) = \partial_{\theta} \mathcal{J}(y, \theta) + \partial_{\theta} \mathcal{N}(y, \theta)^* \xi \quad (1.37)$$

Donc, l'équation (1.35) est également établie par

$$\langle \partial_{\theta} \mathcal{L}(\bar{y}, \bar{\theta}, \xi), \theta - \bar{\theta} \rangle_Z \geq 0 \quad \forall \theta \in K.$$

Finalement, le système d'optimalité formé par les équations (1.33), (1.34) et (1.35) peut être obtenu via l'annulation des dérivées partielles de la fonction de Lagrange par rapport à ξ , y et θ comme suit

$$\begin{aligned}\partial_{\xi} \mathcal{L}(\bar{y}, \bar{\theta}, \xi) = 0 &\Rightarrow \mathcal{N}(\bar{y}, \bar{\theta}) = 0 \\ \partial_y \mathcal{L}(\bar{y}, \bar{\theta}, \xi) = 0 &\Rightarrow \partial_y \mathcal{J}(\bar{y}, \bar{\theta}) + \partial_y \mathcal{N}(\bar{y}, \bar{\theta})^* \xi = 0 \\ (\theta - \bar{\theta})^T \partial_{\theta} \mathcal{L}(\bar{y}, \bar{\theta}, \xi) \geq 0 \quad \forall \theta \in K &\Rightarrow (\theta - \bar{\theta})^T [\partial_{\theta} \mathcal{J}(\bar{y}, \bar{\theta}) + \partial_{\theta} \mathcal{N}(\bar{y}, \bar{\theta})^* \xi] \geq 0 \quad \forall \theta \in K.\end{aligned}$$

En utilisant l'équation (1.37), il est maintenant possible d'évaluer les directions de descente indépendamment de la dimension du paramètre de contrôle. L'algorithme d'optimisation en utilisant l'approche des multiplicateurs de Lagrange est détaillé ci après.

1.3.3 Algorithme de contrôle optimal par l'approche des multiplicateurs de Lagrange

Les étapes de résolution du problème de contrôle optimal en utilisant la méthode des multiplicateurs de Lagrange sont répertoriées dans l'algorithme 5.

Algorithme 5 : Algorithme de contrôle optimal

1 initialisation de l'algorithme : $n = 0$ et $\theta^{(n)} = \theta^{init}$;

2 **while** $E > \varepsilon$ **do**

3 Résoudre l'équation d'état $\mathcal{N}(y^n, \theta^n) = 0$ pour $y^n = y(\theta^n)$;

4 Trouver ξ^n solution de l'équation adjointe

$$\partial_y \mathcal{J}(y(\theta^n), \bar{\theta}) + \partial_y \mathcal{N}(y(\theta^n), \bar{\theta})^* \xi = 0$$

5 Évaluation de la direction de descente par l'équation (1.37)

$$d^n = -\partial_{\theta} \mathcal{L}(y^n, \theta^n, \xi^n, \kappa^n)$$

6 Recherche de manière linéaire du pas de descente σ_n (règle d'Armijo (1.10));

7 Mise à jour des paramètres de contrôle (1.11)

$$\theta^{n+1} = P_K(\theta^n + \sigma_n d^n)$$

 Évaluation du critère de convergence

$$E = \frac{|\mathcal{L}(y^{n+1}, \theta^{n+1}, \xi^{n+1}) - \mathcal{L}(y^n, \theta^n, \xi^n)|}{|\mathcal{L}(y^{n+1}, \theta^{n+1}, \xi^{n+1})|}$$

8 **end**

L'évaluation à chaque itération n de $\partial_{\theta^n} \mathcal{L}$ nécessitera la résolution du problème d'état (1.26) et du problème adjoint (1.32) ce qui engendre des temps de calcul important. Il est à noter que même si le problème d'état est non linéaire, le problème adjoint est toujours linéaire, et donc il peut être beaucoup moins cher à résoudre que le problème direct. De plus, comme la résolution du problème adjoint se fait en remontant le temps, les exigences de stockage pour sa résolution sont malheureusement importantes. En effet, l'opérateur adjoint est une linéarisation de l'opérateur non linéaire. Si le modèle direct est stationnaire, cela ne représente pas une difficulté significative. Cependant, si celui ci dépend du temps, la trajectoire entière de la solution directe doit être disponible pour assembler le système adjoint. L'approche évidente pour rendre toute la trajectoire de la solution disponible consiste à stocker la valeur de chaque variable résolue. Cette approche est la plus simple, et c'est l'option la plus efficace si il y a suffisamment de stockage disponible sur

la machine. Cependant, pour de longues simulations avec de nombreux degrés de liberté, il est généralement impossible de stocker toute la trajectoire de la solution.

Pour s'affranchir de ces difficultés de stockage et de temps de résolution, il est possible d'utiliser les méthodes de réduction de modèles. Ces méthodes permettent de compresser l'information d'un problème physique dans un sous espace de petite dimension. Ainsi, un stockage mémoire très faible et un accès quasi-instantané à la dynamique temporelle du problème deviennent possible. Ces approches seront décrites au chapitre 2.

1.4 CONTRÔLE OPTIMAL DES ÉCOULEMENTS

On s'intéresse dans cette section au problème de contrôle optimal des écoulements. Plus particulièrement, nous allons présenter le problème de minimisation d'une fonctionnelle objectif dont les contraintes sont décrites par les equations de Navier-Stokes.

1.4.1 Problème de Navier-Stokes

Soit Ω un ouvert borné de \mathbb{R}^d ($d = 2, 3$) et $\Gamma = \Gamma_D \cup \Gamma_\sigma$ son bord tel que $\Gamma_D \cap \Gamma_\sigma = \emptyset$. On note par $L^2(\Omega) = [L^2(\Omega)]^d$ l'espace des fonctions réelles de carré intégrable dans Ω , et par $\langle \cdot, \cdot \rangle_{L^2(\Omega)}$ et $\|\cdot\|_{L^2(\Omega)}$ le produit scalaire et la norme associés. On introduit ensuite les sous espaces de $L^2(\Omega)$ suivants :

$$\begin{aligned} H^1(\Omega) &= \{v \in L^2(\Omega) : \nabla v \in L^2(\Omega)\} \\ L_0^2(\Omega) &= \left\{ v \in L^2(\Omega) : \int_{\Omega} v \, dx = 0 \right\} \\ H_{g, \Gamma_D}^1(\Omega) &= \{v \in L^2(\Omega) : v|_{\Gamma_D} = g\} \end{aligned}$$

Pour $p \geq 1$ et V un espace de Hilbert, on définit l'espace

$$L^p(0, T; V) = \left\{ v : (0, T) \longrightarrow V : v \text{ est mesurable et } \int_0^T \|v(t)\|_V^p \, dt < \infty \right\}.$$

Considérons maintenant un fluide Newtonien incompressible de viscosité cinématique constante ν , contenu dans Ω , et soumis à une force volumique externe f . Pour chaque $x \in \Omega$ et $t > 0$, on note par $\mathbf{u} = (u_1, \dots, u_d)(x, t)$ et $p(x, t)$ le champs de vitesse et la pression du fluide. L'équation du mouvement du fluide est décrite par les équations de Navier-Stokes adimensionnel :

$$\begin{cases} \partial_t \mathbf{u} - \frac{1}{Re} \Delta \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} + \nabla p = \mathbf{f}(\theta) & \text{dans } \Omega \times [0, T[\\ \nabla \cdot \mathbf{u} = 0 & \text{dans } \Omega \times [0, T[\\ \mathbf{u} = \mathbf{g} & \text{sur } \Gamma_D \times [0, T[\\ -\frac{1}{Re} \nabla \mathbf{u} \cdot \mathbf{n} + pn = 0 & \text{sur } \Gamma_\sigma \times [0, T[\\ \mathbf{u}(0) = \mathbf{u}_0 & \text{dans } \Omega \end{cases} \quad (1.38)$$

où Re désigne le nombre de Reynolds, et $\theta \in \mathbb{R}^{N_p}$ les paramètres de contrôle².

1.4.2 Formulation variationnelle des équations de Navier-Stokes :

Les formulations variationnelles du problème nécessiteront l'introduction des espaces fonctionnels suivants

$$\begin{aligned}\mathcal{P} &= L_0^2(\Omega), \\ \mathbf{V} &= \mathbf{H}_{g,\Gamma_D}^1(\Omega), \\ \mathbf{V}_0 &= \mathbf{H}_{0,\Gamma_D}^1(\Omega), \\ \mathbf{V}_{div} &= \{\mathbf{v} \in \mathbf{V} : \nabla \cdot \mathbf{v} = 0\} \subset \mathbf{H}^1(\Omega), \\ \mathbf{V}_{0,div} &= \{\mathbf{v} \in \mathbf{V}_0 : \nabla \cdot \mathbf{v} = 0\} \subset \mathbf{H}^1(\Omega), \\ \mathbf{H} &= \overline{\mathbf{V}_{div}}^{L^2(\Omega)} \subset L^2(\Omega)\end{aligned}$$

On définit les formes linéaires, bilinéaires et trinéaires suivantes :

$$\begin{aligned}m : \mathbf{H}^1(\Omega) \times \mathbf{H}^1(\Omega) &\longrightarrow \mathbb{R} \\ (\mathbf{u}, \mathbf{v}) &\longrightarrow m(\mathbf{u}, \mathbf{v}) = \int_{\Omega} \mathbf{u} \mathbf{v} \, dx\end{aligned}$$

$$\begin{aligned}a : \mathbf{H}^1(\Omega) \times \mathbf{H}^1(\Omega) &\longrightarrow \mathbb{R} \\ (\mathbf{u}, \mathbf{v}) &\longrightarrow a(\mathbf{u}, \mathbf{v}) = \int_{\Omega} \nabla \mathbf{u} : \nabla \mathbf{v} \, dx\end{aligned}$$

$$\begin{aligned}c : \mathbf{H}^1(\Omega) \times \mathbf{H}^1(\Omega) \times \mathbf{H}^1(\Omega) &\longrightarrow \mathbb{R} \\ (\mathbf{u}, \mathbf{v}, \mathbf{w}) &\longrightarrow c(\mathbf{u}, \mathbf{v}, \mathbf{w}) = \int_{\Omega} (\mathbf{u} \cdot \nabla) \mathbf{v} \cdot \mathbf{w} \, dx\end{aligned}$$

$$\begin{aligned}d : \mathbf{H}^1(\Omega) \times \mathcal{P} &\longrightarrow \mathbb{R} \\ (\mathbf{u}, q) &\longrightarrow d(\mathbf{u}, q) = - \int_{\Omega} q \nabla \cdot \mathbf{u} \, dx\end{aligned}$$

$$\begin{aligned}b : \mathbf{H}^1(\Omega) &\longrightarrow \mathbb{R} \\ \mathbf{v} &\longrightarrow b(\mathbf{v}) = \int_{\Omega} \mathbf{f} \mathbf{v} \, dx\end{aligned}$$

$$\begin{aligned}d_{\Gamma} : \mathbf{H}^1(\Omega) \times \mathcal{P} &\longrightarrow \mathbb{R} \\ (\mathbf{u}, q) &\longrightarrow d_{\Gamma}(\mathbf{u}, q) = \int_{\Gamma} q \mathbf{u} \cdot \mathbf{n} \, d\sigma\end{aligned}$$

2. Ici les paramètres de contrôle apparaissent dans le terme source des équations de Navier-Stokes (1.38). D'autres situations de contrôle peuvent être envisagées, notamment le contrôle sur le nombre de Reynolds ou sur des paramètres liés aux conditions aux limites sur une ou plusieurs portions du bord Γ .

$$a_\Gamma : \mathbf{H}^1(\Omega) \times \mathbf{H}^1(\Omega) \longrightarrow \mathbb{R}$$

$$(\mathbf{u}, \mathbf{v}) \longrightarrow a_\Gamma(\mathbf{u}, \mathbf{v}) = \int_\Gamma \nabla \mathbf{u} \cdot \mathbf{n} \mathbf{v} \, d\sigma$$

Soient $t \geq 0$ et \mathbf{u}_0 une condition initiale dans \mathbf{H} , pour tout couple de fonctions tests $(\mathbf{v}, q) \in \mathbf{V}_0 \times \mathcal{P}$, les relations suivantes sont vérifiées :

$$\begin{aligned} a_{\Gamma_D}(\mathbf{u}, \mathbf{v}) &= 0 \\ d_{\Gamma_D}(\mathbf{v}, p) &= 0 \\ -\frac{1}{Re} a_{\Gamma_\sigma}(\mathbf{u}, \mathbf{v}) + d_{\Gamma_\sigma}(\mathbf{v}, p) &= 0 \end{aligned}$$

La formulation faible associée au problème (1.38) s'exprime alors comme suit

$$\begin{cases} \text{Trouver } \mathbf{u} \in L^2(0, T; \mathbf{V}) \text{ et } p \in L^2(0, T; \mathcal{P}) \text{ tels que} \\ m(\partial_t \mathbf{u}, \mathbf{v}) + \frac{1}{Re} a(\mathbf{u}, \mathbf{v}) + c(\mathbf{u}, \mathbf{u}, \mathbf{v}) + d(\mathbf{v}, p) = b(\mathbf{v}), \quad \forall \mathbf{v} \in \mathbf{V}_0 \\ d(\mathbf{u}, q) = 0, \quad \forall q \in \mathcal{P} \end{cases} \quad (1.39)$$

Considérons maintenant des fonctions tests à divergence nulle $\mathbf{v} \in \mathbf{V}_{0,div}$, alors

$$d(\mathbf{v}, p) = 0$$

d'où la formulation variationnelle compacte des équations de Navier-Stokes :

$$\begin{cases} \text{Trouver } \mathbf{u} \in L^2(0, T; \mathbf{V}_{div}) \text{ et } p \in L^2(0, T; \mathcal{P}) \text{ tels que} \\ m(\partial_t \mathbf{u}, \mathbf{v}) + \frac{1}{Re} a(\mathbf{u}, \mathbf{v}) + c(\mathbf{u}, \mathbf{u}, \mathbf{v}) = b(\mathbf{v}), \quad \forall \mathbf{v} \in \mathbf{V}_{0,div} \end{cases} \quad (1.40)$$

1.4.3 Formulation du problème de contrôle optimal

Considérons la fonctionnelle objectif

$$\mathcal{J}(\mathbf{u}, \theta) = \frac{1}{2} \int_0^T \int_\Omega |\mathbf{u} - \hat{\mathbf{u}}|^2 \, dx \, dt + \frac{1}{2} \int_\Omega |\mathbf{u}(T) - \hat{\mathbf{u}}(T)|^2 \, dx + \frac{\omega}{2} |\theta|^2 \quad (1.41)$$

où $\hat{\mathbf{u}}$ désigne un champs de vitesse cible, et ω un coefficient de régularisation. En toute évidence, les variables d'état (\mathbf{u}, p) sont liées au paramètre de contrôle θ via l'équation de Navier-Stokes (1.38). En tenant compte de cette relation, le problème de minimisation sous contraintes consiste alors à déterminer θ qui minimise la fonctionnelle objectif (1.41). Le problème de contrôle est alors formulé comme suit

$$\begin{cases} \min_{\theta} \mathcal{J}(\mathbf{u}, \theta) \quad \text{tel que} \\ \mathbf{u} \in L^2(0, T; \mathbf{V}) \text{ et } p \in L^2(0, T; \mathcal{P}) \text{ vérifient} \\ m(\partial_t \mathbf{u}, \mathbf{v}) + \frac{1}{Re} a(\mathbf{u}, \mathbf{v}) + c(\mathbf{u}, \mathbf{u}, \mathbf{v}) + d(\mathbf{v}, p) = b(\mathbf{v}), \quad \forall \mathbf{v} \in \mathbf{V}_0 \\ d(\mathbf{u}, q) = 0, \quad \forall q \in \mathcal{P} \end{cases} \quad (1.42)$$

Le problème sous contraintes 1.42 est ensuite transformé en un problème d'optimisation sans contrainte par la méthode des multiplicateurs de Lagrange. La fonctionnelle de Lagrange dans ce cas s'écrit

$$\begin{aligned} \mathcal{L}(\mathbf{u}, p, \theta, \boldsymbol{\xi}, \pi, \kappa) = & \mathcal{J}(\mathbf{u}, p, \theta) + \int_0^T \left[-m(\mathbf{u}, \partial_t \boldsymbol{\xi}) + \frac{1}{Re} a(\mathbf{u}, \boldsymbol{\xi}) + c(\mathbf{u}, \mathbf{u}, \boldsymbol{\xi}) + d(\boldsymbol{\xi}, p) - b(\boldsymbol{\xi}) \right] dt \\ & + \int_0^T [d_\Gamma(\boldsymbol{\xi}, p) - \nu a_\Gamma(\mathbf{u}, \boldsymbol{\xi})] dt + \int_0^T d(\mathbf{u}, \pi) dt \\ & + [m(\mathbf{u}, \boldsymbol{\xi})]_0^T + \int_0^T \int_{\Gamma_D} \kappa(\mathbf{u} - \mathbf{g}) d\sigma dt \end{aligned}$$

où $\boldsymbol{\xi}$, π et κ sont les multiplicateurs de Lagrange associés respectivement à l'équation de quantité de mouvement, l'équation de continuité et l'équation des conditions aux limites de Dirichlet. Toute solution du problème de minimisation (1.42) satisfait l'équation variationnelle suivante

$$D\mathcal{L}(\mathbf{u}, p, \theta, \boldsymbol{\xi}, \pi, \kappa)(\delta\mathbf{u}, \delta p, \delta\theta, \delta\boldsymbol{\xi}, \delta\pi, \delta\kappa) = 0$$

pour toute direction $(\delta\mathbf{u}, \delta p, \delta\theta, \delta\boldsymbol{\xi}, \delta\pi, \delta\kappa)$. L'équation adjointe est obtenue en satisfaisant l'équation

$$\langle \partial_{\mathbf{u}} \mathcal{L}(\mathbf{u}, p, \theta, \boldsymbol{\xi}, \pi, \kappa), \delta\mathbf{u} \rangle + \langle \partial_p \mathcal{L}(\mathbf{u}, p, \theta, \boldsymbol{\xi}, \pi, \kappa), \delta p \rangle = 0 \quad (1.43)$$

Le terme $\langle \partial_{\mathbf{u}} c(\mathbf{u}, \mathbf{u}, \boldsymbol{\xi}), \delta\mathbf{u} \rangle$ peut s'écrire en intégrant par parties comme suit

$$\begin{aligned} \langle \partial_{\mathbf{u}} c(\mathbf{u}, \mathbf{u}, \boldsymbol{\xi}), \delta\mathbf{u} \rangle &= \int_{\Omega} (\delta\mathbf{u} \cdot \nabla) \mathbf{u} \cdot \boldsymbol{\xi} dx + \int_{\Omega} (\mathbf{u} \cdot \nabla) \delta\mathbf{u} \cdot \boldsymbol{\xi} dx \\ &= \int_{\Omega} (\nabla \mathbf{u})^T \boldsymbol{\xi} \delta\mathbf{u} dx - \int_{\Omega} (\mathbf{u} \cdot \nabla) \boldsymbol{\xi} \delta\mathbf{u} dx + \int_{\Gamma} (\mathbf{u} \cdot \mathbf{n}) \boldsymbol{\xi} \delta\mathbf{u} d\sigma \end{aligned}$$

L'intégration par parties du terme $a(\mathbf{u}, \boldsymbol{\xi})$ permet d'écrire

$$a(\delta\mathbf{u}, \boldsymbol{\xi}) = - \int_{\Omega} \Delta \boldsymbol{\xi} \delta\mathbf{u} dx + \int_{\Gamma} \nabla \boldsymbol{\xi} \cdot \mathbf{n} \delta\mathbf{u} d\sigma$$

De même pour le terme $d(\mathbf{u}, \pi)$, on a

$$d(\delta\mathbf{u}, \pi) = - \int_{\Omega} \pi \nabla \cdot \delta\mathbf{u} dx = \int_{\Omega} \nabla \pi \delta\mathbf{u} dx - \int_{\Gamma} \pi \delta\mathbf{u} \cdot \mathbf{n} d\sigma$$

Après substitution des termes ci dessus, l'équation (1.43) peut alors s'écrire comme suit

$$\begin{aligned} & - \int_0^T \int_{\Omega} \partial_t \boldsymbol{\xi} \delta\mathbf{u} dx dt - \frac{1}{Re} \int_{\Omega} \Delta \boldsymbol{\xi} \delta\mathbf{u} dx dt + \int_0^T \int_{\Omega} (\nabla \mathbf{u})^T \boldsymbol{\xi} \delta\mathbf{u} dx dt - \int_0^T \int_{\Omega} (\mathbf{u} \cdot \nabla) \boldsymbol{\xi} \delta\mathbf{u} dx dt \\ & + \int_0^T \int_{\Gamma} (\mathbf{u} \cdot \mathbf{n}) \boldsymbol{\xi} \delta\mathbf{u} d\sigma dt - \frac{1}{Re} \int_0^T \int_{\Gamma} \nabla \delta\mathbf{u} \cdot \mathbf{n} \boldsymbol{\xi} d\sigma dt + \frac{1}{Re} \int_{\Gamma} \nabla \boldsymbol{\xi} \cdot \mathbf{n} \delta\mathbf{u} d\sigma \\ & + \int_{\Omega} \nabla \pi \delta\mathbf{u} dx - \int_{\Gamma} \pi \delta\mathbf{u} \cdot \mathbf{n} d\sigma + \int_{\Omega} \boldsymbol{\xi}(T) \delta\mathbf{u}(T) dx - \int_{\Omega} \boldsymbol{\xi}(0) \delta\mathbf{u}(0) dx \\ & + \int_0^T \int_{\Omega} (\mathbf{u} - \hat{\mathbf{u}}) \delta\mathbf{u} dx dt + \int_{\Omega} (\mathbf{u}(T) - \hat{\mathbf{u}}(T)) \delta\mathbf{u}(T) dx \\ & - \int_0^T \int_{\Omega} \delta p \nabla \cdot \boldsymbol{\xi} dx dt + \int_0^T \int_{\Gamma} \delta p \boldsymbol{\xi} \cdot \mathbf{n} d\sigma dt + \int_0^T \int_{\Gamma_D} \kappa \delta\mathbf{u} d\sigma dt = 0. \end{aligned}$$

pour toutes directions $\delta \mathbf{u}$ et δp telles que

$$\begin{aligned} -\frac{1}{Re} \nabla \delta \mathbf{u} \cdot \mathbf{n} + \delta p \mathbf{n} &= 0, \quad \text{sur } \Gamma_\sigma \times]0, T] \\ \delta \mathbf{u}(0) &= 0, \quad \text{dans } \Omega \\ \nabla \cdot \delta \mathbf{u} &= 0, \quad \text{dans } \Omega \times]0, T] \\ \delta \mathbf{u} &= 0, \quad \text{sur } \Gamma_D \times]0, T] \end{aligned}$$

Il s'ensuit que l'équation adjointe associée au problème d'optimisation (1.42) s'exprime par

$$\begin{cases} -\partial_t \xi - \nu \Delta \xi + (\nabla \mathbf{u})^T \xi - (\mathbf{u} \cdot \nabla) \xi + \nabla \pi = \hat{\mathbf{u}} - \mathbf{u} & \text{dans } \Omega \times [0, T[\\ \nabla \cdot \xi = 0 & \text{dans } \Omega \times [0, T[\\ \xi = 0 & \text{sur } \Gamma_D \times [0, T[\\ \frac{1}{Re} \nabla \mathbf{u} \cdot \mathbf{n} + \mathbf{u} \cdot \mathbf{n} \xi - \pi \mathbf{n} = 0 & \text{sur } \Gamma_\sigma \times [0, T[\\ \xi(T) = \hat{\mathbf{u}}(T) - \mathbf{u}(T) & \text{dans } \Omega \end{cases} \quad (1.44)$$

Finalement, en utilisant les équations adjointes (1.44), les directions de descente liées au paramètre d'optimisation θ sont évaluées comme suit

$$d_\theta = -\partial_\theta \mathcal{L} = -\int_0^T \int_\Omega \xi \partial_\theta f \, dx \, dt - \omega \theta \quad (1.45)$$

En pratique, le problème de contrôle (1.42) est résolu en utilisant l'algorithme 5. Dans l'algorithme, les directions de descente (étape 6) sont évaluées en utilisant l'équation (1.45), où ξ est solution du système adjoint (1.44) et \mathbf{u} est solution du système d'état (1.38).

1.5 CONCLUSION DU CHAPITRE

Dans ce chapitre, le problème d'optimisation sous contraintes dans sa forme générale a été présenté et les conditions d'optimalités ont été dérivées. En pratique la complexité de ce type de problème est résolue à l'aide de la méthode des multiplicateurs de Lagrange, qui transforme le problème d'optimisation sous contraintes en un problème d'optimisation sans contrainte. L'algorithme utilisé pour la résolution de ce nouveau problème est un algorithme de descente classique dont les directions de descente sont calculées via l'évaluation du gradient de la fonctionnelle. Ce type d'algorithme nécessite la connaissance d'un pas de descente qui doit être soigneusement choisi afin d'éviter les oscillations autour de l'optimum. Il est naturel de chercher le pas optimal qui réalise une descente maximale suivant une direction donnée. Pour des problèmes d'optimisation simples, le pas optimal peut être analytiquement calculé. Malheureusement dans le cas des problèmes que l'on traite, il est impossible de trouver un pas optimal de manière analytique. Pour cela, nous utiliserons une méthode de recherche linéaire basée sur la règle d'Armijo.

La résolution du problème de contrôle optimal avec la méthode classique basée sur l'état adjoint est non réalisable en temps quasi-réel. En effet, pour les problèmes non linéaires instationnaires, l'évaluation du gradient de la fonctionnelle nécessite la résolution de l'équation d'état et de l'équation adjointe, ce qui entraîne des temps de calcul importants ainsi qu'un stockage mémoire volumineux. Il est alors indispensable d'avoir recours à des méthodes qui réduisent considérablement la dimension du problème d'origine. Une solution possible est d'utiliser les méthodes de réduction de modèles. Ces méthodes sont très répandues dans de nombreux domaines d'application et permettent pour les problèmes dynamiques, un faible coût de stockage mémoire, et un accès rapide à la dynamique temporelle. Dans le chapitre suivant, quelques méthodes de réduction de modèles existantes seront présentées.

RÉDUCTION DE MODÈLES ET OPTIMISATION

2

SOMMAIRE

2.1	INTRODUCTION	26
2.2	CONSTRUCTION DES BASES RÉDUITES	27
2.2.1	Méthode BT (Balanced Truncation)	28
2.2.2	Méthode POD (Proper Orthogonal Decomposition)	30
2.2.3	Méthode POD récursive	39
2.2.4	Méthode DMD (Dynamic Modes Decomposition)	40
2.3	CONSTRUCTION DU MODÈLE D'ORDRE RÉDUIT (ROM)	42
2.3.1	Cas général	42
2.3.2	Construction du ROM pour les équations de Navier-Stokes	43
2.4	CONTRÔLE OPTIMAL RÉDUIT	48
2.4.1	Position du problème de contrôle optimal réduit	48
2.4.2	Optimisation réduite "Differentiate Then Reduce"	48
2.4.3	Optimisation réduite "Reduce Then Differentiate"	50
2.4.4	Exemple d'illustration	52
2.5	CONCLUSION DU CHAPITRE	59

2.1 INTRODUCTION

Aujourd'hui, les techniques de réduction de modèles deviennent de plus en plus utilisées dans de nombreuses applications nécessitant la manipulation de données volumineuses et/ou la résolution de problèmes de grande dimension. Ceci s'explique par le fait que ces méthodes facilitent, d'une part, le stockage et le traitement de données de très grande taille et permettent, d'autre part, d'obtenir très rapidement et à faible coût la dynamique temporelle de phénomènes complexes. En effet, les modèles d'ordre réduit (ROM) ont pour but d'approximer la solution d'un phénomène complexe dans un sous-espace vectoriel de faible dimension capable de capturer les caractéristiques dominantes et les propriétés les plus élémentaires du phénomène physique étudié. Ces techniques consistent donc à réduire la dimensionnalité d'un ensemble de données composé d'un très grand nombre de variables (corrélées), tout en conservant autant que possible la variation présente dans cet ensemble. Cela revient à approximer une fonction $h(t, x)$ sur un domaine d'intérêt $[0, T] \times \Omega$ par une combinaison linéaire d'un nombre fini de fonctions de base $\varphi^k(x)$ sous la forme

$$h(t, x) \approx \sum_{k=1}^q \alpha^k(t) \varphi^k(x) \quad (2.1)$$

où q est censé être significativement petit et α^k représentent les coefficients temporels, pouvant être obtenus via la résolution d'un système d'équations différentiels résultant de la projection de Galerkin du modèle complet sur les fonctions de base $\varphi^k(x)$.

La technique de réduction modèle la plus utilisée est la décomposition de Karhunen-Loève, appelée également ACP (Analyse en Composante Principale) ou POD (Proper Orthogonal Decomposition). Dans ce cas, les données sont transformées en un nouvel ensemble de fonctions non corrélées ordonnées de sorte que les premiers éléments conservent la majeure partie de la variation présente dans l'ensemble de données original. Les nouvelles fonctions sont appelées composantes principales. Les dernières composantes identifient les directions dans lesquelles il existe très peu de variations. Ainsi, une fois que les composantes principales sont identifiées, il est possible de réduire la dimension de l'espace d'origine des variables, en ne gardant que les premières composantes principales.

Cette approche a été appliquée dans de nombreux domaines : en micro électro mécanique par Liang et al. [21], en aéroélasticité [22, 23], en dynamique des structures [24][25][26], en détection de dommages [27][28][29], en analyse modale [30], en dynamique des réactions chimiques [31] ...etc. Dans le cadre de la mécanique des fluides, cette approche a été introduite en 1967 par Lumley [32] pour identifier les structures spatialement cohérentes d'un écoulement turbulent. Aubry et al., à la fin des années 80, ont utilisé la base POD pour construire, par projection de Galerkin des équations de Navier-Stokes sur cette base, un modèle d'ordre réduit capable de reproduire la dynamique pariétale de l'écoulement turbulent dans un canal [33]. Depuis ces travaux, l'approche POD/ROM a été appliquée pour étudier de nombreux écoulements, notamment l'écoulement dans une cavité en-

traîné isotherme [34], anisotherme [35] ou encore l'écoulement autour d'un obstacle [36]. Elle a été également utilisée pour étudier la dispersion de polluants dans les écoulements [37, 38, 39, 40] et en interaction fluide structure (FSI) [41, 42]. D'autres applications de la POD en hydrodynamique peuvent être trouvées dans les travaux de Aubry et al. [43], Berkooz et al. [44][45], Glezer et al. [46], Ciliberto et al. [47][48], Arndt et al. [49], Tinney [50], Hong et al. [51], Moin et al. [52], Sirovich et al. [53][54][55], Merzlotta et al. [56], Glegg et al. [57], O'Donnell et al. [58]. Bien que la POD soit l'approche la plus couramment utilisée, d'autres techniques de recherche de bases de faible dimension existent. On peut citer par exemple, la méthode BT (Balanced Truncation) adaptée à la réduction de système de contrôle dont la dynamique est linéarisable [59], la méthode DMD (Dynamic Modes Decomposition) développée par Schmidt [60] ou la méthode CVT (Centroidal Voronoi Tessellations) [61].

Dans ce chapitre, un aperçu de quelques méthodes de réduction de modèles existantes est tout d'abord donné, avec un focus sur la méthode POD dans sa version discrète. La construction du modèle réduit par la projection de Galerkin et la dérivation du modèle réduit associé aux équations de Navier-Stokes isothermes seront détaillées dans la section 2.3.2. Enfin, l'utilisation des modèles d'ordre réduit dans le contexte du contrôle optimal est examinée dans la section 2.4.

2.2 CONSTRUCTION DES BASES RÉDUITES

On rappelle ici que l'objectif de cette section est de chercher à approximer une fonction $h(t, x)$ appartenant à l'espace de Hilbert $L^2(0, T; H^1(\Omega))$ par une combinaison linéaire de produits de fonctions appartenant à l'espace de Hilbert tensoriel $[0, T] \otimes H^1(\Omega)$, comme suit

$$h(t, x) \approx \sum_{k=1}^q \alpha^k(t) \varphi^k(x) \quad (2.2)$$

en espérant raisonnablement que l'approximation devienne exacte quand q s'approche de l'infini. L'espace de Hilbert V est équipé d'un produit scalaire $\langle \cdot, \cdot \rangle$ dont la norme associée est notée $\| \cdot \|$. Les coefficients notés α^k représentent les amplitudes inconnues de l'approximation. Une fois les fonctions de base φ^k sont choisies, les valeurs d'amplitude sont obtenues par un processus de minimisation défini au sens des moindres carrés comme suit

$$\{\alpha^1, \dots, \alpha^q\} = \min_{\{\zeta^1, \dots, \zeta^q\}} \left\| h(t, x) - \sum_{k=1}^q \zeta^k(t) \varphi^k(x) \right\|^2 \quad (2.3)$$

En toute évidence, pour la même fonction $h(t, x)$, la représentation (2.2) n'est pas unique. Le choix des fonctions $\varphi^k(x)$ qui forment une base $\boldsymbol{\varphi}(x)$ pour la fonction $h(t, x)$ est arbitraire. En outre, pour chaque choix fait pour la base, un ensemble d'amplitudes différent correspond. Dans la plupart des cas, la précision cible pour l'approximation peut être

atteinte lorsque q est assez grand. Il est naturel donc de chercher une base capable d'approximer le mieux possible $h(t, x)$ pour un nombre de modes q le plus petit possible. Il existe une liberté totale dans le choix des fonctions de base $\varphi^k(x)$, à condition qu'elles soient linéairement indépendantes. Le choix de l'ensemble orthonormal vérifiant la propriété

$$\langle \varphi^k(x), \varphi^l(x) \rangle = \delta_{kl}$$

présente des avantages et simplifie la tâche de détermination des amplitudes de manière que chaque α^k dépend uniquement de φ^k

$$\alpha^k(t) = \langle h(t, x), \varphi^k(x) \rangle$$

En outre, cette base doit être optimale, dans le sens où pour chaque valeur de q , l'approximation doit être aussi bonne que possible en terme d'erreur au sens des moindres carrés. En d'autres termes, le but est de chercher une suite de fonctions orthonormales telles que les q premiers termes assurent la meilleure approximation à q termes. Une fois trouvée, ces fonctions de bases ordonnées sont appelées les modes propres associés à la fonction $h(t, x)$. Dans les paragraphes suivants, différentes méthodes de construction de bases réduites seront présentées et détaillées.

2.2.1 Méthode BT (Balanced Truncation)

Dans cette section, nous rappelons la méthode BT (Balanced Truncation). Soit le système dynamique linéaire suivant :

$$\begin{cases} \frac{dx}{dt} = Ax(t) + B\theta(t) \\ z(t) = Cx(t) + D\theta(t) \end{cases} \quad (2.4)$$

où $A \in \mathbb{R}^{N_x \times N_x}$ représente la matrice du système dynamique, $B \in \mathbb{R}^{N_x \times d_p}$ la matrice d'entrée, $C \in \mathbb{R}^{d_p \times N_x}$ la matrice de sortie, $D \in \mathbb{R}^{d_p \times d_p}$ la matrice de retour, $x \in \mathbb{R}^{N_x}$ le vecteur d'état d'entrée, $z \in \mathbb{R}^{d_p}$ le vecteur d'état de sortie, et $\theta(t)$ le vecteur d'entrée. La réduction de ce système consiste à produire un système similaire

$$\begin{cases} \frac{dx_q}{dt} = A_q x_q(t) + B_q \theta(t) \\ z_q(t) = C_q x_q(t) + D\theta(t) \end{cases} \quad (2.5)$$

d'ordre nettement inférieur à celui du système originel ($q \ll N_x$), pour lequel $z_q(t)$ est une bonne approximation de $z(t)$ pour les entrées d'intérêt $\theta(t)$. Dans ce cas, $A_q \in \mathbb{R}^{q \times q}$, $B_q \in \mathbb{R}^{q \times d_p}$, $C_q \in \mathbb{R}^{d_p \times q}$, $D \in \mathbb{R}^{d_p \times d_p}$, et la dimension de $\theta(t)$ et z reste inchangés.

Un système linéaire de la forme (2.4) est dit équilibré si les Grammiens

$$W_c = \int_0^\infty e^{At} B B^T e^{A^T t} dt \quad W_o = \int_0^\infty e^{A^T t} C C^T e^{At} dt \quad (2.6)$$

admettent des solutions diagonales identiques

$$W_c = W_o = \Sigma = \text{diag}(\sigma_1, \dots, \sigma_{N_x}). \quad (2.7)$$

où W_c et W_o sont respectivement les opérateurs Grammiens de contrôlabilité et d'observabilité. Il s'avère que tout système peut être transformé en une forme équilibrée en utilisant un changement de base $\tilde{x} = Qx$. Les deux matrices Grammiennes sont transformées par une transformation de congruence comme suit

$$\tilde{W}_c = QW_cQ^T, \quad (2.8)$$

$$\tilde{W}_o = Q^{-T}W_oQ^{-1}, \quad (2.9)$$

$$\tilde{W}_c\tilde{W}_o = QW_cW_oQ^{-1}. \quad (2.10)$$

En utilisant le fait que deux matrices symétriques peuvent être diagonalisées par le biais d'une transformation de congruence appropriée, il est possible de trouver la matrice Q qui résulte en deux matrices diagonales \tilde{W}_c et \tilde{W}_o telles que $\tilde{W}_c = \tilde{W}_o$. Les valeurs propres de W_cW_o restent invariantes par similarité. Ces valeurs propres sont appelées valeurs singulières de Hankel [62]. Elles emmagasinent les informations importantes du système (2.4). En particulier, les petites valeurs propres correspondent aux sous systèmes qui ont un effet faible sur le comportement entrée sortie du système originel. Ces derniers sont presque non observables ou non contrôlables. En pratique, l'astuce pour calculer les valeurs singulières de Hankel est d'observer que les matrices Grammiennes sont des solutions Hermitiennes définies positives et uniques des équations dites de Lyapunov :

$$AW_c + W_cA^T + BB^T = 0, \quad (2.11)$$

$$A^TW_o + W_oA + C^TC = 0. \quad (2.12)$$

Pour un système contrôlable et observable, les deux matrices Grammiennes W_c et W_o sont définies positives. Donc par décomposition de Cholesky, il existe donc L_c et L_o deux matrices triangulaires inférieures telles que

$$W_c = L_cL_c^T \quad \text{et} \quad W_o = L_oL_o^T \quad (2.13)$$

Soit $L_o^TL_c = U\Sigma V^T$ la décomposition en valeurs singulières de $L_o^TL_c$. La matrice de transformation Q s'écrit

$$Q = L_cV\Sigma^{-1/2}, \quad (2.14)$$

$$Q^{-1} = \Sigma^{-1/2}U^TL_o^T \quad (2.15)$$

En utilisant la transformation par similarités, on introduit les matrices suivantes

$$\tilde{A} = Q^{-1}AQ, \quad (2.16)$$

$$\tilde{B} = Q^{-1}B, \quad (2.17)$$

$$\tilde{C} = CQ. \quad (2.18)$$

Le modèle d'ordre réduit est finalement obtenu par troncature du système originel, en partitionnant les matrices Σ , \tilde{A} , \tilde{B} et \tilde{C} comme suit

$$\Sigma = \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \quad \tilde{A} = \begin{bmatrix} \tilde{A}_{11} & \tilde{A}_{12} \\ \tilde{A}_{22} & \tilde{A}_{22} \end{bmatrix} \quad \tilde{B} = \begin{bmatrix} \tilde{B}_1 \\ \tilde{B}_2 \end{bmatrix} \quad \tilde{C} = \begin{bmatrix} \tilde{C}_1 & \tilde{C}_2 \end{bmatrix} \quad (2.19)$$

telles que $\Sigma_1 \in \mathbb{R}^{q \times q}$ contient les valeurs singulières ayant une contribution significative dans le système, $\tilde{A}_{11} \in \mathbb{R}^{q \times q}$, $\tilde{B}_1 \in \mathbb{R}^{q \times d_p}$, $\tilde{C}_1 \in \mathbb{R}^{d_p \times q}$. Le modèle réduit (2.5) s'écrit alors

$$\begin{cases} \frac{dx_q}{dt} = A_{11}x_q(t) + B_1\theta(t) \\ z_q(t) = C_1x_q(t) + D\theta(t) \end{cases} \quad (2.20)$$

L'un des aspects attractifs du modèle réduit ainsi calculé est que l'erreur engendrée par la troncature est bornée [63][64] et s'exprime par :

$$\sigma_q \leq \|H(s) - \tilde{H}_q(s)\|_\infty \leq 2 \sum_{q=q+1}^{N_x} \sigma_q \quad (2.21)$$

où s est une fréquence complexe, et les fonctions H et \tilde{H} sont les fonctions de transfert données par

$$\begin{aligned} H(s) &= D + C(sI - A)^{-1}B, \\ \tilde{H}(s) &= \tilde{D} + \tilde{C}(sI - \tilde{A})^{-1}\tilde{B}. \end{aligned}$$

Cependant, la mise en œuvre numérique de cette méthode nécessite des temps de calculs importants, principalement à cause de la résolution des équations de Lyapunov. D'autre part, cette approche est applicable uniquement aux systèmes linéaires de la forme des équations (2.4). Ceci est handicapant pour une utilisation en mécanique des fluides, car dans ce cas les systèmes doivent être linéarisables.

2.2.2 Méthode POD (Proper Orthogonal Decomposition)

Étant donné que les applications traitées dans ce manuscrit sont de nature numérique, seule la version discrète de la POD sera présentée. On trouve généralement cette approche dans la littérature sous le nom de l'Analyse en Composante Principale (PCA), bien que certains auteurs l'appellent également décomposition de Karhunen-Loève discrète. On désignera par la suite par A la matrice éléments finis symétrique définie positive, qui sert pour définir la norme subordonnée $\|\cdot\|_A$ liée au produit scalaire choisi pour la construction des fonctions de base $\varphi^1, \dots, \varphi^q$. On choisira éventuellement A la matrice de masse qui correspond au produit scalaire usuel de $L^2(\Omega)$. Si on choisit le produit scalaire euclidien, l'opérateur A correspond à la matrice identité.

a) Minimisation de l'erreur d'approximation

Supposons que nous avons un ensemble de N_s vecteurs u_l de dimension N_x , concaténés par colonnes dans une matrice X . Notre objectif est de trouver la représentation la plus précise de X dans un sous-espace W de dimension $q < N_s$. Si nous désignons par $\varphi^1, \varphi^2, \dots, \varphi^q$ la base de W orthonormée par rapport à la norme subordonnée $\|\cdot\|_A$, alors chaque vecteur de l'ensemble d'origine peut être écrit comme

$$u_l \approx \sum_{k=1}^q \alpha^{lk} \varphi^k, \quad l = 1, \dots, N_s \quad (2.22)$$

où α^{lk} sont les amplitudes correspondantes au l -ième vecteur dans le nouveau sous-espace W . Si $\boldsymbol{\varphi}$ désigne la matrice formée par les éléments de la base orthonormée φ^k qui engendrent W , et $\boldsymbol{\alpha}$ la matrice formée par les amplitudes α^{lk} , l'équation (2.22) peut être écrite sous forme matricielle comme

$$u_l \approx \boldsymbol{\varphi} \cdot \boldsymbol{\alpha}_l, \quad l = 1, \dots, N_s \quad (2.23)$$

L'erreur de cette approximation au sens des moindres carrés peut être exprimée par

$$E = \sum_{l=1}^{N_s} \left\| u_l - \sum_{k=1}^q \alpha^{lk} \varphi^k \right\|_A^2$$

Le but est maintenant de trouver une base $\boldsymbol{\varphi}$ formée par l'ensemble des vecteurs orthonormés $\varphi^1, \varphi^2, \dots, \varphi^q$ qui minimisent cette erreur. Le problème de minimisation s'écrit :

$$\min_{\substack{\boldsymbol{\alpha}, \boldsymbol{\varphi} \\ \boldsymbol{\varphi}^T \mathbf{A} \boldsymbol{\varphi} = \mathbf{I}_q}} \sum_{l=1}^{N_s} \|u_l - \boldsymbol{\varphi} \cdot \boldsymbol{\alpha}_l\|_A^2 \quad (2.24)$$

En développant l'expression de E on a

$$\begin{aligned} E &= \sum_{l=1}^{N_s} \|u_l\|_A^2 - 2 \sum_{l=1}^{N_s} u_l^T \mathbf{A} \left(\sum_{k=1}^q \alpha^{lk} \varphi^k \right) + \sum_{l=1}^{N_s} \sum_{k=1}^q \alpha^{lk^2} \\ &= \sum_{l=1}^{N_s} \|u_l\|_A^2 - 2 \sum_{l=1}^{N_s} \sum_{k=1}^q \alpha^{lk} u_l^T \mathbf{A} \varphi^k + \sum_{l=1}^{N_s} \sum_{k=1}^q \alpha^{lk^2} \end{aligned} \quad (2.25)$$

La dérivée partielle de E par rapport à α^{ij} s'exprime par :

$$\partial_{\alpha^{ij}} E = -2u_i^T \mathbf{A} \varphi^j + 2\alpha^{ij}$$

L'optimum de E au point α^{ij} est donc atteint lorsque

$$\alpha^{ij} = u_i^T \mathbf{A} \varphi^j$$

En substituant cette valeur optimale dans l'équation (2.25), on obtient :

$$E = \sum_{l=1}^{N_s} \|u_l\|_A^2 - 2 \sum_{l=1}^{N_s} \sum_{k=1}^q \left(u_l^T \mathbf{A} \varphi^k \right) u_l^T \mathbf{A} \varphi^k + \sum_{l=1}^{N_s} \sum_{k=1}^q \left(u_l^T \mathbf{A} \varphi^k \right)^2 \quad (2.26)$$

Étant donné que

$$u_l^T \mathbf{A} \varphi^k = \varphi^{kT} \mathbf{A} u_l$$

le deuxième terme de l'équation (2.26) peut être réécrit comme suit

$$2 \sum_{l=1}^{N_s} \sum_{k=1}^q \left(u_l^T \mathbf{A} \varphi^k \right) u_l^T \mathbf{A} \varphi^k = 2 \sum_{l=1}^{N_s} \sum_{k=1}^q \varphi^{kT} \mathbf{A} u_l u_l^T \mathbf{A} \varphi^k \quad (2.27)$$

D'autre part, le troisième terme de (2.26) peut être exprimé par

$$\sum_{l=1}^{N_s} \sum_{k=1}^q \left(u_l^T \mathbf{A} \varphi^k \right)^2 = \sum_{l=1}^{N_s} \sum_{k=1}^q \left(u_l^T \mathbf{A} \varphi^k \right) u_l^T \mathbf{A} \varphi^k = \sum_{l=1}^{N_s} \sum_{k=1}^q \varphi^{kT} \mathbf{A} u_l u_l^T \mathbf{A} \varphi^k \quad (2.28)$$

Substituant (2.27) et (2.28) dans (2.26), l'erreur totale à minimiser devient

$$E = \sum_{l=1}^{N_s} \|u_l\|^2 - \sum_{l=1}^{N_s} \sum_{k=1}^q \varphi^{kT} A u_l u_l^T A \varphi^k \quad (2.29)$$

ou encore

$$\begin{aligned} E &= \sum_{l=1}^{N_s} \|u_l\|^2 - \sum_{k=1}^q \varphi^{kT} A \left(\sum_{l=1}^{N_s} u_l u_l^T \right) A \varphi^k \\ &= \sum_{l=1}^{N_s} \|u_l\|^2 - \sum_{k=1}^q \varphi^{kT} A^{\frac{1}{2}} D A^{\frac{1}{2}} \varphi^k \end{aligned} \quad (2.30)$$

où D , est appelée matrice de covariance, est définie par :

$$D = A^{\frac{1}{2}} X X^T A^{\frac{1}{2}} \quad (2.31)$$

Le premier terme de l'équation (2.30) est constant et ne dépend que de l'ensemble des vecteurs initiaux (mais pas de la base choisie). Il s'ensuit alors que le problème de minimisation de l'erreur E , revient à un problème de maximisation du deuxième terme de l'équation (2.30)

$$\max_{\{\psi, \psi^T \psi = I_d\}} \sum_{k=1}^q \psi^{kT} D \psi^k \quad \text{avec} \quad \psi = A^{\frac{1}{2}} \varphi \quad (2.32)$$

Afin de résoudre ce problème de maximisation sous contraintes, on peut utiliser la méthode des multiplicateurs de Lagrange qui, comme mentionné au chapitre 1, permet de convertir le problème (2.32) en un problème de maximisation sans contrainte en introduisant la fonction de Lagrange

$$\mathcal{L}(\psi, \Lambda) = \sum_{k=1}^q \psi^{kT} D \psi^k - \sum_{k=1}^q \sum_{l=1}^q \lambda_{kl} (\psi^{kT} \psi^l - \delta_{lk}) \quad (2.33)$$

où $\Lambda = (\lambda_{ij}) \in \mathbb{R}^{q \times q}$. Sachant que la matrice D est symétrique, on peut écrire

$$\begin{aligned} \partial_{\psi^j} \mathcal{L}(\psi, \Lambda) &= 2D\psi^j - \sum_{l=1}^q \sum_{k=1}^q \lambda_{kl} \psi^l \delta_{kj} - \sum_{l=1}^q \sum_{k=1}^q \lambda_{kl} \psi^k \delta_{lj} \\ &= 2D\psi^j - \sum_{l=1}^q (\lambda_{jl} + \lambda_{lj}) \psi^l \end{aligned}$$

D'après la condition d'optimalité du Lagrangien, on a

$$D\psi^j = \frac{1}{2} \sum_{l=1}^q (\lambda_{jl} + \lambda_{lj}) \psi^l \quad (2.34)$$

Pour $q = 1$, on a $l = 1$ et l'équation (2.34) se réduit à

$$D\psi^1 = \lambda_1 \psi^1, \quad \text{avec} \quad \lambda_1 = \lambda_{11} \quad (2.35)$$

Pour $q \geq 1$, la condition d'optimalité du premier ordre du Lagrangien est

$$D\psi^j = \lambda_j \psi^j, \quad \forall j = 1, \dots, q, \quad \text{avec } \lambda_j = \lambda_{jj}. \quad (2.36)$$

Nous souhaitons montrer que les conditions d'optimalité pour la base $\{\psi^1, \psi^2, \dots, \psi^q, \psi^{q+1}\}$ de rang $q + 1$ sont données par

$$D\psi^j = \lambda_j \psi^j, \quad \forall j = 1, \dots, q + 1 \quad (2.37)$$

Par hypothèse, l'équation (2.36) est déjà vérifiée, il reste juste à montrer que

$$D\psi^{q+1} = \lambda_{q+1} \psi^{q+1} \quad (2.38)$$

D'après l'équation (2.34) on peut écrire

$$D\psi^{q+1} = \frac{1}{2} \sum_{k=1}^{q+1} (\lambda_{q+1,k} + \lambda_{k,q+1}) \psi^k \quad (2.39)$$

On a aussi

$$\begin{aligned} \lambda_{q+1} \psi^{lT} \psi^{q+1} &= \psi^{lT} D\psi^{q+1} \\ &= \frac{1}{2} \sum_{k=1}^{q+1} (\lambda_{q+1,k} + \lambda_{k,q+1}) \psi^{lT} \psi^k \\ &= \lambda_{q+1,l} + \lambda_{l,q+1} \end{aligned}$$

Étant donné que les vecteurs $\{\psi^1, \psi^2, \dots, \psi^q, \psi^{q+1}\}$ forment une base

$$\psi^{q+1T} \psi^l = 0, \quad \forall l \in 1, \dots, q$$

on déduit que

$$\lambda_{q+1,l} = -\lambda_{l,q+1}, \quad \forall l \in 1, \dots, q \quad (2.40)$$

Par substitution de (2.40) dans l'équation (2.39), on aboutit à

$$\begin{aligned} D\psi^{q+1} &= \frac{1}{2} \sum_{l=1}^q (\lambda_{q+1,l} + \lambda_{l,q+1}) \psi^l + \lambda_{q+1,q+1} \psi^{q+1} \\ &= \lambda_{q+1} \psi^{q+1} \end{aligned}$$

d'où la vérification de (2.37). En résumé, les conditions d'optimalités sont satisfaites si ψ^j et λ_j sont les vecteurs et valeurs propres correspondantes aux problèmes aux valeurs propres suivants

$$D\psi^j = \lambda_j \psi^j, \quad \forall j = 1, \dots, q. \quad (2.41)$$

En substituant (2.41) dans l'équation (2.30), l'erreur d'approximation devient

$$\begin{aligned} E &= \sum_{l=1}^{N_s} \|u_l\|^2 - \sum_{k=1}^q \psi^{kT} D\psi^k \\ &= \sum_{l=1}^{N_s} \|u_l\|^2 - \sum_{k=1}^q \psi^{kT} (\lambda_k \psi^k) \\ &= \sum_{l=1}^{N_s} \|u_l\|^2 - \sum_{k=1}^q \lambda_k \end{aligned} \quad (2.42)$$

On conclut finalement que l'erreur d'approximation est minimisée, si et seulement si la base $\boldsymbol{\varphi}$ est composée des q vecteurs propres qui correspondent aux q plus grandes valeurs propres du problème (2.41). L'erreur d'approximation dans ce cas peut être exprimée par le ratio entre les valeurs propres considérées et toutes les autres valeurs propres du problème (2.41), soit :

$$\mathcal{R}^q = \frac{\sum_{i=1}^q \lambda_i}{\sum_{i=1}^{N_s} \lambda_i} \quad (2.43)$$

Les étapes de construction de la base POD en utilisant la matrice de covariance sont résumées dans l'algorithme 6.

Algorithme 6 : Algorithme POD par covariance

- 1 Construire la matrice de covariance $\mathbf{D} \in \mathbb{R}^{N_x \times N_x}$ telle que

$$\mathbf{D} = \mathbf{A}^{\frac{1}{2}} \mathbf{X} \mathbf{X}^T \mathbf{A}^{\frac{1}{2}} \quad \text{où} \quad \mathbf{X} = [u_1 \ u_2 \ \cdots \ u_{N_s}]$$

- 2 Calculer les valeurs propres $\lambda_1 > \cdots > \lambda_{N_x}$ de \mathbf{D} et la matrice des vecteurs propres associés $\boldsymbol{\psi} = [\psi^1 \ \cdots \ \psi^{N_x}]$
- 3 Calculer la base POD $\boldsymbol{\varphi}$ en résolvant le système linéaire $\mathbf{A}^{\frac{1}{2}} \boldsymbol{\varphi} = \boldsymbol{\psi}$
- 4 Tronquer la base en ne gardant que les q premiers modes tels que $\mathcal{R}^q \leq \varepsilon$, où ε est fixé par l'utilisateur,
- 5 Calculer les coefficients temporels par projection :

$$\alpha^i = \mathbf{X}^T \mathbf{A} \boldsymbol{\varphi}^i, \quad \forall i \in \{1, \dots, N_x\}$$

Un aspect négatif lié à la dérivation des composantes principales par cette approche, est la grande taille ($N_x \times N_x$) du problème aux valeurs propres (2.41). La construction et résolution numérique de ce problème peut s'avérer très coûteuse, voire impossible. C'est typiquement le cas lorsque N_x représente le nombre de degrés de libertés et N_s le nombre d'instant temporels issus de la discrétisation d'un problème physique (généralement $N_x \gg N_s$). Ce fait justifie l'intérêt d'introduire la méthode des Snapshots [65], qui permet de chercher la même base en résolvant un problème aux valeurs propres de taille $N_s \times N_s$ beaucoup plus petite.

b) Méthode des Snapshots

Soit \mathbf{X} la matrice composée de N_s vecteurs d'origine u_l de dimension N_x . Supposons que le premier vecteur de base $\boldsymbol{\varphi}^1$ s'exprime comme combinaison linéaire de tous les u_l , c'est à dire

$$\boldsymbol{\varphi}^1 = \mathbf{X} v_1$$

où les coefficients inconnus de cette combinaison forment un vecteur v_1 de dimension N_s . En supposant que ce vecteur est unitaire (c'est à dire $v_1^T v_1 = 1$), le vecteur φ^1 qui contient les plus grandes variations de l'ensemble de vecteurs formant la matrice X est celui qui maximise la longueur

$$\delta = \varphi^{1T} A \varphi^1$$

Ce critère apparaît évident, puisque le vecteur inconnu φ^1 est défini comme combinaison linéaire de tous les vecteurs collectés dans la matrice X . Par conséquent, pour une matrice X fixe, le vecteur inconnu est défini de manière unique par l'ensemble des coefficients qui forment le vecteur v_1 . Une modification de ces coefficients sous la contrainte de former un vecteur unitaire changera évidemment non seulement la direction de φ^1 , mais aussi sa longueur. La direction qui correspond à celle avec la plus grande variance de tous les vecteurs de l'ensemble aura la plus grande longueur δ . Trouver φ^1 revient alors à résoudre le problème de maximisation sous contraintes suivant

$$\max_{\{v_1, v_1^T v_1 = 1\}} \left(v_1^T X^T A X v_1 \right) \quad (2.44)$$

Encore une fois, la méthode des multiplicateurs de Lagrange peut être utilisée pour simplifier la résolution de ce problème. On introduit la fonction de Lagrange correspondante

$$\mathcal{L}(v_1, \lambda_1) = v_1^T C v_1 - \lambda_1 (v_1^T v_1 - 1) \quad (2.45)$$

où

$$C = X^T A X$$

est la matrice de corrélation temporelle, et λ_1 est un multiplicateur de Lagrange. L'annulation de la différentielle de (2.45) par rapport à v_1 donne

$$C v_1 - \lambda_1 v_1 = 0 \quad (2.46)$$

En multipliant l'équation (2.46) par v_1^T on trouve

$$v_1^T C v_1 = \lambda_1$$

Ceci montre que la solution du problème (2.44) est obtenue en prenant la plus grande valeur propre λ_1 de la matrice C et le vecteur propre associé v_1 . De manière analogue on cherche φ^k tel que

$$\varphi^k = X v_k$$

L'expression générale pour construire ce k -ième vecteur de base optimale est donnée par

$$C v_k - \lambda_k v_k = 0 \quad (2.47)$$

Jusqu'à présent, seuls les vecteurs v_k ont été forcés à être unitaires. Par conséquent, cette dérivation n'aboutira pas forcément à des vecteurs de base normalisés φ^k . Comme il est

plus pratique de travailler avec une base orthonormée, les vecteurs v_k doivent être calibrés par des facteurs de normalisations z_k

$$\hat{v}_k = v_k z_k$$

La formule générale pour le k -ième vecteur propre devient alors

$$\hat{\phi}^k = \mathbf{X} \hat{v}_k = \mathbf{X} v_k z_k \quad (2.48)$$

où les $\hat{\phi}^k$ correspondent aux vecteurs de base normalisés. Alors les facteurs de normalisation sont alors obtenus en forçant le critère suivant

$$\hat{\phi}^{kT} \mathbf{A} \hat{\phi}^k = 1 \quad (2.49)$$

En substituant (2.48) dans l'équation (2.49) le critère devient

$$(\mathbf{X} v_k z_k)^T \mathbf{A} \mathbf{X} v_k z_k = 1 \quad (2.50)$$

Après regroupement des termes, et en utilisant le fait que z_k et un facteur de normalisation on écrit

$$v_k^T z_k \mathbf{X}^T \mathbf{A} \mathbf{X} v_k z_k = z_k^2 v_k^T \mathbf{C} v_k = 1$$

En utilisant maintenant (2.47), l'expression ci-dessus devient

$$z_k^2 v_k^T \lambda_k v_k = z_k^2 \lambda_k v_k^T v_k = 1$$

Puisque les v_k sont unitaires, cette equation s'écrit finalement

$$z_k^2 \lambda_k = 1$$

Le facteur de normalisation recherché est donc

$$z_k = \lambda_k^{-1/2}$$

Finalement, les fonctions de la base orthonormale POD sont définies par

$$\hat{\phi}^k = \mathbf{X} v_k \lambda_k^{-1/2} \quad (2.51)$$

D'après (2.47), v_k est solution du problème

$$\mathbf{X}^T \mathbf{A} \mathbf{X} v_k = \lambda_k v_k$$

En utilisant (2.51), on peut écrire

$$\mathbf{X}^T \mathbf{A} \hat{\phi}^k = \sqrt{\lambda_k} v_k$$

On multiplie les deux côtés de l'équation par $\mathbf{A}^{\frac{1}{2}} \mathbf{X}$

$$\mathbf{A}^{\frac{1}{2}} \mathbf{X} \mathbf{X}^T \mathbf{A} \hat{\phi}^k = \lambda_k \mathbf{A}^{\frac{1}{2}} \hat{\phi}^k$$

soit

$$DA^{\frac{1}{2}}\hat{\varphi}^k = \lambda_k A^{\frac{1}{2}}\hat{\varphi}^k$$

Sachant d'après (2.32) que $A^{\frac{1}{2}}\hat{\varphi}^k = \psi^k$, on retrouve bien que la méthode basée sur la matrice de covariance D et la méthode basée sur la matrice de corrélation temporelle C résultent en la même base POD. L'utilisateur est alors libre de décider laquelle parmi ces deux méthodes est la plus appropriée pour son application. Cependant, on notera que la matrice D est de taille $N_x \times N_x$, tandis que la matrice C est de taille $N_s \times N_s$. Comme il est toujours préférable de travailler sur des matrices de taille petite, il est donc raisonnable de choisir l'approche basée sur la matrice C dans les cas où $N_s \ll N_x$ et l'approche basée sur la matrice D dans le cas opposé.

Les étapes de construction de la base POD en utilisant la matrice de corrélation sont résumées dans l'algorithme 7.

Algorithme 7 : Algorithme POD par corrélation

- 1 Construire la matrice de corrélation C par

$$C = X^T A X \quad \text{où} \quad X = [u_1 \ u_2 \ \cdots \ u_{N_s}]$$

- 2 Calculer les valeurs propres $\lambda_1 > \cdots > \lambda_{N_s}$ de C et la matrice des vecteurs propres associés $V = [v_1 \ \cdots \ v_{N_s}]$.
- 3 Calculer la base POD $\varphi = [\varphi^1 \ \cdots \ \varphi^{N_s}]$ comme suit

$$\varphi^i = X v_i \lambda_i^{-1/2}, \quad \forall i \in \{1, \dots, N_s\}$$

- 4 Tronquer la base en ne gardant que les q premiers modes tels que $\mathcal{R}^q \leq \varepsilon$, où ε est fixé par l'utilisateur.
- 5 Calculer les coefficients temporels par projection

$$\alpha^i = X^T A \varphi^i, \quad \forall i \in \{1, \dots, N_s\}$$

c) POD par l'approche SVD

Une autre façon d'obtenir une base POD est d'utiliser la décomposition en valeurs singulières (SVD - Singular Value Decomposition). La SVD est dotée de deux propriétés importantes. Tout d'abord, cette décomposition existe pour toute matrice qu'elle soit carrée, rectangulaire, singulière, non singulière, creuse ou dense. Donc, peu importe le type de la matrice, nous savons qu'elle admet une décomposition en valeurs singulières. La deuxième propriété utile de la SVD est l'optimalité. C'est la méthode qui permet d'obtenir la meilleure approximation de rang faible d'une matrice quelconque au sens de la norme euclidienne (norme-2). Pour plus de détails sur la SVD, le lecteur intéressé peut se référer à [66].

La SVD consiste à décomposer une matrice quelconque Y de taille $N_x \times N_s$ en un produit

de trois matrices

$$Y = U\Sigma V^T \quad (2.52)$$

où les colonnes de U et V sont orthonormales et Σ est une matrice rectangulaire dont uniquement les éléments de la diagonale, appelés valeurs singulières, sont non nuls et vérifient

$$\Sigma_{kk} = \sigma_k, \quad \sigma_1 > \sigma_2 > \dots > \sigma_r > 0, \quad r = \min(N_x, N_s)$$

Le rang de la matrice Y est égal au nombre de valeurs singulières non nulles. Le théorème suivant permet de quantifier l'erreur d'approximation de la matrice Y par la matrice Y_k de rang inférieur.

Théorème 2.1 (Schmidt-Mirsky, Eckart-Young) *Pour une décomposition SVD de la matrice Y , si $q < \text{rang}(Y)$ et*

$$Y_q = \sum_{i=1}^q \sigma_i u_i v_i^T,$$

alors

$$\|Y - Y_q\| = \min_{\text{rang}(B) \leq q} \|Y - B\|_2 = \sigma_{q+1}, \quad (2.53)$$

où u_i et v_i sont respectivement les vecteurs colonnes de U et V .

L'équation (2.53) indique que la distance au sens de la norme euclidienne (norme-2) entre la matrice approximée Y_k de rang k et la matrice originel Y est exactement égale à σ_{k+1} . Supposons maintenant que nous cherchons une matrice de rang inférieur qui approche la matrice Y . Cette matrice peut être obtenue par troncature de la décomposition SVD de la matrice Y , en gardant que les q premières valeurs singulières et les q premières colonnes des matrices U et V . L'approximation d'ordre q de Y est alors donnée par

$$Y \approx U_q \Sigma_q V_q^T \quad (2.54)$$

L'optimalité de cette approximation repose sur le fait qu'aucune autre matrice de rang q ne peut être plus proche de Y au sens des moindres carrés (racine carrée de la somme des carrés de tous les éléments).

Considérons maintenant que $Y = A^{\frac{1}{2}}X$. En multipliant (2.52) du côté gauche par $X^T A^{\frac{1}{2}}$ on a :

$$X^T A X = V \Sigma^T U^T U \Sigma V^T$$

En tenant compte de l'orthonormalité de U , on obtient :

$$X^T A X = V \Sigma^2 V^T$$

Or d'après le paragraphe précédent, $X^T A X$ correspond à la matrice de corrélation temporelle C , d'où

$$C = V \Sigma^2 V^T$$

La matrice Σ^2 est une matrice diagonale de taille $N_x \times N_s$, dont les éléments de la diagonale sont égaux au carré des valeurs singulières, soit

$$\Sigma_{kk}^2 = \sigma_k^2$$

Finalement, en prenant en compte l'orthogonalité de V on peut écrire :

$$CV = V\Sigma^2 \quad (2.55)$$

L'équation (2.55) montre que les colonnes de V sont les vecteurs propres de la matrice de corrélation C , et les valeurs singulières de la matrice $A^{\frac{1}{2}}X$ sont les racines carrées des valeurs propres de la matrice C . De la même manière, en multipliant (2.52) du côté droit par U^T on montre que les colonnes de U sont les vecteurs propres de la matrice de covariance $D = A^{\frac{1}{2}}XX^T A^{\frac{1}{2}}$, et la base POD $\boldsymbol{\varphi}$ dans ce cas est obtenue en résolvant le système linéaire $A^{\frac{1}{2}}\boldsymbol{\varphi} = U$.

2.2.3 Méthode POD récursive

Contrairement à la méthode POD classique où uniquement deux paramètres ($N_p = 2$) entrent en jeu dans la forme séparée d'une fonction, la méthode POD récursive [67] est destinée à généraliser cette décomposition pour des fonctions multi-paramétriques ($N_p > 2$).

Considérons par exemple le cas simple d'une fonction à trois variables suivante

$$h : X \times Y \times Z \longrightarrow \mathbb{R}$$

En utilisant la méthode POD classique, la fonction $h(x, y, z)$ peut être approximée par la relation

$$h(x, y, z) \approx h_q(x, y, z) = \sum_{k=1}^q \xi_k(x) \cdot v_k(y, z)$$

On applique une deuxième fois la méthode POD sur les modes v_k

$$v_k(y, z) = \sum_{j=1}^{q_k} \varphi_j^{(k)}(y) \cdot \psi_j^{(k)}(z)$$

on obtient la décomposition suivante

$$h_{q, q_k}(x, y, z) = \sum_{k=1}^q \sum_{j=1}^{q_k} \xi_k(x) \cdot \varphi_j^{(k)}(y) \cdot \psi_j^{(k)}(z)$$

Cette approche peut être appliquée pour décomposer des fonctions tenseurs d'ordre D en une somme de D produit tensoriel de fonctions de dimension 1. Cette reformulation permettra de réduire les exigences de stockage et faciliter l'accès aux données. Toutefois, la construction d'une telle décomposition peut s'avérer très coûteuse lorsque l'ordre D du tenseur est grand.

2.2.4 Méthode DMD (Dynamic Modes Decomposition)

Considérons la matrice de snapshots $\mathbf{X} = [u_1 \ u_2 \ u_3 \ \cdots \ u_{N_s}]$, où $u_k \in \mathbb{R}^{N_x}$, et supposons qu'il existe un opérateur linéaire A qui assure approximativement la liaison entre deux vecteurs consécutifs, c'est à dire

$$u_{k+1} \approx Au_k$$

Cette hypothèse nous permet de formuler l'ensemble \mathbf{X} comme une suite de Krylov

$$\mathcal{K}_{N_s}(A) = \{u_1, Au_1, A^2u_1, \dots, A^{N_s-1}u_1\}$$

Le but de la méthode DMD, introduite par Schmid [60], est d'extraire les caractéristiques dynamiques du processus décrit par l'ensemble $\mathcal{K}_{N_s}(A)$. On souhaite alors calculer les valeurs propres et vecteurs propres de l'opérateur A qui est inconnu par hypothèse. L'idée de la méthode repose sur l'hypothèse qu'à partir d'un nombre de snapshots critique (ici N_s), les vecteurs de l'ensemble \mathbf{X} deviennent linéairement dépendants. Cela revient à considérer que l'ajout de vecteurs supplémentaires ne va pas améliorer le sous espace vectoriel engendré par les vecteurs u_i , $i = 1, \dots, N_s$. Quand cette condition est atteinte, on peut exprimer u_{N_s} en fonction des u_i , $i = 1, \dots, N_s - 1$ comme suit

$$u_{N_s} = \sum_{k=1}^{N_s-1} s_k u_k + r$$

où $r \in \mathbb{R}^{N_x}$ est un résidu. Cette expression s'écrit sous forme matricielle

$$u_{N_s} = \mathbf{Z}\mathbf{s} + r \quad (2.56)$$

avec $\mathbf{s}^T = [s_1, s_2, \dots, s_{N_s-1}]$ et \mathbf{Z} la matrice construite à partir de \mathbf{X} comme suit

$$\mathbf{Z} = [u_1 \ u_2 \ \cdots \ u_{N_s-1}]$$

En introduisant les matrices \mathbf{Y} et \mathbf{S} suivantes

$$\mathbf{Y} = [u_2 \ u_3 \ \cdots \ u_{N_s}] \quad \mathbf{S} = \begin{pmatrix} 0 & & & s_1 \\ 1 & 0 & & \\ & \ddots & \ddots & \\ & & 1 & 0 & s_{N_s-2} \\ & & & 1 & s_{N_s-1} \end{pmatrix} \quad (2.57)$$

on peut écrire

$$\mathbf{Y} = \mathbf{Z}\mathbf{S} + r e_{N_s-1}^T \quad (2.58)$$

où $e_{N_s-1} \in \mathbb{R}^{N_s-1}$ est le $(N_s - 1)$ -ième vecteur de la base euclidienne unitaire. De plus, on a par hypothèse que $\mathbf{Y} = \mathbf{A}\mathbf{Z}$, d'où

$$\mathbf{A}\mathbf{Z} = \mathbf{Y} = \mathbf{Z}\mathbf{S} + r e_{N_s-1}^T \quad (2.59)$$

Les coefficients s_1, \dots, s_{N_s-1} formant la dernière colonne de la matrice S , seront déterminés en calculant le minimum au sens des moindres carrés du résidu r , c'est à dire :

$$\min_s \|Zs - u_{N_s}\|^2 \quad (2.60)$$

Une manière pour calculer s consiste à considérer la décomposition QR de Z , ce qui implique que

$$s = R^{-1}Q^T u_{N_s}$$

Finalement, on peut écrire que

$$AQ = QH \quad \text{avec} \quad H = RSR^{-1}$$

Ainsi, on peut voir que les valeurs propres de S représentent quelques valeurs propres de A . Les vecteurs propres de A qui correspondent aux modes propres DMD sont alors calculés comme suit

$$\varphi^i = Qh_i$$

où h_i est le i -ème vecteur propre de H . En pratique, bien que la décomposition ci-dessus, basée sur la matrice S soit mathématiquement correcte, l'algorithme qui en résulte rencontre des problèmes de conditionnement et est souvent incapable d'extraire plus que quelques premiers modes dynamiques dominants [60].

Une procédure d'implémentation plus robuste qui résulte en une matrice \tilde{S} similaire¹ à S est alors choisie pour contourner ce problème. Le minimum au sens des moindres carrés du résidu r (2.60) est calculé en utilisant la décomposition SVD comme suit $Z = U\Sigma V^T$. En substituant cette expression dans l'équation (2.59), on obtient

$$U^T A U = U^T Y V \Sigma^{-1} \equiv \tilde{S}$$

Les vecteurs propres de A qui correspondent aux modes propres DMD sont

$$\varphi^i = U v_i$$

où v_i est le i -ème vecteur propre de \tilde{S} .

Les différentes étapes de la méthode DMD sont résumés comme suit :

1. Stocker les snapshots dans deux matrices

$$Z = [u_1 \ u_2 \ \cdots \ u_{N_s-1}] \quad Y = [u_2 \ u_3 \ \cdots \ u_{N_s}]$$

2. Calculer la décomposition en valeurs singulières tronquée de Z

$$Z = U\Sigma V^T$$

3. Définir la matrice réduite \tilde{S}

$$\tilde{S} = U^T Y V \Sigma^{-1}$$

1. Deux matrices A et B sont dites similaires si il existe une matrice P inversible telle que $A = P^{-1}BP$.

4. Calculer les vecteurs et valeurs propres de \tilde{S}

$$\tilde{S}v = v\lambda$$

5. Le mode DMD φ^i correspondant à la valeur propre DMD λ_i est alors donné par

$$\varphi^i = \mathbf{U}v_i$$

Récemment, la DMD a été appliquée entre autres, pour étudier l'écoulement derrière une membrane souple [60], l'écoulement autour d'un cylindre [68], l'écoulement autour des trains à grande vitesse [69], les interactions entre les ondes de choc et la couche limite turbulente [70], l'écoulement dans une cavité [60] [71] et divers jets [72] [60][73]...etc

2.3 CONSTRUCTION DU MODÈLE D'ORDRE RÉDUIT (ROM)

2.3.1 Cas général

Soit V l'espace de Hilbert tel que $V = L^2(0, T; H^1(\Omega))$. L'espace V peut être identifié à l'espace produit $\mathcal{T} \otimes \mathcal{V}$, où $\mathcal{T} = L^2(0, T)$ et $\mathcal{V} = H^1(\Omega)$. Considérons maintenant un système dynamique dont la solution $y = y(x, t)$ évolue dans V en suivant l'équation

$$\frac{dy}{dt} = X(y) \quad (2.61)$$

où X désigne un champs vectoriel défini sur V . En particulier, dans le cas où $y = y(x, t)$ est gouverné par une equation aux dérivées partielles, définie dans un domaine spatial Ω , V désignera un espace fonctionnel sur Ω et X un opérateur différentiel spatial. Soit $\tilde{\mathcal{V}}$ un sous-espace vectoriel de \mathcal{V} de dimension q , sur lequel la solution $y \in V$ est approximée comme suit

$$y(t, x) \approx y_q(t, x) = \sum_{j=1}^q \alpha^j(t) \varphi^j(x) \quad (2.62)$$

où $\{\varphi^j \in \mathcal{V}, j = 1, \dots, q\}$ est une base de $\tilde{\mathcal{V}}$ orthonormale par rapport au produit scalaire de L^2 , c'est à dire :

$$\int_{\Omega} \varphi^i \varphi^j dx = \delta_{ij}$$

En remplaçant y par l'expression de y_q dans l'équation (2.61), on obtient

$$\sum_{j=1}^q \frac{d\alpha^j}{dt} \varphi^j = X(y_q) + r \quad (2.63)$$

où r est le résidu lié à l'erreur d'approximation. En supposant que

$$\int_{\Omega} r \varphi^i dx = 0, \quad \forall i = 1, \dots, q$$

et en tenant compte de l'orthonormalité des modes φ^i , le système d'équations différentielles ordinaires de dimension q résultant de la projection de Galerkin de l'équation (2.63) sur les φ^i s'exprime

$$\frac{d\alpha^i}{dt} = \int_{\Omega} X \left(\sum_{j=1}^q \alpha^j \varphi^j \right) \varphi^i dx, \quad \forall i = 1, \dots, q \quad (2.64)$$

Ce système d'équations différentielles ordinaires de faible dimension q , constitue ce que l'on appelle modèle d'ordre réduit (ROM). La solution de ce modèle réduit nous donne les coefficients temporels α^j et la formule (2.62) l'expression de la solution approximée.

2.3.2 Construction du ROM pour les équations de Navier-Stokes

a) Position du problème

Dans ce paragraphe, on s'intéresse à la dérivation des équations du modèle réduit associé au problème de Navier-Stokes. On considère donc l'écoulement isotherme d'un fluide Newtonien incompressible de viscosité cinématique constante ν régi par les équations

$$\begin{cases} \partial_t \mathbf{u} - \frac{1}{Re} \Delta \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} + \nabla p = \mathbf{f} & \text{dans } \Omega \times [0, T[\\ \nabla \cdot \mathbf{u} = 0 & \text{dans } \Omega \times [0, T[\\ \mathbf{u} = \mathbf{g} & \text{sur } \Gamma_D \times [0, T[\\ \partial_n \mathbf{u} = 0 & \text{sur } \Gamma_N \times [0, T[\\ -\frac{1}{Re} \partial_n \mathbf{u} + p \mathbf{n} = 0 & \text{sur } \Gamma_\sigma \times [0, T[\\ \mathbf{u}(0) = \mathbf{u}_0 & \text{dans } \Omega \end{cases} \quad (2.65)$$

Le domaine spatial Ω est un ouvert borné connexe de \mathbb{R}^d , avec $d = 2, 3$. Nous supposons que la frontière Γ du domaine Ω est divisée en trois partitions Γ_D , Γ_N et Γ_σ . Sur Γ_D , la vitesse est fixée à une valeur $\mathbf{g}(x)$ donnée indépendante du temps. Les forces exercées en chaque point de la frontière Γ_σ par unité de surface sont nulles et une condition de Neumann homogène est imposée sur la frontière Γ_N . On introduit les champs moyens de vitesse et de pressions $\bar{\mathbf{u}}$ et \bar{p} qui vérifient

$$\begin{cases} \mathbf{u}(t, x) = \bar{\mathbf{u}}(x) + \tilde{\mathbf{u}}(t, x) \\ p(t, x) = \bar{p}(x) + \tilde{p}(t, x) \\ \bar{\mathbf{u}} = \mathbf{g}, \quad \text{sur } \Gamma_D \times [0, T[\\ \partial_n \bar{\mathbf{u}} = 0, \quad \text{sur } \Gamma_N \times [0, T[\\ \frac{1}{Re} \partial_n \bar{\mathbf{u}} = \bar{p} \mathbf{n}, \quad \text{sur } \Gamma_\sigma \times [0, T[\end{cases}$$

Les équations (5.23) deviennent alors

$$\begin{cases} \partial_t \tilde{\mathbf{u}} - \frac{1}{Re} \Delta \tilde{\mathbf{u}} + \bar{\mathbf{u}} \cdot \nabla \tilde{\mathbf{u}} + \tilde{\mathbf{u}} \cdot \nabla \bar{\mathbf{u}} + \tilde{\mathbf{u}} \cdot \nabla \tilde{\mathbf{u}} + \nabla \tilde{p} = \tilde{\mathbf{f}} & \text{dans } \Omega \times [0, T[\\ \nabla \cdot \tilde{\mathbf{u}} = 0 & \text{dans } \Omega \times [0, T[\\ \tilde{\mathbf{u}} = 0 & \text{sur } \Gamma_D \times [0, T[\\ -\frac{1}{Re} \partial_n \tilde{\mathbf{u}} + \tilde{p} \mathbf{n} = 0 & \text{sur } \Gamma_\sigma \times [0, T[\\ \tilde{\mathbf{u}}(0) = \tilde{\mathbf{u}}_0 = \mathbf{u}_0 - \bar{\mathbf{u}}_0 & \text{dans } \Omega \end{cases} \quad (2.66)$$

avec

$$\tilde{\mathbf{f}} = \mathbf{f} + \frac{1}{Re} \Delta \bar{\mathbf{u}} - \bar{\mathbf{u}} \cdot \nabla \bar{\mathbf{u}} - \nabla \bar{p}$$

Pour $t \geq 0$ et $\mathbf{u}_0 \in \mathbf{H}$ une condition initiale, la formulation faible associée au problème (2.66) peut s'écrire comme suit

$$\begin{cases} \text{Trouver } \tilde{\mathbf{u}} \in L^2(0, T; \mathbf{V}) \text{ et } \tilde{p} \in L^2(0, T; \mathcal{P}) \text{ tels que} \\ m(\partial_t \tilde{\mathbf{u}}, \mathbf{v}) + \frac{1}{Re} a(\tilde{\mathbf{u}}, \mathbf{v}) + c(\bar{\mathbf{u}}, \tilde{\mathbf{u}}, \mathbf{v}) + c(\tilde{\mathbf{u}}, \bar{\mathbf{u}}, \mathbf{v}) + c(\tilde{\mathbf{u}}, \tilde{\mathbf{u}}, \mathbf{v}) + d(\mathbf{v}, \tilde{p}) + d_{\Gamma_N}(\mathbf{v}, \tilde{p}) = b(\mathbf{v}) \\ d(\tilde{\mathbf{u}}, \mathbf{q}) = 0 \\ \forall (\mathbf{v}, \mathbf{q}) \in \mathbf{V}_0 \times \mathcal{P} \end{cases} \quad (2.67)$$

Les espaces fonctionnels \mathbf{H} , \mathbf{V} , \mathbf{V}_0 et \mathcal{P} ainsi que les formes multilinéaires $m(\cdot, \cdot)$, $a(\cdot, \cdot)$, $d(\cdot, \cdot)$, $d_{\Gamma_N}(\cdot, \cdot)$, $c(\cdot, \cdot, \cdot)$ et $b(\cdot)$ sont définis dans la section 1.4 du chapitre 1.

b) Construction du modèle d'ordre réduit

Considérons une base POD de vitesse $\boldsymbol{\varphi}$ de dimension q_u construite à partir d'un ensemble de champs fluctuants de vitesses $\{\tilde{\mathbf{u}}_1, \tilde{\mathbf{u}}_2, \dots, \tilde{\mathbf{u}}_{N_s}\}$ pris à des instants discrets $t_k \in \{t_1, t_2, \dots, t_{N_s}\}$. La reconstruction du champs fluctuant de vitesse sur le sous espace vectoriel engendré par $\boldsymbol{\varphi}$ est obtenue comme suit

$$\tilde{\mathbf{u}} \approx \sum_{i=1}^{q_u} \alpha_u^i \boldsymbol{\varphi}_u^i$$

En substituant cette expression dans la première équation de (2.67) avec $\mathbf{v} = \boldsymbol{\varphi}_u^i$ (pour $i = 1, \dots, q_u$), on obtient

$$\sum_{j=1}^{q_u} \mathbf{M}_{ij}^{(u)} \frac{d\alpha_u^j}{dt} + \sum_{j=1}^{q_u} \left[\frac{1}{Re} \mathbf{R}_{ij}^{(u)} + \bar{\mathbf{C}}_{ij}^{(u)} \right] \alpha_u^j + \sum_{j=1}^{q_u} \sum_{k=1}^{q_u} \mathbf{C}_{ijk}^{(u)} \alpha_u^j \alpha_u^k + [\mathbf{B}_i^{(u)} + \mathbf{D}_i^{(u)}] = \tilde{\mathbf{F}}_i^{(u)}, \quad \forall i = 1, \dots, q_u \quad (2.68)$$

où

$$\begin{aligned}
\mathbf{M}_{ij}^{(u)} &= m(\varphi_u^j, \varphi_u^i) = \int_{\Omega} \varphi_u^i \varphi_u^j dx & \mathbf{R}_{ij}^{(u)} &= a(\varphi_u^j, \varphi_u^i) = \int_{\Omega} \nabla \varphi_u^i : \nabla \varphi_u^j dx \\
\overline{\mathbf{C}}_{ij}^{(u)} &= c(\overline{\mathbf{u}}, \varphi_u^j, \varphi_u^i) + c(\varphi_u^j, \overline{\mathbf{u}}, \varphi_u^i) = \int_{\Omega} (\overline{\mathbf{u}} \cdot \nabla) \varphi_u^j \cdot \varphi_u^i dx + \int_{\Omega} (\varphi_u^j \cdot \nabla) \overline{\mathbf{u}} \cdot \varphi_u^i dx \\
\mathbf{C}_{ijk}^{(u)} &= c(\varphi_u^j, \varphi_u^k, \varphi_u^i) = \int_{\Omega} (\varphi_u^j \cdot \nabla) \varphi_u^k \cdot \varphi_u^i dx & \tilde{\mathbf{F}}_i^{(u)} &= b(\varphi_u^i) = \int_{\Omega} \tilde{\mathbf{f}} \varphi_u^i dx \\
\mathbf{B}_i^{(u)} &= d(\varphi_u^i, \tilde{p}) = - \int_{\Omega} \tilde{p} \nabla \cdot \varphi_u^i dx & \mathbf{D}_i^{(u)} &= d_{\Gamma_N}(\varphi_u^i, \tilde{p}) = \int_{\Gamma_N} \tilde{p} \varphi_u^i \cdot \mathbf{n} d\sigma
\end{aligned}$$

Étant donné que les modes POD de vitesse sont par construction à divergence nulle, l'équation de conservation de masse $\sum_{i=1}^{q_u} \alpha_u^i d(\varphi_u^i, q) = 0$ est vérifiée pour tout q quelles que soient les valeurs α_u^i . De plus, le terme lié au gradient de pression est numériquement nul, d'où

$$\mathbf{B}_i^{(u)} = 0, \quad \forall i = 1, \dots, q_u$$

Le système d'équations différentielles constituant le modèle réduit s'exprime alors comme suit

$$\begin{cases}
\sum_{j=1}^{q_u} \mathbf{M}_{ij}^{(u)} \frac{d\alpha_u^j}{dt} + \sum_{j=1}^{q_u} \left[\frac{1}{Re} \mathbf{R}_{ij}^{(u)} + \overline{\mathbf{C}}_{ij}^{(u)} \right] \alpha_u^j + \sum_{j=1}^{q_u} \sum_{k=1}^{q_u} \mathbf{C}_{ijk}^{(u)} \alpha_u^j \alpha_u^k + \mathbf{D}_i^{(u)} = \tilde{\mathbf{F}}_i^{(u)}, & \forall i = 1, \dots, q_u \\
\sum_{j=1}^{q_u} \mathbf{M}_{ij}^{(u)} \alpha_u^j(0) = \int_{\Omega} \varphi_u^i \tilde{\mathbf{u}}_0 dx, & \forall i = 1, \dots, q_u
\end{cases} \quad (2.69)$$

Une fois que les sous espaces vectoriels réduits engendrés par les bases réduites POD sont déterminés, le calcul des matrices réduites $\mathbf{M}^{(u)}$, $\mathbf{R}^{(u)}$, $\overline{\mathbf{C}}^{(u)}$, $\mathbf{B}^{(u)}$, $\mathbf{D}_i^{(u)}$, $\tilde{\mathbf{F}}^{(u)}$ et du tenseur $\mathbf{C}^{(u)}$ se fait dans la phase offline [74][75]. Ainsi, la résolution du système d'équations différentielles ordinaires devient très rapide. Cependant dans certains cas, l'évaluation du terme $\sum_{j=1}^{q_u} \sum_{k=1}^{q_u} \mathbf{C}_{ijk}^{(u)} \alpha_u^j \alpha_u^k$ en chaque itération de temps peut être coûteuse à cause de la dimension du tenseur $\mathbf{C}^{(u)}$ qui est égale au nombre de modes de vitesse gardés au cube. Pour réduire le coût de calcul de ce terme, il est possible d'utiliser par exemple la méthode EIM-DEIM [76][77].

c) Traitement du terme de pression

Comme la pression fluctuante est indéterminée, la présence du terme de pression $\mathbf{D}_i^{(u)}$ dans l'équation du modèle d'ordre réduit (2.69) présente un obstacle dans sa résolution. La plupart des travaux existant néglige ce terme dont la contribution est généralement petite. Toutefois, comme il a été souligné par Noack et al. [78], ce terme ne peut pas être négligé dans de nombreuses applications. Pour palier à ce problème, Bergmann et al. [79] ont proposé de construire une base POD $\boldsymbol{\varphi} = [\boldsymbol{\varphi}_u^T \ \boldsymbol{\varphi}_p^T]^T$ à partir de l'ensemble des snapshots $\mathbf{X} = [\mathbf{X}_u^T \ \mathbf{X}_p^T]^T$ de vitesse et de pression de telle sorte que les coefficients temporels pour la vitesse et la pression soient identiques, c'est à dire, $p = \sum_{i=1}^q \alpha^i \varphi_u^i$ et

$\mathbf{u} = \sum_{i=1}^q \alpha^i \varphi_p^i$. Ainsi, le terme $D_i^{(u)}$ s'écrit

$$D_i^{(u)} = \sum_{l=1}^{q_p} \alpha^l \int_{\Gamma_N} \varphi_p^l \varphi_u^i \cdot \mathbf{n} \, d\sigma$$

Il est également possible, comme pour les méthodes de prédiction correction introduites par Chorin [80] et Temam [81], d'introduire une équation de Poisson pour tenir compte du terme de pression dans le modèle d'ordre réduit [82][83][84]. Une autre approche proposée par Leblond et al. [85] et Tallet et al. [86], consiste à créer une base POD $\boldsymbol{\varphi}_p = \{\varphi_p^1, \dots, \varphi_p^{q_p}\}$ de telle sorte que la pression fluctuante soit reconstruite comme suit $\tilde{p} \approx \sum_{i=1}^{q_p} \alpha_p^i \varphi_p^i$. Les coefficients temporels de vitesse α_u et de pression α_p sont alors solutions d'un modèle d'ordre réduit construit en minimisant à chaque instant t^{n+1} la norme L^2 au carré du résidu $\mathcal{R}es^{(n+1)}$ qui apparaît dans les équations de quantité de mouvement discrétisé en temps. Dans ce cas, le résidu discrétisé en temps est donné par

$$\begin{aligned} \mathcal{R}es^{(n+1)} &= \sum_{j=1}^{q_u} \varphi_u^j \frac{\alpha_u^{j,n+1} - \alpha_u^{j,n}}{\delta t} - \sum_{j=1}^{q_u} \frac{1}{Re} \Delta \varphi_u^j \alpha_u^{j,n+1} + \left[(\bar{\mathbf{u}} \cdot \nabla) \varphi_u^j + (\varphi_u^j \cdot \nabla \bar{\mathbf{u}}) \right] \alpha_u^{j,n} \\ &\quad + \sum_{j=1}^{q_u} \sum_{k=1}^{q_u} (\varphi_u^k \cdot \nabla) \varphi_u^j \alpha_u^{j,n} \alpha_u^{k,n} + \sum_{l=1}^{q_p} \nabla \varphi_p^l \alpha_p^{l,n+1} - \tilde{\mathbf{F}}_{i,n+1}^{(u)} \end{aligned}$$

Les variations de $\|\mathcal{R}es^{(n+1)}\|_{L^2(\Omega)}^2$ par rapport à α_u^i pour $i = 1, \dots, q_u$ et α_p^m pour $m = 1, \dots, q_p$ s'expriment comme suit

$$\begin{aligned} \partial_{\alpha_u^i} \|\mathcal{R}es^{(n+1)}\|_{L^2(\Omega)}^2 &= 2 \int_{\Omega} \mathcal{R}es^{(n+1)} \left(\varphi_u^i - \frac{1}{Re} \Delta \varphi_u^i \right) dx \\ \partial_{\alpha_p^m} \|\mathcal{R}es^{(n+1)}\|_{L^2(\Omega)}^2 &= 2 \int_{\Omega} \mathcal{R}es^{(n+1)} \nabla \varphi_p^m dx \end{aligned}$$

En notant $\psi_u^i = \varphi_u^i - \frac{1}{Re} \Delta \varphi_u^i$, les conditions d'optimalité du problème $\min_{\alpha_u, \alpha_p} \|\mathcal{R}es^{(n+1)}\|_{L^2(\Omega)}^2$ résultant de l'annulation de $\partial_{\alpha_u^i} \|\mathcal{R}es^{(n+1)}\|_{L^2(\Omega)}^2$ et $\partial_{\alpha_p^m} \|\mathcal{R}es^{(n+1)}\|_{L^2(\Omega)}^2$ conduisent au modèle réduit suivant

$$\left\{ \begin{aligned} &\sum_{j=1}^{q_u} \mathbf{M}_{ij}^{(u)} \frac{\alpha_u^{j,n+1} - \alpha_u^{j,n}}{\delta t} - \sum_{j=1}^{q_u} \frac{1}{Re} \mathbf{R}_{ij}^{(u)} \alpha_u^{j,n+1} + \bar{\mathbf{C}}_{ij}^{(u)} \alpha_u^{j,n} \\ &\quad + \sum_{j=1}^{q_u} \sum_{k=1}^{q_u} \mathbf{C}_{ijk}^{(u)} \alpha_u^{j,n} \alpha_u^{k,n} + \sum_{l=1}^{q_p} \mathbf{K}_{il}^{(u)} \alpha_p^{l,n+1} = \tilde{\mathbf{F}}_{i,n+1}^{(u)}, \quad \forall i = 1, \dots, q_u \\ &\sum_{j=1}^{q_u} \mathbf{M}_{mj}^{(p)} \frac{\alpha_u^{j,n+1} - \alpha_u^{j,n}}{\delta t} - \sum_{j=1}^{q_u} \frac{1}{Re} \mathbf{R}_{mj}^{(p)} \alpha_u^{j,n+1} + \bar{\mathbf{C}}_{mj}^{(p)} \alpha_u^{j,n} \\ &\quad + \sum_{j=1}^{q_u} \sum_{k=1}^{q_u} \mathbf{C}_{mjk}^{(p)} \alpha_u^{j,n} \alpha_u^{k,n} + \sum_{l=1}^{q_p} \mathbf{K}_{ml}^{(p)} \alpha_p^{l,n+1} = \tilde{\mathbf{F}}_{m,n+1}^{(p)}, \quad \forall m = 1, \dots, q_p \\ &\sum_{j=1}^{q_u} \mathbf{M}_{ij}^{(u)} \alpha_u^j(0) = \int_{\Omega} \varphi_u^i \tilde{\mathbf{u}}_0 dx, \quad \forall i = 1, \dots, q_u \end{aligned} \right. \quad (2.70)$$

où

$$\begin{aligned}
\mathbf{M}_{ij}^{(u)} &= m(\varphi_u^j, \psi_u^i) = \int_{\Omega} \varphi_u^j \psi_u^i dx & \mathbf{R}_{ij}^{(u)} &= a(\varphi_u^j, \psi_u^i) = \int_{\Omega} \Delta \varphi_u^j \psi_u^i dx \\
\bar{\mathbf{C}}_{ij}^{(u)} &= c(\bar{\mathbf{u}}, \varphi_u^j, \psi_u^i) + c(\varphi_u^j, \bar{\mathbf{u}}, \psi_u^i) = \int_{\Omega} (\bar{\mathbf{u}} \cdot \nabla) \varphi_u^j \cdot \psi_u^i dx + \int_{\Omega} (\varphi_u^j \cdot \nabla) \bar{\mathbf{u}} \cdot \psi_u^i dx \\
\mathbf{C}_{ijk}^{(u)} &= c(\varphi_u^j, \varphi_u^k, \psi_u^i) = \int_{\Omega} (\varphi_u^j \cdot \nabla) \varphi_u^k \cdot \psi_u^i dx \\
\mathbf{K}_{il}^{(u)} &= \int_{\Omega} \nabla \varphi_p^l \cdot \psi_u^i dx & \tilde{\mathbf{F}}_i^{(u)} &= \int_{\Omega} \tilde{\mathbf{f}} \psi_u^i dx
\end{aligned}$$

et

$$\begin{aligned}
\mathbf{M}_{mj}^{(p)} &= m(\varphi_u^j, \nabla \varphi_p^m) = \int_{\Omega} \varphi_u^j \nabla \varphi_p^m dx & \mathbf{R}_{mj}^{(p)} &= a(\varphi_u^j, \nabla \varphi_p^m) = \int_{\Omega} \Delta \varphi_u^j \nabla \varphi_p^m dx \\
\bar{\mathbf{C}}_{mj}^{(p)} &= c(\bar{\mathbf{u}}, \varphi_u^j, \nabla \varphi_p^m) + c(\varphi_u^j, \bar{\mathbf{u}}, \nabla \varphi_p^m) = \int_{\Omega} (\bar{\mathbf{u}} \cdot \nabla) \varphi_u^j \cdot \nabla \varphi_p^m dx + \int_{\Omega} (\varphi_u^j \cdot \nabla) \bar{\mathbf{u}} \cdot \nabla \varphi_p^m dx \\
\mathbf{C}_{mjk}^{(p)} &= c(\varphi_u^j, \varphi_u^k, \nabla \varphi_p^m) = \int_{\Omega} (\varphi_u^j \cdot \nabla) \varphi_u^k \cdot \nabla \varphi_p^m dx \\
\mathbf{K}_{ml}^{(p)} &= \int_{\Omega} \nabla \varphi_p^l \cdot \nabla \varphi_p^m dx & \tilde{\mathbf{F}}_m^{(p)} &= \int_{\Omega} \tilde{\mathbf{f}} \nabla \varphi_p^m dx
\end{aligned}$$

d) Stabilisation du modèle d'ordre réduit

En utilisant la POD pour la construction de la base spatiale, seul un nombre relativement faible de modes est gardé pour la construction du ROM. Ainsi, le ROM ne prend pas en compte l'interaction avec les modes non résolus souvent représentatifs de petites échelles du phénomène étudié. Bien que très petites, ces échelles peuvent jouer un rôle important dans la résolution du ROM. Ainsi leur non prise en compte résulte en un ROM qui peut être instable. Plusieurs techniques ont été développées pour palier au problème de stabilité des modèles réduits. L'approche la plus élémentaire et la plus utilisée est de considérer que l'influence des petites échelles peut être modélisée par une viscosité supplémentaire dite artificielle. Cette viscosité est introduite par l'utilisateur de façon à ce que les résultats obtenus soient le plus proche possible de ceux obtenus dans les simulations utilisées pour la construction de la base réduite. En particulier, cet ajout peut être par l'introduction d'une viscosité turbulente de type Heisenberg [87][88], ou d'une viscosité spectrale évanescence [89]. Une variante de cette approche introduite par Rempfer [90] consiste à attribuer à chaque mode POD un certain niveau de dissipation. Giovanni et al. [84] proposent une méthode de stabilisation appelée SUP-ROM. Cette méthode consiste à enrichir le sous espace vectoriel de vitesse par un ensemble de fonctions de bases de telle sorte que cet enrichissement garantisse la vérification d'une version réduite de la condition inf-sup. Bergmann et al. [79], proposent quant à eux, différentes méthodes de stabilisation basées sur l'évaluation des résidus du modèle complet : enrichissement de la base par ajout des résidus, formulation de Petrov-Galerkin du modèle réduit (SUPG : streamline upwind Petrov Galerkin) et formulation VMS (Variational Multiscale). Toutes ces approches agissent sur l'équation du ROM par modification ou introduction de termes supplémentaires. Il est également possible d'améliorer la structure et le contenu de la base POD sans modifier le ROM. Ainsi, Iollo et al. [91][92] suggèrent de définir le problème POD en utilisant la

norme dans l'espace de Sobolev $H^1(\Omega)$, afin d'introduire la partie dissipative directement dans les équations du modèle réduit.

Dans notre cas, le modèle réduit utilisé a été construit en projetant l'équation de quantité de mouvement sur les fonctions de base de vitesse, puis sur le gradient des fonctions de base de pression afin de compléter le système d'équations différentielles. Le ROM construit ainsi a montré un bon comportement sans aucun besoin d'être stabilisé.

2.4 CONTRÔLE OPTIMAL RÉDUIT

Dans un algorithme d'optimisation sous contraintes EDP (contrôle optimal), les modèles complets (résolus par éléments finis, volumes finis ou différences finies) permettant d'obtenir les variables d'état et leurs variables adjointes, assurent une bonne prédiction des directions de descentes. L'inconvénient majeur de ces modèles complets est qu'ils sont très coûteux en temps de calcul et en stockage mémoire. En toute évidence, une seule itération de l'algorithme de descente nécessitera au moins la résolution de deux systèmes linéaires de grande taille. Dans le cas où les contraintes sont décrites par une EDP non linéaire, le coût augmente davantage. Une solution possible permettant de réduire ces exigences est d'utiliser les techniques de réduction de modèles. Dans cette section, l'optimisation à l'échelle réduite est décrite pour un problème d'optimisation non linéaire général.

2.4.1 Position du problème de contrôle optimal réduit

L'objectif de cette section est de réduire drastiquement la dimensionnalité du problème d'optimisation défini dans la section 1.3, en utilisant les modèles d'ordre réduit construit par la méthode POD/Galerkin. En reprenant le même cadre fonctionnel et les définitions de la section 1.3, le problème de contrôle optimal dans son cadre général est défini par

$$\min \mathcal{J}(y, \theta) \text{ sur } (y, \theta) \in V \times K \quad (2.71)$$

sous la contrainte

$$\mathcal{N}(y, \theta) = 0 \quad (2.72)$$

On distinguera deux sortes d'optimisation réduites, l'optimisation réduite "Reduce Then Differentiate", et l'optimisation réduite "Differentiate Then Reduce".

2.4.2 Optimisation réduite "Differentiate Then Reduce"

L'optimisation réduite "Differentiate Then Reduce" consiste tout simplement à supposer que les variables d'état et adjointes peuvent être représentées dans deux sous espaces vectoriels de dimension finie distincts, engendrés par deux bases φ et ψ . Le problème d'optimisation (dont l'état y et l'état adjoint ζ sont solutions de deux équations aux dérivées partielles) est alors ramené à un problème d'optimisation réduit simple à

résoudre, dont la variable d'état $\tilde{\alpha}$ et la variable adjointe ζ sont cette fois-ci solutions de deux équations différentielles ordinaires de tailles réduite.

On suppose donc que $y = y(\theta)$ et $\zeta = \zeta(\theta)$ peuvent être approximés en utilisant une décomposition spatio-temporelle d'ordre q et r respectivement comme suite

$$u(t, x, \theta) \approx y_q(t, x, \theta) = \sum_{k=1}^q \alpha^k(t, \theta) \varphi^k(x) \quad \forall \theta \in K \quad (2.73)$$

$$\zeta(t, x, \theta) \approx \zeta_r(t, x, \theta) = \sum_{k=1}^r \zeta^k(t, \theta) \psi^k(x) \quad \forall \theta \in K \quad (2.74)$$

avec

$$\begin{aligned} \alpha &= \{\alpha^1, \dots, \alpha^q\} & \zeta &= \{\zeta^1, \dots, \zeta^r\} \\ \varphi &= \{\varphi^1, \dots, \varphi^q\} & \psi &= \{\psi^1, \dots, \psi^r\} \end{aligned}$$

Les ensembles α et ζ sont les coefficients temporels déterminés en résolvant les modèles d'ordre réduit issus de la projection de Galerkin des équations du modèle complet d'état et de l'état adjoint sur les bases réduites φ et ψ supposées fixes par rapport aux changements du paramètre de contrôle θ . Ainsi, dans la suite, pour une base φ fixée, on notera $\mathcal{N}_\varphi : \mathcal{T}^q \times K \rightarrow \mathcal{T}^q$ l'application telle que $\mathcal{N}_\varphi(\alpha, \theta) = 0$ correspond au modèle réduit de l'équation d'état donné comme suit

$$\left\langle \mathcal{N} \left(\sum_{k=1}^q \alpha^k \varphi^k, \theta \right), \varphi^i \right\rangle_{L^2(\Omega)} = 0, \quad \forall i = 1, \dots, q \quad (2.75)$$

De même, on notera $\mathcal{M}_\psi : \mathcal{T}^r \times \mathcal{T}^q \times K \rightarrow \mathcal{T}^r$ l'application telle que $\mathcal{M}_\psi(\zeta, \alpha, \theta) = 0$ correspond au modèle réduit de l'équation adjointe donnée par

$$\left\langle \partial_y \mathcal{J} \left(\sum_{k=1}^q \alpha^k \varphi^k, \theta \right) + \partial_y \mathcal{N} \left(\sum_{k=1}^q \alpha^k \varphi^k, \theta \right)^* \sum_{k=1}^r \zeta^k \psi^k, \psi_l \right\rangle_{L^2(\Omega)} = 0, \quad \forall l \in \{1, \dots, r\} \quad (2.76)$$

L'optimisation réduite "Differentiate Then Reduce" consiste tout d'abord à extraire les conditions d'optimalités pour le problème complet (section 1.3), puis en utilisant les approximations de l'état et de l'état adjoint (2.73)(2.74), déterminer la direction de descente d_n à l'itération n par la relation suivante

$$d_n = - \left(\partial_\theta \mathcal{J}(y_q^{(n)}, \theta^{(n)}) + \partial_\theta \mathcal{N}(y_q^{(n)}, \theta^{(n)})^* \zeta_r^{(n)} \right) \quad (2.77)$$

L'algorithme d'optimisation réduite "Differentiate Then Reduce" est décrit dans l'algorithme 8.

Remarque 2.1 *Plusieurs choix peuvent être réalisés pour la construction des bases réduites φ et ψ . Il est par exemple possible de choisir une seule base spatiale pour approcher l'état et l'état adjoint. Cette base pourra être construite par exemple en utilisant la méthode POD sur l'ensemble des snapshots issus de plusieurs simulations du problème d'état et du problème adjoint [93].*

Algorithme 8 : Algorithme d'optimisation réduit "Differentiate Then Reduce".

```

1 Initialisation de l'algorithme :  $n = 0$  et  $\theta^{(n)} = \theta^{init}$ ;
2 while  $E > \varepsilon$  do
3   Résoudre l'équation d'état
      
$$\mathcal{N}_\varphi(\tilde{\alpha}^{(n)}, \theta^{(n)}) = 0$$

4   Connaissant  $\tilde{\alpha}^{(n)}$ , trouver  $\zeta^{(n)}$  solution de l'équation adjointe
      
$$\mathcal{M}_\psi(\zeta^{(n)}, \tilde{\alpha}^{(n)}, \theta^{(n)}) = 0$$

5   Reconstruction de  $u_q^{(n)}$  et  $\xi_r^{(n)}$  en utilisant les expressions (2.73) et (2.74).
6   Évaluation de la direction de descente  $d_n = -\partial_\theta \mathcal{L}(u_q^{(n)}, \theta^{(n)}, \xi_r^{(n)})$ 
7   Recherche linéaire du pas  $\sigma_n$  (Armijo)
8   Mise à jour des paramètres de contrôle
      
$$\theta^{(n+1)} = P_K(\theta^{(n)} + \sigma_n d_n)$$

9   Critère de convergence
      
$$E = \frac{|\mathcal{J}_\varphi(\tilde{\alpha}^{(n+1)}, \theta^{(n+1)}) - \mathcal{J}_\varphi(\tilde{\alpha}^{(n)}, \theta^{(n)})|}{|\mathcal{J}_\varphi(\tilde{\alpha}^{(n+1)}, \theta^{(n+1)})|}, \quad \text{où } \mathcal{J}_\varphi(\tilde{\alpha}, \theta) = \mathcal{J}\left(\sum_{i=1}^q \tilde{\alpha}^i \varphi^i, \theta\right)$$

10 end

```

2.4.3 Optimisation réduite "Reduce Then Differentiate"

Le problème de l'optimisation réduite "Reduce Then Differentiate" consiste à partir de la connaissance de l'approximation de l'état $y \approx y_q = \sum_{i=1}^q \alpha^i \varphi^i$, à reformuler le problème d'optimisation sous forme réduite et à dériver les conditions nécessaires d'optimalités associées.

En substituant l'expression de y_q dans le problème d'optimisation (2.71)(2.72), le problème de contrôle optimal réduit revient à

$$\min \mathcal{J}_\varphi(\tilde{\alpha}, \theta) \quad \text{sur } (\tilde{\alpha}, \theta) \in \mathcal{T} \times K \quad (2.78)$$

sous la contrainte réduite

$$\mathcal{N}_\varphi(\tilde{\alpha}, \theta) = 0 \quad (2.79)$$

où \mathcal{J}_φ est telle que

$$\mathcal{J}_\varphi(\tilde{\alpha}, \theta) = \mathcal{J}(y_q, \theta)$$

Le problème de contrôle réduit (2.78) (2.79) est résolu en utilisant la méthode des multiplicateurs de Lagrange. La fonction de Lagrange associée à la contrainte réduite (2.79)

s'écrit :

$$\begin{aligned} \mathcal{L}_\varphi : \mathcal{T} \times Z \times \mathcal{T} &\longrightarrow \mathbb{R} \\ (\tilde{\alpha}, \theta, \beta) &\longrightarrow \mathcal{L}(\tilde{\alpha}, \theta, \beta) = \mathcal{J}_\varphi(\tilde{\alpha}, \theta) + \langle \beta, \mathcal{N}_\varphi(\tilde{\alpha}, \theta) \rangle_{\mathcal{T}} \end{aligned}$$

La différentielle du Lagrangien par rapport à la variable d'état $\tilde{\alpha}$, suivant la direction $\delta\tilde{\alpha} \in \mathcal{T}$, est donnée par

$$\begin{aligned} \langle \partial_{\tilde{\alpha}} \mathcal{L}_\varphi(\tilde{\alpha}, \theta, \beta), \delta\tilde{\alpha} \rangle_{\mathcal{T}} &= \langle \partial_{\tilde{\alpha}} \mathcal{J}_\varphi(\tilde{\alpha}, \theta), \delta\tilde{\alpha} \rangle_{\mathcal{T}} + \langle \beta, \partial_{\tilde{\alpha}} \mathcal{N}_\varphi(\tilde{\alpha}, \theta) \delta\tilde{\alpha} \rangle_{\mathcal{T}} \\ &= \langle \partial_{\tilde{\alpha}} \mathcal{J}_\varphi(\tilde{\alpha}, \theta), \delta\tilde{\alpha} \rangle_{\mathcal{T}} + \langle \partial_{\tilde{\alpha}} \mathcal{N}_\varphi(\tilde{\alpha}, \theta)^* \beta, \delta\tilde{\alpha} \rangle_{\mathcal{T}} \end{aligned}$$

La condition d'optimalité $\partial_{\tilde{\alpha}} \mathcal{L}_\varphi(\tilde{\alpha}, \bar{\theta}, \beta) = 0$ permet d'extraire l'équation adjointe. Ainsi la variable adjointe β sera solution du problème

$$\partial_{\tilde{\alpha}} \mathcal{N}_\varphi(\tilde{\alpha}, \theta)^* \beta + \partial_{\tilde{\alpha}} \mathcal{J}_\varphi(\tilde{\alpha}, \theta) = 0 \quad (2.80)$$

En prenant maintenant la différentielle du Lagrangien par rapport θ suivant la direction $\delta\theta \in Z$, on a

$$\langle \partial_\theta \mathcal{L}_\varphi(\tilde{\alpha}, \theta, \beta), \delta\theta \rangle_Z = \langle \partial_\theta \mathcal{J}_\varphi(\tilde{\alpha}, \theta), \delta\theta \rangle_Z + \langle \beta, \partial_\theta \mathcal{N}_\varphi(\tilde{\alpha}, \theta) \delta\theta \rangle_{\mathcal{T}} \quad (2.81)$$

$$= \langle \partial_\theta \mathcal{J}_\varphi(\tilde{\alpha}, \theta), \delta\theta \rangle_Z + \langle \partial_\theta \mathcal{N}_\varphi(\tilde{\alpha}, \theta)^* \beta, \delta\theta \rangle_Z \quad (2.82)$$

On peut en déduire la condition d'optimalité réduite

$$(\bar{\theta} - \theta)^T (\partial_\theta \mathcal{J}_\varphi(\tilde{\alpha}, \theta) + \partial_\theta \mathcal{N}_\varphi(\tilde{\alpha}, \theta)^* \beta) \geq 0 \quad \forall \bar{\theta} \in K. \quad (2.83)$$

L'algorithme d'optimisation réduite "Reduce Then Differentiate" est résumé dans algo. 9.

Algorithme 9 : Algorithme d'optimisation réduit "Reduce Then Differentiate".

1 Initialisation de l'algorithme : $n = 0$ et $\theta^{(n)} = \theta^{init}$;

2 **while** $E > \varepsilon$ **do**

3 Résoudre l'équation d'état réduite

$$\mathcal{N}_\varphi(\tilde{\alpha}^{(n)}, \theta^{(n)}) = 0$$

4 Connaissant $\tilde{\alpha}^{(n)}$, trouver $\beta^{(n)}$ solution de l'équation adjointe

$$\partial_{\tilde{\alpha}} \mathcal{N}_\varphi(\tilde{\alpha}^{(n)}, \theta^{(n)})^* \beta^{(n)} + \partial_{\tilde{\alpha}} \mathcal{J}_\varphi(\tilde{\alpha}^{(n)}, \theta^{(n)}) = 0$$

5 Évaluation de la direction de descente $d_n = -\partial_\theta \mathcal{L}_\varphi(\tilde{\alpha}^n, \theta^n, \beta^n)$

6 Recherche linéaire du pas σ_n (Armijo)

7 Mise à jour des paramètres de contrôle

$$\theta^{(n+1)} = P_K(\theta^{(n)} + \sigma_n d_n)$$

8 Critère de convergence

$$E = \frac{|\mathcal{J}_\varphi(\tilde{\alpha}^{(n+1)}, \theta^{(n+1)}) - \mathcal{J}_\varphi(\tilde{\alpha}^{(n)}, \theta^{(n)})|}{|\mathcal{J}_\varphi(\tilde{\alpha}^{(n+1)}, \theta^{(n+1)})|}, \quad \text{où } \mathcal{J}_\varphi(\tilde{\alpha}, \theta) = \mathcal{J}\left(\sum_{i=1}^q \tilde{\alpha}^i \varphi^i, \theta\right)$$

9 **end**

2.4.4 Exemple d'illustration

a) Position du problème :

Le but de ce paragraphe est de tester la robustesse des approches de contrôle optimal "Reduce Then Differentiate" et "Differentiate Then Reduce" sur le problème modèle de contrôle de l'équation de réaction diffusion non linéaire en agissant sur les forces extérieures. Soit $\Omega = [0, 1] \times [0, 1]$ un domaine borné et Γ sa frontière. Le problème étudié est

$$\begin{cases} \partial_t u - \nu \Delta u + \frac{1}{2} u^2 = f_c & \text{dans } \Omega \times]0, T] \\ u = g & \text{sur } \Gamma \times]0, T] \\ u(0) = u_0 & \text{dans } \Omega. \end{cases} \quad (2.84)$$

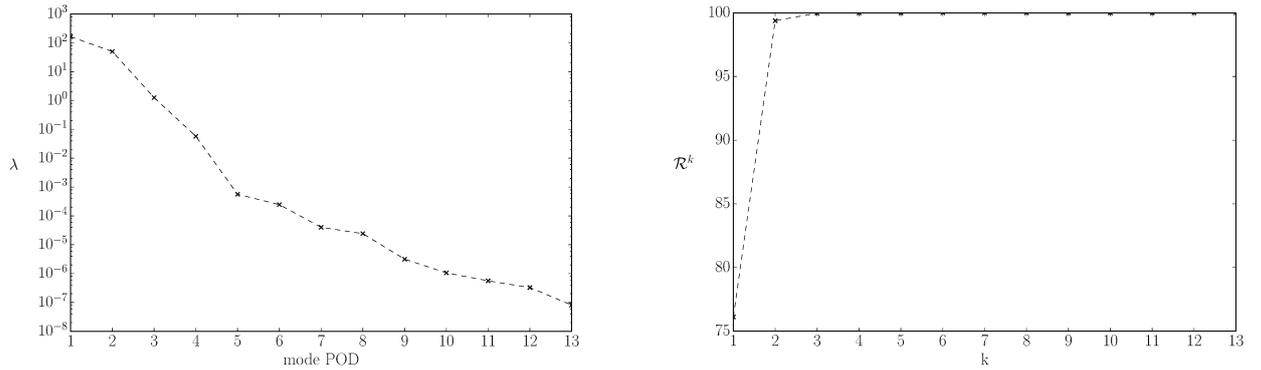
où $u_0 \in L^2(\Omega)$ est une condition initiale et $f_c \in L^2(]0, T], L^2(\Omega))$ désigne les forces extérieures appliquées sur le domaine Ω . Le terme source f_c est défini par la fonction Gaussienne centrée en un point (c_x, c_y) comme

$$f(t, x, y) = \sqrt{2} \cos\left(\frac{2\pi}{T_0} t\right) \exp\left(-\frac{(x - c_x)^2}{2\tilde{\sigma}_x^2} - \frac{(y - c_y)^2}{2\tilde{\sigma}_y^2}\right) \quad (2.85)$$

La fonction g des conditions aux limites de Dirichlet est donnée par l'expression

$$g(t, x) = \frac{1}{3} \cos(x^2 + y^2) \quad (2.86)$$

Pour ce cas d'étude, le coefficient de diffusion ν est choisi égal à 0.01, $T_0 = 2.5$, $\tilde{\sigma}_x = 0.3$, $\tilde{\sigma}_y = 0.7$, $c_x = 0.3$, $c_y = 0.7$. Le modèle complet est discrétisé et résolu par la méthode des éléments finis (Fenics [94]) sur l'intervalle temporel $]0, 10]$ en utilisant un maillage non uniforme composé de 5511 degrés de libertés.



(a) Valeurs propres de la matrice de corrélation

(b) Ratio \mathcal{R}^k entre les valeurs propres gardées et toutes les valeurs propres POD.

FIGURE 2.1 – Illustration des propriétés des valeurs propres POD.

b) Modèle d'ordre réduit

La base POD associée à ce problème a été construite par la méthode des snapshots en utilisant 500 clichés uniformément répartis sur l'intervalle temporel $[0, 10]$. Les valeurs propres POD, rangées par ordre décroissant sont tracées sur la figure 2.1a. La contribution des modes POD² dans la représentation de l'ensemble des snapshots est montrée sur la figure 2.1b. On constate que très peu de modes capturent la quasi totalité de l'information. Les q modes conservés correspondent alors aux q premières valeurs propres les plus significatives du problème POD. Ces modes sont tracés dans l'ordre selon leurs contribution dans la figure 2.2. Dans ce cas d'étude, les 6 premiers modes ont été conservés ($q = 6$). En effectuant la projection de Galerkin des équations (2.84) sur ces modes, on aboutit au système différentiel de petite taille suivant vérifié par les coefficients temporels

2. L'expression du ratio \mathcal{R}^k est donnée par
$$\mathcal{R}^k = \frac{\sum_{i=1}^k \lambda_i}{\sum_{i=1}^{N_s} \lambda_i}$$

$\alpha = \{\alpha^1, \dots, \alpha^q\}$, soit

$$\begin{cases} \sum_{j=1}^q \mathbf{M}_{ij} \frac{d\alpha^j}{dt} + \nu \sum_{j=1}^q \mathbf{R}_{ij} \alpha^j + \frac{1}{\epsilon} \sum_{j=1}^q \mathbf{B}_{ij} \alpha^j + \frac{1}{2} \sum_{j,k=1}^q \mathbf{N}_{ijk} \alpha^j \alpha^k = \mathbf{F}_i + \frac{1}{\epsilon} \mathbf{G}_i \\ \sum_{j=1}^q \mathbf{M}_{ij} \alpha^j(0) = \langle u_0, \varphi_i \rangle_{L^2(\Omega)} \\ \forall i = 1, \dots, q \end{cases} \quad (2.87)$$

Les coefficients du système dynamique réduit liés aux équations (2.84) sont

$$\begin{aligned} \mathbf{M}_{ij} &= \int_{\Omega} \varphi_j \varphi_i dx & \mathbf{N}_{ijk} &= \int_{\Omega} \varphi_j \varphi_k \varphi_i dx & \mathbf{F}_i &= \int_{\Omega} f \varphi_i dx \\ \mathbf{R}_{ij} &= \int_{\Omega} \nabla \varphi_j \cdot \nabla \varphi_i dx & \mathbf{B}_{ij} &= \int_{\Gamma} \varphi_j \varphi_i d\sigma & \mathbf{G}_i &= \int_{\Gamma} g \varphi_i d\sigma \end{aligned}$$

Les conditions aux limites de Dirichlet sont forcées dans le ROM au sens faible en utilisant

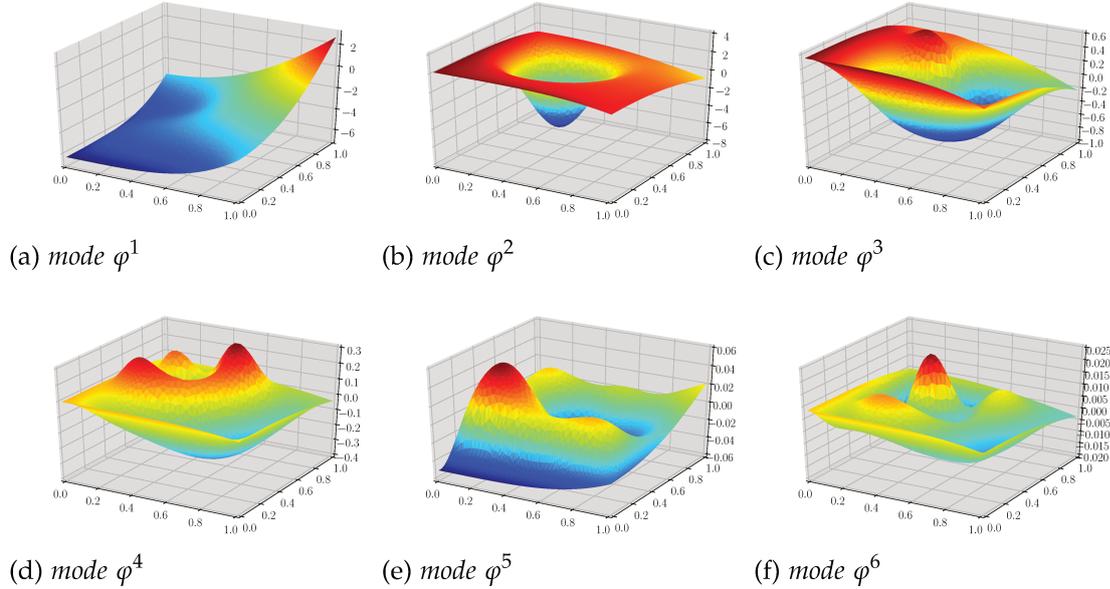


FIGURE 2.2 – Fonctions de base générées par la méthode POD.

la méthode de pénalisation [95]. Cette technique consiste à écrire la solution u sur Γ comme

$$u|_{\Gamma} = g - \epsilon \nu \frac{\partial u}{\partial n}$$

où ϵ est un réel qui tend vers 0. Quand $\epsilon \rightarrow 0$ la condition aux limites sur Γ va tendre vers une condition de Dirichlet. Par conséquent, on peut remplacer les conditions de Dirichlet par la condition de Neumann suivante

$$\nu \frac{\partial u}{\partial n} = \frac{u|_{\Gamma} - g}{\epsilon}$$

Soit u_q l'approximation de u en gardant q modes, l'erreur relative est mesurée en utilisant l'erreur de reconstruction moyenne $\varepsilon_{\%}^T$ définie par

$$\varepsilon_{\%}^T = 100 \times \frac{\left(\int_0^T \|u_q - u\|_{L^2(\Omega)}^2 dt \right)^{\frac{1}{2}}}{\left(\int_0^T \|u\|_{L^2(\Omega)}^2 dt \right)^{\frac{1}{2}}}$$

L'erreur de reconstruction du modèle réduit en fonction du nombre de modes est tracée dans la figure 2.3. On constate que très peu de modes assurent une bonne reconstruction de u . Ainsi, 3 modes assurent une erreur inférieure à 2% et 6 modes une erreur inférieure à 0.1%. De plus, l'erreur commise par le ROM est très proche de celle obtenue par POD.

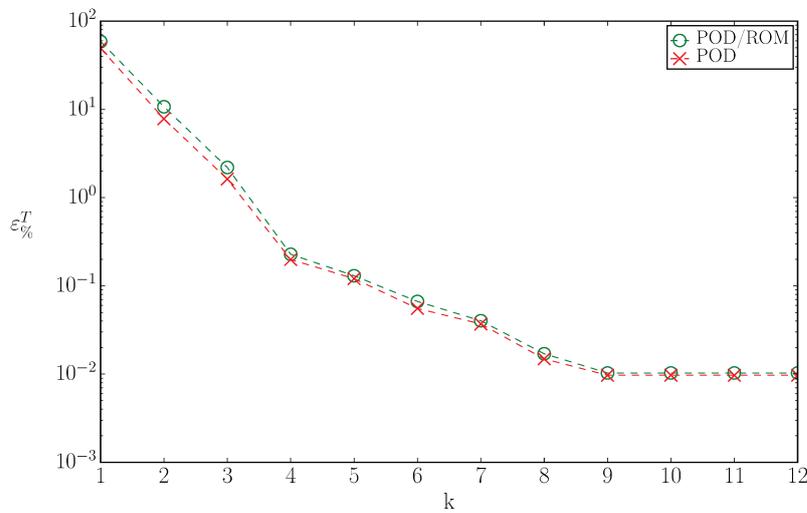


FIGURE 2.3 – Pourcentage d'erreur en fonction du nombre de modes considérés pour la reconstruction.

Écriture du problème de contrôle optimal réduit

Considérons maintenant le problème d'optimisation sous contraintes suivant

$$\min \mathcal{J}(u, \theta) \text{ sur } (u, \theta) \in V \times K \quad \text{tel que } u \text{ satisfait l'équation (2.84)} \quad (2.88)$$

où $V = L^2(0, T, H^1(\Omega))$ et $K = \{\theta \in Z / 0 < \theta < 1\}$ avec $Z = \mathbb{R}^2$. Pour $\kappa > 0$ un paramètre de régularisation et \hat{u} la solution cible associée à un paramètre $\hat{\theta} \in K$, on définit la fonctionnelle objectif $\mathcal{J} : V \times Z \rightarrow \mathbb{R}$ par

$$\mathcal{J}(u, \theta) = \frac{1}{2} \int_0^T \int_{\Omega} |u - \hat{u}|^2 dx dt + \frac{1}{2} \int_{\Omega} |u(T) - \hat{u}(T)|^2 dx + \frac{\kappa}{2} |\theta|^2 \quad (2.89)$$

où \hat{u} correspond à la solution du problème (2.84) pour \hat{c}_x et \hat{c}_y . Une base POD d'ordre $\hat{q} = 6$ est ensuite construite et \hat{u} est approximée comme suit

$$\hat{u} \approx \hat{u}_{\hat{q}} = \sum_{i=1}^{\hat{q}} \hat{\alpha}^i \hat{\varphi}^i \quad (2.90)$$

La fonctionnelle réduite s'écrit

$$\begin{aligned} \mathcal{J}_\varphi(\alpha, \theta) = & \frac{1}{2} \sum_{k=1}^q \sum_{j=1}^q \mathbf{I}_{k,j} \left(\int_0^T \alpha^k \alpha^j dt + \alpha^k(T) \alpha^j(T) \right) \\ & + \frac{1}{2} \sum_{l=1}^{\hat{q}} \sum_{q=1}^{\hat{q}} \hat{\mathbf{I}}_{l,q} \left(\int_0^T \hat{\alpha}^l \hat{\alpha}^q dt + \hat{\alpha}^l(T) \hat{\alpha}^q(T) \right) \\ & - \sum_{k=1}^q \sum_{l=1}^{\hat{q}} \mathbf{L}_{kl} \left(\int_0^T \alpha^k \hat{\alpha}^l dt + \alpha^k(T) \hat{\alpha}^l(T) \right) + \frac{\kappa}{2} |\theta|^2. \end{aligned} \quad (2.91)$$

telle que

$$\mathbf{I}_{kj} = \int_{\Omega} \varphi^k \varphi^j dx \quad \hat{\mathbf{I}}_{lq} = \int_{\Omega} \hat{\varphi}^l \hat{\varphi}^q dx \quad \mathbf{L}_{kl} = \int_{\Omega} \varphi^k, \hat{\varphi}^l dx$$

On rappelle que l'objectif de cette étude est de comparer les deux approches "Reduce Then Differentiate" et "Differentiate Then Reduce".

Approche "Differentiate Then Reduce" :

En notant ζ la variable adjointe associée à l'équation contrainte (2.84), le problème adjoint s'écrit

$$\begin{cases} \partial_t \zeta + \nu \Delta \zeta - u \zeta = u - \hat{u} & \text{in } \Omega \times]0, T] \\ \zeta = 0 & \text{on } \Gamma \times]0, T] \\ \zeta(T) = \hat{u}(T) - u(T) & \text{in } \Omega. \end{cases} \quad (2.92)$$

De manière identique au problème d'état, un ensemble de snapshots obtenu en résolvant l'équation adjointe a été considéré pour construire la base POD associée au paramètre $(c_x, c_y) = (0.3, 0.7)$. La variable adjointe est ensuite approximée dans cette base comme suit

$$\zeta \approx \sum_{i=1}^{q^*} \zeta^i \psi^i$$

En injectant cette expression dans l'équation adjointe (2.92), puis en effectuant la projection de Galerkin sur la base POD adjointe de taille $q^* = 6$, le modèle réduit de l'équation adjointe s'écrit

$$\begin{cases} \sum_{j=1}^{q^*} \mathbf{M}_{ij}^* \frac{d\zeta^j}{dt} - \nu \sum_{j=1}^{q^*} \mathbf{R}_{ij}^* \zeta^j - \sum_{j,k=1}^{q^*} \mathbf{N}_{ijk}^* \alpha^j \zeta^k = \mathbf{F}_i^* \\ \sum_{j=1}^{q^*} \mathbf{M}_{ij}^* \alpha^j(0) = -\mathbf{F}_i^*(T) \\ \forall i = 1, \dots, q^* \end{cases} \quad (2.93)$$

avec

$$\begin{aligned} \mathbf{M}_{ij}^* &= \int_{\Omega} \psi^j \psi^i dx & \mathbf{N}_{ijk}^* &= \int_{\Omega} \varphi^k \psi^j \psi^i dx \\ \mathbf{R}_{ij}^* &= \int_{\Omega} \nabla \psi^j \nabla \psi^i dx & \mathbf{F}_i^* &= \sum_{k=1}^q \alpha^k \int_{\Omega} \varphi^k \psi^i dx - \sum_{l=1}^{\hat{q}} \hat{\alpha}^l \int_{\Omega} \hat{\varphi}^l \psi^i dx \end{aligned}$$

Les directions de descente sont alors données par :

$$d_{c_x} = -\sum_{j=1}^{q^*} \frac{\sqrt{2}}{\sigma_x^2} \int_0^T \zeta^j \cos\left(\frac{2\pi}{T_0}t\right) dt \int_{\Omega} \psi^j(x - c_x) \exp\left(-\frac{(x - c_x)^2}{2\sigma_x^2} - \frac{(y - c_y)^2}{2\sigma_y^2}\right) dx - \kappa c_x$$

$$d_{c_y} = -\sum_{j=1}^{q^*} \frac{\sqrt{2}}{\sigma_y^2} \int_0^T \zeta^j \cos\left(\frac{2\pi}{T_0}t\right) dt \int_{\Omega} \psi^j(y - c_y) \exp\left(-\frac{(x - c_x)^2}{2\sigma_x^2} - \frac{(y - c_y)^2}{2\sigma_y^2}\right) dx - \kappa c_y$$

Approche "Reduce Then Differentiate" :

Le problème réduit adjoint "Reduce Then Differentiate" quant à lui est obtenu directement à partir du modèle réduit (2.87). Il est donné par le modèle réduit suivant

$$\begin{cases} \sum_{i=1}^q M_{ij} \frac{d\beta^i}{dt} - \nu \sum_{i=1}^q R_{ij} \beta^i - \sum_{i,k=1}^q N_{ijk} \alpha^k \beta^i = \sum_{i=1}^q I_{ji} \alpha^i - \sum_{i=1}^{\hat{q}} L_{ji} \hat{\alpha}^i \\ \sum_{i=1}^q M_{ij} \beta^i(T) = \sum_{i=1}^{\hat{q}} L_{ji} \hat{\alpha}^i(T) - \sum_{i=1}^q I_{ji} \alpha^i(T) \\ \forall j = 1, \dots, q \end{cases}$$

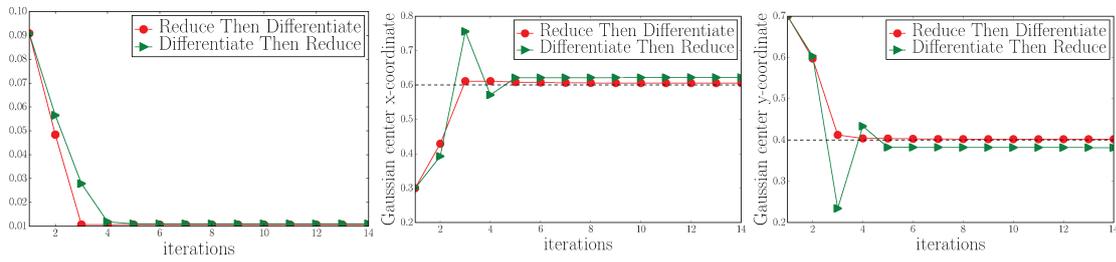
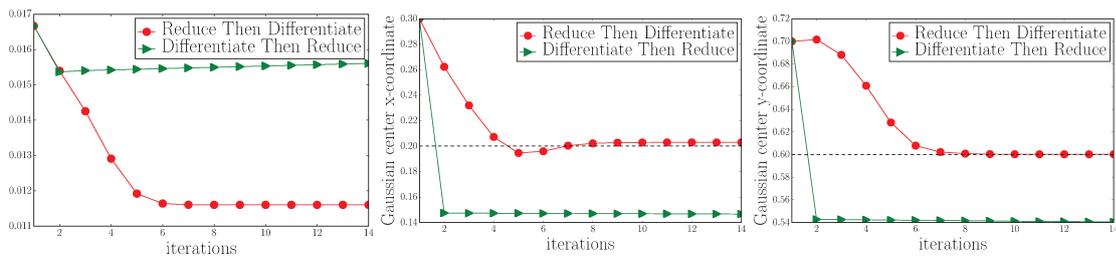
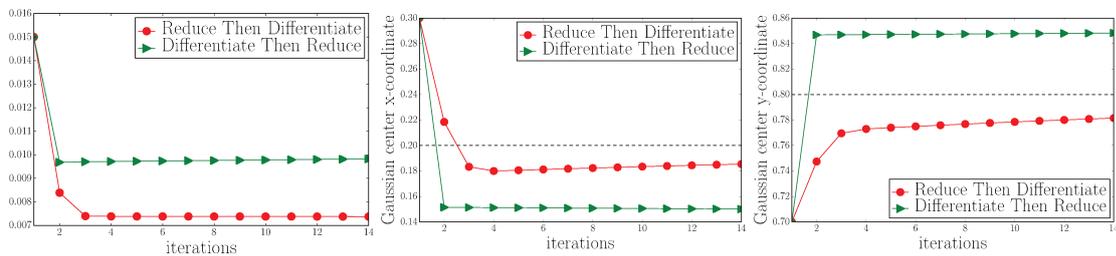
Les directions de descente dans ce cas s'écrivent :

$$d_{c_x} = \frac{\sqrt{2}}{\sigma_x^2} \sum_{j=1}^q \int_0^T \cos\left(\frac{2\pi}{T_0}t\right) \beta^j dt \int_{\Omega} \varphi^j(x - c_x) \exp\left(-\frac{(x - c_x)^2}{2\sigma_x^2} - \frac{(y - c_y)^2}{2\sigma_y^2}\right) dx - \kappa c_x$$

$$d_{c_y} = \frac{\sqrt{2}}{\sigma_y^2} \sum_{j=1}^q \int_0^T \cos\left(\frac{2\pi}{T_0}t\right) \beta^j dt \int_{\Omega} \varphi^j(x - c_y) \exp\left(-\frac{(x - c_x)^2}{2\sigma_x^2} - \frac{(y - c_y)^2}{2\sigma_y^2}\right) dx - \kappa c_y$$

Comparaison des deux approches :

Nous choisissons pour la présente étude de construire les problèmes de contrôle optimal "Reduce Then Differentiate" et "Differentiate Then Reduce" en utilisant la base POD associée au paramètre $(c_x, c_y) = (0.3, 0.7)$. Trois cas tests en considérant trois solutions cibles obtenues pour les paramètres $(c_x, c_y) = (0.4, 0.8)$, $(c_x, c_y) = (0.2, 0.6)$ et $(c_x, c_y) = (0.2, 0.8)$ ont été étudiés. On constate que l'approche "Differentiate Then Reduce" converge moins bien que l'approche "Reduce Then Differentiate". Cela peut être dû aux erreurs cumulées par les modèles réduits de l'équation d'état ainsi que celui de l'équation adjointe. Ces erreurs affectent la qualité de prédiction des directions de descentes déterminées par le gradient de la fonction de Lagrange. Le domaine de validité de la base POD associée au paramètre $(c_x, c_y) = (0.3, 0.7)$ est évaluée en calculant l'erreur relative pour différents points dans l'espace des paramètres admissibles. La figure 2.7 montre que le ROM peut prédire la dynamique temporelle juste sur un voisinage du paramètre pour lequel la base POD a été construite. Plus on s'éloigne de ce voisinage plus l'erreur relative tend à augmenter. Par conséquent, dans le contrôle optimal réduit, l'utilisation d'une base POD construite pour un paramètre qui est loin du paramètre cible, risque de causer une mauvaise prédiction des paramètres de contrôle. C'est notamment le cas pour la base POD construite pour $(c_x, c_y) = (0.2, 0.8)$ (figure 2.6).

(a) évolution de \mathcal{J}_φ (b) évolution du paramètre c_x (c) évolution du paramètre c_y FIGURE 2.4 – Comportement de la de l’algorithme de contrôle optimal réduit. Le paramètre optimal est $\theta = (0.6, 0.4)$.(a) évolution de \mathcal{J}_φ (b) évolution du paramètre c_x (c) évolution du paramètre c_y FIGURE 2.5 – Comportement de la de l’algorithme de contrôle optimal réduit. Le paramètre optimal est $\theta = (0.2, 0.6)$.(a) évolution de \mathcal{J}_φ (b) évolution du paramètre c_x (c) évolution du paramètre c_y FIGURE 2.6 – Comportement de la de l’algorithme de contrôle optimal réduit. Le paramètre optimal est $\theta = (0.2, 0.8)$.

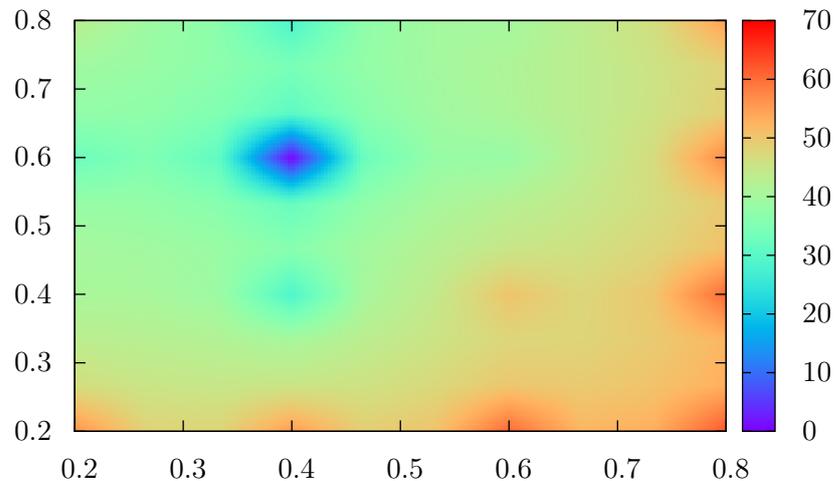


FIGURE 2.7 – Erreur moyenne entre le modèle complet et le modèle réduit construit par projection de Galerkin des équations du modèle complet sur la base POD construite pour le paramètre $\theta = (0.4, 0.6)$

2.5 CONCLUSION DU CHAPITRE

Dans ce chapitre, différentes techniques de réduction de modèles ont été rappelées, avec un focus sur la méthode POD en raison de sa capacité à produire une base spatiale optimale au sens de la moyenne quadratique. La dynamique temporelle est obtenue ensuite par la résolution d'un système d'équations différentielles de petite taille obtenu par la projection de Galerkin de l'équation gouvernant le problème d'origine sur les fonctions de base POD.

Les modèles réduits ont alors été utilisés pour formuler un problème de contrôle optimal réduit dont la résolution est beaucoup plus rapide que le problème de contrôle optimal classique.

La POD étant une méthode a posteriori, elle dépend alors fortement des informations fournis par les simulations ou les données expérimentales. La capacité d'une base spatiale fixée à reproduire la dynamique du problème pour les paramètres de contrôle susceptibles d'être balayés par l'algorithme d'optimisation, a été testée pour le cas du problème de réaction-diffusion. Les résultats ont confirmé que le choix de la base POD est cruciale pour assurer un bon comportement des algorithmes de contrôle réduit "état-état" ou "état-adjoint". Une solution proposée par Bergmann et al. [96], consiste à utiliser la méthode des régions de confiance, qui fournit un indice sur la validité de la base et la met à jour si nécessaire en résolvant le modèle complet. Cette approche, bien qu'efficace, reste tout de même coûteuse en temps de calcul. Par conséquent, il est plus intéressant de chercher des méthodes de mise à jour de bases réduites moins coûteuses, qui éviteront le recours

à la résolution du modèle complet. Les chapitres suivants 3 et 4 seront consacrés à la présentation des approches permettant d'adapter rapidement les bases réduites en fonction du changement des paramètres de contrôle. Comme les résultats sur l'exemple du contrôle de l'équation de réaction-diffusion ont montré que le contrôle optimal réduit par l'approche "état-état" est plus efficace que l'approche "état-adjoint", c'est cette approche qui sera adoptée par la suite dans les applications au chapitre 5.

ADAPTATION DES MODÈLES D'ORDRE RÉDUIT

3

SOMMAIRE

3.1	INTRODUCTION	62
3.2	MÉTHODES CLASSIQUES D'INTERPOLATION	63
3.3	MÉTHODE ITSGM POUR L'ADAPTATION DE BASES RÉDUITES	66
3.3.1	Quelques rappels de géométrie différentielle	66
3.3.2	Interpolation des bases POD	70
3.3.3	Coût asymptotique de la méthode ITSGM	73
3.4	ADAPTATION DE BASES RÉDUITES PAR PGD	73
3.4.1	PGD basée sur le critère d'orthogonalité de Galerkin	74
3.4.2	Mise en forme de la base PGD	76
3.5	CONCLUSION DU CHAPITRE	79

3.1 INTRODUCTION

Dans le cadre du contrôle optimal, l'utilisation de l'approche POD/ROM nécessite une attention particulière. En effet, la capacité des fonctions de bases POD à reproduire la dynamique du phénomène étudié est dépendante de l'information originellement contenue dans l'ensemble des snapshots utilisés pour leur construction. Supposons que ces snapshots correspondent à un paramètre de contrôle $\theta_{const} \in \mathbb{R}^d$. Dans la plupart des cas, la base POD ainsi construite, n'est plus représentative de la dynamique du phénomène pour une perturbation du paramètre θ_{const} . Ainsi, dans le cadre du contrôle optimal, une seule et même base, construite pour un paramètre de contrôle donné θ_{const} , ne sera pas capable de prédire la dynamique du phénomène pour l'ensemble des paramètres θ_{algo} balayé par l'algorithme de contrôle. Il s'avère donc nécessaire de construire des bases POD robustes vis à vis de la variation des paramètres de contrôle, ou de développer des techniques d'adaptation de bases POD.

Pour augmenter le domaine de validité de la base POD, il est possible de construire une base POD à partir de snapshots provenant de simulations correspondantes à différentes valeurs $\theta_{const}^1, \dots, \theta_{const}^{N_p}$ du paramètre de contrôle. Cette approche a notamment été appliquée avec succès par Akman et al. [93] dans le contrôle optimal des équations de réaction-convection diffusion, par Tallet et al. [35] pour le contrôle de l'écoulement anisotherme dans une cavité entraînée et par Mifsud et al. [97] dans le contexte de la réduction de modèle de l'aérodynamique des armes. L'un des inconvénients majeurs de cette approche, notée par la suite MPS (Multiple Parametrized Snapshots) est la construction coûteuse du problème aux valeurs propres permettant de calculer la base MPS-POD. En effet, la taille de la matrice de corrélation dans ce cas est $N_p N_s \times N_p N_s$ où N_s désigne le nombre de snapshots pour un paramètre donné et N_p le nombre de paramètres. Bien que la base POD/MPS permette une flexibilité dans l'algorithme de contrôle vis à vis du changement de paramètres, elle reste tout de même optimale uniquement pour les points de l'échantillonnage, et peut pour certains paramètres de contrôle ne pas être représentative du phénomène physique.

Une autre approche possible, consiste à interpoler un ensemble de bases POD $\varphi_1, \dots, \varphi_{N_b}$ construites pour différentes valeurs $\theta_1, \dots, \theta_{N_b}$ du paramètre de contrôle, pour obtenir la base associée au paramètre de contrôle souhaité. Les techniques d'interpolation standards (RBF, Lagrange, Spline ...) s'avèrent peu efficaces car elles n'assurent pas l'obtention d'une base, et même si elles résultent en une base, cette dernière n'est pas en général représentative du phénomène étudié. Pour contourner cette difficulté Amsallem et al. [98] ont proposé une approche d'interpolation de bases fondée sur les résultats de la géométrie différentielle, et plus particulièrement, sur les propriétés de la variété de Grassmann. Cette approche, notée par la suite ITSGM (Interpolation on the Tangent Subspace of the Grassmann Manifold) a été appliquée avec succès dans le contexte de l'aéroélasticité. Une fois qu'une approximation de la base souhaitée est obtenue par interpolation, un nouveau modèle d'ordre réduit est construit via la projection de Galerkin de cette dernière sur les

équations du modèle complet.

Au lieu d'interpoler les bases puis de construire le ROM, Amsallem et al. [99] ont proposé de construire le modèle réduit associé à chaque base de l'échantillonnage des bases POD, puis d'interpoler les matrices réduites résultantes de la projection de Galerkin. La méthodologie est similaire à celle de l'interpolation des bases, sauf que cette fois-ci, la variété considérée est celle des matrices symétriques définies positives. Cette méthode a été appliquée avec succès pour l'adaptation des modèles d'ordre réduit issus des problèmes linéaires de la dynamique structurelle.

Ce chapitre débute par un bref rappel des méthodes d'interpolation usuelles qui seront utilisées dans ce manuscrit. La deuxième section concerne la description de la méthode d'interpolation de bases réduite ITSGM. Finalement, la troisième section sera consacrée à la description d'une méthode d'adaptation de bases réduites qui consiste à enrichir, en utilisant la méthode PGD (Proper Generalized Decomposition) la base associée à un paramètre donné, pour qu'elle soit valide pour un nouveau paramètre.

3.2 MÉTHODES CLASSIQUES D'INTERPOLATION

Supposons qu'il existe une fonction $f(x)$ inconnue pour laquelle une distribution de $n + 1$ données $f(x_0), \dots, f(x_n) \in \mathbb{R}^{N_x} \times \dots \times \mathbb{R}^{N_x}$ en $n + 1$ points distincts $x_0, \dots, x_n \in \mathbb{R}^{d_p} \times \dots \times \mathbb{R}^{d_p}$ est connue. Le problème d'interpolation consiste à construire une fonction $g(x)$ qui coïncide avec $f(x)$ aux points x_0, \dots, x_n , i.e.,

$$g(x_k) = f(x_k), \quad 0 \leq k \leq n, \quad (3.1)$$

Une manière simple pour obtenir cette fonction est de relier les valeurs connues par une ligne droite. Bien que ce soit une solution légitime, il est parfois préférable de chercher une fonction approchée qui soit plus régulière. Par exemple il est possible de chercher $g(x)$ dans la classe des fonctions polynomiales. De manière alternative, la fonction $g(x)$ peut être une fonction trigonométrique ou polynomiale régulière par morceaux etc ...

Dans ce qui suit, nous allons rapidement rappeler les méthodes usuelles d'interpolation qui seront utilisées dans ce manuscrit. Sans perte de généralités, ces méthodes seront détaillées pour le cas $d_p = 1$.

a) Interpolation de Lagrange :

L'interpolation de Lagrange suppose que $g(x)$ est recherchée sous la forme d'une combinaison linéaire de polynômes de degrés $0, \dots, n$ comme suit

$$g_n(x) = \sum_{k=0}^n f(x_k) l_k^n$$

où l_k^n sont des polynômes de degré au plus n . En imposant la condition (3.1), et en tenant compte de la condition

$$l_k^n(x_i) = \delta_i^k, \quad i, k = 0, \dots, n$$

où δ_i^k est l'indice de Kronecker défini par

$$\delta_i^k = \begin{cases} 1, & i = k \\ 0, & i \neq k \end{cases}$$

les polynômes l_k^n de degré n peuvent être construits comme suit

$$l_k^n = \frac{(x - x_0) \cdots (x - x_{k-1})(x - x_{k+1}) \cdots (x - x_n)}{(x_k - x_0) \cdots (x_k - x_{k-1})(x_k - x_{k+1}) \cdots (x_k - x_n)}$$

b) Interpolation par spline

Une courbe spline $S(x)$ est une fonction polynomiale par morceaux définie sur un intervalle $[x_L, x_R]$ divisé en sous-intervalles $[x_{k-1}, x_k]$. Supposons que nous disposons de $n + 1$ données $f(x_0), \dots, f(x_n)$ telles que

$$x_L = x_0 < x_1 < \cdots < x_n = x_R$$

La fonction spline d'interpolation $S(x)$ est alors déterminée sur chaque intervalle $[x_{k-1}, x_k]$ par un polynôme cubique. On note $S_k(x)$ la restriction de $S(x)$ sur l'intervalle $[x_{k-1}, x_k]$. Étant donné que $S_k(x)$ est un polynôme cubique déterminé par 4 coefficients, le nombre total de paramètres à déterminer est alors $4n$. Ces paramètres sont déterminés en vérifiant les conditions suivantes :

- (I). Sur chaque intervalle $[x_{k-1}, x_k]$, $S(x)$ doit vérifier $S_k(x_k) = f(x_k)$ et $S_k(x_{k+1}) = f(x_{k+1})$.
- (II). $S'(x)$ doit être continue en tous nœuds intérieurs, c'est à dire, pour tout $k = 1, \dots, n - 1$ on doit avoir $S'_{k-1}(x_k) = S'_k(x_k)$.
- (III). $S''(x)$ doit être continue en tous nœuds intérieurs, c'est à dire, $S''_{k-1}(x_k) = S''_k(x_k)$.
- (IV). L'un des choix de conditions au limites aux points x_L et x_R doit être considérée
 - (a). Les conditions d'Hermite

$$S'(x_L) = f'(x_L), \quad S'(x_R) = f'(x_R)$$

Cela suppose que $f'(x_L)$ et $f'(x_R)$ sont connus.

- (b). Les conditions périodiques

$$S'(x_L) = S'(x_R), \quad S''(x_L) = S''(x_R)$$

- (c). Les conditions naturelles

$$S''(x_L) = S''(x_R)$$

c) Interpolation RBF

L'interpolation RBF (Radial Basis Function) consiste à chercher une solution sous la forme

$$g_n(x) = \sum_{k=0}^n \lambda_k \phi(d(x, x_k)),$$

avec $\lambda_k \in \mathbb{R}$. $d(x, y) = \|x - y\|$ est une métrique dans l'espace euclidien, et $\phi : [0, \infty[\rightarrow \mathbb{R}$ est la fonction de base radiale. Tenons compte de la condition (3.1), on obtient alors le système linéaire vérifié par les λ_k :

$$A_\phi \lambda = f$$

où

$$A_\phi \in \mathbb{R}^{n+1 \times n+1} \quad \text{est définie par} \quad A_{\phi,ij} = \phi(d(x_i, x_j))$$

L'existence d'un unique interpolant exige alors que la matrice d'interpolation soit non singulière. En particulier, l'unicité est assuré en choisissant par exemple l'une des fonctions de bases suivantes :

Soit $r > 0$ et $\varepsilon > 0$

$\phi(r) = e^{-\varepsilon r^2}$	fonction Gaussienne
$\phi(r) = \frac{1}{\sqrt{r^2 + \varepsilon^2}}$	fonction multi-quadratique inverse
$\phi(r) = \sqrt{r^2 + \varepsilon^2}$	fonction multi-quadratique
$\phi(r) = r$	fonction linéaire
$\phi(r) = r^3$	fonction cubique

où r et ε sont deux nombres réels strictement positifs. Dans ce manuscrit, nous utiliserons la méthode RBF en choisissant comme fonction radiale la fonction Gaussienne.

d) Interpolation IDW

La méthode d'interpolation IDW (Inverse Distance Weighting) part du principe que les observations ont un poids plus ou moins important selon leurs éloignement. Ainsi, plus l'observation est éloignée, moins sa contribution sera importante. Étant donné un échantillonnage x_0, \dots, x_n , la forme générale qui permet de trouver la fonction d'interpolation $g(x)$ en utilisant l'interpolation IDW est donnée par

$$g(x) = \begin{cases} \frac{\sum_{k=0}^n \omega_k(x) f(x_k)}{\sum_{k=0}^n \omega_k}, & \text{si } d(x, x_k) \neq 0, \forall k \in \{0, \dots, n\} \\ f(x_k), & \text{si } d(x, x_k) = 0 \end{cases}$$

où

$$\omega_k(x) = \frac{1}{d(x, x_k)^p}$$

est la fonction poids IDW [100], p un nombre réel positif et $d(\cdot, \cdot)$ une métrique bien définie.

e) Limitations

Bien que les méthodes d'interpolation usuelles permettent parfois d'obtenir des approximations légitimes de données, elles ne sont pas tout le temps robustes en réduction de modèles. En effet, d'un point de vue mathématique, une interpolation classique d'un ensemble de bases réduites de dimension q ne formera pas forcément une base de dimension q . Un soin particulier doit donc être apporté dans le cas de l'interpolation de bases réduites. Le but du paragraphe suivant, est de surmonter cette difficulté en utilisant les outils de la géométrie différentielle.

3.3 MÉTHODE ITSGM POUR L'ADAPTATION DE BASES RÉDUITES

3.3.1 Quelques rappels de géométrie différentielle

a) Rappel sur les variétés différentielles

Une variété \mathcal{M} de dimension q est un espace topologique qui ressemble localement à un espace Euclidien de dimension q . Cette variété est dite différentielle si elle permet la définition des dérivées des trajectoires. La dérivée d'une trajectoire $\mathcal{Y}(t)$ en un point $[\boldsymbol{\varphi}] \in \mathcal{M}$ est le vecteur noté $\dot{\mathcal{Y}}(t)$ qui vit dans $\mathcal{T}_S\mathcal{M}$ appelé espace tangent à la variété \mathcal{M} au point $[\boldsymbol{\varphi}]$.

Une variété Riemannienne est une variété différentielle munie d'un produit scalaire $\langle \cdot, \cdot \rangle_{[\boldsymbol{\varphi}]}$ défini de manière unique sur tout espace tangent $\mathcal{T}_S\mathcal{M}$. Avec ce produit scalaire, la longueur d'une trajectoire $\mathcal{Y} : [0, T] \rightarrow \mathcal{M}$ est définie par

$$L(\mathcal{Y}) = \int_0^T \|\dot{\mathcal{Y}}(t)\|_{\mathcal{Y}(t)} dt \quad (3.2)$$

où $\|\cdot\|_{\mathcal{Y}(t)}$ est la norme induite par le produit scalaire $\langle \cdot, \cdot \rangle_{\mathcal{Y}(t)}$ sur $\mathcal{T}_{\mathcal{Y}(t)}\mathcal{M}$. La distance $d([\boldsymbol{\varphi}], [\boldsymbol{\varphi}'])$ entre deux points de la variété est celle qui réalise le minimum des distances de toutes les trajectoires possibles entre les deux points $[\boldsymbol{\varphi}]$ et $[\boldsymbol{\varphi}']$.

Le chemin qui minimise la distance entre deux points dans la variété est appelée géodésique. Ce chemin est aussi une trajectoire définie par une équation différentielle de second ordre [101]. Par conséquent, une géodésique est définie de façon unique par deux conditions initiales. Ces dernières sont typiquement la solution initiale et la vitesse initiale [102]. Une géodésique peut être paramétrée par la variable scalaire $t \in [0, 1]$ comme $\mathcal{Y} : [0, 1] \rightarrow \mathcal{M}$. Dans ce cas, $\mathcal{Y}(0)$ est la solution initiale et $\mathcal{Y}(1)$ est la solution finale de la géodésique. L'unicité de la géodésique permet de définir l'application exponentielle¹ $\text{Exp}_{[\boldsymbol{\varphi}]} : \mathcal{T}_S\mathcal{M} \rightarrow \mathcal{M}$ qui relie l'espace tangent en un point $[\boldsymbol{\varphi}]$ d'une variété différentielle à elle-même. La définition de l'application exponentielle géodésique est donnée par :

Soit $\mathcal{Y}(t)$ une géodésique sur la variété différentielle \mathcal{M} , déterminée de manière unique par les conditions initiales $\mathcal{Y}(0)$ et $\dot{\mathcal{Y}}(0)$. L'exponentielle de $\dot{\mathcal{Y}}(0)$, notée $\text{Exp}_{\mathcal{Y}_0}(\dot{\mathcal{Y}}_0)$ est

1. L'exponentielle usuelle est un cas particulier

par définition

$$\text{Exp}_{\mathcal{Y}_0}(\dot{\mathcal{Y}}_0) = \mathcal{Y}(1) \quad (3.3)$$

b) Variété de Grassmann $\mathcal{G}(q, N_x)$

Ces dernières années, le calcul sur la variété de Grassmann [103][104] a suscité l'intérêt de la communauté scientifique par de nombreuses applications, notamment en "subspace tracking" [105], en "clustering" [106], en "sparse coding" [107][108], et récemment en réduction de modèles [98]. Dans ce paragraphe nous allons donc présenter brièvement la dérivation des formules permettant le calcul pratique dans cette variété.

Soit $\boldsymbol{\varphi} \in \mathbb{R}_*^{N_x \times q}$ où $q \ll N_x$ et $\mathbb{R}_*^{N_x \times q}$ l'ensemble des matrices de rang maximal. Les colonnes de $\boldsymbol{\varphi}$ forment une base dans \mathbb{R}^{N_x} d'un sous espace $[\boldsymbol{\varphi}]$ de dimension ² q . L'ensemble de tout ces sous espaces $[\boldsymbol{\varphi}]$ de dimension q forme ce qu'on appelle la variété de Grassmann $\mathcal{G}(q, N_x)$. Les matrices $\boldsymbol{\varphi} \in \mathbb{R}_*^{N_x \times q}$ qui représentent ces sous espaces appartiennent à la variété non compacte de Stiefel³ $\mathcal{ST}(q, N_x)$ [101][109]. Cependant, il est possible de considérer ces matrices dans la variété de Stiefel compacte qui est un sous-ensemble de $\mathcal{ST}(q, N_x)$ contenant uniquement les matrices qui sont orthonormales

$$\mathcal{ST}(q, N_x) = \left\{ \boldsymbol{\varphi} \in \mathbb{R}^{N_x \times q} / \boldsymbol{\varphi}^T \boldsymbol{\varphi} = I_q \right\}$$

Les quatre principales représentations matricielles de la variété de Grassmann $\mathcal{G}(q, N_x)$ sont :

(I). Représentation via les matrices de rang maximal [110] :

$$\mathcal{G}(q, N_x) \cong \mathbb{R}_*^{N_x \times q} / GL(q)$$

(II). Représentation orthogonale [101] :

$$\mathcal{G}(q, N_x) \cong \mathcal{O}(N_x) / \mathcal{O}(q) \times \mathcal{O}(N_x - q)$$

(III). Représentation via la variété de Stiefel [101] :

$$\mathcal{G}(q, N_x) \cong \mathcal{ST}(q, N_x) / \mathcal{O}(q)$$

(IV). Représentation via les matrices symétriques idempotentes [111] :

$$\mathcal{G}(q, N_x) \cong \left\{ \mathbf{P} \in \mathbb{R}^{q \times q} : \mathbf{P}^T = \mathbf{P}, \mathbf{P}^2 = \mathbf{P}, \text{rang}(\mathbf{P}) = q \right\}$$

La représentation (IV) est très répandue pour les tâches d'apprentissage sur la variété de Grassmann, par exemple, en analyse discriminante [112], pour les méthodes du noyau de Grassmann [113], et en représentation Grassmannienne de rang faible [114]. Dans les représentations (I), (II), et (III), un point $[\boldsymbol{\varphi}]$ de la variété de Grassmann $\mathcal{G}(q, N_x)$ est identifié à une classe d'équivalence suivant les espaces quotients associés. Par exemple, dans

2. Les colonnes de ces matrices pourraient être construites à partir d'une base POD, dans ce cas N_x indique le nombre de degrés de liberté du modèle complet et q la taille de la base d'ordre réduit.

3. La variété non compacte de Stiefel est composée de toutes les matrices de taille $N_x \times q$ et de rang q .

la représentation (III), un point $[\boldsymbol{\varphi}] \in \mathcal{G}(q, N_x)$ est vu comme une matrice représentante $\boldsymbol{\varphi} \in \mathcal{ST}(q, N_x)$ issu de la classe d'équivalence $[\boldsymbol{\varphi}]$ donnée par

$$[\boldsymbol{\varphi}] = \{\boldsymbol{\varphi} \mathbf{Q} : \forall \mathbf{Q} \in \mathcal{O}(q)\}$$

Comme les matrices représentatives $\boldsymbol{\varphi}$ sont considérées dans la variété de Stiefel compacte, chaque point de la variété de Grassmann est alors défini par la classe d'équivalence suivante

$$[\boldsymbol{\varphi}] = \{\boldsymbol{\varphi} \mathbf{Q} : \boldsymbol{\varphi}^T \boldsymbol{\varphi} = I_q, \mathbf{Q} \in \mathcal{O}(q)\}$$

c) Espace tangent en un point de la variété $\mathcal{G}(q, N_x)$

En chaque point $[\boldsymbol{\varphi}] = \text{span}(\boldsymbol{\varphi})$ de $\mathcal{G}(q, N_x)$, il existe un espace tangent de même dimension noté $\mathcal{T}_{[\boldsymbol{\varphi}]} \mathcal{G}(q, N_x)$, pour lequel le point d'origine est le point de tangence. Chaque point de cet espace tangent peut être représenté par une matrice $\mathbf{H} \in \mathbb{R}^{N_x \times q}$. Plus concrètement, l'espace tangent peut être représenté par l'ensemble suivant [103]

$$\mathcal{T}_{[\boldsymbol{\varphi}]} \mathcal{G}(q, N_x) = \{\mathbf{H} \in \mathbb{R}^{N_x \times q} : \boldsymbol{\varphi}^T \mathbf{H} = 0\}$$

d) Géodésique dans $\mathcal{G}(q, N_x)$

Pour le cas d'une variété de Grassmann, la paramétrisation de la géodésique est donnée par le résultat [101][109] suivant

Soit \mathcal{Y}_0 un point de la variété de Grassmann $\mathcal{G}(q, N_x)$ représenté par la matrice de rang maximal $\boldsymbol{\varphi} \in \mathcal{ST}(q, N_x)$, et χ le point dans l'espace tangent $\mathcal{T}_{\mathcal{Y}_0} \mathcal{G}(q, N_x)$, représenté par la matrice $\mathbf{H} \in \mathbb{R}^{N_x \times q}$ dont la décomposition en valeurs singulières est $\mathbf{H} = \mathbf{U} \boldsymbol{\Sigma} \mathbf{V}^T$. La représentation paramétrique sur $\mathcal{G}(q, N_x)$ de la géodésique ayant les conditions initiales $\mathcal{Y}(0) = \mathcal{Y}_0$ et $\dot{\mathcal{Y}}(0) = \chi$ est donnée pour tout $\mathbf{Q} \in \mathcal{O}(q)$ par

$$\mathcal{Y}(t) = \text{span} \left\{ \boldsymbol{\varphi} \mathbf{V} \cos(t \boldsymbol{\Sigma}) \mathbf{Q}^T + \mathbf{U} \sin(t \boldsymbol{\Sigma}) \mathbf{Q}^T \right\} \quad 0 \leq t \leq 1 \quad (3.4)$$

Pour simplifier, un choix possible est de considérer par exemple $\mathbf{Q} = I_q$ ou $\mathbf{Q} = \mathbf{V}$.

e) Application exponentielle géodésique $\text{Exp}_{[\boldsymbol{\varphi}]}$

Un intérêt pratique de la variété de Grassmann est le fait que c'est une variété différentielle sur laquelle une formule explicite de l'application $\text{Exp}_{[\boldsymbol{\varphi}]}$ peut être formulée. La formule explicite pour l'application exponentielle entre un point de l'espace tangent à la variété elle-même est établi par le résultat [115] suivant

Soient $\boldsymbol{\varphi} \in \mathcal{ST}(q, N_x)$ une matrice de rang maximal dont les colonnes engendrent le sous-espace $[\boldsymbol{\varphi}] \in \mathcal{G}(q, N_x)$, et χ un point de l'espace tangent $\mathcal{T}_{[\boldsymbol{\varphi}]} \mathcal{G}(q, N_x)$ représenté par la matrice \mathbf{H} où $\mathbf{H} = \mathbf{U} \boldsymbol{\Sigma} \mathbf{V}^T$ est sa décomposition en valeurs singulières. L'application Exponentielle $\text{Exp}_{[\boldsymbol{\varphi}]}$ qui relie χ à un sous-espace $[\boldsymbol{\varphi}']$ de dimension q représenté par la matrice de rang maximale $\boldsymbol{\varphi}' \in \mathcal{ST}(q, N_x)$ est donnée par

$$\boldsymbol{\varphi}' = \boldsymbol{\varphi} \mathbf{V} \cos(\boldsymbol{\Sigma}) \mathbf{Q}^T + \mathbf{U} \sin(\boldsymbol{\Sigma}) \mathbf{Q}^T, \quad \forall \mathbf{Q} \in \mathcal{O}(q) \quad (3.5)$$

f) Application logarithme géodésique $\text{Log}_{[\varphi]}$

Maintenant que l'application exponentielle est définie, il est important pour les applications pratiques de définir son inverse. L'application inverse est appelée application logarithmique et associe un voisinage $\mathcal{V}_{[\varphi]}$ de $[\varphi] \in \mathcal{G}(q, N_x)$ à l'espace tangent $\mathcal{T}_{[\varphi]}\mathcal{G}(q, N_x)$ [102][116].

Dans le cas de la variété de Grassmann, il n'existe pas d'expression explicite pour l'application $\text{Log}_{\mathcal{G}}$. Toutefois, il est possible d'évaluer l'opération $\text{Log}_{[\varphi]}$ par la méthode décrite ci-après [101][117].

Soient φ, φ' un couple de matrices dans la variété de Stiefel compacte représentative des deux classes d'équivalences $[\varphi], [\varphi'] \in \mathcal{G}(q, N_x)$. Nous cherchons à trouver une matrice H représentative de $\chi \in \mathcal{T}_{[\varphi]}\mathcal{G}(q, N_x)$ vérifiant

$$\text{Exp}_{[\varphi]}(\chi) = [\varphi']$$

Écrivons H tel que $Z\Theta W^T = H$ est sa décomposition en valeurs singulières. En utilisant l'expression de l'application exponentielle (3.5), φ' peut être exprimée comme suit

$$\varphi' = \varphi W \cos(\Theta) Q^T + Z \sin(\Theta) Q^T$$

Maintenant, en tenant compte de la condition $\varphi^T H = 0$, car H appartient à l'espace tangent, on peut écrire

$$W \cos(\Theta) Q^T = \varphi^T \varphi' \quad \text{et} \quad Z \sin(\Theta) Q^T = \varphi' - \varphi \varphi^T \varphi'$$

d'où

$$Z \sin(\Theta) Q^T (W \cos(\Theta) Q^T)^{-1} = (\varphi' - \varphi \varphi^T \varphi') (\varphi^T \varphi')^{-1} \quad (3.6)$$

La matrice W étant une matrice orthogonale alors $W^{-1} = W^T$, l'équation (3.6) devient donc

$$Z \tan(\Theta) W^T = (I_{N_x} - \varphi \varphi^T) \varphi' (\varphi^T \varphi')^{-1}$$

En appliquant la SVD, le terme du second membre s'écrit

$$(I_{N_x} - \varphi \varphi^T) \varphi' (\varphi^T \varphi')^{-1} = U \Sigma V^T$$

Ainsi, par identification, Z, W et Θ sont tels que $Z = U, W = V$ et $\Theta = \arctan(\Sigma)$. Le calcul pratique de l'application logarithmique s'effectue donc de la façon suivante :

Soient φ et φ' deux matrices orthogonales dont les colonnes engendrent $[\varphi]$ et $[\varphi'] \in \mathcal{V}_{[\varphi]}$ respectivement. L'image de $[\varphi']$ par l'application logarithmique, $\chi = \text{Log}_{[\varphi]}([\varphi']) \in \mathcal{T}_{[\varphi]}\mathcal{G}(q, N_x)$, est représentée par la matrice H donnée par

$$H = U \arctan(\Sigma) V^T \quad (3.7)$$

où U, V et Σ sont les matrices associées à la décomposition en valeurs singulières suivante :

$$(I - \varphi \varphi^T) \varphi' (\varphi^T \varphi')^{-1} = U \Sigma V^T$$

3.3.2 Interpolation des bases POD

On considère un échantillonnage $\{\boldsymbol{\varphi}_1, \dots, \boldsymbol{\varphi}_{N_b}\}$ de bases POD de dimension q construites pour différents paramètres $\mathcal{P}_{N_b} = \{\theta_1, \dots, \theta_{N_b}\}$. On souhaite maintenant interpoler ces bases réduites de façon à obtenir la base pour le paramètre $\tilde{\theta} \notin \mathcal{P}_{N_b}$. Comme expliqué brièvement auparavant, les bases POD peuvent être vues comme des représentants de points appartenant à la variété de Grassmann $\mathcal{G}(q, N_x)$. La variété de Grassmann n'est pas un espace vectoriel, par conséquent, l'utilisation directe des méthodes d'interpolation usuelles sur les bases réduites s'avère souvent inefficace. Toutefois, une propriété intéressante concernant la variété de Grassmann, est que l'espace tangent $\mathcal{T}_{[\boldsymbol{\varphi}]}\mathcal{G}(q, N_x)$ en un point $[\boldsymbol{\varphi}]$ quelconque est un espace vectoriel, dans lequel l'utilisation des méthodes traditionnelles d'interpolation est légitime. La méthode consiste donc tout d'abord à choisir un point⁴ de référence $[\boldsymbol{\varphi}_{\theta_0}]$ qui sera l'origine de l'espace tangent à la variété $\mathcal{G}(q, N_x)$ en ce point. Ensuite, par l'application logarithme géodésique, on cherche les vitesses initiales χ_i des géodésiques entre le point $[\boldsymbol{\varphi}_{\theta_0}]$ et les autres points $[\boldsymbol{\varphi}_i]$ de l'échantillonnage. L'étape suivante consiste à calculer par une méthode d'interpolation standard la vitesse initiale $\tilde{\chi}$ correspondante au paramètre cible $\tilde{\theta}$. Finalement, le sous espace approximé $[\boldsymbol{\varphi}_{\tilde{\theta}}]$ est obtenu en appliquant l'application exponentielle géodésique. Cette technique est illustrée sur la figure 3.1.

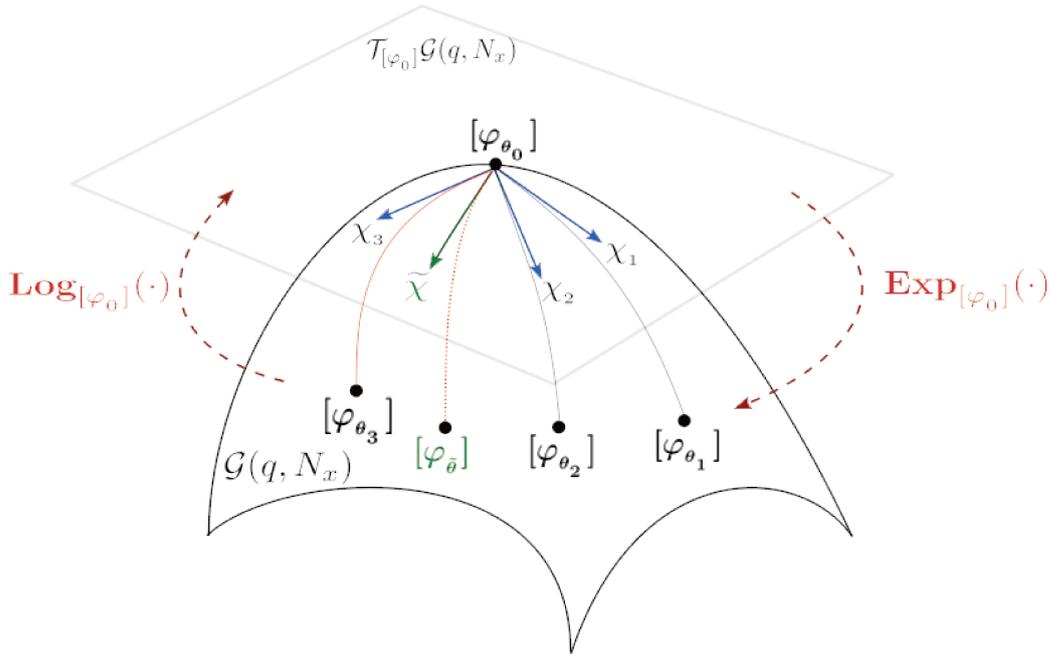


FIGURE 3.1 – Illustration de l'interpolation des bases réduites. Étape 1 : calculer les vitesses initiales $\chi_k \in \mathcal{T}_{[\boldsymbol{\varphi}_{\theta_0}]} \mathcal{G}(q, N_x)$ images par l'application $\text{Log}_{[\boldsymbol{\varphi}_{\theta_0}]}$ des sous espaces $[\boldsymbol{\varphi}_{\theta_k}] \in \mathcal{G}(q, N_x)$; étape 2 : se placer dans l'espace tangent $\mathcal{T}_{[\boldsymbol{\varphi}_{\theta_0}]} \mathcal{G}(q, N_x)$ et interpoler les χ_k afin d'obtenir la vitesse initiale approximée $\tilde{\chi}$; étape 3 : déterminer le sous espace interpolé $[\boldsymbol{\varphi}_{\tilde{\theta}}]$ image par l'application $\text{Exp}_{[\boldsymbol{\varphi}_{\theta_0}]}$ de $\tilde{\chi} \in \mathcal{T}_{[\boldsymbol{\varphi}_{\theta_0}]} \mathcal{G}(q, N_x)$.

4. $[\boldsymbol{\varphi}]$ désigne dans ce paragraphe le sous espace vectoriel de dimension q engendré par a base POD $\boldsymbol{\varphi}_{\theta_j}$, qui elle a été construite pour un paramètre θ_j .

Afin de simplifier la compréhension de la méthode d'interpolation de bases, le cas particulier ($N_b = 2$) de deux bases $\boldsymbol{\varphi}_1$ et $\boldsymbol{\varphi}_2$ correspondantes à deux paramètres scalaires θ_1 et θ_2 sera présenté dans le prochain paragraphe. Ce cas particulier est appelé dans la littérature Subspace Angle Interpolation method (SAIM).

a) Cas particulier de deux bases (SAIM)

Soient $\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2 \in \mathbb{R}^{N_x \times q}$ deux bases POD construites en deux paramètres scalaires distincts θ_1 et θ_2 . On souhaite calculer la base $\tilde{\boldsymbol{\varphi}}$ correspondante à un paramètre $\tilde{\theta}$. Les deux bases $\boldsymbol{\varphi}_1$ et $\boldsymbol{\varphi}_2$ sont des représentants de deux sous espaces vectoriels $[\boldsymbol{\varphi}_1]$ et $[\boldsymbol{\varphi}_2]$ de $\mathcal{G}(q, N_x)$. Les étapes d'adaptation de $[\boldsymbol{\varphi}_1]$ et $[\boldsymbol{\varphi}_2]$ pour un nouveau paramètre $\tilde{\theta}$ différent de θ_1 et θ_2 , consiste dans un premier lieu à choisir un point de référence dans $\mathcal{G}(q, N_x)$ (ici ça sera $[\boldsymbol{\varphi}_1]$ ou bien $[\boldsymbol{\varphi}_2]$). L'étape suivante repose sur l'utilisation de l'application logarithmique afin de pouvoir déterminer la vitesse initiale de la géodésique $\mathcal{Y}(t)$ qui les relie.

Supposons ici que $[\boldsymbol{\varphi}_1]$ est choisi comme point de référence, alors la géodésique dans $\mathcal{G}(q, N_x)$ satisfait $\mathcal{Y}(0) = [\boldsymbol{\varphi}_1]$ et $\mathcal{Y}(1) = [\boldsymbol{\varphi}_2]$. On note $\mathcal{T}_{[\boldsymbol{\varphi}_1]}$ l'espace tangent à $\mathcal{G}(q, N_x)$ au point $[\boldsymbol{\varphi}_1]$ tel que $\dot{\mathcal{Y}}(t) \in \mathcal{T}_{[\boldsymbol{\varphi}_1]}$. Par définition de l'application logarithmique, on a

$$\dot{\mathcal{Y}}(0) = \text{Log}_{[\boldsymbol{\varphi}_1]}[\boldsymbol{\varphi}_2]$$

D'après (3.7), la matrice représentante de $\dot{\mathcal{Y}}(0) \in \mathcal{T}_{[\boldsymbol{\varphi}_1]}$ peut être écrite comme suit

$$\mathbf{H} = U \arctan(\Sigma) V^T$$

avec U , V et Σ sont les matrices de la décomposition en valeurs singulières suivante

$$(I - \boldsymbol{\varphi}_1 \boldsymbol{\varphi}_1^T) \boldsymbol{\varphi}_2 (\boldsymbol{\varphi}_1^T \boldsymbol{\varphi}_2)^{-1} = U \Sigma V^T$$

Maintenant qu'on s'est placé sur l'espace tangent, la vitesse initiale correspondante à $\tilde{\theta}$ est approximée en utilisant une méthode d'interpolation standard. La géodésique $\tilde{\mathcal{Y}}(t)$ dans $\mathcal{G}(q, N_x)$ entre $[\boldsymbol{\varphi}_1] \in \mathcal{G}(q, N_x)$ et $[\tilde{\boldsymbol{\varphi}}] \in \mathcal{G}(q, N_x)$ satisfait $\tilde{\mathcal{Y}}(0) = [\boldsymbol{\varphi}_1]$ et $\tilde{\mathcal{Y}}(1) = [\tilde{\boldsymbol{\varphi}}]$. Étant donné que $\mathcal{T}_{[\boldsymbol{\varphi}_1]}$ est un espace plat et que $[\boldsymbol{\varphi}_1]$ est son origine (c'est à dire : $\text{Log}_{[\boldsymbol{\varphi}_1]}([\boldsymbol{\varphi}_1]) = 0$), la condition initiale sur la dérivée $\dot{\tilde{\mathcal{Y}}}(t) \in \mathcal{T}_{[\boldsymbol{\varphi}_1]}$ peut être interpolée en utilisant les deux points $\text{Log}_{[\boldsymbol{\varphi}_1]}([\boldsymbol{\varphi}_1]) = 0$ et $\text{Log}_{[\boldsymbol{\varphi}_1]}([\boldsymbol{\varphi}_2]) = \dot{\mathcal{Y}}(0)$ comme suit

$$\dot{\tilde{\mathcal{Y}}}(0) = \tilde{r} \dot{\mathcal{Y}}(0)$$

où

$$\tilde{r} = \frac{\tilde{\theta} - \theta_1}{\theta_2 - \theta_1}$$

Ainsi, la matrice représentative de $\dot{\tilde{\mathcal{Y}}}(0)$ est calculée de la façon suivante

$$\tilde{\mathbf{H}} = U \tilde{r} \arctan(\Sigma) V^T$$

puis, ayant trouvé l'approximation de la vitesse initiale de la cible, on utilise l'application exponentielle pour trouver une approximation d'une base représentante du sous espace

cible.

En utilisant le résultat (3.4), la géodésique sur $\mathcal{G}(q, N_x)$ ayant les conditions initiales $\mathcal{Y}(0)$ et $\tilde{\mathcal{Y}}(0)$ est donnée par

$$\tilde{\mathcal{Y}}(t) = \text{Vect} \{ \boldsymbol{\varphi}_1 V \cos(t\tilde{r} \arctan(\Sigma)) + U \sin(t\tilde{r} \arctan(\Sigma)) \} \quad 0 \leq t \leq 1$$

d'où

$$\tilde{\mathcal{Y}}(1) = \text{Vect} \{ \boldsymbol{\varphi}_1 V \cos(\tilde{r} \arctan(\Sigma)) + U \sin(\tilde{r} \arctan(\Sigma)) \}$$

Finalement, l'expression de la base adaptée en utilisant deux bases POD $\boldsymbol{\varphi}_1$ et $\boldsymbol{\varphi}_2$ construites pour deux paramètres scalaires différents θ_1 et θ_2 est

$$\tilde{\boldsymbol{\varphi}} = \boldsymbol{\varphi}_1 V \cos \left[\left(\frac{\tilde{\theta} - \theta_1}{\theta_2 - \theta_1} \right) \arctan(\Sigma) \right] + U \sin \left[\left(\frac{\tilde{\theta} - \theta_1}{\theta_2 - \theta_1} \right) \arctan(\Sigma) \right]$$

b) Cas général (ITSGM)

En partant du même principe que pour l'interpolation de deux bases (SAIM), l'approche d'interpolation est étendue au cas général d'un nombre arbitraire N_b de bases POD. Pour commencer, un point $[\boldsymbol{\varphi}_{\theta_0}]$ de la variété de Grassmann $\mathcal{G}(q, N_x)$ est choisi pour être le point de référence de l'espace tangent. Ensuite, les géodésiques entre le point de référence $[\boldsymbol{\varphi}_{\theta_0}]$ et les autres points $[\boldsymbol{\varphi}_{\theta_i}] \in \mathcal{G}(q, N_x)$ sont calculées. Bien évidemment, toutes ces géodésiques partagent le point $[\boldsymbol{\varphi}_{\theta_0}]$ comme position initiale, seule la vitesse initiale est différente. Les dérivées initiales, qui appartiennent toutes à l'espace tangent $\mathcal{T}_{[\boldsymbol{\varphi}_{\theta_0}]} \mathcal{G}(q, N_x)$, sont finalement interpolées afin d'approximer la dérivée initiale qui détermine la géodésique entre le point de référence et le point cible. Finalement, en appliquant l'application exponentielle, une base représentative du sous espace interpolé $[\boldsymbol{\varphi}_{\tilde{\theta}}]$ peut être calculée. Les étapes de la méthode ITSGM [98] sont détaillées dans l'algorithme 10.

Algorithme 10 : Algorithme ITSGM

- 1 Choisir une base de référence $\boldsymbol{\varphi}_{i_0}$ considérée comme représentant du sous espace $[\boldsymbol{\varphi}]_{i_0} \in \mathcal{G}(q, N_x)$
- 2 Pour tout $i = 1, \dots, N_b$ et $i \neq i_0$, déterminer un représentant \mathbf{H}_i de $\chi_i \in \mathcal{T}_{[\boldsymbol{\varphi}_{\theta_0}]} \mathcal{G}(q, N_x)$, image par l'application logarithmique $\text{Log}_{[\boldsymbol{\varphi}_{\theta_0}]}$ de $[\boldsymbol{\varphi}_{\theta_i}]$

$$\mathbf{H}_i = U_i \arctan(\Sigma_i) V_i^T \quad \text{avec} \quad (I - \boldsymbol{\varphi}_{\theta_0} \boldsymbol{\varphi}_{\theta_0}^T) \boldsymbol{\varphi}_{\theta_i} (\boldsymbol{\varphi}_{\theta_0}^T \boldsymbol{\varphi}_{\theta_i})^{-1} = U_i \Sigma_i V_i^T \quad (3.8)$$

- 3 Trouver $\tilde{\mathbf{H}}$ représentant de $\tilde{\chi} \in \mathcal{T}_{[\boldsymbol{\varphi}_{\theta_0}]} \mathcal{G}(q, N_x)$ en interpolant l'ensemble des \mathbf{H}_i
- 4 En utilisant la valeur approximée de $\tilde{\mathbf{H}}$ et par l'application exponentielle $\text{Exp}_{[\boldsymbol{\varphi}_{\theta_0}]}$, calculer $\boldsymbol{\varphi}_{\tilde{\theta}}$ représentant du sous espace $[\boldsymbol{\varphi}_{\tilde{\theta}}] \in \mathcal{G}(q, N_x)$ cibles

$$\boldsymbol{\varphi}_{\tilde{\theta}} = \boldsymbol{\varphi}_{\theta_0} \tilde{V} \cos(\tilde{\Sigma}) + \tilde{U} \sin(\tilde{\Sigma}) \quad \text{avec} \quad \tilde{\mathbf{H}} = \tilde{U} \tilde{\Sigma} \tilde{V}^T \quad (3.9)$$

Remarque : Une fois que la base spatiale pour le paramètre cible $\tilde{\theta}$ est calculée par interpolation, le système dynamique d'équations différentielles ordinaire (ROM) décrivant

l'évolution temporelle est obtenu en effectuant la projection de Galerkin des équations du modèle complet sur la base interpolée. Le modèle réduit ainsi construit est de faible dimension et permet un accès rapide à la dynamique temporelle du problème.

3.3.3 Coût asymptotique de la méthode ITSGM

Rappelons ici que N_x signifie le nombre de degrés de libertés du problème complet, N_s le nombre d'instants temporels (c'est à dire le nombre de snapshots), N_b le nombre de bases POD de l'échantillonnage (supposées de même dimension q) et d_p la dimension des paramètres d'interpolation. Dans le but de simplifier, le coût de calcul associé à la méthode ITSGM sera présenté dans le cas où la dimension du paramètre d'interpolation est $d_p = 1$. Dans ce qui suit, le coût pour chaque étape de l'algorithme d'interpolation est détaillé :

- **Étape 2** : le calcul par l'application logarithmique nécessite les opérations suivantes :
 - $2(N_b - 1)$ multiplications matricielles, soit $2(N_b - 1) \times \mathcal{O}(N_x q^2)$.
 - $(N_b - 1)$ inversions de matrices, soit $(N_b - 1) \times \mathcal{O}(q^3)$.
 - $(N_b - 1)$ décompositions SVD tronquées d'ordre q effectuées pour un coût asymptotique égal à $(N_b - 1) \times \mathcal{O}(N_x q^2)$.
- **Étape 3** : l'interpolation (par exemple de Lagrange) dans l'espace tangent est implémentée pour un coût asymptotique égale à $N_b \times \mathcal{O}(N_x q)$.
- **Étape 4** : le calcul par l'application exponentielle nécessite les opérations suivantes :
 - une décomposition SVD tronquée d'ordre q effectuée à un coût asymptotique de $\mathcal{O}(N_x q^2)$.
 - des multiplications matricielles, soit un coût égal à $\mathcal{O}(N_x q^2) + 2q\mathcal{O}(N_x)$.

Comme q est très petit devant N_x ($q \ll N_x$), les termes en $\mathcal{O}(q^3)$ sont négligés. De même, en supposant que q est de l'ordre de la dizaine, les termes en $\mathcal{O}(N_x q)$ sont négligés devant ceux en $\mathcal{O}(N_x q^2)$. Le coût asymptotique estimé pour les calculs effectués dans la méthode ITSGM s'élève donc à $(3N_b - 1) \times \mathcal{O}(N_x q^2)$. Cette méthode est par conséquent efficace en terme de temps de calcul, car sa complexité est fonction linéaire du nombre de degrés de libertés du problème complet.

3.4 ADAPTATION DE BASES RÉDUITES PAR PGD

La méthode PGD (Proper Generalized Decomposition) est une méthodes d'approximation basées sur des produits tensoriels. Elle consiste à approximer une solution $y \in V$

d'un problème $\mathcal{N}(y) = 0$ par la représentation séparée suivante

$$y(\theta_1, \theta_2, \dots, \theta_{N_p}) \approx y_q = \sum_{k=1}^q x_k^{(1)}(\theta_1) \otimes \dots \otimes x_k^{(N_p)}(\theta_{N_p}), \quad x_k^j \in V_j \quad (3.10)$$

où V est identifié à un espace vectoriel généré par le produit tensoriel de q espaces vectoriels V_j . Les variables θ_j peuvent représenter le temps t , la variable d'espace x , un paramètre de contrôle, etc. La PGD est une méthode itérative qui consiste à chaque itération, à ajouter à la forme séparée (3.10) un nouveau jeu de fonctions $(x_{q+1}^{(1)}(\theta_1), \dots, x_{q+1}^{(N_p)}(\theta_{N_p}))$ calculé en résolvant les problèmes obtenus en injectant la forme séparée d'ordre $q + 1$ dans les équations régissant le problème d'origine et en projetant ces équations sur chaque fonction $x_{q+1}^{(k)}$. En supposant que le nombre de degrés de liberté noté N_θ pour discrétiser chacune des variables θ_j , est le même, le nombre total d'inconnus avec la PGD est $N_\theta \times q \times N_p$ au lieu de $N_\theta^{N_p}$ avec une approche traditionnelle. Ainsi, si il faut q produit de fonctions pour obtenir une représentation correcte de la solution y , nous aurons à chaque itération PGD à résoudre un problème dont le nombre de degrés de liberté est $N_p \times N_\theta$. Comme q est généralement petit (quelques dizaines voire une centaine), cette démarche est plus rapide que les méthodes classiques.

La méthode PGD a initialement été proposée par Ladevèze et al. [118] [119][120] pour réduire le temps de simulation des problèmes de la mécanique des structures non linéaire. Dans ce cas, la méthode est connue sous le nom de "radial time-space approximation" et consiste à chercher la solution sous forme séparée en espace et temps. La PGD a également été utilisée par Nouy et al. [121] [122] pour la résolution des problèmes stochastiques. Dans ce contexte, l'approche est appelée "Generalized Decomposition". Ammar et al. ont étendu la PGD à la résolution de problèmes multidimensionnels dans le cadre de la théorie cinétique des fluides complexes afin de résoudre les modèles de systèmes polymériques [123] [124]. Depuis ces travaux, la PGD a été appliqué pour résoudre rapidement et de manière efficace de nombreux problèmes de transfert de chaleur [125], de mécanique des fluides [126] [127] [128] [129] [130], de chimie quantique [131], de mécanique des solides [132] [133] [134], etc. Dans le contexte du contrôle et de l'optimisation, la PGD a été déjà utilisée par Chinesta et al. [135] [136]. Dans ces travaux, la séparation des variables est étendue en plus de l'espace et le temps aux paramètres de contrôle. Cette forme séparée construite en dehors de la boucle de contrôle (phase offline) est censée représenter correctement la variation des paramètres de contrôle à l'intérieur de l'algorithme d'optimisation. Cet abaque numérique est utilisé par la suite dans la boucle d'optimisation ou de contrôle pour calculer rapidement un optimum. Dans cette thèse, l'approche considérée est différente, la forme séparée considérée est spatio-temporelle, et la PGD intervient dans l'algorithme de contrôle comme un correcteur de la base spatiale par rapport au changement des paramètres de contrôle.

3.4.1 PGD basée sur le critère d'orthogonalité de Galerkin

Considérons l'application $\mathcal{N} : V \times Z \rightarrow W$ décrivant une equation au dérivées partielles où $V = L^2(0, T; H^1(\Omega))$ et $W = L^2(0, T; L^2(\Omega))$. V et W sont identifiés respectivement à $\mathcal{T} \otimes \mathcal{V}$ et $\mathcal{T} \otimes \mathcal{H}$, où $\mathcal{T} = L^2(0, T)$, $\mathcal{V} = H^1(\Omega)$ et $\mathcal{H} = L^2(\Omega)$.

Soit $y(\theta)$ la solution de $\mathcal{N}(y, \theta) = 0$, on suppose que nous disposons d'une décomposition spatio-temporelle y_q d'ordre q telle que

$$y(t, x, \theta) \approx y_q(t, x, \theta) = \sum_{k=1}^q \alpha^k(t, \theta) \varphi^k(x, \theta) \quad (3.11)$$

L'approche PGD considérée ici vise à enrichir cette décomposition en ajoutant a l'approximation précédente le produit de fonctions $\tilde{\alpha} \tilde{\varphi}$ afin d'approximer la solution associée à un nouveau paramètre cible $\tilde{\theta}$, soit :

$$y(t, x, \tilde{\theta}) \approx y_q(t, x, \tilde{\theta}) = \sum_{k=1}^q \alpha^k(t, \theta) \varphi^k(x, \theta) + \tilde{\alpha} \tilde{\varphi}.$$

Les nouvelles fonctions $(\tilde{\alpha}, \tilde{\varphi}) \in \mathcal{T} \times \mathcal{V}$ sont définies de manière optimale lorsqu'elles satisfont le critère de Galerkin double suivant

$$\langle \mathcal{N}(y_q + \tilde{\alpha} \tilde{\varphi}, \theta), \tilde{\alpha} \varphi^* + \alpha^* \tilde{\varphi} \rangle_W = 0, \quad \forall (\alpha^* \varphi^*) \in \mathcal{T} \times \mathcal{V} \quad (3.12)$$

Soit une fonction temporelle $\tilde{\alpha} \in \mathcal{T}$ connue et fixée, on définit l'application $f_x^{(q)} : \mathcal{T} \rightarrow \mathcal{V}$ qui à $\tilde{\alpha}$ associe une fonction spatiale $\tilde{\varphi}$ par

$$\langle \mathcal{N}(y_q + \tilde{\alpha} \tilde{\varphi}, \tilde{\alpha} \varphi^*) \rangle_V = 0, \quad \forall \varphi^* \in \mathcal{V} \quad (3.13)$$

De la même manière, pour une fonction spatiale $\tilde{\varphi} \in \mathcal{V}$ connue et fixée, l'application $f_t^{(q)} : \mathcal{V} \rightarrow \mathcal{T}$ qui à $\tilde{\varphi}$ associe une fonction temporelle $\tilde{\alpha}$ peut être définie par

$$\langle \mathcal{N}(y_q + \tilde{\alpha} \tilde{\varphi}, \alpha^* \tilde{\varphi}) \rangle_V = 0, \quad \forall \alpha^* \in \mathcal{T} \quad (3.14)$$

Le nouveau couple $(\tilde{\alpha}, \tilde{\varphi}) \in \mathcal{T} \times \mathcal{V}$ satisfait les equations (3.13) et (3.14) si et seulement si

$$\tilde{\varphi} = f_x^{(q)}(\tilde{\alpha}) \quad \text{et} \quad \tilde{\alpha} = f_t^{(q)}(\tilde{\varphi}) \quad (3.15)$$

On définit $g_x^{(q)} : \mathcal{V} \rightarrow \mathcal{V}$ l'application composée définie par

$$g_x^{(q)} \stackrel{\text{def}}{=} f_x^{(q)} \circ f_t^{(q)} \quad (3.16)$$

et $g_t^{(q)} : \mathcal{T} \rightarrow \mathcal{T}$ l'application composée définie par

$$g_t^{(q)} \stackrel{\text{def}}{=} f_t^{(q)} \circ f_x^{(q)} \quad (3.17)$$

En utilisant l'application (3.16), pour $\tilde{\alpha} = f_t^{(q)}(\tilde{\varphi})$ le problème (3.15) est alors équivalent au problème du point fixe

$$\tilde{\varphi} = g_x^{(q)}(\tilde{\varphi}) \quad (3.18)$$

De même pour $\tilde{\varphi} = f_x^{(q)}(\tilde{\alpha})$, en utilisant cette fois ci l'application (3.4.1), le problème (3.15) peut être amené au problème de point fixe suivant

$$\tilde{\alpha} = g_t^{(q)}(\tilde{\alpha}) \quad (3.19)$$

Par conséquent, pour une décomposition d'ordre q de y , la nouvelle fonction optimale $\tilde{\alpha} \in \mathcal{T}$ (resp. $\tilde{\varphi} \in \mathcal{V}$) peut être définie comme le point fixe du problème (3.19) (resp. (3.18)). Les étapes pour calculer un couple optimal $(\tilde{\alpha}, \tilde{\varphi}) \in \mathcal{T} \times \mathcal{V}$ sont résumées dans l'algorithme de type point fixe (Algorithme 11) appliqué à l'opérateur $g_t^{(q)}$.

Algorithme 11 : PGD (Algorithme du point fixe)

- 1 Actualiser les coefficients temporels $\alpha_q = \{\alpha^1, \dots, \alpha^q\}$ en résolvant le modèle réduit construit avec la base associée au paramètre θ .

$$\left\langle \mathcal{N}\left(\sum_{j=1}^q \alpha^j \varphi^j, \tilde{\theta}\right), \varphi^i \right\rangle_{L^2(\Omega)} = 0, \quad i = 1, \dots, q$$

- 2 $m = 0$ et M_{max} choisi par l'utilisateur

- 3 **while** $m \leq M_{max}$ **or** $\|y_{q+1} - y_q\| < \varepsilon \|y_{q+1}\|$ **do**

- 4 $\tilde{\alpha}^0$ arbitraire;

- 5 **for** $k \leftarrow 0$ **to** k_{max} **do**

- 6 calculer $\tilde{\varphi}^{(k+1)} = f_x^{(q)}(\tilde{\alpha}^{(k)})$;

- 7 calculer $\tilde{\alpha}^{(k+1)} = f_t^{(q)}(\tilde{\varphi}^{(k+1)})$;

- 8 Vérifier la convergence $|\sigma^{k+1} - \sigma^k| < \epsilon_u |\sigma^{k+1}|$ où $\sigma^k = \|\tilde{\alpha}^k\|$;

- 9 **end**

- 10 Enrichissement : $\varphi^{q+1} = \tilde{\varphi}^{(k+1)}$ et $\boldsymbol{\varphi} = \{\varphi^1, \dots, \varphi^{q+1}\}$

- 11 Update : Trouver $\alpha_{q+1} = \{\alpha^1, \dots, \alpha^{q+1}\}$ solutions de l'équation du modèle réduit

$$\left\langle \mathcal{N}\left(\sum_{j=1}^{q+1} \alpha^j \varphi^j, \tilde{\theta}\right), \varphi^i \right\rangle_{L^2(\Omega)} = 0, \quad i = 1, \dots, q+1$$

- 12 Reconstruction de la solution

$$y_{q+1}(t, x, \tilde{\theta}) = \sum_{k=1}^q \alpha^k(t, \tilde{\theta}) \varphi^k(x)$$

- 13 $q \leftarrow q + 1$

- 14 $m \leftarrow m + 1$

- 15 **end**
-

3.4.2 Mise en forme de la base PGD

Après un enrichissement par la méthode PGD, l'élément rajouté à la base spatiale $\boldsymbol{\varphi}$ peut apporter une contribution faible ou présenter des dépendances linéaires avec la base initiale. Ainsi, l'ensemble des fonctions de base $\boldsymbol{\varphi} = \{\varphi^1, \dots, \varphi^q, \tilde{\varphi}\}$ ne formera pas forcément une base de rang minimal. De plus, en effectuant de multiples enrichissements, la taille de cet ensemble devient grande entraînant un coût numérique supplémentaire dans la résolution du modèle réduit obtenu par projection de Galerkin. Afin d'avoir une base de rang minimal tout en gardant le même ordre de précision de l'approximation, il est primordial de réorganiser les modes spatiaux de manière à supprimer les relations de liaisons entre le dernier élément ajouté et les modes existants. Pour ce faire, le procédé de Gram-Schmidt est tout d'abord appliqué sur la famille $\boldsymbol{\varphi}$ de manière à ce que le nouvel ensemble de fonctions de base $\boldsymbol{\phi} = \{\phi^1, \dots, \phi^{q+1}\}$ vérifie

$$\int_{\Omega} \phi^i \phi^j dx = \delta_{ij}, \quad \forall j = 1, \dots, q+1$$

Par la suite, les coefficients temporels sont actualisés en résolvant l'équation du modèle d'ordre réduit construit en utilisant la base $\boldsymbol{\phi}$. Les nouveaux coefficients temporels $\boldsymbol{\zeta} = \{\zeta^1, \dots, \zeta^{q+1}\}$ sont ainsi obtenus. L'ensemble des solutions en des instants discrets $\{t_1, t_2, \dots, t_{N_s}\}$ est reconstruit comme suit

$$\mathbf{X} = \{y_{q+1}(t_1), \dots, y_{q+1}(t_{N_s})\} = \left\{ \sum_{j=1}^{q+1} \zeta^j(t_1) \phi^j, \dots, \sum_{j=1}^{q+1} \zeta^j(t_q) \phi^j \right\}$$

En utilisant la méthode POD par corrélation, le problème aux valeurs propres à résoudre s'écrit

$$\mathbf{C}\mathbf{V} = \lambda\mathbf{V} \quad \text{où} \quad \mathbf{C}_{ij} = \int_{\Omega} y(t_i) y(t_j) dx = \sum_{k=1}^{q+1} \zeta^k(t_i) \zeta^k(t_j)$$

La base améliorée de rang minimal $\boldsymbol{\psi}$ est recherchée telle que

$$\psi^i = \sum_{l=1}^{N_s} \sum_{j=1}^{q+1} \phi^j \zeta^j(t_l) \mathbf{V}_{li}, \quad i = 1, \dots, q+1 \quad (3.20)$$

En posant $\mathbf{K}_{ji} = \sum_{l=1}^{N_s} \zeta^j(t_l) \mathbf{V}_{li}$, on est donc amené à résoudre le problème aux valeurs propres suivant

$$\mathbf{C}^{(red)} \mathbf{K} = \lambda \mathbf{K} \quad (3.21)$$

où $\mathbf{C}^{(red)} \in \mathbb{R}^{(q+1) \times (q+1)}$ et

$$\mathbf{C}_{ij}^{(red)} = \sum_{l=1}^{N_s} \zeta^i(t_l) \zeta^j(t_l)$$

La résolution du problème (3.21) conduit à $q+1$ valeurs propres dont on ne garde que celles qui ont une contribution significative. Ainsi, on ne garde que les r premières valeurs propres telles que

$$\sum_{i=1}^r \lambda_i < \eta \sum_{i=1}^{q+1} \lambda_i$$

où η est le critère de sélection des modes propres fixé par l'utilisateur. La base de rang minimal est finalement reconstruite par la relation

$$\psi^i = \sum_{j=1}^{q+1} \phi^j \mathbf{K}_{ji}, \quad i = 1, \dots, r$$

Les étapes de réduction de la taille de la base PGD sont détaillées dans l'algorithme 12. Cet algorithme remplacera l'étape 9 de l'algorithme PGD (Algo. 11).

Algorithme 12 : Réduction de taille de la base PGD

- 1 Appliquer le procédé de Gram-Schmidt à la famille $\{\phi^1, \dots, \phi^q, \tilde{\phi}\}$ et obtenir la famille de vecteurs de base $\{\phi^1, \dots, \phi^{q+1}\}$ telle que

$$\int_{\Omega} \phi^i \phi^j dx = \delta_{ij}, \quad \forall j = 1, \dots, q+1$$

- 2 Actualiser l'ensemble des coefficients temporels $\{\alpha^1, \dots, \alpha^q, \tilde{\alpha}\}$ en résolvant l'équation du modèle d'ordre réduit construit en utilisant la base orthonormalisée ϕ , et obtenir la famille de coefficients temporels $\{\zeta^1, \dots, \zeta^{q+1}\}$.
- 3 Construire la matrice de corrélation réduite $\mathbf{C}^{(red)}$ telle que

$$\mathbf{C}_{ij}^{(red)} = \sum_{l=1}^{N_s} \zeta^i(t_l) \zeta^j(t_l)$$

- 4 Déterminer le rang minimal r , qui pour η fixé à l'avance par l'utilisateur vérifie

$$\sum_{i=1}^r \lambda_i < \eta \sum_{i=1}^{q+1} \lambda_i$$

- 5 Reconstruire les fonctions de la base de rang r par la relation

$$\psi^i = \sum_{j=1}^{q+1} \phi^j \mathbf{K}_{ji}, \quad i = 1, \dots, r$$

3.5 CONCLUSION DU CHAPITRE

Dans la première partie de ce chapitre, la méthode d'interpolation de bases réduites ITSGM (Interpolation on Tangent Subspace of Grassman Manifold) a été introduite. Contrairement aux méthodes d'interpolation usuelles (Lagrange, RBF, spline ...) cette méthode fondée sur des résultats de géométrie différentielle permet de donner un sens au problème d'interpolation des bases POD en passant par le sous espace tangent en un point de la variété de Grassmann. Une fois que la base interpolée est calculée, la dynamique temporelle associée au nouveau paramètre est obtenue en résolvant le système d'équations différentielles de petite taille, issu de la projection de Galerkin du modèle complet sur cette base. La seconde partie de ce chapitre a été consacrée à la description de l'approche d'adaptation de base fondée sur la méthode PGD (Proper Generalized Decomposition). La robustesse de ces deux approches à l'intérieur d'un algorithme de contrôle optimal sera testée au chapitre 5.

CONSTRUCTION D'UN MODÈLE RÉDUIT NON INTRUSIF

4

SOMMAIRE

4.1	INTRODUCTION	82
4.2	MÉTHODE NIMR	83
4.2.1	Position du problème	83
4.2.2	Description de la méthode	83
4.3	MÉTHODE HNIMR	88
4.4	COÛT ASYMPTOTIQUE DES MÉTHODES NIMR ET HNIMR	90
4.5	APPLICATION SUR LE PROBLÈME DE L'ÉCOULEMENT AUTOUR D'UN CYLINDRE	91
4.5.1	Position du problème	91
4.5.2	Construction des bases réduites d'apprentissage	93
4.5.3	Mise en œuvre des méthodes NIMR et HNIMR	94
4.5.4	Évaluation des performances des méthodes NIMR et HNIMR	97
4.6	CONCLUSION DU CHAPITRE	105

4.1 INTRODUCTION

L'ensemble des méthodes présentées au chapitre précédent sont dites intrusives, car elles nécessitent pour prédire la dynamique temporelle, le recours aux équations du modèle réduit, qui eux ont été construites à partir du modèle complet via la projection de Galerkin. De plus en plus d'auteurs proposent des techniques de réduction de modèle dites non-intrusives, dans le sens où elles ne nécessitent pas d'avoir recours aux équations régissant le problème de base. Ces approches peuvent être très utiles notamment lorsque les snapshots sont issus de données expérimentales. Ainsi, dans le contexte de la réduction de modèle non-intrusive, Xiao et al. [137][138] suggèrent de construire un ensemble d'hyper surfaces pour la représentation de la dynamique temporelle, puis d'utiliser la méthode RBF sur des grilles de Smolyak afin de les interpoler. La base spatiale réduite, quant à elle, est obtenue en appliquant la POD sur l'ensemble des snapshots également interpolés par RBF. L'approche a été utilisée pour étudier l'écoulement 2D autour d'un cylindre lorsque le nombre de Reynolds varie. Toujours sur le même cas d'étude, Shinde et al. [139] proposent d'approximer les fonctions de bases spatiales et temporelles par une interpolation linéaire. Cette approche a également été appliquée par Joyner [140] pour la détection de dommages par courant de Foucault (eddy current damage detection). Pour ce cas, il a été observé que les résultats obtenus par la méthode POD/interpolation sont comparables à ceux obtenus par la méthode POD/Galerkin.

Dans ce manuscrit, et dans le contexte de la réduction de modèle non-intrusive, nous proposons une méthode originale basée sur l'interpolation des bases spatiales et temporelles par la méthode ITSGM. Considérons la décomposition SVD (Singular Value Decomposition) d'un ensemble de matrices $\mathbf{X}_k = \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^T$, où \mathbf{X}_k représente la matrice des snapshots. \mathbf{U}_k et \mathbf{V}_k sont respectivement les bases spatiale et temporelle, et $\mathbf{\Sigma}_k$ la matrice diagonale contenant les valeurs singulières. La méthode proposée notée par la suite NIMR (Non Intrusive Model Reduction) s'effectue alors en deux étapes. La première étape est l'étape d'interpolation, qui consiste à interpoler les valeurs singulières (en utilisant par exemple Lagrange, RBF, Spline ...) et les bases spatiales et temporelles (en utilisant la méthode ITSGM). La deuxième étape est l'étape de classement, qui a pour but de classer selon l'ordre des valeurs singulières interpolées, les modes propres spatiaux et temporels obtenus par la méthode ITSGM. Ce classement s'effectue par la résolution d'un problème de minimisation sous contraintes dont la solution peut analytiquement être déterminée. Une variante de cette méthode, notée sous l'acronyme HNIMR (Hyper Non Intrusive Model Reduction), sera également présentée. Cette variante permet de réduire d'avantage l'ordre du problème d'interpolation par NIMR et par conséquent d'accélérer le temps de calcul.

Ce chapitre commence par une description détaillée des méthodes de réduction de modèle non intrusive NIMR et HNIMR proposées. La robustesse et précision de ces méthodes sera

testée par la suite sur le problème d'écoulement incompressible d'un fluide Newtonien autour d'un cylindre.

4.2 MÉTHODE NIMR

4.2.1 Position du problème

Supposons que pour un ensemble de paramètres $Y_\theta = \{\theta_i \in \mathbb{R}^{d_p}, i = 1, \dots, N_b\}$ nous disposons, pour chaque paramètre θ_i , de l'ensemble des solutions $\mathbf{X}(\theta_i)$ d'un problème physique non linéaire en des instants discrets t_i . En particulier, pour chaque paramètre θ_i , $\mathbf{X}_i = \mathbf{X}(\theta_i) \in \mathbb{R}^{N_x \times N_s}$ désignera la matrice constituée par concaténation des snapshots issus de la simulation numérique ou de l'expérience. La question qui se pose dans le cas où les équations gouvernant ce problème sont numériquement coûteuses ou ne sont pas disponibles est : "Existe-t-il un moyen robuste permettant d'approximer la solution $\tilde{\mathbf{X}}$ décrivant ce même problème pour un paramètre $\tilde{\theta} \notin Y_\theta$ sans avoir recours aux équations du modèle complet ou à l'expérience ?". Pour répondre à cette question, il est possible d'obtenir l'approximation $\tilde{\mathbf{X}}$ en utilisant les méthodes d'interpolation polynomiales usuelles sur les éléments de l'ensemble $\{\mathbf{X}_j\}_{j=1, \dots, N_b}$. Cette approche, appliquée directement, peut s'avérer efficace pour les problèmes à caractère linéaire ou faiblement non linéaire. Toutefois, pour le cas des problèmes non linéaires, elle ne garantit pas en général une bonne approximation de l'interpolée $\tilde{\mathbf{X}}$. Malgré cette difficulté, les méthodes d'interpolation restent attirantes à cause de leur simplicité et de leur rapidité. L'objectif de cette section est donc de contourner cette difficulté en dérivant une méthode d'interpolation basée sur l'interpolation ITSGM présentée au chapitre précédent. Cette méthode est appelée NIMR (Non Intrusive Model Reduction) et sa dérivation est détaillée dans ce qui suit.

4.2.2 Description de la méthode

Pour $i = 1, \dots, N_b$, nous effectuons la décomposition SVD tronquée d'ordre q des matrices de snapshots comme suit

$$\mathbf{X}_i \approx \mathbf{U}_i \Sigma_i \mathbf{V}_i^T$$

où $\mathbf{U}_i \in \mathbb{R}^{N_x \times q}$ et $\mathbf{V}_i \in \mathbb{R}^{N_s \times q}$ sont deux matrices vérifiant

$$\mathbf{U}_i^T \mathbf{U}_i = \mathbf{V}_i^T \mathbf{V}_i = \mathbf{I}_q$$

et $\Sigma_i = \text{diag}(\sigma_{i,1}, \sigma_{i,2}, \dots, \sigma_{i,q})$, est la matrice diagonale contenant les valeurs singulières

$$\sigma_{i,j} = \sqrt{\lambda_j(\mathbf{X}_i^T \mathbf{X}_i)} \geq \sigma_{i,j+1}, \text{ pour } j = 1, \dots, q-1$$

- L'approximation des valeurs singulières $\{\tilde{\sigma}_1, \dots, \tilde{\sigma}_q\} = \text{diag}(\tilde{\Sigma})$ correspondante au nouveau paramètre $\tilde{\theta}$ est assez simple à obtenir. En effet, il suffit juste d'interpoler point par

point l'ensemble des valeurs singulières $\sigma_{i,1}, \dots, \sigma_{i,q}$ connues pour l'ensemble des paramètres dans Y_θ .

- Les matrices colonnes \mathbf{U}_i et \mathbf{V}_i appartiennent à la variété de Stiefel, $\mathbf{U}_i \in \mathcal{ST}(q, N_x)$ et $\mathbf{V}_i \in \mathcal{ST}(q, N_s)$. Par conséquent, leur interpolation point par point n'a pas de sens et ne garantit pas que le résultat soit forcément un élément de la variété de Stiefel. Il est donc plus judicieux de traiter ces éléments comme des représentants de sous espaces vectoriels afin de donner un sens à l'interpolation, et ce en appliquant la méthode ITSGM. L'algorithme ITSGM (Algo. 10) du chapitre précédent est alors appliqué sur l'ensemble des $[\mathbf{U}_i]$ puis sur l'ensemble des $[\mathbf{V}_i]$ et une approximation de deux représentants $\tilde{\mathbf{U}}$ et $\tilde{\mathbf{V}}$ des sous espaces vectoriels $[\tilde{\mathbf{U}}]$ et $[\tilde{\mathbf{V}}]$ est obtenue.

- La matrice $\tilde{\mathbf{X}}$ peut alors être exprimée comme suit

$$\tilde{\mathbf{X}} = \tilde{\mathbf{U}}\mathbf{K}\tilde{\Sigma}\mathbf{Q}^T\tilde{\mathbf{V}}^T$$

où $\mathbf{K}, \mathbf{Q} \in [\mathcal{O}(q)]^2$ sont deux matrices orthogonales qui servent de calibrage. L'introduction de \mathbf{K} et \mathbf{Q} dans l'approximation a une grande importance, et le bon choix de ces deux matrices va assurer la combinaison adéquate des vecteurs colonnes de $\tilde{\mathbf{U}}$ et de $\tilde{\mathbf{V}}$ qui correspond au rangement des valeurs singulières interpolées dans la matrice $\tilde{\Sigma}$.

- La deuxième étape de la méthode NIMR consiste à ajuster le signe des modes de chaque base POD de l'échantillonnage par rapport à une base de référence. Cette base de référence est choisie telle que la distance entre le sous espace vectoriel qu'elle engendre et le sous espace vectoriel interpolé par la méthode ITSGM est minimale. La fonction $dist([\mathbf{Y}], [\mathbf{Z}])$ désigne la métrique mesurant la distance entre deux points $[\mathbf{Y}]$ et $[\mathbf{Z}]$ appartenant à la variété de Grassmann $\mathcal{G}(q, N_x)$.

$$dist([\mathbf{Y}], [\mathbf{Z}]) = \sum_{i=1}^q \arccos(s_i)$$

où s_i sont les valeurs singulières de la matrice $\mathbf{Y}^T\mathbf{Z}$. Les étapes de changement de signes des bases \mathbf{U}_k et \mathbf{V}_k pour $k = 1, \dots, N_b$, sont détaillés dans l'algorithme (Algo. 13).

Algorithme 13 : Ajustement de l'échantillonnage des bases POD.

```

1 for  $k = 1, \dots, N_b$  do
2   for  $j = 1, \dots, q$  and  $j \neq j_0$  do
3     • Ajustement des signes des modes spatiaux :
4     Si  $\|\mathbf{U}_{j_0}^j - \mathbf{U}_k^j\|_2 > \|\tilde{\mathbf{U}}^j + \mathbf{U}_k^j\|_2 \Rightarrow$  multiplier le  $j^{\text{ème}}$  mode  $\mathbf{U}_k^j$  de la base  $\mathbf{U}_k$  par  $-1$ .
5     • Ajustement des signes des modes temporels :
6     Si  $\|\mathbf{V}_{j_0}^j - \mathbf{V}_k^j\|_2 > \|\tilde{\mathbf{V}}^j + \mathbf{V}_k^j\|_2 \Rightarrow$  multiplier le  $j^{\text{ème}}$  mode  $\mathbf{V}_k^j$  de la base  $\mathbf{V}_k$  par  $-1$ .
7   end
8 end
```

- Une fois que les bases ajustés sont déterminées, l'étape suivante de la méthode NIMR consiste à proposer un critère permettant de retrouver les matrices de passage orthogo-

nales \mathbf{K} et \mathbf{Q} les mieux adaptées au classement des valeurs singulières interpolées. Elles sont choisies comme solutions des deux problèmes de minimisation suivants :

$$\mathbf{K} = \operatorname{argmin}_{\delta \mathbf{K} \in \mathcal{O}(q)} \sum_{i=1}^{N_b} \omega_i \|\tilde{\mathbf{U}} \delta \mathbf{K} - \mathbf{u}_i\|_F^2 \quad \text{et} \quad \mathbf{Q} = \operatorname{argmin}_{\delta \mathbf{Q} \in \mathcal{O}(q)} \sum_{i=1}^{N_b} \kappa_i \|\tilde{\mathbf{V}} \delta \mathbf{Q} - \mathbf{v}_i\|_F^2 \quad (4.1)$$

où $\|\cdot\|_F$ est la norme de Frobenius et ω_i et κ_i sont des coefficients de pondération tels que

$$\sum_{i=1}^{N_b} \omega_i = \sum_{i=1}^{N_b} \kappa_i = 1 \quad (4.2)$$

Le choix de ces coefficients est inspiré des poids de l'interpolation IDW. Ils sont donnés par

$$\omega_i = \frac{\operatorname{dist}([\tilde{\mathbf{U}}], [\mathbf{u}_i])^{-m}}{\sum_{k=1}^{N_b} \operatorname{dist}([\tilde{\mathbf{U}}], [\mathbf{u}_k])^{-m}} \quad \text{et} \quad \kappa_i = \frac{\operatorname{dist}([\tilde{\mathbf{V}}], [\mathbf{v}_i])^{-l}}{\sum_{k=1}^{N_b} \operatorname{dist}([\tilde{\mathbf{V}}], [\mathbf{v}_k])^{-l}} \quad (4.3)$$

avec $m, l > 1$ deux réels arbitraires. Trouver \mathbf{K} et \mathbf{Q} revient donc à résoudre les problèmes d'optimisation sous contraintes donnés par l'équation (4.1). Étant donné que ces deux problèmes sont similaires, nous nous contenterons d'illustrer les démarches de la résolution uniquement pour trouver \mathbf{K} .

- La recherche de \mathbf{K} comme solution du problème de minimisation sous contrainte d'orthogonalité peut être simplifié en introduisant la fonction de Lagrange suivante

$$\begin{aligned} \mathcal{L}(\mathbf{K}, \mathbf{R}) &= \sum_{i=1}^{N_b} \omega_i \|\tilde{\mathbf{U}} \mathbf{K} - \mathbf{u}_i\|_F^2 + \operatorname{Trace} \left(\mathbf{R}(\mathbf{K}^T \mathbf{K} - I_q) \right) \\ &= \sum_{i=1}^{N_b} \omega_i \|\tilde{\mathbf{U}} \mathbf{K}\|_F^2 - 2 \sum_{i=1}^{N_b} \omega_i \langle \tilde{\mathbf{U}} \mathbf{K}, \mathbf{u}_i \rangle_F + \sum_{i=1}^{N_b} \omega_i \|\mathbf{u}_i\|_F^2 + \operatorname{Trace} \left(\mathbf{R}(\mathbf{K}^T \mathbf{K} - I_q) \right) \end{aligned}$$

où \mathbf{R} est une matrice symétrique qui joue le rôle du multiplicateur de Lagrange. En tenant compte de (4.2) on peut écrire

$$\begin{aligned} \mathcal{L}(\mathbf{K}, \mathbf{R}) &= \operatorname{Trace} \left(\mathbf{K}^T \underbrace{\tilde{\mathbf{U}}^T \tilde{\mathbf{U}}}_{I_q} \mathbf{K} \right) - 2 \sum_{i=1}^{N_b} \omega_i \operatorname{Trace} \left(\mathbf{u}_i^T \tilde{\mathbf{U}} \mathbf{K} \right) \\ &\quad + \sum_{i=1}^{N_b} \omega_i \operatorname{Trace} \left(\underbrace{\mathbf{u}_i^T \mathbf{u}_i}_{I_q} \right) + \operatorname{Trace} \left(\mathbf{R}(\mathbf{K}^T \mathbf{K} - I_q) \right) \\ &= \operatorname{Trace} \left(\mathbf{K}^T \mathbf{K} \right) - 2 \sum_{i=1}^{N_b} \omega_i \operatorname{Trace} \left(\mathbf{u}_i^T \tilde{\mathbf{U}} \mathbf{K} \right) + \operatorname{Trace} \left(\mathbf{R}(\mathbf{K}^T \mathbf{K} - I_q) \right) + q \end{aligned}$$

En utilisant maintenant les identités suivantes

$$\begin{aligned} \frac{d}{d\mathbf{B}} \operatorname{Trace}(\mathbf{A} \mathbf{B}^T \mathbf{B}) &= \mathbf{B}(\mathbf{A} + \mathbf{A}^T) \\ \frac{d}{d\mathbf{B}} \operatorname{Trace}(\mathbf{A} \mathbf{B}) &= \mathbf{A}^T \end{aligned}$$

la différentielle de \mathcal{L} par rapport à \mathbf{K} s'écrit

$$\partial_{\mathbf{K}}\mathcal{L}(\mathbf{K}, \mathbf{R}) = 2\mathbf{K} - 2\sum_{i=1}^{N_b}\omega_i\tilde{\mathbf{U}}^T\mathbf{u}_i + 2\mathbf{K}\mathbf{R} \quad (4.4)$$

Par annulation de la différentielle de $\partial_{\mathbf{K}}\mathcal{L}(\mathbf{K}, \mathbf{R})$, on aboutit à

$$\mathbf{K}(I_q + \mathbf{R}) = \sum_{i=1}^{N_b}\omega_i\tilde{\mathbf{U}}^T\mathbf{u}_i \quad (4.5)$$

En utilisant l'équation (4.5), on peut extraire la relation suivante

$$(I_q + \mathbf{R})^T \mathbf{K}^T \mathbf{K} (I_q + \mathbf{R}) = \sum_{i=1}^{N_b} \sum_{j=1}^{N_b} \omega_i \omega_j \mathbf{u}_i^T \tilde{\mathbf{U}} \tilde{\mathbf{U}}^T \mathbf{u}_j \quad (4.6)$$

Étant donné que \mathbf{K} est orthogonale et que \mathbf{R} est symétrique, on peut écrire

$$I_q + \mathbf{R} = \left(\underbrace{\sum_{j=1}^{N_b} \sum_{i=1}^{N_b} \omega_i \omega_j \mathbf{u}_i^T \tilde{\mathbf{U}} \tilde{\mathbf{U}}^T \mathbf{u}_j}_M \right)^{\frac{1}{2}} \quad (4.7)$$

On note bien que la matrice M est par construction symétrique définie positive, ceci permet de calculer $M^{\frac{1}{2}}$ via la décomposition aux valeurs propres de M comme suit

$$M^{\frac{1}{2}} = P\lambda^{\frac{1}{2}}P^T \quad (4.8)$$

où λ est la matrice des valeurs propres de M et P est la matrice des vecteurs propres associés. Ainsi, on a

$$I_q + \mathbf{R} = P\lambda^{\frac{1}{2}}P^T \quad (4.9)$$

En injectant l'expression (4.9) dans l'équation (4.5), on trouve l'expression de la matrice de passage \mathbf{K} comme suit

$$\mathbf{K} = \tilde{\mathbf{U}}^T \left(\sum_{i=1}^{N_b} \omega_i \mathbf{u}_i \right) P\lambda^{-\frac{1}{2}}P^T$$

• En suivant la même démarche, on peut montrer que la matrice \mathbf{Q} est donnée par

$$\mathbf{Q} = \tilde{\mathbf{V}}^T \left(\sum_{i=1}^{N_b} \kappa_i \mathbf{v}_i \right) \mathbf{H}\eta^{-\frac{1}{2}}\mathbf{H}^T$$

où \mathbf{H} et η vérifient la décomposition aux valeurs propres

$$\sum_{j=1}^{N_b} \sum_{i=1}^{N_b} \kappa_i \kappa_j \mathbf{v}_i^T \tilde{\mathbf{V}} \tilde{\mathbf{V}}^T \mathbf{v}_j = \mathbf{H}\eta\mathbf{H}^T$$

- Les étapes de la méthode NIMR sont résumées dans l'algorithme 14.

Algorithme 14 : Algorithme NIMR.

Étape offline :

- 1 Pour $i = 1, \dots, N_b$, approximer la matrice de snapshots X_i en utilisant la décomposition SVD tronquée d'ordre q , soit :

$$X_i \approx \mathbf{U}_i \Sigma_i \mathbf{V}_i^T$$

Étape online :

- 2 Interpoler les valeurs singulières $\sigma_{i,1}, \dots, \sigma_{i,q}$ et construire la matrice $\tilde{\Sigma}$ pour le paramètre cible $\tilde{\theta}$.
- 3 Calculer $\tilde{\mathbf{U}}$ et $\tilde{\mathbf{V}}$ les matrices représentant les sous-espaces vectoriels $[\tilde{\mathbf{U}}]$ et $[\tilde{\mathbf{V}}]$ associés au nouveau paramètre $\tilde{\theta}$ et obtenus en appliquant l'interpolation ITSGM respectivement sur les sous-espaces $[\mathbf{U}_i]$ et $[\mathbf{V}_i]$, $i = 1, \dots, N_b$.
- 4 Construire les nouvelles bases \mathbf{U}_k et \mathbf{V}_k pour $k = 1, \dots, N_b$ par l'algorithme (Algo. 13).
- 5 Calculer les coefficients de pondération ω_j et κ_j définis par l'équation (4.3).
- 6 Calculer λ la matrice diagonale des valeurs propres, et \mathbf{P} la matrice des vecteurs propres associés, vérifiant la décomposition aux valeurs propres suivante

$$\sum_{j=1}^{N_b} \sum_{i=1}^{N_b} \omega_i \omega_j \mathbf{U}_i^T \tilde{\mathbf{U}} \tilde{\mathbf{U}}^T \mathbf{U}_j = \mathbf{P} \lambda \mathbf{P}^T$$

- 7 Calculer η la matrice diagonale des valeurs propres, et \mathbf{H} la matrice des vecteurs propres associés, vérifiant la décomposition aux valeurs propres suivante

$$\sum_{j=1}^{N_b} \sum_{i=1}^{N_b} \kappa_i \kappa_j \mathbf{V}_i^T \tilde{\mathbf{V}} \tilde{\mathbf{V}}^T \mathbf{V}_j = \mathbf{H} \eta \mathbf{H}^T$$

- 8 Calculer les matrices orthogonales \mathbf{K} et \mathbf{Q} par les relations

$$\mathbf{K} = \tilde{\mathbf{U}}^T \left(\sum_{i=1}^{N_b} \omega_i \mathbf{U}_i \right) \mathbf{P} \lambda^{-\frac{1}{2}} \mathbf{P}^T$$

$$\mathbf{Q} = \tilde{\mathbf{V}}^T \left(\sum_{i=1}^{N_b} \kappa_i \mathbf{V}_i \right) \mathbf{H} \eta^{-\frac{1}{2}} \mathbf{H}^T$$

- 9 Reconstruire la matrice des snapshots interpolée

$$\tilde{\mathbf{X}} = \tilde{\mathbf{U}} \mathbf{K} \tilde{\Sigma} \mathbf{Q}^T \tilde{\mathbf{V}}^T$$

Remarque 4.1 L'étape offline de l'algorithme 14 correspond à l'étape de construction des bases réduites qui se fait une fois pour toute en dehors de l'algorithme NIMR. Cette étape n'est

donc pas effectué à chaque fois que le paramètre d'interpolation change, seules les étapes 2 à 9 sont exécutées.

4.3 MÉTHODE HNIMR

La méthode HNIMR n'est rien d'autre que la méthode NIMR mais en dimension hyper-réduite. C'est une sorte de compression de données dont l'objectif est de diminuer davantage la dimensionnalité du problème d'interpolation. Supposons que pour $i = 1, \dots, q$, nous disposons des décompositions en valeurs singulières tronquées d'ordre q des matrices de snapshots \mathbf{X}_i données par

$$\mathbf{X}_i \approx \mathbf{U}_i \boldsymbol{\Sigma}_i \mathbf{V}_i^T$$

où $\mathbf{U}_i \in \mathbb{R}^{N_x \times q}$ et $\mathbf{V}_i \in \mathbb{R}^{N_s \times q}$ et $\boldsymbol{\Sigma}_i = \text{diag}(\sigma_{i,1}, \sigma_{i,2}, \dots, \sigma_{i,q})$ telle que

$$\sigma_{i,j} = \sqrt{\lambda_j(\mathbf{X}_i^T \mathbf{X}_i)} \geq \sigma_{i,j+1}, \text{ pour } j = 1, \dots, q-1$$

La méthode HNIMR consiste tout d'abord à construire les matrices $\text{col}(\mathbf{U})$ et $\text{col}(\mathbf{V})$ par concaténation par bloc des bases \mathbf{U}_i et \mathbf{V}_i , soient

$$\text{col}(\mathbf{U}) = [\mathbf{U}_1 \ \mathbf{U}_2 \ \dots \ \mathbf{U}_{N_b}] \quad \text{et} \quad \text{col}(\mathbf{V}) = [\mathbf{V}_1 \ \mathbf{V}_2 \ \dots \ \mathbf{V}_{N_b}]$$

On effectue ensuite les décompositions en valeurs singulières tronquées de $\text{col}(\mathbf{U})$ et $\text{col}(\mathbf{V})$ d'ordre r et s respectivement comme suit

$$\text{col}(\mathbf{U}) = \boldsymbol{\varphi} \boldsymbol{\varrho} \mathbf{W}^T \quad \text{et} \quad \text{col}(\mathbf{V}) = \boldsymbol{\alpha} \Theta \mathbf{Z}^T$$

où $r, s \leq qN_b$, $\boldsymbol{\varphi} \in \mathbb{R}^{N_x \times r}$, $\mathbf{W} \in \mathbb{R}^{qN_b \times r}$, $\boldsymbol{\alpha} \in \mathbb{R}^{N_s \times s}$, $\mathbf{Z} \in \mathbb{R}^{qN_b \times s}$, $\boldsymbol{\varrho} = \text{diag}(\rho_1, \rho_2, \dots, \rho_r)$ et $\Theta = \text{diag}(\theta_1, \theta_2, \dots, \theta_s)$. En rangeant les matrices \mathbf{W} et \mathbf{Z} sous forme de matrices lignes par blocs

$$\mathbf{W} = \begin{bmatrix} W_1 \\ W_2 \\ \vdots \\ W_{N_b} \end{bmatrix} \quad \text{et} \quad \mathbf{Z} = \begin{bmatrix} Z_1 \\ Z_2 \\ \vdots \\ Z_{N_b} \end{bmatrix}$$

avec $W_i \in \mathbb{R}^{q \times r}$ et $Z_i \in \mathbb{R}^{q \times s}$, on peut approximer encore les matrices de snapshots \mathbf{X}_i comme suit

$$\mathbf{X}_i \approx \boldsymbol{\varphi} \boldsymbol{\varrho} W_i^T \boldsymbol{\Sigma}_i Z_i \Theta \boldsymbol{\alpha}^T$$

L'avantage de cette écriture est l'apparition d'un nouvel ensemble de matrices de snapshots réduites $\mathbf{Y}_i = W_i^T \boldsymbol{\Sigma}_i Z_i$ de taille $r \times s$ hyper réduite. Ces matrices réduites formeront le nouvel ensemble de snapshots paramétrés sur lequel la méthode NIMR va être appliquée. Soit donc $\mathbf{Y}_i = \mathbf{N}_i \boldsymbol{\pi}_i \mathbf{J}_i^T$ la décomposition en valeurs singulières tronquée d'ordre m de \mathbf{Y}_i et écrivons

$$\mathbf{X}_i \approx \boldsymbol{\varphi} \boldsymbol{\varrho} \mathbf{N}_i \boldsymbol{\pi}_i \mathbf{J}_i^T \Theta \boldsymbol{\alpha}^T \quad (4.10)$$

où $N_i \in \mathbb{R}^{r \times m}$, $J_i \in \mathbb{R}^{s \times m}$ sont maintenant les bases réduites et $\pi_{i,j}$ les valeurs singulières telles que

$$\pi_{i,j} = \sqrt{\lambda_j(Y_i^T Y_i)} \geq \pi_{i,j+1}, \text{ pour } j = 1, \dots, m-1$$

Par analogie avec la section précédente, les valeurs singulières π_i sont tout d'abord interpolées en utilisant une méthode d'interpolation usuelle et la matrice $\tilde{\pi}$ des valeurs singulières approximée est obtenue. L'interpolation ITSGM sera ensuite appliquée sur l'ensemble des sous-espaces vectoriels $[N_i]$ puis l'ensemble des sous-espaces vectoriels $[J_i]$ afin d'obtenir deux représentants des classes d'équivalence $[\tilde{N}]$ et $[\tilde{J}]$ respectivement. Finalement, les matrices de passage orthogonales K et Q permettant de classer les vecteurs des bases interpolées \tilde{N} et \tilde{J} seront recherchées comme solutions des problèmes d'optimisation

$$K = \operatorname{argmin}_{\delta K \in \mathcal{O}(q)} \sum_{i=1}^{N_b} \omega_i \|\tilde{N} \delta K - N_i\|_F^2 \quad \text{et} \quad Q = \operatorname{argmin}_{\delta Q \in \mathcal{O}(q)} \sum_{i=1}^{N_b} \kappa_i \|\tilde{J} \delta Q - J_i\|_F^2 \quad (4.11)$$

où ω_i et κ_i sont des coefficients de pondération telles que

$$\sum_{i=1}^{N_b} \omega_i = \sum_{i=1}^{N_b} \kappa_i = 1$$

L'expression de K et Q dans ce cas est

$$K = \tilde{N}^T \left(\sum_{i=1}^{N_b} \omega_i N_i \right) P \lambda^{-\frac{1}{2}} P^T$$

$$Q = \tilde{J}^T \left(\sum_{i=1}^{N_b} \kappa_i J_i \right) H \eta^{-\frac{1}{2}} H^T$$

avec P , λ , H , η qui sont obtenues par les décompositions en valeurs propres suivantes

$$\sum_{j=1}^{N_b} \sum_{i=1}^{N_b} \omega_i \omega_j N_i^T \tilde{N} \tilde{N}^T N_j = P \lambda P^T$$

$$\sum_{j=1}^{N_b} \sum_{i=1}^{N_b} \kappa_i \kappa_j J_i^T \tilde{J} \tilde{J}^T J_j = H \eta H^T$$

On en déduit que

$$\tilde{Y} = \tilde{N} K \tilde{\pi} Q^T \tilde{J}^T$$

Finalement, la solution approximée sera reconstruite comme suit

$$\tilde{X} \approx \varphi \varrho \tilde{Y} \Theta \alpha^T \quad (4.12)$$

Les étapes de la méthode HNIMR sont résumés dans l'algorithme 15.

Algorithme 15 : Algorithme HNIMR.

Étape offline :

- 1 Pour $i = 1, \dots, N_b$, approximer la matrice de snapshots $\mathbf{X}_i \in \mathbb{R}^{N_x \times N_s}$ en utilisant la décomposition SVD tronquée d'ordre q , soit :

$$\mathbf{X}_i \approx \mathbf{U}_i \boldsymbol{\Sigma}_i \mathbf{V}_i^T$$

- 2 Construire les matrices $\text{col}(\mathbf{U})$ et $\text{col}(\mathbf{V})$ par concaténation par bloc

$$\text{col}(\mathbf{U}) = [\mathbf{U}_1 \ \mathbf{U}_2 \ \cdots \ \mathbf{U}_{N_b}] \quad \text{et} \quad \text{col}(\mathbf{V}) = [\mathbf{V}_1 \ \mathbf{V}_2 \ \cdots \ \mathbf{V}_{N_b}]$$

et calculer leurs décompositions SVD tronquées d'ordre r et s respectivement :

$$\text{col}(\mathbf{U}) = \boldsymbol{\varphi} \boldsymbol{\varrho} \mathbf{W}^T \quad \text{col}(\mathbf{V}) = \boldsymbol{\alpha} \Theta \mathbf{Z}^T$$

- 3 Extraire les matrices blocs $W_i \in \mathbb{R}^{q \times r}$ et $Z_i \in \mathbb{R}^{q \times s}$ tels que

$$\mathbf{W} = \begin{bmatrix} W_1 \\ W_2 \\ \vdots \\ W_{N_b} \end{bmatrix} \quad \text{et} \quad \mathbf{Z} = \begin{bmatrix} Z_1 \\ Z_2 \\ \vdots \\ Z_{N_b} \end{bmatrix}$$

- 4 Construire les matrices de snapshots hyper réduites $\mathbf{Y}_i = W_i^T \boldsymbol{\Sigma}_i Z_i$ et calculer leur décompositions SVD tronquées d'ordre m :

$$\mathbf{Y}_i = N_i \boldsymbol{\pi}_i \mathbf{J}_i^T, i = 1, \dots, N_b$$

Étape online :

- 5 Pour un nouveau paramètre d'interpolation, interpoler les matrices \mathbf{Y}_i en utilisant les étapes 2 à 7 de l'algorithme NIMR (Algorithme 14) pour obtenir

$$\tilde{\mathbf{Y}} = \tilde{N} \tilde{K} \tilde{\boldsymbol{\pi}} \tilde{Q}^T \tilde{\mathbf{J}}^T$$

- 6 Reconstruire la matrice des snapshots interpolée

$$\tilde{\mathbf{X}} \approx \boldsymbol{\varphi} \boldsymbol{\varrho} \tilde{\mathbf{Y}} \Theta \boldsymbol{\alpha}^T \tag{4.13}$$

4.4 COÛT ASYMPTOTIQUE DES MÉTHODES NIMR ET HNIMR

Rappelons que N_x , N_s , N_b signifient respectivement, le nombre de degrés de liberté du problème complet, le nombre de snapshots et le nombre de bases POD (supposées de même dimension q). Comme pour le calcul de la complexité de la méthode ITSGM, le

coût de calcul associé aux méthodes NIMR et HNIMR sera présenté lorsque le paramètre d'interpolation est de dimension $N_p = 1$:

- **Étape 2** : L'interpolation des valeurs singulières est instantanée et nécessite un coût asymptotique égale à $N_b \times \mathcal{O}(q)$.
- **Étape 3** : La méthode ITSGM est appliquée deux fois avec un coût asymptotique qui s'élève à $(3N_b - 1) \times [\mathcal{O}(N_x q^2) + \mathcal{O}(N_s q^2)]$
- **Étape 5, 6 et 7** : Ces étapes nécessitent les opérations suivantes :
 - N_b multiplications matricielles avec un coût asymptotique de $N_b \times \mathcal{O}(N_x q^2)$.
 - N_b multiplications matricielles avec un coût asymptotique de $N_b \times \mathcal{O}(N_s q^2)$.
 - calcul de deux décompositions aux valeurs propres avec un coût asymptotique égale à $2\mathcal{O}(q^3)$.
- **Étape 8** : multiplications matricielles d'un coût asymptotique de $2 [\mathcal{O}(q^3) + \mathcal{O}(q^2)]$.

Une estimation du coût asymptotique total pour les calculs effectués dans la méthode NIMR dans le cas $d_p = 1$ s'élève donc à $6N_b \times \mathcal{O}(\max(N_x, N_s)q^2)$. La méthode NIMR est alors efficace d'un point de vue numérique, car sa complexité est une fonction linéaire du maximum entre le nombre de degrés de libertés du problème complet et le nombre de snapshots.

Le coût des calculs peut davantage être réduit en considérant la version hyper réduite HNIMR. Dans ce cas, le coût asymptotique est considérablement réduit et atteint au maximum $6N_b \times \mathcal{O}(\max(r, s)q^2)$, où r et s sont très petits devant N_x et N_s (ordre de la dizaine).

4.5 APPLICATION SUR LE PROBLÈME DE L'ÉCOULEMENT AUTOUR D'UN CYLINDRE

4.5.1 Position du problème

Nous considérons l'écoulement d'un fluide dans un canal, autour d'un cylindre circulaire de diamètre D (Figure 4.1). Le domaine d'étude Ω est un domaine rectangulaire de hauteur $H = 30D$ et de longueur $45D$. Le centre du cylindre est situé à $L_1 = 10D$ de la frontière gauche et à $H/2$ de la paroi basse. Une vitesse d'intensité u_∞ est appliquée à l'entrée du canal et une condition de type outlet ($pn - \frac{1}{Re} \nabla \mathbf{u} \cdot \mathbf{n} = 0$) à sa sortie. Sur les parois horizontales du canal, les conditions aux limites imposées sur la vitesse sont telles que $v = 0$ (composante verticale) et $\partial_y u = 0$ (composante horizontale). Le fluide est supposé isotherme et incompressible et sa dynamique est régie par les équations de

Navier-Stokes suivantes

$$\left\{ \begin{array}{ll} \partial_t \mathbf{u} - \frac{1}{Re} \Delta \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} + \nabla p = 0 & \text{dans } \Omega \times [0, T[\\ \nabla \cdot \mathbf{u} = 0 & \text{dans } \Omega \times [0, T[\\ \mathbf{u} = \mathbf{u}_\infty & \text{sur } \Gamma_{in} \times [0, T[\\ \mathbf{u} = 0 & \text{sur } \Gamma_{cylindre} \times [0, T[\\ u_2 = 0 & \text{sur } \Gamma_{noslip} \times [0, T[\\ \partial_n u_1 = 0 & \text{sur } \Gamma_{noslip} \times [0, T[\\ -\frac{1}{Re} \partial_n \mathbf{u} + p \mathbf{n} = 0 & \text{sur } \Gamma_{out} \times [0, T[\\ \mathbf{u}(0) = \mathbf{u}_0 & \text{dans } \Omega \end{array} \right.$$

Les valeurs du nombre de Reynolds $Re = \mathbf{u}_\infty D / \nu$ considérées dans cette étude varient entre 90 et 450. Les simulations numériques ont été effectuées avec Fenics [94] (éléments finis de Taylor-Hood $\mathbb{P}_2/\mathbb{P}_1$) en utilisant un maillage non uniforme, comprenant 85124 degrés de libertés pour la vitesse et 10694 pour la pression, raffiné autour du cylindre et le long de la zone du sillage (Figure 4.2). La condition initiale considérée pour toutes les valeurs du nombre de Reynolds correspond à la solution à un instant donné du régime periodique de l'écoulement à $Re = 100$. Le pas de temps est fixé à 0.01.

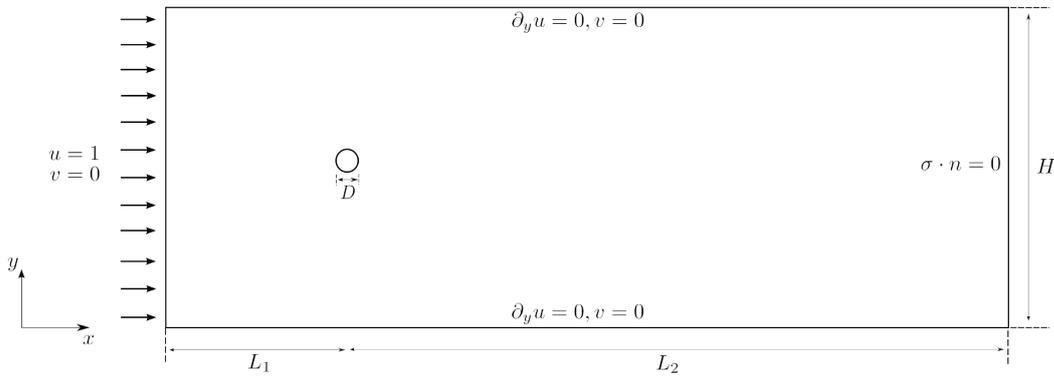


FIGURE 4.1 – Domaine spatial et conditions au limites pour le cas d'étude de l'écoulement autour d'un cylindre circulaire.

Afin de tester la pertinence des méthodes NIMR et HNIMR de manière quantitative, les erreurs relatives, le nombre de Strouhal ainsi que les coefficients aérodynamiques (coefficient de traînée C_D et de portance C_L) sont calculés. Le nombre de Strouhal est défini par $S_t = \frac{D f_s}{\mathbf{u}_\infty}$ où f_s représente la fréquence d'oscillation de l'écoulement. Les coefficients adimensionnels de traînée C_D et de portance C_L sont donnés par les expressions suivantes

$$C_D = \frac{F_D}{\frac{1}{2} \rho U_\infty^2 D} ; C_L = \frac{F_L}{\frac{1}{2} \rho U_\infty^2 D}$$

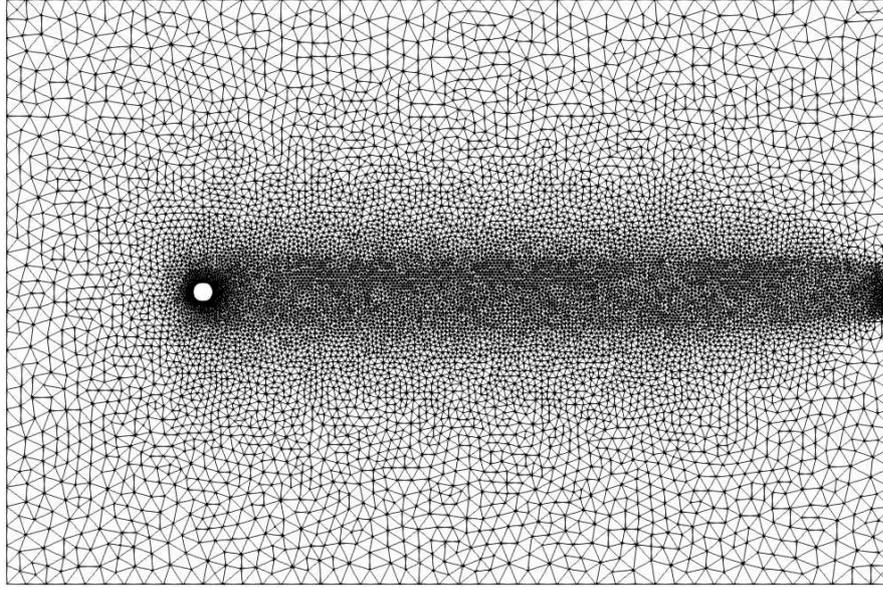


FIGURE 4.2 – Maillage éléments finis du domaine fluide pour le cas de l'écoulement autour d'un cylindre.

où ρ est la masse volumique du fluide et F_D, F_L représentent respectivement les forces de traînée et de portance exercées par le fluide sur le cylindre définies par la relation suivante

$$\begin{pmatrix} F_D \\ F_L \end{pmatrix} = \int_{\Gamma_{\text{cylindre}}} \left(\mu(\nabla \mathbf{u} + \nabla^T \mathbf{u}) - p\mathbf{I} \right) \cdot \mathbf{n} \, d\sigma$$

où \mathbf{u} et p sont la vitesse et la pression dimensionnelles et μ la viscosité dynamique du fluide.

4.5.2 Construction des bases réduites d'apprentissage

Un échantillonnage d'écoulements paramétrés¹ est construit pour les valeurs du nombre de Reynolds comprises entre $Re = 90$ et $Re = 450$ en considérant un pas $\Delta Re = 30$, soit $Re = \{90, 120, 150, 180, \dots, 450\}$. Les variables de vitesse et de pression sont ensuite décomposées en un champ moyen² et un champ fluctuant comme suit

$$\begin{cases} \mathbf{u}(t, x, Re) = \bar{\mathbf{u}}(x) + \tilde{\mathbf{u}}(t, x, Re) \\ p(t, x, Re) = \bar{p}(x) + \tilde{p}(t, x, Re) \end{cases}$$

où les champs moyens $\bar{\mathbf{u}}$ et \bar{p} correspondent à la moyenne de tous les snapshots issus des solutions du modèle complet pour les différentes valeurs du nombre de Reynolds de

1. Les équations de Navier-Stokes sont résolues sous forme adimensionnelle. Lorsque le nombre de Reynolds varie, u_∞ n'est pas modifié, la variation se fait par l'intermédiaire de la viscosité cinématique ν .

2. Les conditions aux limites sont portées par le champ moyen.

l'échantillonnage considéré, soient :

$$\bar{\mathbf{u}} = \frac{1}{N_b N_s} \sum_{i=1}^{N_b} \sum_{j=1}^{N_s} \mathbf{u}(t_j, x, Re_i) \quad \bar{p} = \frac{1}{N_b N_s} \sum_{i=1}^{N_b} \sum_{j=1}^{N_s} p(t_j, x, Re_i)$$

N_s est le nombre de Snapshots pour chaque nombre de Reynolds et N_b le nombre des valeurs du nombre de Reynolds de l'échantillonnage. Les champs fluctuants sont ensuite approximés sur deux sous-espaces vectoriels de dimension finie $\tilde{V} = \text{span}\{\varphi_u^1, \dots, \varphi_u^{q_u}\}$ et $\tilde{P} = \text{span}\{\varphi_p^1, \dots, \varphi_p^{q_p}\}$ comme suit

$$\tilde{\mathbf{u}}(t, x, Re) = \sum_{i=1}^{q_u} \alpha_u^i(t, Re) \varphi_u^i(x, Re) \quad (4.14)$$

$$\tilde{p}(t, x, Re) = \sum_{l=1}^{q_p} \alpha_p^l(t, Re) \varphi_p^l(x, Re) \quad (4.15)$$

$\boldsymbol{\varphi} = \{\varphi_u^1, \dots, \varphi_u^{q_u}\}$ est la base POD de vitesse et $\{\varphi_p^1, \dots, \varphi_p^{q_p}\}$ la base POD de pression. Ces deux bases ont été calculées en considérant 500 snapshots ($N_s = 500$) régulièrement répartis entre les instants $t_i = 7$ et $t_f = 12$, ce qui représente environs 8 périodes du détachement tourbillonnaire de l'écoulement. La contribution des modes POD dans l'approximation de $\tilde{\mathbf{u}}$ et \tilde{p} est déterminée par le ratio

$$\mathcal{R}^k = \frac{\sum_{i=1}^k \lambda_i}{\sum_{i=1}^{N_s} \lambda_i}$$

où les λ_i sont les valeurs propres POD. Les figures 4.3 et 4.4 sur lesquelles sont tracé les valeurs propres POD et leurs ratios pour les différents nombres de Reynolds d'apprentissages, indiquent que très peu de modes POD sont capables de capturer la quasi totalité de l'énergie. Ainsi, pour ce cas d'étude, les sous espaces vectoriels \tilde{V} et \tilde{P} sont construits à partir des modes propres associés aux premières valeurs propres ayant une contribution significative, soient $q_u = 10$ et $q_p = 8$. Pour chaque nombre de Reynolds d'apprentissage, les modèles réduits seront construits en gardant le même nombre de modes, à savoir 10 modes de vitesse et 8 modes de pression.

4.5.3 Mise en œuvre des méthodes NIMR et HNIMR

Si $\mathbf{X}_u(Re)$ et $\mathbf{X}_p(Re)$ désignent les matrices formées par concaténation des vecteurs snapshots obtenus pour un nombre de Reynolds donné

$$\mathbf{X}_u(Re) = [\mathbf{u}_{q_u}^1(Re), \dots, \mathbf{u}_{q_u}^{N_s}(Re)] \quad \mathbf{X}_p(Re) = [p_{q_p}^1(Re), \dots, p_{q_p}^{N_s}(Re)]$$

alors l'approximation (4.14) peut s'écrire sous forme matricielle comme suit

$$\begin{cases} \mathbf{X}_u(Re) \approx \boldsymbol{\varphi}_u(Re) \boldsymbol{\alpha}_u(Re)^T = \boldsymbol{\varphi}_u(Re) \boldsymbol{\Sigma}_u(Re) \mathbf{V}_u(Re)^T \\ \mathbf{X}_p(Re) \approx \boldsymbol{\varphi}_p(Re) \boldsymbol{\alpha}_p(Re)^T = \boldsymbol{\varphi}_p(Re) \boldsymbol{\Sigma}_p(Re) \mathbf{V}_p(Re)^T \end{cases} \quad (4.16)$$

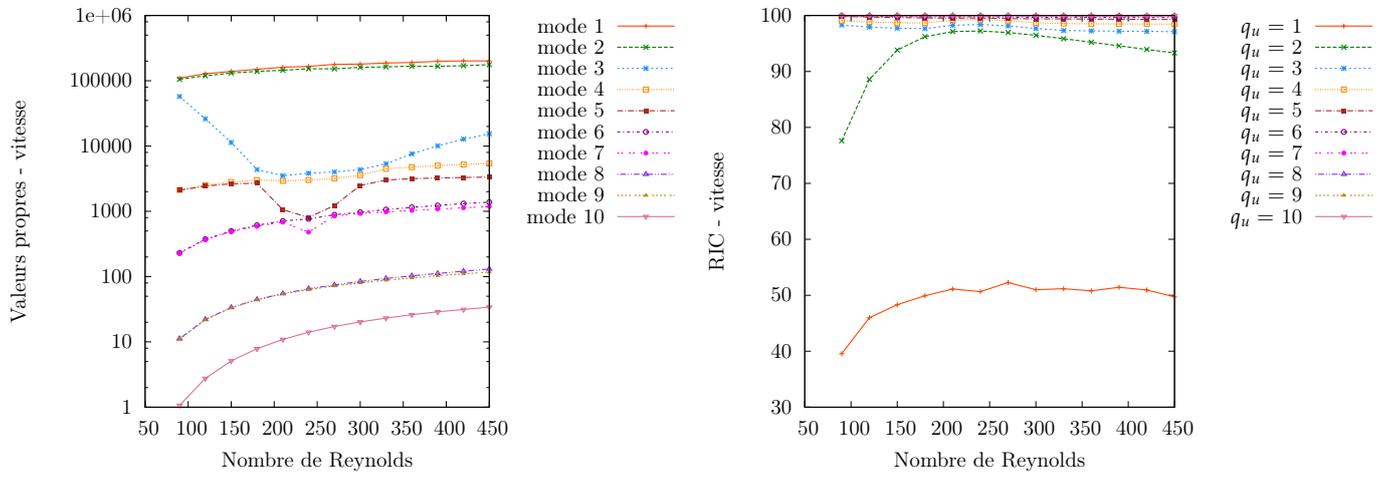


FIGURE 4.3 – Valeurs propres POD de vitesse et ratio en fonction du nombre de Reynolds.

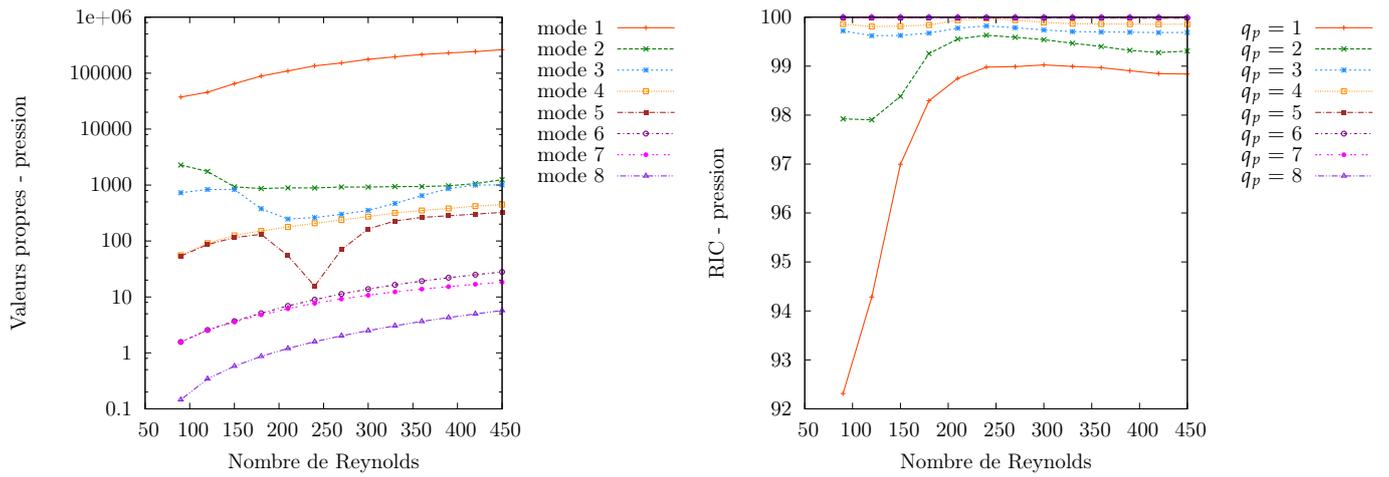


FIGURE 4.4 – Valeurs propres POD de pression et Ratio en fonction du nombre de Reynolds.

où $\boldsymbol{\varphi}_u(Re)$ et $\boldsymbol{\varphi}_p(Re)$ sont normalisés au sens de la norme euclidienne et les matrices $\Sigma_u(Re)$, $\Sigma_p(Re)$, $\mathbf{V}_u(Re)$ et $\mathbf{V}_p(Re)$ sont telles que

$$\begin{aligned}\Sigma_u(Re) &= \text{diag}(\|\alpha_u^1(Re)\|, \dots, \|\alpha_u^{q_u}(Re)\|) & \Sigma_p(Re) &= \text{diag}(\|\alpha_p^1(Re)\|, \dots, \|\alpha_p^{q_p}(Re)\|) \\ \mathbf{V}_u(Re) &= \frac{\boldsymbol{\alpha}_u(Re)}{\|\boldsymbol{\alpha}_u(Re)\|} & \mathbf{V}_p(Re) &= \frac{\boldsymbol{\alpha}_p(Re)}{\|\boldsymbol{\alpha}_p(Re)\|}\end{aligned}$$

L'intérêt de cette décomposition³ est d'extraire les valeurs singulières des coefficients temporels afin de les normaliser et par suite se ramener à la décomposition SVD des matrices $\mathbf{X}_u(Re)$ et $\mathbf{X}_p(Re)$. Considérons maintenant N_b valeurs du nombre de Reynolds pour lesquelles nous construisons les deux ensembles de matrices de snapshots

$$\mathcal{D}^{(u)} = \{\mathbf{X}_u(Re_1), \mathbf{X}_u(Re_2), \dots, \mathbf{X}_u(Re_{N_b})\} \quad \text{et} \quad \mathcal{D}^{(p)} = \{\mathbf{X}_p(Re_1), \mathbf{X}_p(Re_2), \dots, \mathbf{X}_p(Re_{N_b})\}$$

En utilisant les décompositions en valeurs singulières (4.16), les ensembles suivants sont formés

$$\mathcal{D}_x^{(u)} = \{\boldsymbol{\varphi}_u(Re_1), \boldsymbol{\varphi}_u(Re_2), \dots, \boldsymbol{\varphi}_u(Re_{N_b})\} \quad \text{et} \quad \mathcal{D}_x^{(p)} = \{\boldsymbol{\varphi}_p(Re_1), \boldsymbol{\varphi}_p(Re_2), \dots, \boldsymbol{\varphi}_p(Re_{N_b})\}$$

$$\mathcal{D}_t^{(u)} = \{\mathbf{V}_u(Re_1), \mathbf{V}_u(Re_2), \dots, \mathbf{V}_u(Re_{N_b})\} \quad \text{et} \quad \mathcal{D}_t^{(p)} = \{\mathbf{V}_p(Re_1), \mathbf{V}_p(Re_2), \dots, \mathbf{V}_p(Re_{N_b})\}$$

$$\mathcal{D}_\sigma^{(u)} = \{\Sigma_u(Re_1), \Sigma_u(Re_2), \dots, \Sigma_u(Re_{N_b})\} \quad \text{et} \quad \mathcal{D}_\sigma^{(p)} = \{\Sigma_p(Re_1), \Sigma_p(Re_2), \dots, \Sigma_p(Re_{N_b})\}$$

Nous rappelons que l'objectif de la NIMR est de construire les matrices de snapshots $\mathbf{X}_u(\widetilde{Re})$ et $\mathbf{X}_p(\widetilde{Re})$ approchées correspondant à une nouvelle valeur du nombre de Reynolds $\widetilde{Re} \neq Re_i$ pour $i = 1, \dots, N_b$. La première étape consiste à interpoler les ensembles des matrices singulières $\mathcal{D}_\sigma^{(u)}$ et $\mathcal{D}_\sigma^{(p)}$ en utilisant l'interpolation spline cubique. L'adaptation des bases spatiales et temporelles pour le nouveau nombre de Reynolds est ensuite établie en appliquant la méthode ITSGM sur les ensembles $\mathcal{D}_x^{(u)}$, $\mathcal{D}_x^{(p)}$, $\mathcal{D}_t^{(u)}$ et $\mathcal{D}_t^{(p)}$. L'interpolation utilisée dans l'espace tangent à la variété de Grassmann est celle de Lagrange. L'étape d'ajustement des bases interpolées $\widetilde{\boldsymbol{\varphi}}_u$, $\widetilde{\boldsymbol{\varphi}}_p$, $\widetilde{\mathbf{V}}_u$ et $\widetilde{\mathbf{V}}_p$, est effectuée en calculant les matrices orthogonales de petite taille \mathbf{Q}_u et \mathbf{K}_u pour la vitesse et \mathbf{Q}_p et \mathbf{K}_p pour la pression par résolution des deux problèmes d'optimisation sous contraintes (4.1), où les puissances ont été choisies égales à 3 (c'est à dire $m = l = 3$).

Remarque 4.2 *Pour le cas de l'écoulement autour d'un cylindre, les modes temporels sont périodiques et possèdent des conditions initiales différentes sur la phase de l'échantillonnage. Les résultats de la méthode NIMR et HNIMR peuvent par conséquent être déphasés par rapport aux résultats de référence issus du modèle complet. Afin de procéder à une étude comparative avec les résultats de la POD, les résultats sont post-traités pour supprimer ce déphasage.*

3. Cette écriture exprime simplement le lien entre la méthode POD et la décomposition SVD.

4.5.4 Évaluation des performances des méthodes NIMR et HNIMR

L'objectif est d'illustrer la capacité des méthodes ITSGM/ROM, NIMR et HNIMR à prédire la dynamique de l'écoulement pour des nombres de Reynolds n'appartenant pas à l'ensemble des nombres de Reynolds d'apprentissage. Le pourcentage d'erreur moyen $\bar{\varepsilon}_u\%$ entre la solution du modèle complet \mathbf{u} et son approximation \mathbf{u}_{q_u} est défini par la relation

$$\bar{\varepsilon}_u\% = 100 \times \frac{\left(\int_0^T \|\mathbf{u} - \mathbf{u}_{q_u}\|_{L^2(\Omega)}^2 dt \right)^{\frac{1}{2}}}{\left(\int_0^T \|\mathbf{u}\|_{L^2(\Omega)}^2 dt \right)^{\frac{1}{2}}}$$

Les mêmes relations sont utilisés pour mesurer l'erreur d'approximation de la pression p . L'influence du nombre de bases N_b et du pas ΔRe entre deux nombres de Reynolds de l'échantillonnage sur la méthode NIMR a été testée pour $Re = 160$ et $Re = 280$ en considérant $2 \leq N_b \leq 6$ et $20 \leq \Delta Re \leq 50$. Les figures 4.5 et 4.6 indiquent que l'évolution de l'erreur relative moyenne pour la vitesse et la pression ne suit pas de règle bien établi en fonction de la variation de N_b et de ΔRe . Par conséquent, il est difficile d'affirmer à partir de ces résultats quel est le choix optimal du nombre de bases N_b et du pas de Reynolds ΔRe . Par la suite, nous prendrons $N_b = 3$ et $\Delta Re = 30$.

Les nombres de Reynolds d'apprentissage seront donc ceux considérés au paragraphe 4.5.2, à savoir 90, 120, 150, 180, 210, 240, 270, 300, 330, 360, 390, 420 et 450. Pour $Re_{cible} = 195$, les modes spatiaux de la vitesse et de pression obtenus par la méthode NIMR sont comparés aux modes POD⁴ sur les figures 4.7 et 4.8. On constate que les modes NIMR reclassés sont très proches de ceux obtenus par la POD. Les matrices de calibration permettent bien de reclasser les modes de la base interpolée selon l'ordre et la contribution des modes POD. De même, les coefficients temporels obtenus par NIMR sont quasi-identiques à ceux de la POD (voir les figures 4.9 et 4.10). Les coefficients aérodynamiques obtenus par le modèle complet (FULL), la méthode NIMR et la méthode ITSGM/ROM⁵ pour $Re_{cible} = 195$ sont tracés sur la figure 4.11. Une bonne concordance des résultats obtenus par la méthode NIMR avec ceux obtenus par le modèle complet est observée. Ceci est confirmé par les erreurs relatives moyennes répertoriées dans le tableau 4.1. En effet, la méthode NIMR permet d'obtenir une approximation très satisfaisante sur la vitesse et la pression (0.47% d'erreur sur la vitesse 3.23% sur la pression).

Différents nombre de Reynolds $90 < Re_i < 450$ distincts des nombres de Reynolds d'apprentissage ont été considéré par la suite afin de tester et valider l'approche NIMR. Les coefficients aérodynamiques $\bar{C}_D, C_{L,max}, C_{L,rms}, S_t$ sont tracés en fonction du nombre

4. POD désigne ici les au fonctions de base POD de référence calculées à partir des snapshots issus du modèle complet pour le nombre de Reynolds cible Re_{cible} .

5. ITSGM/ROM est le modèle réduit interpolé obtenu par projection de Galerkin de l'équation du modèle complet sur la base interpolé par ITSGM.

de Reynolds sur la figure 4.12, et comparés aux résultats⁶ issus du modèle complet et de l'approche ITSGM/ROM. Une bonne concordance entre les résultats du modèle complet et les résultats de la méthode NIMR est observée. L'erreur relative moyenne entre la solution de référence et la solution obtenue par interpolation est représentée sur la figure 4.13. On constate que quel que soit le nombre de Reynolds considéré, l'erreur relative moyenne de la méthode NIMR ne dépasse pas 1.5% pour la vitesse et 6% pour la pression. Ces résultats sont satisfaisants : le modèle réduit obtenu par la méthode NIMR est capable de reproduire avec pertinence un écoulement à un nombre de Reynolds différent de ceux considérés dans la construction des écoulements d'apprentissage. Bien que la méthode NIMR soit légèrement moins précise que la méthode ITSGM/ROM, elle est beaucoup plus rapide en terme de temps de calcul. En effet, les temps de simulation réels des méthodes ITSGM/ROM, NIMR et HNIMR répertoriés au tableau 4.2, montrent que la méthode NIMR est environ 6 fois plus rapide que la méthode ITSGM/ROM. La version hyper réduite HNIMR présente des gains encore plus importants en étant 30 fois plus rapide que la méthode NIMR et 170 fois plus rapide que la méthode ITSGM/ROM. Cette différence importante de temps de calcul entre l'approche intrusive ITSGM/ROM et les approches non intrusives NIMR et HNIMR s'explique par le coût relativement élevé lié au calcul des coefficients du modèle réduit obtenu par projection de Galerkin.

Méthode	\bar{C}_D	$C_{L,max}$	$C_{L,rms}$	S_t	$\bar{\varepsilon}_u^{\%}$	$\bar{\varepsilon}_p^{\%}$
POD	1.307	0.672	0.474	0.176	0.03426%	0.03631%
NIMR	1.301	0.675	0.480	0.175	0.47239%	3.23544%
ITSGM/ROM	1.304	0.694	0.474	0.176	0.11703%	0.36774%

TABLE 4.1 – Comparaison des coefficients aérodynamiques obtenus par les méthodes POD, ITSGM/ROM et NIMR pour le nombre de Reynolds cible $Re_{cible} = 195$

ITSGM/ROM	NIMR	HNIMR
9.5 sec	1.51 sec	0.05 sec

TABLE 4.2 – Temps de calcul réels en utilisant les méthodes de réduction de modèle ITSGM/ROM, NIMR et HNIMR pour la prédiction de l'écoulement autour d'un cylindre

6. Les résultats du coefficient de traînée moyen \bar{C}_D obtenus par modèle complet montrent qu'il y a un comportement convexe de ce coefficient en fonction de la variation du nombre de Reynolds, avec un minimum compris entre $Re = 160$ et $Re = 230$. Ce comportement a été déjà observé par Fabiane et al. [141].

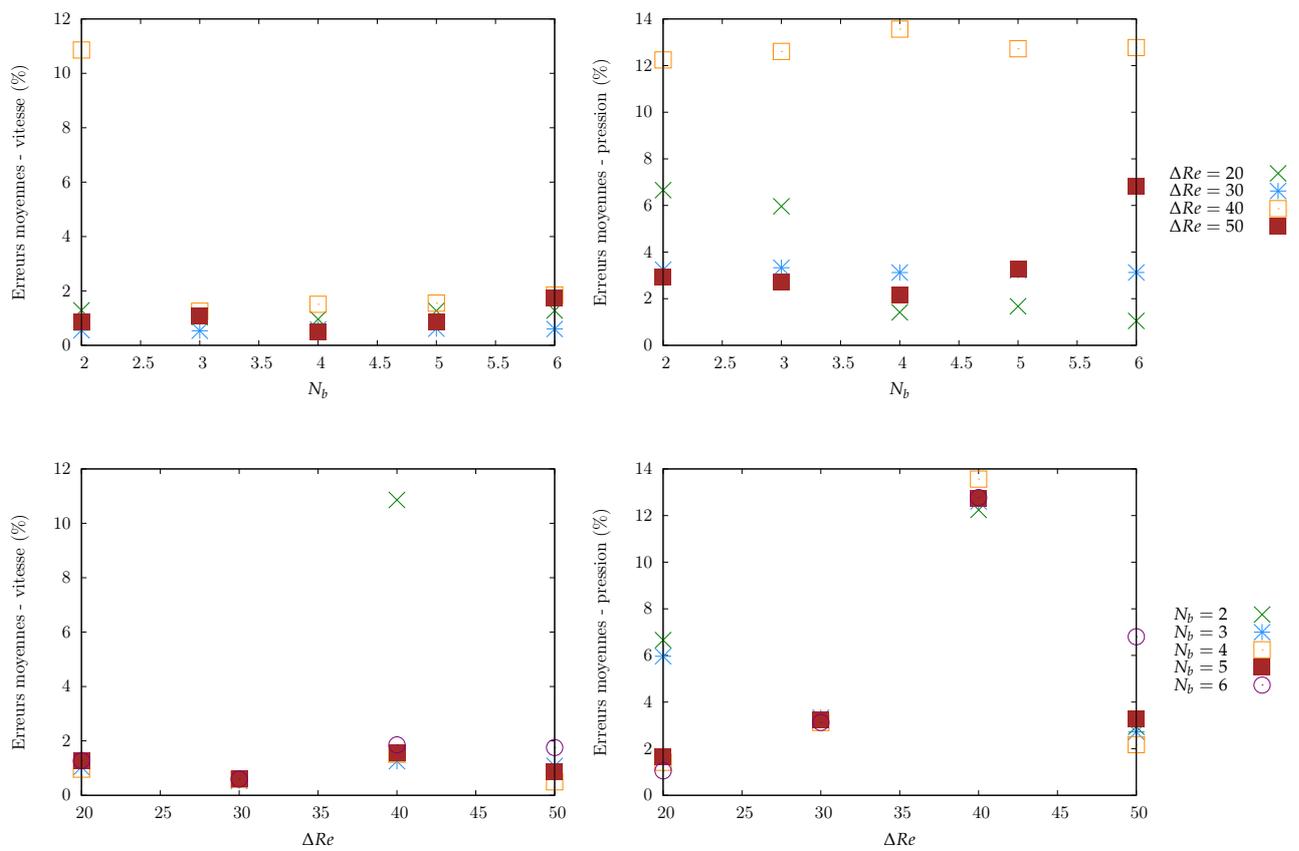


FIGURE 4.5 – Influence du pas de Reynolds ΔRe et du nombre de bases N_b sur la méthode NIMR pour $Re = 160$.

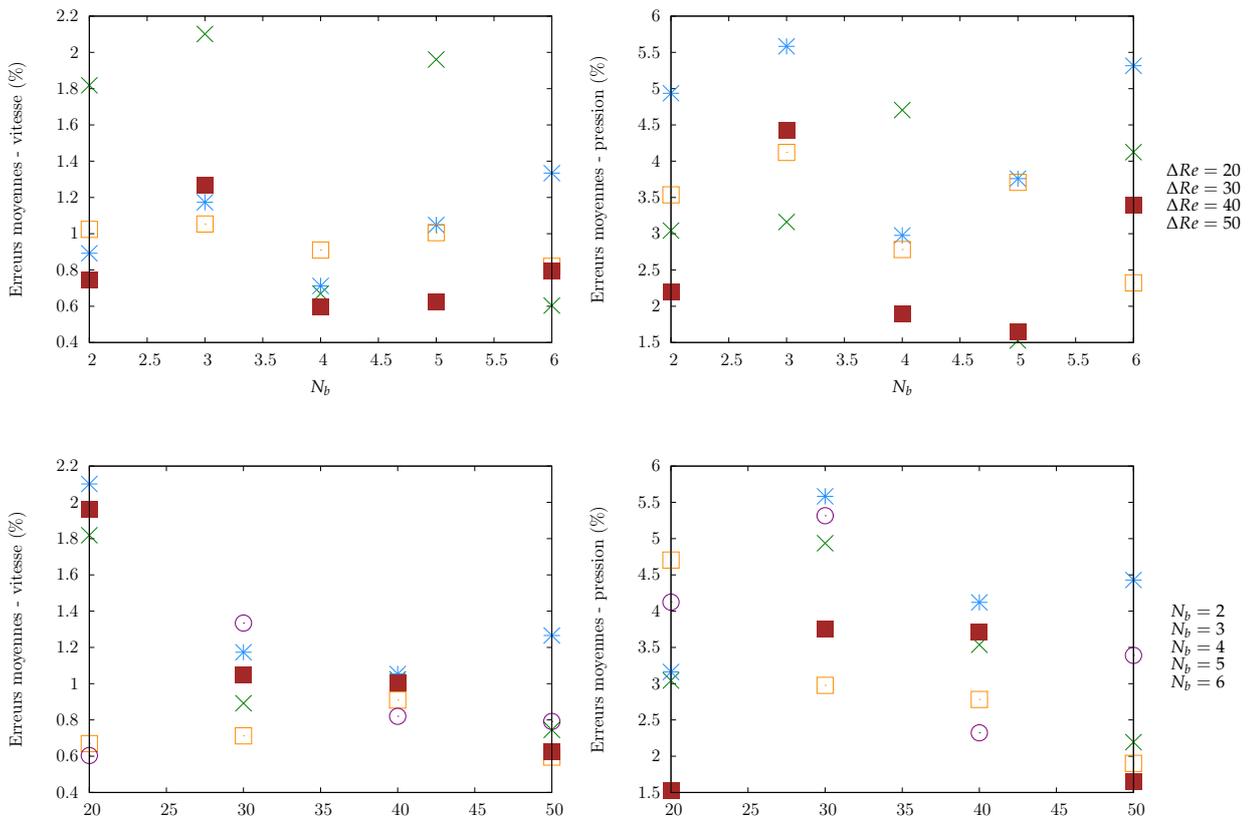


FIGURE 4.6 – Influence du pas de Reynolds ΔRe et du nombre de bases N_b sur la méthode NIMR pour $Re = 280$.

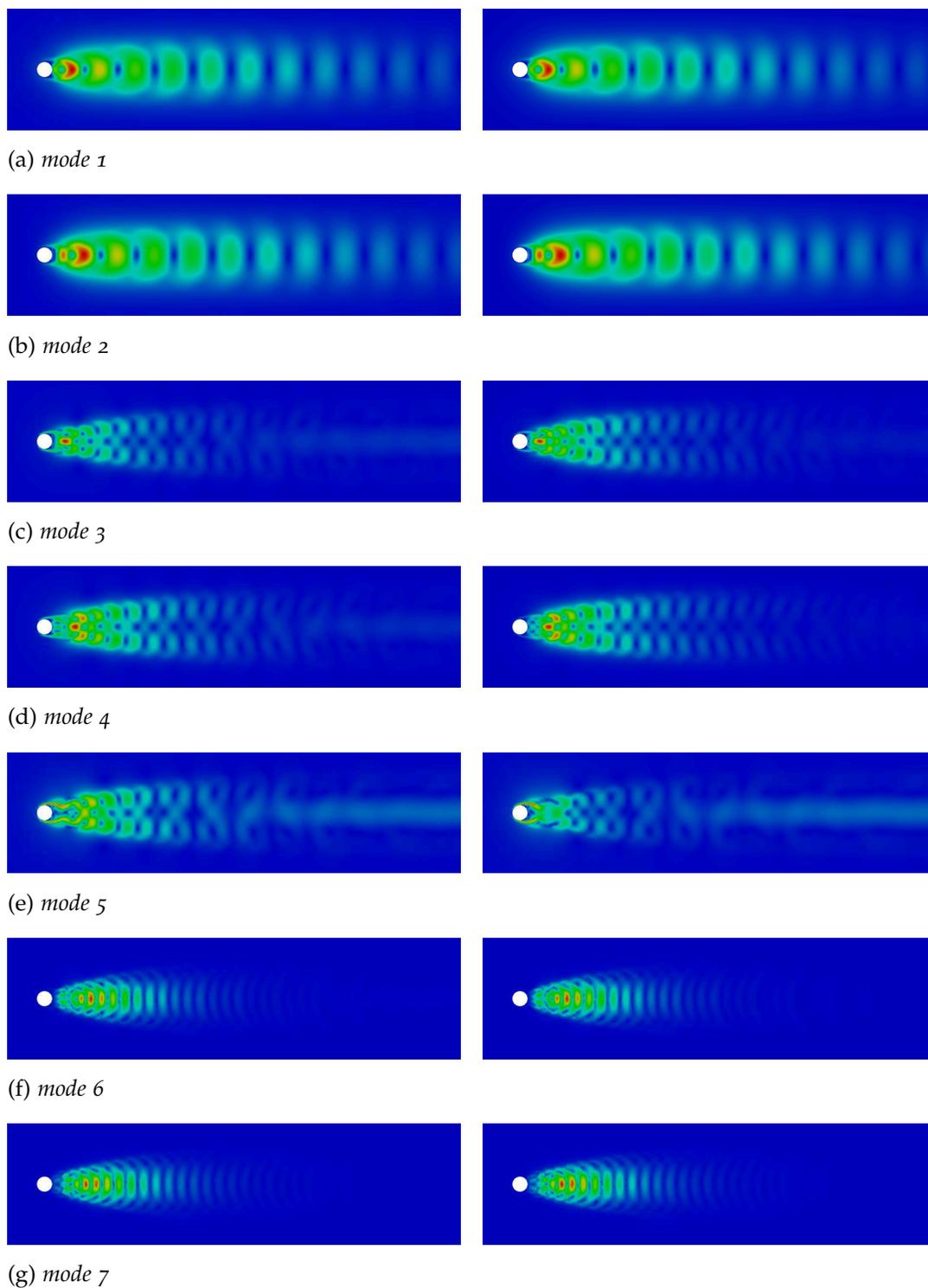


FIGURE 4.7 – Comparaison des modes de vitesse issues de la NIMR (gauche) avec les modes POD (droite) pour $Re_{cible} = 195$.

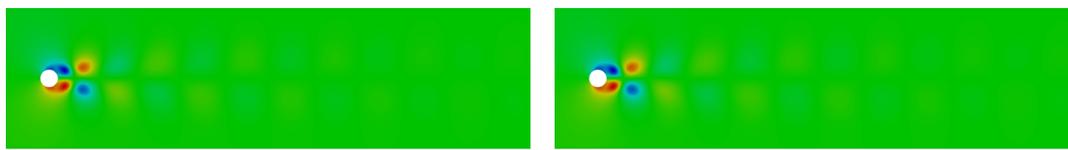
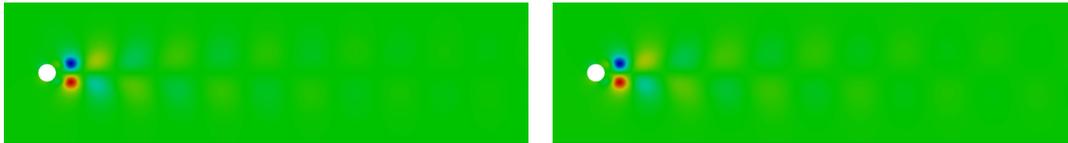
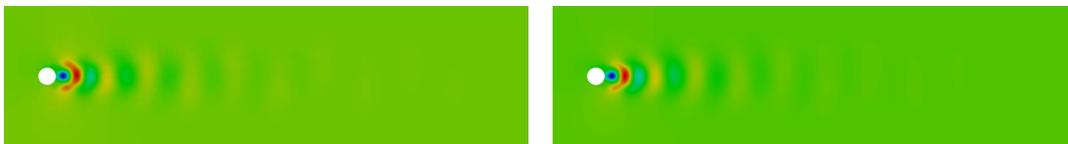
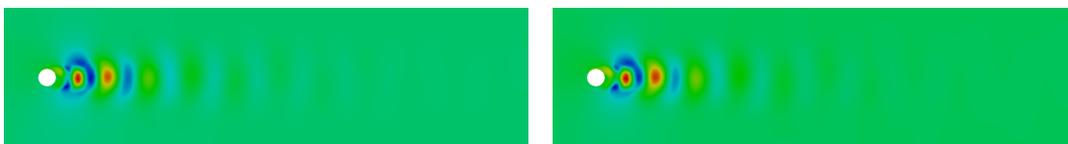
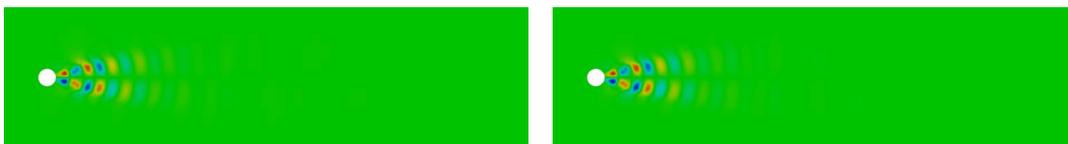
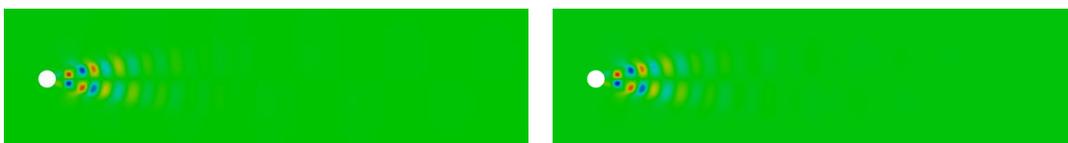
(a) *mode 1*(b) *mode 2*(c) *mode 3*(d) *mode 4*(e) *mode 5*(f) *mode 6*(g) *mode 7*

FIGURE 4.8 – Comparaison des modes de pression issues de la NIMR (gauche) avec les modes POD (droite) pour $Re_{cible} = 195$.

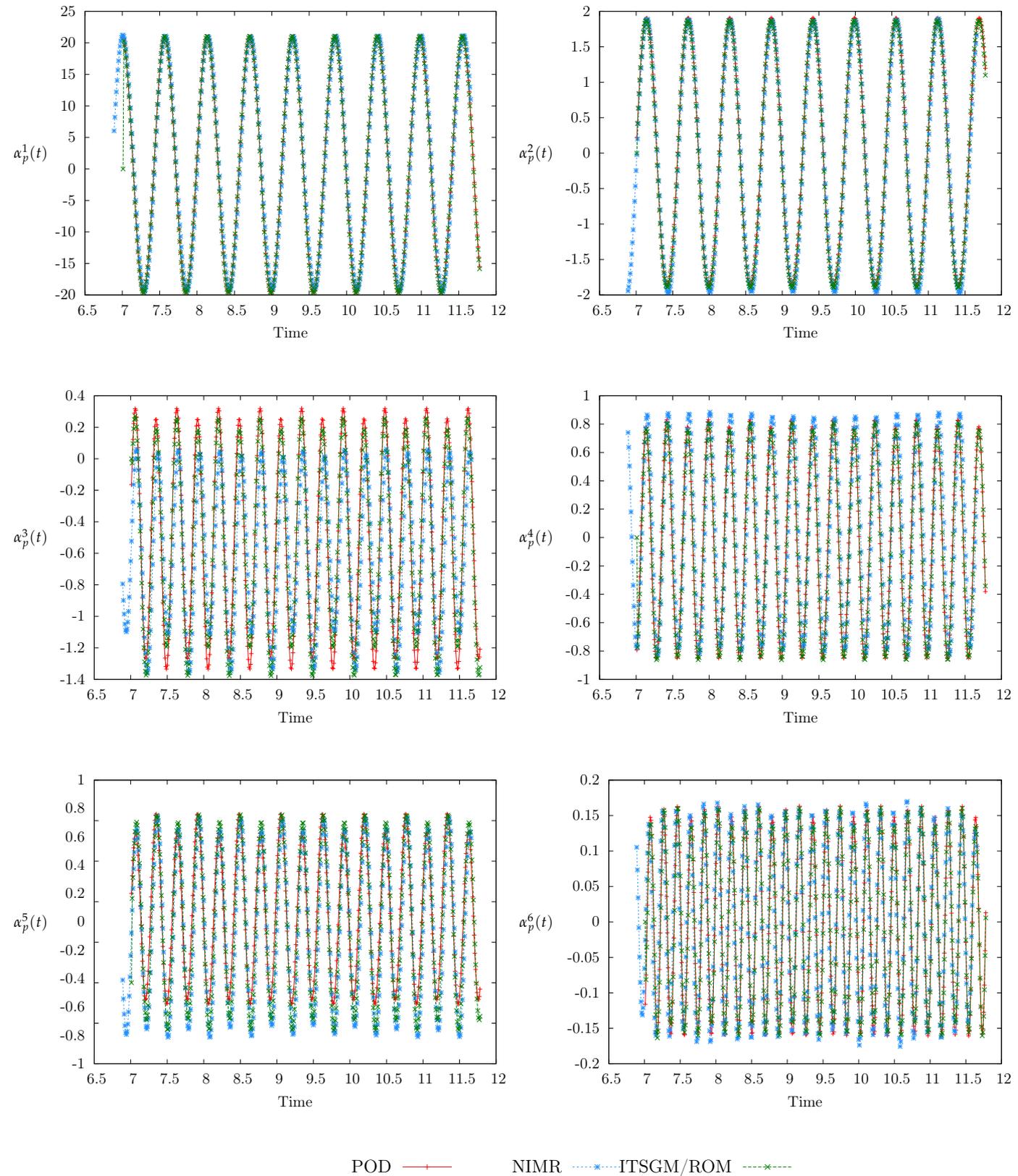


FIGURE 4.10 – Comparaison des coefficients temporels de pression obtenus par ITSGM/ROM et NIMR avec les coefficients temporels POD pour le nombre de Reynolds cible $Re_{cible} = 195$.

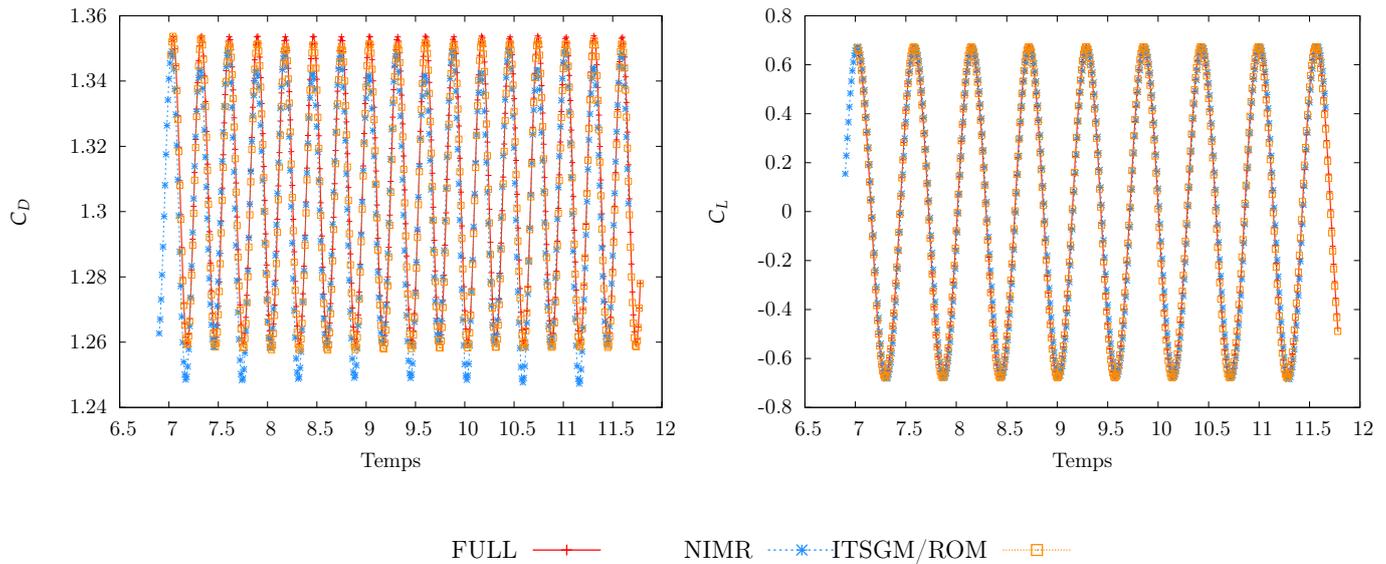


FIGURE 4.11 – Comparaison des coefficients C_D et C_L obtenues par les méthodes ITSGM/ROM et NIMR avec les coefficients du modèle complet pour le nombre de Reynolds cible $Re_{cible} = 195$.

4.6 CONCLUSION DU CHAPITRE

Comme il a été déjà souligné au chapitre 3, en utilisant la méthode ITSGM pour l'interpolation des bases POD, la nécessité de recourir aux équations du modèle complet pour reconstruire la dynamique temporelle met en évidence l'intrusivité de cette approche. En effet, une fois que la base interpolée est calculée, la dynamique temporelle associée au nouveau problème est obtenue en résolvant le système d'équations différentielles ordinaires, issu de la projection de Galerkin du modèle complet sur la base interpolée. Cette approche de réduction de modèle est alors dite intrusive vu sa dépendance aux équations du modèle complet. Il est alors impossible de reconstruire la dynamique temporelle du problème sans avoir accès aux équations de base. Dans le but de contourner ce problème, ce chapitre a été dédié à la construction de la nouvelle méthode d'interpolation non intrusive appelée NIMR (Non Intrusive Model Reduction) et sa variante HNIMR (Hyper Non Intrusive Model Reduction). Cette méthode a été appliquée et validée sur le problème de l'écoulement d'un fluide autour d'un cylindre lorsque le nombre de Reynolds change. Il s'ensuit que les résultats obtenus sont en accord avec les résultats du modèle complet (d'erreur moyenne relative inférieure à 1.5% et 6% respectivement pour la vitesse et la pression). L'approche s'avère également très rapide en temps de calcul. En effet, l'interpolation des bases réduites par la méthode NIMR se fait en moins de 2 secondes et en moins de 0.05 secondes pour l'approche HNIMR. Dans le chapitre suivant, le potentiel des approches de réduction de modèle non intrusive (NIMR et HNIMR), ainsi que des approches intrusives du chapitre précédent (ITSGM, MPS et PGD) sera illustré sur les problèmes de contrôle optimal réduit des équations de réaction diffusion, de Burgers et de Navier-Stokes.

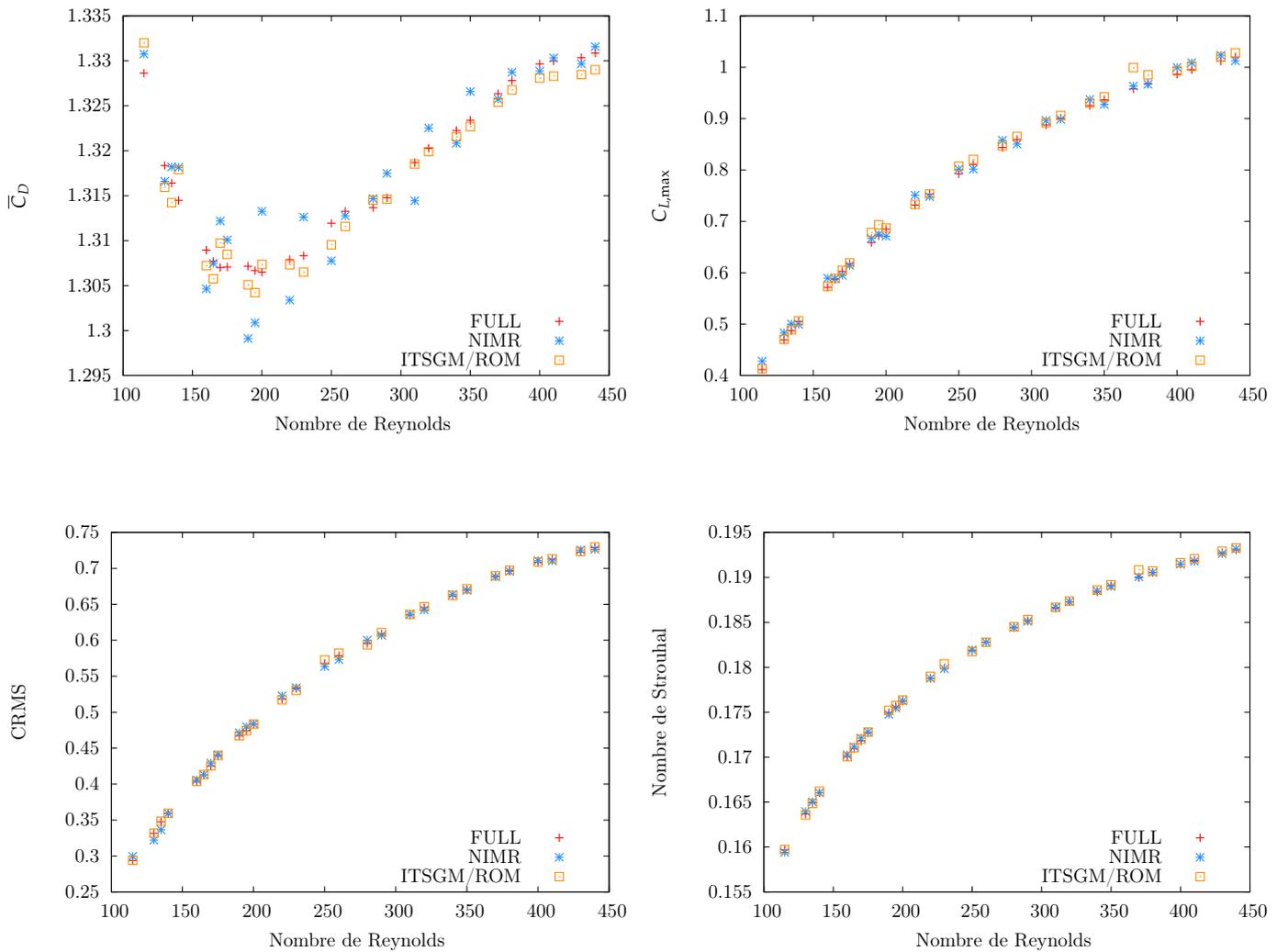


FIGURE 4.12 – Coefficients aérodynamiques pour différentes valeurs du nombre de Reynolds, obtenus par le modèle complet (FULL) et les méthodes ITSGM/ROM et NIMR considérés pour $\Delta Re = 30$ et $N_b = 3$.

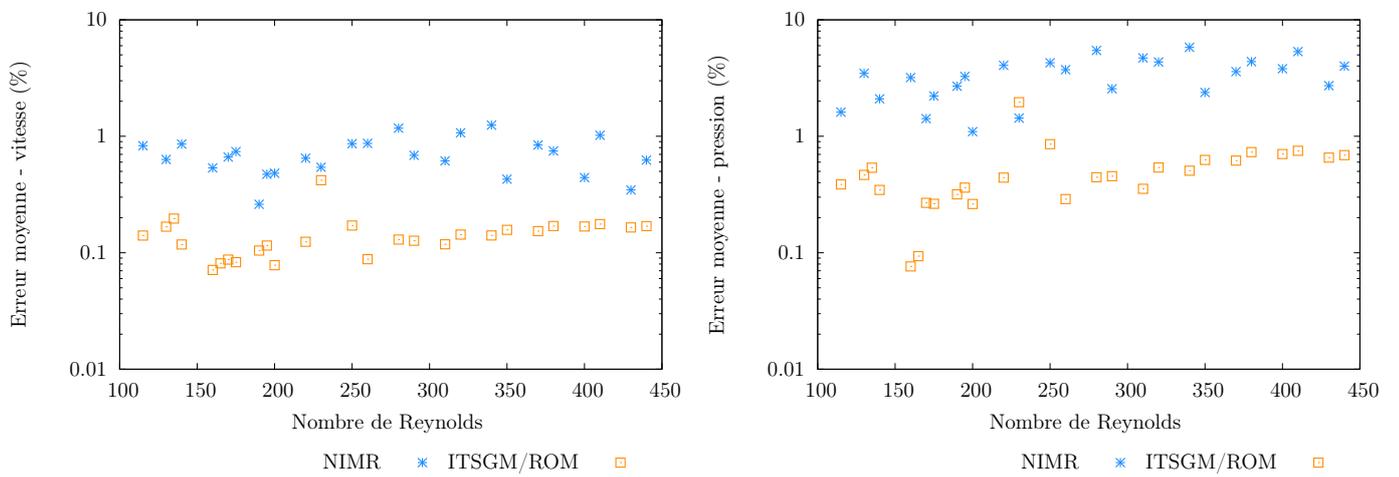


FIGURE 4.13 – Pourcentage d'erreur moyen entre la solution du modèle complet et les solutions obtenues par les méthodes NIMR et ITSGM/ROM pour $\Delta Re = 30$ et $N_b = 3$.

CONTRÔLE OPTIMAL BASÉ SUR DES MODÈLES RÉDUITS ADAPTATIFS

SOMMAIRE

5.1	INTRODUCTION	110
5.2	CONTRÔLE OPTIMAL RÉDUIT AVEC ADAPTATION DE LA BASE RÉDUITE	110
5.2.1	Algorithme de contrôle optimal réduit avec phase d'adaptation	110
5.2.2	Application au contrôle de l'équation de réaction-diffusion	113
5.2.3	Application au contrôle de l'équation de Burgers	122
5.2.4	Application au contrôle de l'équation de Navier-Stokes	128
5.3	CONTRÔLE OPTIMAL RÉDUIT EN UTILISANT LES ALGORITHMES GÉNÉTIQUES . . .	134
5.3.1	Principe des algorithmes génétiques	134
5.3.2	Représentation génétique	135
5.3.3	Opérateurs génétiques	136
5.3.4	Application au contrôle de l'écoulement d'un fluide autour d'un cylindre . .	138
5.4	CONCLUSION DU CHAPITRE	144

5.1 INTRODUCTION

Le contrôle optimal et la réduction de modèle par POD ont été couplés dans plusieurs applications de la physique dans le but de réduire le temps de calcul et les exigences de stockage mémoire. Le contrôle optimal basé sur les équations adjointes d'un modèle réduit (voir section 2.3.2) a notamment été utilisé en mécanique des fluides pour réduire les instabilités qui apparaissent dans le sillage de l'écoulement autour d'un cylindre fixe [142] ou en rotation [143] en agissant respectivement sur la vitesse d'entrée et sur la vitesse de rotation du cylindre ; pour contrôler la taille de la zone de recirculation présente en aval de l'écoulement dans une marche descente en agissant sur l'intensité d'un jet auxiliaire [144, 145] ; pour contrôler l'écoulement autour d'un profil d'aile [146] [147] ; pour contrôler l'écoulement dans une cavité entraînée différentiellement chauffée [148]. L'approche a également été appliquée pour le contrôle des équations paraboliques [149] et des équations de Burgers [150, 151]. Ces différentes applications ont montré que le contrôle par les modèles d'ordre réduit permet un gain significatif en temps de calcul ainsi qu'en stockage mémoire. Cependant, la réduction du temps de calcul dépend fortement de la qualité du modèle réduit et de sa capacité à prédire les directions de la descente avec une précision suffisante. Pour améliorer l'approche du contrôle optimal par modèle réduit, Leibfritz et al. [152] ont combiné la POD avec la NSDP (nonlinear semidefinite programming) et la méthode des régions de confiance IPCTR (interior point constraint trust region). Agarwal et al. [153] et Bergmann et al. [154, 96] s'appuient sur une technique similaire, qui non seulement restreint le modèle réduit à sa région de validité, mais qui fournit un algorithme robuste globalement convergent.

L'objectif de ce chapitre est dans un premier temps de tester les méthodes ITSGM, PGD et MPS dans l'algorithme de contrôle optimal basé sur l'équation adjointe réduite. Après avoir décrit les modifications algorithmiques engendrées par la phase d'adaptation de bases, la robustesse de l'approche est testée pour le contrôle optimal des équations de réactions diffusion non linéaires et puis des équations de Burgers, et enfin l'écoulement autour d'un cylindre régi par les équations de Navier-Stokes. La seconde partie concerne le contrôle optimal en utilisant un algorithme génétique en conjonction avec les méthodes de réduction de modèle non intrusives NIMR et HNIMR. La pertinence de l'approche en terme de précision et de temps de calcul est évaluée pour le problème de contrôle optimal de l'écoulement d'un fluide autour d'un cylindre.

5.2 CONTRÔLE OPTIMAL RÉDUIT AVEC ADAPTATION DE LA BASE RÉDUITE

5.2.1 Algorithme de contrôle optimal réduit avec phase d'adaptation

Le problème de contrôle optimal complet que l'on souhaite résoudre s'écrit

$$\min \mathcal{J}(y, \theta) \text{ sur } (y, \theta) \in V \times K \quad \text{tel que } \mathcal{N}(y, \theta) = 0 \quad (5.1)$$

où K un est ensemble convexe fermé de $Z = \mathbb{R}^{d_p}$ et $V = L^2(0, T; H^1(\Omega))$. Soit θ un paramètre fixe, supposons que pour un paramètre de contrôle $\tilde{\theta}$ voisin de θ la variable d'état $y(\tilde{\theta})$ est approximée dans le sous espace vectoriel de dimension q , engendré par l'ensemble des fonctions de bases $\boldsymbol{\varphi}_\theta = \{\varphi_\theta^1, \dots, \varphi_\theta^q\}$, soit :

$$y(t, x, \tilde{\theta}) \approx y_q(t, x, \tilde{\theta}) = \sum_{k=1}^q \alpha^k(t, \tilde{\theta}) \varphi_\theta^k(x) \quad (5.2)$$

Les coefficients temporels $\tilde{\boldsymbol{\alpha}}(t, \tilde{\theta}) = \{\alpha^1(t, \tilde{\theta}), \dots, \alpha^q(t, \tilde{\theta})\}$ peuvent être déterminés en résolvant le système d'équations différentiels obtenu par projection de Galerkin de l'équation des contraintes sur la base réduite $\boldsymbol{\varphi}_\theta$, soit

$$\left\langle \mathcal{N} \left(\sum_{k=1}^q \alpha^k(t, \tilde{\theta}) \varphi_\theta^k(x), \tilde{\theta} \right), \varphi_\theta^i(x) \right\rangle_{L^2(\Omega)} = 0, \quad \forall i = 1, \dots, q$$

L'équation du modèle réduit ci dessus s'écrit de manière compacte comme suite

$$\mathcal{N}_{\boldsymbol{\varphi}_\theta}(\tilde{\boldsymbol{\alpha}}, \tilde{\theta}) = 0 \quad (5.3)$$

Pour $\theta \in K$ fixé, on définit l'application

$$\begin{aligned} \mathcal{F}_{\boldsymbol{\varphi}_\theta} : K &\longrightarrow \mathcal{T} \\ \tilde{\theta} &\longrightarrow \tilde{\boldsymbol{\alpha}} \end{aligned}$$

où $\mathcal{T} = L^2(0, T)$ et $\tilde{\boldsymbol{\alpha}}$ est solution du modèle réduit $\mathcal{N}_{\boldsymbol{\varphi}_\theta}(\tilde{\boldsymbol{\alpha}}, \tilde{\theta}) = 0$. L'application $\mathcal{F}_{\boldsymbol{\varphi}_\theta}$ permet d'obtenir à partir de la base réduite $\boldsymbol{\varphi}_\theta$ construite pour le paramètre θ , la solution du modèle réduit pour tout paramètre $\tilde{\theta}$ dans le voisinage de $\theta \in K$. Cette solution est donc obtenue par résolution du modèle réduit (5.3). En substituant (5.2) dans le problème de contrôle optimal (5.1) et en utilisant l'expression des contraintes réduites (5.3), on peut formuler le problème de contrôle optimal réduit suivant

$$\min \mathcal{J}_{\boldsymbol{\varphi}_\theta}(\tilde{\boldsymbol{\alpha}}, \tilde{\theta}) \quad \text{sur} \quad (\tilde{\boldsymbol{\alpha}}, \tilde{\theta}) \in \mathcal{T} \times K \quad (5.4)$$

tel que

$$\tilde{\boldsymbol{\alpha}} = \mathcal{F}_{\boldsymbol{\varphi}_\theta}(\tilde{\theta}) \quad \text{c'est à dire} \quad \mathcal{N}_{\boldsymbol{\varphi}_\theta}(\tilde{\boldsymbol{\alpha}}, \tilde{\theta}) = 0 \quad (5.5)$$

On rappelle que dans ce cas (voir section 2.4), la fonction de Lagrange associée à la contrainte réduite (5.5) est définie par

$$\begin{aligned} \mathcal{L}_{\boldsymbol{\varphi}_\theta} : \mathcal{T} \times Z \times \mathcal{T} &\longrightarrow \mathbb{R} \\ (\tilde{\boldsymbol{\alpha}}, \tilde{\theta}, \boldsymbol{\beta}) &\longrightarrow \mathcal{L}(\tilde{\boldsymbol{\alpha}}, \tilde{\theta}, \boldsymbol{\beta}) = \mathcal{J}_{\boldsymbol{\varphi}_\theta}(\tilde{\boldsymbol{\alpha}}, \tilde{\theta}) + \langle \boldsymbol{\beta}, \mathcal{N}_{\boldsymbol{\varphi}_\theta}(\tilde{\boldsymbol{\alpha}}, \tilde{\theta}) \rangle_{\mathcal{T}} \end{aligned}$$

où $\boldsymbol{\beta}$ est un multiplicateur de Lagrange. Les conditions d'optimalités pour le problème d'optimisation sous contrainte réduite (5.4)(5.5) sont données par

$$\begin{aligned} \mathcal{N}_{\boldsymbol{\varphi}_\theta}(\tilde{\boldsymbol{\alpha}}, \tilde{\theta}) = 0 &\Rightarrow \text{équation des contraintes réduite} \\ \partial_{\tilde{\boldsymbol{\alpha}}} \mathcal{N}_{\boldsymbol{\varphi}_\theta}(\tilde{\boldsymbol{\alpha}}, \tilde{\theta})^* \boldsymbol{\beta} + \partial_{\tilde{\boldsymbol{\alpha}}} \mathcal{J}_{\boldsymbol{\varphi}_\theta}(\tilde{\boldsymbol{\alpha}}, \tilde{\theta}) &= 0 \Rightarrow \text{équation adjointe} \quad (5.6) \\ \left\langle \partial_{\tilde{\theta}} \mathcal{J}_{\boldsymbol{\varphi}_\theta}(\tilde{\boldsymbol{\alpha}}, \tilde{\theta}) + \partial_{\tilde{\theta}} \mathcal{N}_{\boldsymbol{\varphi}_\theta}(\tilde{\boldsymbol{\alpha}}, \tilde{\theta})^* \boldsymbol{\beta}, \bar{\theta} - \tilde{\theta} \right\rangle_Z &\geq 0 \quad \forall \bar{\theta} \in K \Rightarrow \text{inéquation du minimum de } \mathcal{J}_{\boldsymbol{\varphi}_\theta} \end{aligned}$$

et les directions de descente sont évaluées par la relation suivante

$$d_\theta = -\partial_{\tilde{\theta}} \mathcal{L}_{\varphi_\theta}(\tilde{\alpha}, \tilde{\theta}, \beta) = -\partial_{\tilde{\theta}} \mathcal{J}_{\varphi_\theta}(\tilde{\alpha}, \tilde{\theta}) - \partial_{\tilde{\theta}} \mathcal{N}_{\varphi_\theta}(\tilde{\alpha}, \tilde{\theta})^* \beta \quad (5.7)$$

Les étapes de résolution de ce problème d'optimisation sont résumés dans l'algorithme 9. Contrairement à cette approche où la base réduite est supposée fixe, la stratégie développée avec phase d'adaptation de bases, consiste à résoudre une séquence de problèmes d'optimisation de type (5.4)(5.5) entrecoupée par une phase d'adaptation de la base réduite. Autrement dit, pour un paramètre candidat $\theta^{(n)}$ fourni par l'algorithme de contrôle réduit à l'itération n , la base réduite $\varphi_{\theta^{(n)}}$ est fixée et une solution $\theta_\infty^{(n)}$ du problème (5.4)(5.5) associé est trouvée. Dans une seconde étape, la nouvelle base réduite $\varphi_{\theta^{(n+1)}}$ est mise à jour en utilisant la méthode ITSGM ou la méthode PGD de telle sorte qu'elle soit la plus adaptée au paramètre convergé $\theta^{(n+1)} = \theta_\infty^{(n)}$. Les étapes du contrôle optimal réduit avec phase d'adaptation sont illustrées dans l'algorithme 16.

Algorithme 16 : Algorithme de contrôle optimal réduit avec adaptation de bases.

```

1 initialiser  $k = 0$ , et choisir  $\theta^{(k)}$  arbitraire;
2 résoudre le modèle réduit (5.5)  $\rightarrow \tilde{\alpha}^{(0)}$ ;
3 prendre  $\tilde{\alpha}^{(-1)} = \tilde{\alpha}^{(0)}$ ;
4 while  $|\mathcal{J}_{\varphi_{\theta^{(k)}}}(\tilde{\alpha}^{(k)}, \theta^{(k)}) - \mathcal{J}_{\varphi_{\theta^{(k)}}}(\tilde{\alpha}^{(k-1)}, \theta^{(k-1)})| \geq \epsilon |\mathcal{J}_{\varphi_{\theta^{(k)}}}(\alpha_m^{(k)}, \theta^{(k)})|$  do
5   initialiser  $l = 0$ ,  $\tilde{\theta}^{(l)} = \theta^{(k)}$ ;
6   while  $|\mathcal{J}_{\varphi_{\theta^{(k)}}}(\tilde{\alpha}^{(l)}, \tilde{\theta}^{(l)}) - \mathcal{J}_{\varphi_{\theta^{(k)}}}(\tilde{\alpha}^{(l-1)}, \tilde{\theta}^{(l-1)})| \geq \epsilon_r |\mathcal{J}_{\varphi_{\theta^{(k)}}}(\tilde{\alpha}^{(l)}, \tilde{\theta}^{(l)})|$  do
7      $l \leftarrow l + 1$ ;
8     pour  $\tilde{\theta} = \tilde{\theta}^{(l-1)}$ , résoudre le modèle réduit (5.5)  $\rightarrow \tilde{\alpha}^{(l)}$ ;
9     pour  $\tilde{\alpha} = \tilde{\alpha}^{(l)}$  et  $\tilde{\theta} = \tilde{\theta}^{(l-1)}$ , résoudre le problème adjoint réduit (5.6)  $\rightarrow \zeta_m^{(l)}$ ;
10    calculer la direction de descente  $d^{(l)}$  par l'équation (5.7);
11    calculer le nouveau paramètre de contrôle par  $\tilde{\theta}^{(l)} = P_K(\tilde{\theta}^{(l-1)} + \sigma_l d^{(l)})$  tel
    que  $\sigma_l > 0$  est déterminé par la règle d'Armijo;
12  end
13   $\theta^{(k)} \leftarrow \tilde{\theta}^{(l)}$ ;
14  adapter la base  $\varphi_{\theta^{(k)}}$  pour le nouveau paramètre  $\theta^{(k)}$  en utilisant la méthode
    ITSGM ou la méthode PGD;
15   $k \leftarrow k + 1$ ;
16 end

```

Dans le cas où la méthode PGD est choisie pour actualiser la base réduite (étape 14), un seul enrichissement PGD à chaque itération est suffisant pour obtenir une précision satisfaisante des directions de descente et par suite à assurer une convergence rapide de l'algorithme de contrôle réduit. Après plusieurs itérations de l'algorithme de contrôle réduit, la fonctionnelle et les paramètres de contrôle stagnent et l'algorithme 16 agit exactement comme l'algorithme PGD classique [130]. En effet, dans ce cas les paramètres de contrôles

sont fixes et faire une itération supplémentaire de l'algorithme de contrôle réduit revient à faire un enrichissement PGD supplémentaire.

Les méthodes ITSGM et MPS sont des méthodes a posteriori car elles dépendent des snapshots préalablement calculés et stockés. La précision maximale des solutions obtenues en utilisant ces méthodes est donc déterminée par le choix initial de la distribution des paramètres d'apprentissage, où les snapshots utilisés pour construire les bases POD sont calculés. Dans une boucle de contrôle réduit, les méthodes MPS et ITSGM atteindront leurs précision maximale après un certain nombre d'itérations. Au-delà de ces itérations, les paramètres de contrôle ainsi que la fonctionnelle coût vont stagner, signifiant qu'un contrôle optimal approximatif est atteint. Un moyen possible pour améliorer la précision des résultats obtenus par les méthodes ITSGM et MPS est d'utiliser la méthode PGD dans la phase d'adaptation lorsque les indices de stagnation des paramètres de contrôle apparaissent. Dans ce qui suit, ces combinaisons de méthodes seront mentionnées sous les acronymes ITSGM/PGD et MPS/PGD. Dans la suite de ce chapitre, ces méthodes d'adaptation de bases réduites seront testées dans l'algorithme de contrôle optimal réduit (Algo. 16) pour le contrôle optimal des équations de réaction-diffusion, des équations de Burgers¹ et des équations de Navier-Stokes. Dans le cas où le contrôle est effectué pour atteindre une solution cible u , la formule utilisée pour le calcul de l'erreur relative à l'issue du contrôle entre la solution convergée u_∞ et la solution cible est donnée par

$$\varepsilon_{\%}^T = 100 \times \frac{\left(\int_0^T \|u_\infty - u\|_{L^2(\Omega)}^2 dt \right)^{\frac{1}{2}}}{\left(\int_0^T \|u\|_{L^2(\Omega)}^2 dt \right)^{\frac{1}{2}}}$$

5.2.2 Application au contrôle de l'équation de réaction-diffusion

a) Problème de contrôle optimal complet

Soient $\Omega \subset \mathbb{R}^2$ un domaine borné et $\Gamma = \Gamma_D \cup \Gamma_N$ sa frontière, le problème de réaction diffusion étudié s'écrit

$$\begin{cases} \partial_t u - \nu \Delta u + \frac{1}{2} u^2 = f & \text{in } \Omega \times]0, T] \\ u = g & \text{on } \Gamma_D \times]0, T] \\ \partial_n u = 0 & \text{on } \Gamma_N \times]0, T] \\ u(0) = u_0 & \text{in } \Omega. \end{cases} \quad (5.8)$$

où $u_0 \in L^2(\Omega)$ est une condition initiale et $f \in L^2(]0, T], L^2(\Omega))$ désigne les forces extérieures appliqués sur le domaine Ω . le terme f est défini par la fonction Gaussienne de

¹. Les résultats concernant le contrôle optimal réduit de l'équation de réaction-diffusion et l'équation de Burgers ont fait l'objet de la publication [155].

rayon $\tilde{\sigma}$ centrée au point (c_x, c_y) comme suit

$$f(t, x, y, c_x, c_y) = \lambda \cos\left(\frac{2\pi}{T_0}t\right) \exp\left(-\frac{(x - c_x)^2 + (y - c_y)^2}{2\tilde{\sigma}^2}\right) \quad (5.9)$$

La fonction g des conditions aux limites de Dirichlet est donnée par l'expression

$$g(t, x) = \eta(t) \cos(x^2 + y^2) \quad (5.10)$$

Dans la suite, on s'intéresse aux paramètres de contrôle c_x , c_y , $\tilde{\sigma}$ et ν . Nous allons présenter trois cas tests du contrôle réduit de l'équation de réaction-diffusion. Le premier cas test consiste à agir sur c_x et c_y coordonnées du centre de la fonction Gaussienne du terme source f . Dans le 2^{ème} cas, le contrôle est effectué en faisant varier le coefficient de diffusion ν et le rayon de la fonction Gaussienne $\tilde{\sigma}$. Finalement, dans le 3^{ème} cas, tous ces paramètres seront variés.

On considère le problème d'optimisation sous contraintes suivant

$$\min \mathcal{J}(u, \theta) \text{ sur } (u, \theta) \in V \times K \quad \text{tel que } u \text{ satisfait l'équation (5.8)} \quad (5.11)$$

où $V = L^2(0, T, H_{g, \Gamma_D}^1(\Omega))$ et $K \subset Z$ avec $Z = \mathbb{R}^{d_p}$. Pour $\kappa > 0$ un paramètre de régularisation, et \hat{u} la solution cible associée à un paramètre $\hat{\theta} \in K$, on définit la fonctionnelle objectif $\mathcal{J} : V \times Z \rightarrow \mathbb{R}$ par

$$\mathcal{J}(u, \theta) = \frac{1}{2} \int_0^T \int_{\Omega_c} |u(T) - \hat{u}(T)|^2 dx dt + \frac{1}{2} \int_{\Omega_c} |u - \hat{u}|^2 dx + \frac{\kappa}{2} |\theta|^2 \quad (5.12)$$

où Ω_c est le domaine de contrôle décrit dans la figure 5.1.

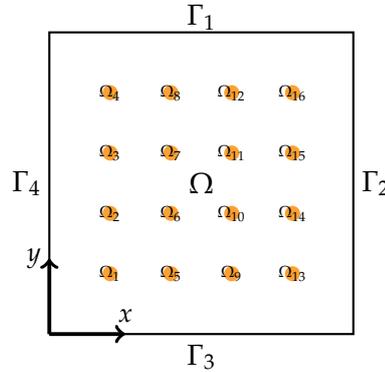


FIGURE 5.1 – Domaine spatiale Ω et sous domaines $\Omega_c = \bigcup_{i=1}^{16} \Omega_i$ sur lesquelles le contrôle est effectué.

b) Écriture du problème de contrôle optimal réduit

Supposons maintenant que $u(\theta)$ et \hat{u} sont approximés dans deux sous espaces vectoriels engendrés respectivement par les familles de vecteurs de base POD $\varphi = \{\varphi^1, \dots, \varphi^q\}$ et $\hat{\varphi} = \{\hat{\varphi}^1, \dots, \hat{\varphi}^{\hat{q}}\}$, soient :

$$u(t, x, \theta) \approx \sum_{k=1}^q \alpha^k(t, \theta) \varphi^k(x) \quad \hat{u}(x, t) \approx \sum_{j=1}^{\hat{q}} \hat{\alpha}^j(t) \hat{\varphi}^j(x) \quad (5.13)$$

Le modèle réduit de l'équation de réaction-diffusion s'écrit

$$\begin{cases} \sum_{j=1}^q \mathbf{M}_{ij} \frac{d\alpha^j}{dt} + \nu \sum_{j=1}^q \mathbf{R}_{ij} \alpha^j + \frac{1}{\epsilon} \sum_{j=1}^q \mathbf{B}_{ij} \alpha^j + \frac{1}{2} \sum_{j,k=1}^q \mathbf{N}_{ijk} \alpha^j \alpha^k = \mathbf{F}_i + \frac{1}{\epsilon} \mathbf{G}_i \\ \sum_{j=1}^q \mathbf{M}_{ij} \alpha^j(0) = \int_{\Omega} u_0 \varphi_i dx \\ \forall i = 1, \dots, q \end{cases} \quad (5.14)$$

où $\epsilon \rightarrow 0$ est le paramètre de pénalisation des conditions aux limites [95] (voir chapitre 2, sous section 2.3.2). Les coefficients du modèle réduit \mathbf{M}_{ij} , \mathbf{R}_{ij} , \mathbf{B}_{ij} , \mathbf{q} , \mathbf{N}_{ijk} , \mathbf{F}_i et \mathbf{G}_i issus de la projection de Galerkin sont calculés comme suit

$$\begin{aligned} \mathbf{M}_{ij} &= \int_{\Omega} \varphi_j \varphi_i dx & \mathbf{N}_{ijk} &= \int_{\Omega} \varphi_j \varphi_k \varphi_i dx & \mathbf{F}_i &= \int_{\Omega} f \varphi_i dx \\ \mathbf{R}_{ij} &= \int_{\Omega} \nabla \varphi_j \nabla \varphi_i dx & \mathbf{B}_{ij} &= \int_{\Gamma} \varphi_j \varphi_i d\sigma & \mathbf{G}_i &= \int_{\Gamma} g \varphi_i d\sigma \end{aligned}$$

En utilisant le modèle d'ordre réduit (5.14), on dérive le modèle réduit adjoint associé suivant

$$\begin{cases} \sum_{i=1}^q \mathbf{M}_{ij} \frac{d\beta^i}{dt} - \nu \sum_{i=1}^q \mathbf{R}_{ij} \beta^i - \sum_{i,k=1}^q \mathbf{N}_{ijk} \alpha^k \beta^i = \sum_{i=1}^q \mathbf{I}_{ji} \alpha^i - \sum_{i=1}^{\hat{q}} \mathbf{L}_{ji} \hat{\alpha}^i \\ \sum_{i=1}^q \mathbf{M}_{ij} \beta^i(T) = \sum_{i=1}^{\hat{q}} \mathbf{L}_{ji} \hat{\alpha}^i(T) - \sum_{i=1}^q \mathbf{I}_{ji} \alpha^i(T) \\ \forall j = 1, \dots, q \end{cases}$$

où \mathbf{I}_{kj} sont les coefficients de la fonctionnelle coût réduite obtenus en injectant les approximations (5.13) dans la fonctionnelle objectif (5.12). Ces coefficients sont obtenus par les identités suivantes

$$\mathbf{I}_{kj} = \int_{\Omega_c} \varphi^k \varphi^j dx \quad \mathbf{L}_{kl} = \int_{\Omega_c} \varphi^k \hat{\varphi}^l dx$$

Finalement, en utilisant le problème adjoint réduit, on détermine les directions de descente dans les directions des variables de contrôle c_x , c_y , ν et $\tilde{\sigma}$ comme suit

$$d_{c_x} = -\frac{\lambda}{\tilde{\sigma}^2} \sum_{j=1}^q \int_0^T \cos\left(\frac{2\pi}{T_0} t\right) \beta^j dt \int_{\Omega} \varphi^j(x - c_x) \exp\left(-\frac{(x - c_x)^2 + (y - c_y)^2}{2\tilde{\sigma}^2}\right) dx - \kappa c_x$$

$$d_{c_y} = -\frac{\lambda}{\tilde{\sigma}^2} \sum_{j=1}^q \int_0^T \cos\left(\frac{2\pi}{T_0} t\right) \beta^j dt \int_{\Omega} \varphi^j(x - c_y) \exp\left(-\frac{(x - c_x)^2 + (y - c_y)^2}{2\tilde{\sigma}^2}\right) dx - \kappa c_y$$

$$d_{\tilde{\sigma}} = -\lambda \sum_{j=1}^q \int_0^T \cos\left(\frac{2\pi}{T_0} t\right) \beta^j dt \int_{\Omega} \frac{(x - c_x)^2 + (y - c_y)^2}{\tilde{\sigma}^3} \exp\left(-\frac{(x - c_x)^2 + (y - c_y)^2}{2\tilde{\sigma}^2}\right) \varphi_j(x) dx - \kappa \tilde{\sigma}$$

$$d_{\nu} = -\sum_{i=1}^q \sum_{j=1}^q \mathbf{R}_{ij} \int_0^T \alpha^i \beta^j dt - \kappa \nu$$

c) Test 1 : contrôle de $\theta = (c_x, c_y)$

Pour ce cas test, l'équation de réaction-diffusion (5.8) est considérée en choisissant $\Gamma_D = \Gamma_1 \cup \Gamma_2 \cup \Gamma_3 \cup \Gamma_4$ et $\Gamma_N = \emptyset$. Le coefficient de diffusion ν est choisi égal à 0.01, $\eta(t) = \frac{1}{3}$, $\lambda = \sqrt{2}$, $T_0 = 2.5$ et $\tilde{\sigma} = 0.15$. Dans ce premier test, l'objectif est de contrôler la solution u à l'intérieur du sous domaine Ω_c en agissant, par l'intermédiaire du paramètre de contrôle $\theta = (c_x, c_y)$, sur le centre de la fonction Gaussienne (5.9). Pour la mise en œuvre du contrôle optimal réduit avec phase d'adaptation, 16 points uniformément répartis dans le domaine $K = \Omega$ ont été considérés afin de construire un échantillonnage Y_θ du paramètre de contrôle θ (figure 5.2). Une base POD φ_θ de taille 6 pour chaque paramètre $\theta \in Y_\theta$ est calculée en utilisant un ensemble de 400 snapshots uniformément répartis sur l'intervalle $]0, T]$ avec $T = 5$. La réunion de tous ces snapshots paramétrés est ensuite considérée pour la construction de la base MPS/POD (i.e., 1600 snapshots ont été utilisés). Le modèle complet permettant d'avoir les snapshots a été résolu par éléments finis (Fenics) sur un maillage non uniforme composé de 5511 degrés de liberté, où le pas de temps considéré est $\delta t = 0.005$.

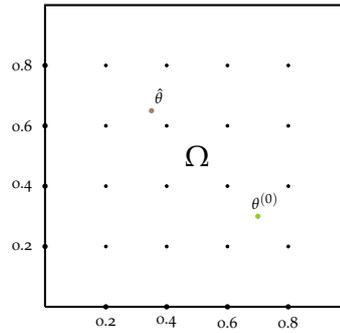


FIGURE 5.2 – Distribution des paramètres de contrôles θ dans le domaine des solutions admissibles.

Soit $\hat{u} = u(\hat{\theta})$ la solution du problème (5.8) construite pour le paramètre cible $\hat{\theta} = (0.35, 0.65)$ que l'on souhaite atteindre par l'algorithme de contrôle optimal réduit. En prenant comme point de départ de l'algorithme de contrôle le paramètre $\theta^{(0)} = (0.7, 0.3)$, l'objectif est de retrouver le paramètre de contrôle cible $\hat{\theta}$ qui minimise la fonctionnelle (5.12). Les solutions en trois instants associés aux paramètres initiaux et cible sont tracées sur la figure 5.3. On constate clairement l'influence de θ sur la solution $u(\theta)$ qui apparaît comme une excitation externe localement appliquée dans le domaine Ω .

Étant donné les notions locales sur lesquelles la méthode d'interpolation ITSGM est fondée, et afin d'obtenir une bonne approximation par cette interpolation, il est préférable de sélectionner uniquement les bases POD correspondants à des paramètres dans un voisinage assez petit du paramètre cible. Ainsi dans notre cas, seulement les 4 bases POD voisines sont considérées. On note également que pour la méthode MPS l'adaptation de la base à chaque itération n'est pas nécessaire, car la base MPS/POD est par hypothèse construite de façon qu'elle soit valide pour tous les paramètres dans Y_θ .

La figure 5.4 montre pour chaque méthode utilisée dans le contrôle réduit l'évolution

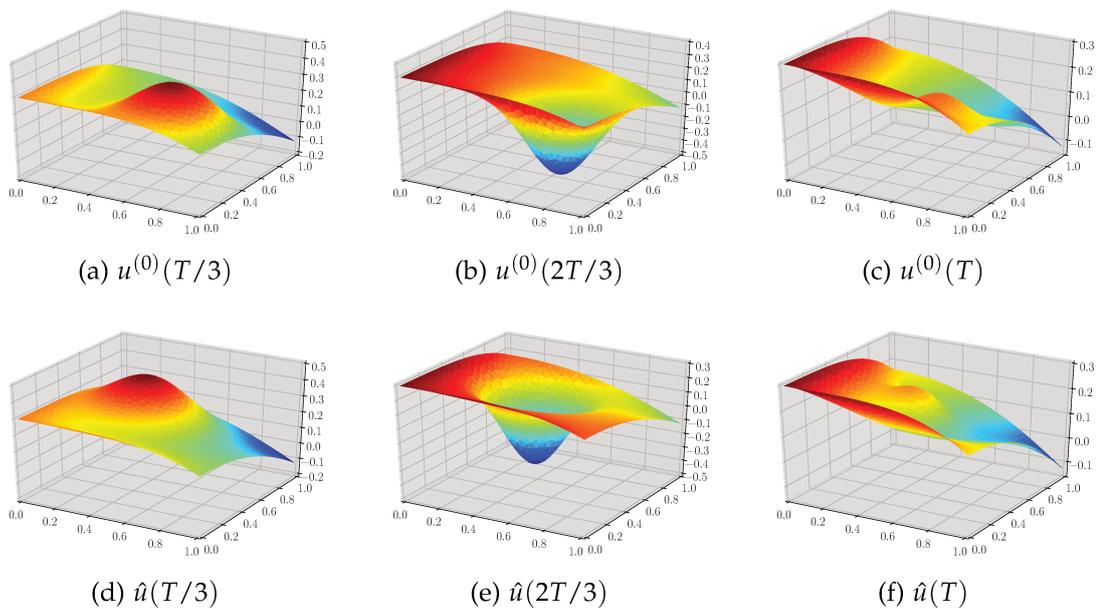


FIGURE 5.3 – Représentation de la solution cible \hat{u} et de la solution initiale $u^{(0)}$ aux instants $T/3$, $2T/3$ et T .

de la fonctionnelle coût. En terme de convergence de l'algorithme de contrôle réduit, la méthode PGD se comporte mieux que les méthodes ITSGM et MPS. À l'itération 9, le contrôle réduit par PGD donne une approximation suffisante des paramètres cibles (c_x, c_y) . Au delà de l'itération 9, les paramètres de contrôle stagnent et la PGD continue à minimiser l'erreur d'approximation de la solution jusqu'à l'itération 18, où la fonctionnelle commence à se stabiliser, indiquant que la méthode PGD a atteint sa précision maximale. Pour les méthodes ITSGM et MPS, la précision maximale est atteinte aux itérations 8 et 10 respectivement. Cela se traduit par la stagnation de la fonctionnelle et des paramètres de contrôle. La stagnation de la fonctionnelle indique que la méthode utilisée pour l'adaptation des modèles réduits a atteint sa précision maximale. Étant donné que la méthode PGD apporte une meilleure précision par rapport aux méthodes ITSGM et MPS, une amélioration de la précision obtenue par ces dernières peut être envisagée en les combinant avec la méthode PGD. Autrement dit, la PGD est appliquée à partir de l'itération où la fonctionnelle pour les méthodes ITSGM et MPS ne varie plus. Ceci entraîne une chute de la fonctionnelle ce qui se traduit par une augmentation de la précision de la solution. Les figures 5.5 montrent que l'algorithme de contrôle optimal réduit en utilisant les méthodes ITSGM, PGD, MPS ou encore les variantes ITSGM/PGD, MPS/PGD, permet de retrouver le paramètre cible $\hat{\theta}$. Le temps CPU est très petit (rapport entre 94 et 134 par rapport au modèle complet) comparé à celui nécessaire dans le cas où le problème de contrôle serait résolu par le modèle complet (tableau 5.1). D'autre part, l'erreur relative entre la solution cible et la solution convergée obtenue par l'algorithme de contrôle réduit, est petite (entre 0.3% et 6%), ce qui prouve la robustesse des méthodes d'adaptation de modèles réduit utilisées à l'intérieur de l'algorithme réduit pour cet exemple.

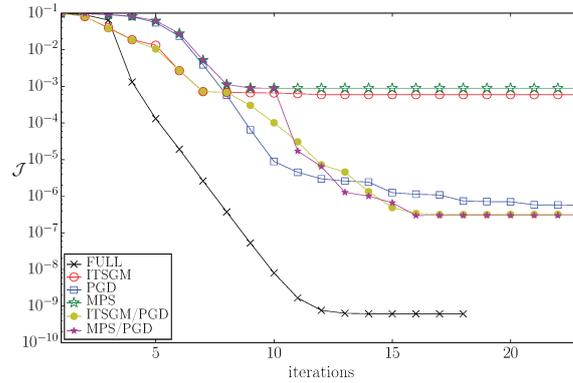
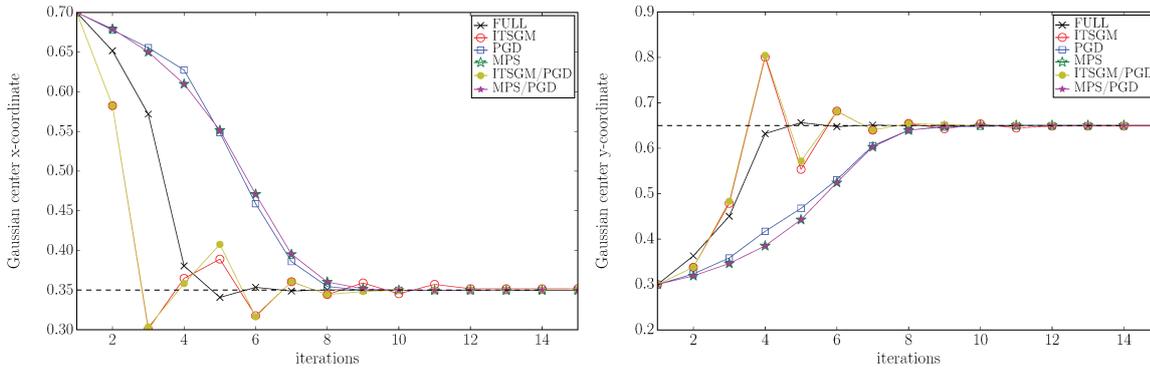


FIGURE 5.4 – Évolution de la fonctionnelle objectif en fonction des itérations de contrôle.

(a) Évolution du paramètre c_x (b) Évolution du paramètre c_y FIGURE 5.5 – Évolution des paramètres c_x et c_y en fonction des itérations de contrôle.

Méthode/itération	$\mathcal{T}^{FULL} / \mathcal{T}_{online}^{method}$	$\theta^{(\infty)}$	$\mathcal{J}(\theta^{(\infty)})$	$\varepsilon_{\%}^T$
FULL (iter.12)	—	(0.350, 0.650)	7.6×10^{-10}	0.003%
ITSGM (iter.12)	98	(0.352, 0.649)	5.9×10^{-04}	5.471%
MPS (iter.12)	134	(0.350, 0.651)	8.9×10^{-04}	6.616%
PGD (iter.12)	94	(0.350, 0.649)	2.9×10^{-06}	0.392%
ITSGM-PGD (iter.12)	100	(0.350, 0.650)	7.2×10^{-06}	0.584%
MPS-PGD (iter.12)	131	(0.350, 0.650)	6.4×10^{-06}	0.541%

TABLE 5.1 – Rapport de temps CPU entre le contrôle optimal classique et le contrôle optimal réduit avec phase d'adaptation, paramètre de contrôle convergé $\theta^{(\infty)}$, valeurs des fonctionnelles et pourcentages d'erreurs de reconstruction entre la solution cible \hat{u} et la solution $u(\theta^{(\infty)})$ obtenue à l'issue de l'algorithme de contrôle.

d) Test 2 : contrôle de $\theta = (\nu, \tilde{\sigma})$

Considérons le problème d'optimisation où l'on souhaite contrôler la solution u à l'intérieur du sous domaine Ω_c en agissant cette fois-ci sur le coefficient de diffusion ν et sur le rayon de la fonction Gaussienne du terme source $\tilde{\sigma}$. Le paramètre de contrôle dans ce cas est noté $\theta = (\nu, \tilde{\sigma})$. Pour ce cas nous posons $\eta(t) = \cos(t)$, $\lambda = 2\sqrt{2}$, $T_0 = 2$, $c_x = 0.55$, $c_y = 0.53$, $\Gamma_D = \Gamma_1 \cup \Gamma_3$ et $\Gamma_N = \Gamma_2 \cup \Gamma_4$. Le modèle complet est résolu sur l'intervalle temporel $]0, 5]$ par éléments finis en considérant un maillage non uniforme constitué de 33025 degrés de libertés² et un pas de temps $\delta t = 10^{-4}$. Les points d'échantillonnage de l'ensemble Y_θ sur lequel les snapshots ont été calculés pour construire les bases POD et MPS/POD sont représentées sur la figure 5.6. L'ensemble contenant les paramètres admissibles est donné par

$$K = \left\{ (\nu, \sigma) \in \mathbb{R}_{+*}^2 / 10^{-4} < \nu < 1 \text{ et } 0.05 < \sigma < 0.5 \right\} \quad (5.15)$$

Soit $\theta^{(0)} = (0.0045, 0.1)$ le paramètre de départ de l'algorithme de contrôle réduit 9, cherchons à retrouver par l'algorithme de contrôle réduit le paramètre cible $\hat{\theta} = (0.001, 0.35)$ et la solution associée pour laquelle la fonctionnelle \mathcal{J} est minimisée. Les figures 5.7 et 5.8

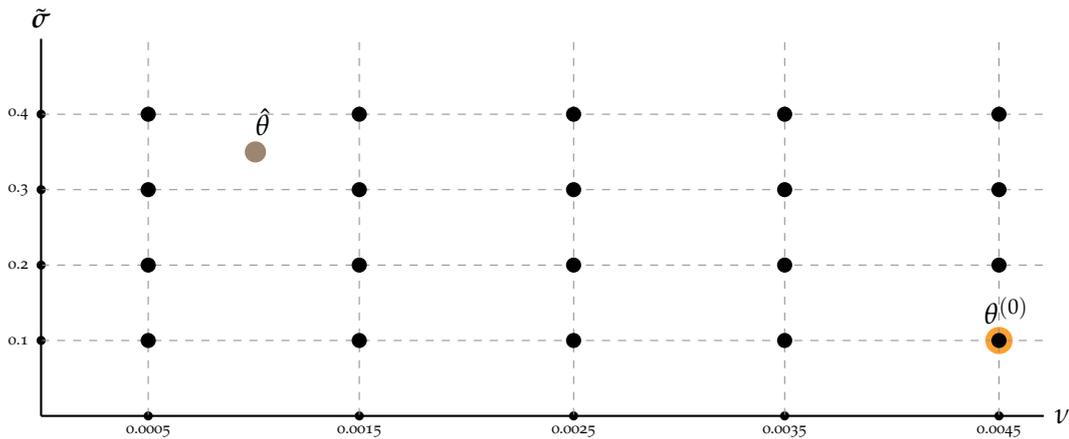


FIGURE 5.6 – Échantillonnage des paramètres de contrôles θ .

montrent que la stratégie du contrôle optimale réduit en utilisant les méthodes ITSGM, PGD, MPS ou encore les variantes ITSGM/PGD, MPS/PGD, permet de retrouver le paramètre cible $\hat{\theta}$ en diminuant drastiquement le temps de calcul CPU. Le temps du contrôle complet (FULL) a été divisé par 85 en utilisant la méthode PGD et 136 en utilisant la méthode MPS/PGD (tableau 5.2). De plus, l'erreur relative entre la solution cible et la solution à l'issue de l'algorithme de contrôle réduit, est petite (entre 2% et 5%).

e) Test 3 : contrôle de $\theta = (c_x, c_y, \nu, \tilde{\sigma})$

Comme il a été souligné dans les deux tests ci-dessus, le contrôle optimal réduit en uti-

2. Le maillage dans ce cas d'étude est plus raffiné que celui du test 1 à cause de la dynamique importante induite par les conditions aux limites de Neumann et les petites valeurs que peut avoir le coefficient de diffusion.

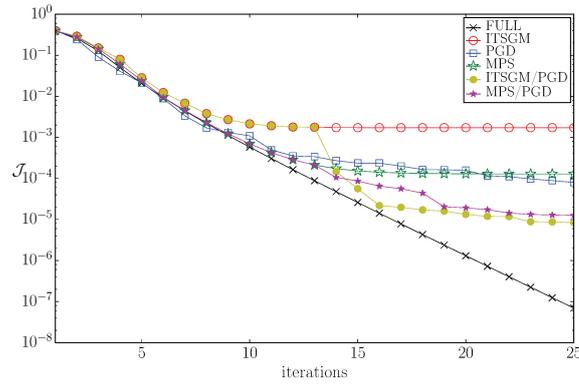
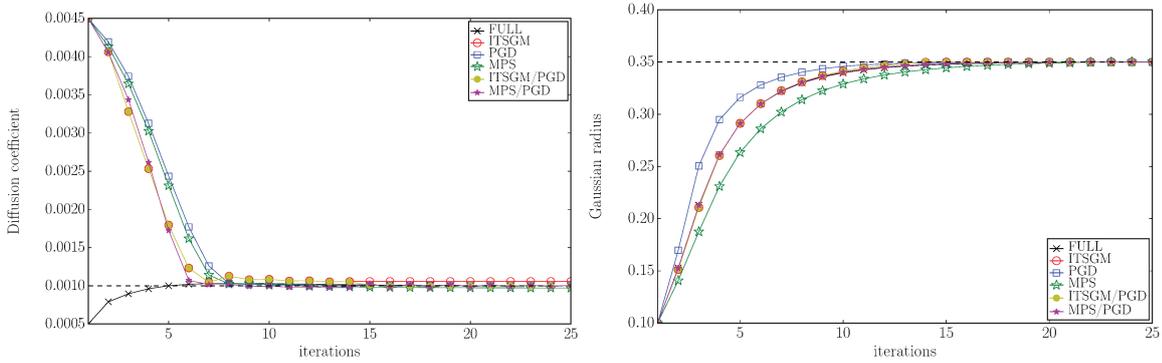


FIGURE 5.7 – Évolution de la fonctionnelle objectif en fonction des itérations.

(a) Évolution du coefficient de diffusion ν (b) Évolution du rayon de la Gaussienne $\tilde{\sigma}$ FIGURE 5.8 – Évolution des paramètres ν et $\tilde{\sigma}$ en fonction des itérations de contrôle.

Méthode/itération	$\mathcal{T}^{FULL} / \mathcal{T}^{method}_{online}$	$\theta^{(\infty)}$	$\mathcal{J}(\theta^{(\infty)})$	$\varepsilon_{\%}^T$
FULL (iter.12)	—	(0.0010, 0.345)	1.6×10^{-04}	1.428%
ITSGM (iter.12)	107	(0.0011, 0.347)	1.7×10^{-03}	4.502%
MPS (iter.12)	111	(0.0010, 0.338)	1.0×10^{-03}	3.676%
PGD (iter.12)	85	(0.0010, 0.348)	3.5×10^{-04}	2.191%
ITSGM-PGD (iter.12)	107	(0.0011, 0.347)	1.7×10^{-03}	4.502%
MPS-PGD (iter.12)	136	(0.0010, 0.345)	2.8×10^{-04}	1.893%

TABLE 5.2 – Rapport de temps CPU entre le contrôle optimal classique et le contrôle optimal réduit avec phase d'adaptation, paramètre de contrôle convergé $\theta^{(\infty)}$, valeurs des fonctionnelles et pourcentages de d'erreurs de reconstruction entre la solution cible \hat{u} et la solution $u(\theta^{(\infty)})$ obtenue à l'issue de l'algorithme de contrôle.

lisant les méthodes ITSGM et MPS nécessite une phase offline de création d'un échantillonnage d'ensembles de snapshots paramétrés. Dans certains cas, lorsque le contrôle est effectué sur un nombre de paramètres d_p supérieur à 2, la phase offline peut être très coûteuse et les approches ITSGM et MPS perdent leur intérêt dans le contrôle optimal réduit. Étant donné que la PGD est une méthode a priori qui vise à diminuer l'erreur de l'approximation, le but de ce paragraphe est alors de montrer que la méthode PGD présente des gains indépendamment du nombre de paramètres de contrôle. Dans ce troisième cas test, nous allons donc considérer le même problème de contrôle que le cas test 2, en agissant en plus sur le coefficient de diffusion et le centre de la fonction Gaussienne. Le paramètre de contrôle s'écrit alors $\theta = (\nu, \tilde{\sigma}, c_x, c_y)$.

On considère \hat{u} la solution cible qui correspond au paramètre $\hat{\theta} = (0.001, 0.35, 0.3, 0.7)$ telle que \hat{u} est approximée sur un sous-espace de dimension 10 représenté par la base POD associée. L'ensemble des solutions admissibles K est donné par

$$K = \left\{ (\nu, \tilde{\sigma}, c_x, c_y) \in \mathbb{R}_+^2 \times \mathbb{R}^2 \ / \ 10^{-4} < \nu < 1, \ 0 < \tilde{\sigma} < 1, \ 0 < c_x < 1 \text{ et } 0 < c_y < 1 \right\} \quad (5.16)$$

En démarrant d'une solution initiale $u^{(0)} = u(\theta^{(0)})$ telle que $\theta^{(0)} = (0.005, 0.1, 0.6, 0.3)$, nous cherchons à atteindre le paramètre cible $\hat{\theta}$ en minimisant la fonctionnelle \mathcal{J} sous la contrainte de l'équation (5.8). L'allure en trois instants des solutions initiale et cible est représenté sur la figure 5.9. Ces figures montrent que la dynamique entre la solution de départ et la solution cible est très riche. Les figures 5.10 et 5.11 montrent que l'algorithme

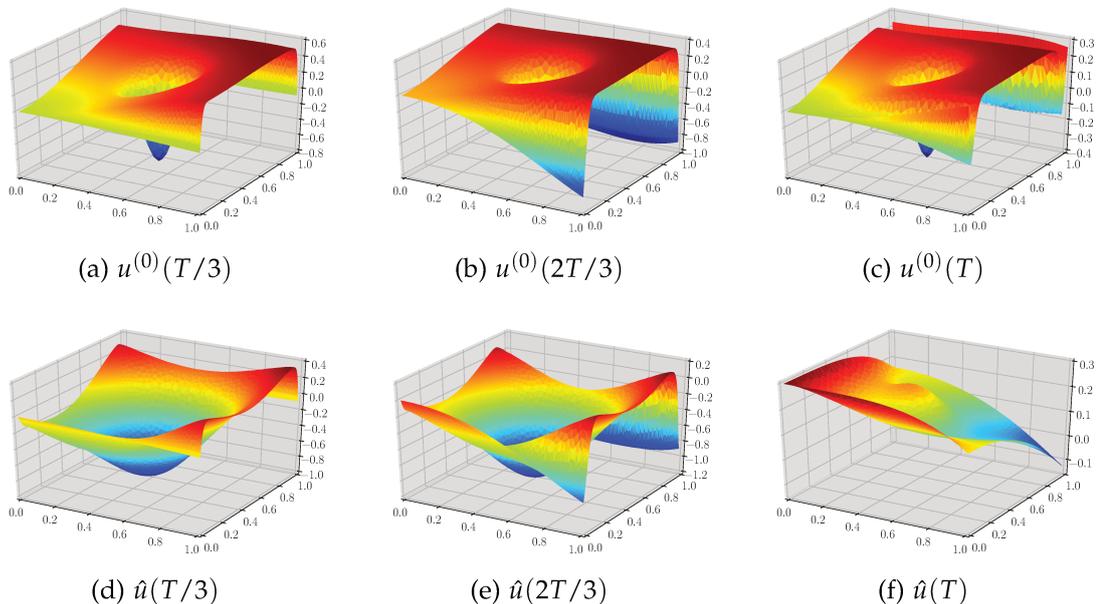


FIGURE 5.9 – Représentation de la solution cible \hat{u} et initiale $u^{(0)}$ aux instants $T/3, 2T/3$ et T .

de contrôle réduit en utilisant la méthode PGD pour l'actualisation des bases réduites, parvient à atteindre correctement les paramètres cibles du problème. Le temps CPU nécessaire pour que l'algorithme PGD fournisse une solution avec une erreur relative de 2.8% (resp. 1.13%), est réduit de 94 fois (resp. 73 fois) par rapport au temps CPU du

problème de contrôle optimal classique basé sur le modèle complet (voir le tableau 5.3). Ainsi, environ 98% du temps CPU a été économisé en utilisant la PGD dans l'approche du contrôle optimal réduit. L'approche du contrôle réduit en utilisant la PGD peut donc réduire drastiquement le temps CPU avec une bonne précision.

Méthode/itération	$\mathcal{T}^{FULL} / \mathcal{T}^{PGD}$	$\mathcal{J}(\theta^{(\infty)})$	$\varepsilon_{\%}^T$
FULL (iter.25)	—	1.05×10^{-5}	0.34%
PGD (iter.20)	94	5.69×10^{-4}	2.8%
PGD (iter.25)	73	9.53×10^{-5}	1.13%

TABLE 5.3 – Rapport de temps CPU entre le contrôle optimal classique et le contrôle optimal réduit en utilisant la PGD dans la phase d'adaptation de bases réduites, paramètre de contrôle convergé $\theta^{(\infty)}$, valeurs des fonctionnelles et pourcentages de d'erreurs de reconstruction entre la solution cible \hat{u} et la solution $u(\theta^{(\infty)})$ obtenues à l'issue de l'algorithme de contrôle.

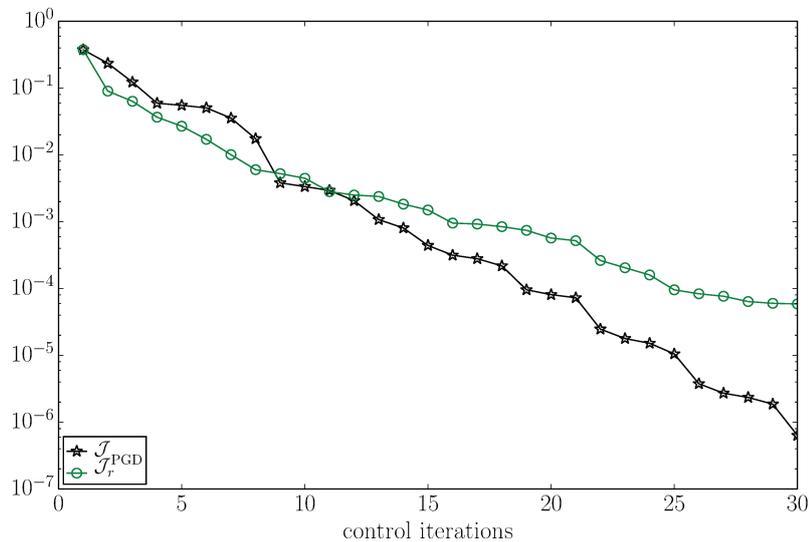


FIGURE 5.10 – Évolution de la fonctionnelle \mathcal{J} .

5.2.3 Application au contrôle de l'équation de Burgers

a) Position du problème

Dans cette section, la stratégie du contrôle optimal réduit adaptatif est appliquée sur le problème de contrôle optimal des équations de Burgers en deux dimensions. Soient $T > 0$ et $\Omega \subset \mathbb{R}^2$ un domaine carré de côté d et de bord Γ . Pour une condition initiale $\mathbf{u}_0 \in [L^2(\Omega)]^2$ et un terme source $\mathbf{f} \in L^2(]0, T], [L^2(\Omega)]^2)$, l'équation de Burgers s'écrit

$$\begin{cases} \partial_t \mathbf{u} - \nu \Delta \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} = \mathbf{f} & \text{dans } \Omega \times]0, T] \\ \mathbf{u} = \mathbf{0} & \text{sur } \Gamma \times]0, T] \\ \mathbf{u}(0) = \mathbf{u}_0 & \text{dans } \Omega. \end{cases} \quad (5.17)$$

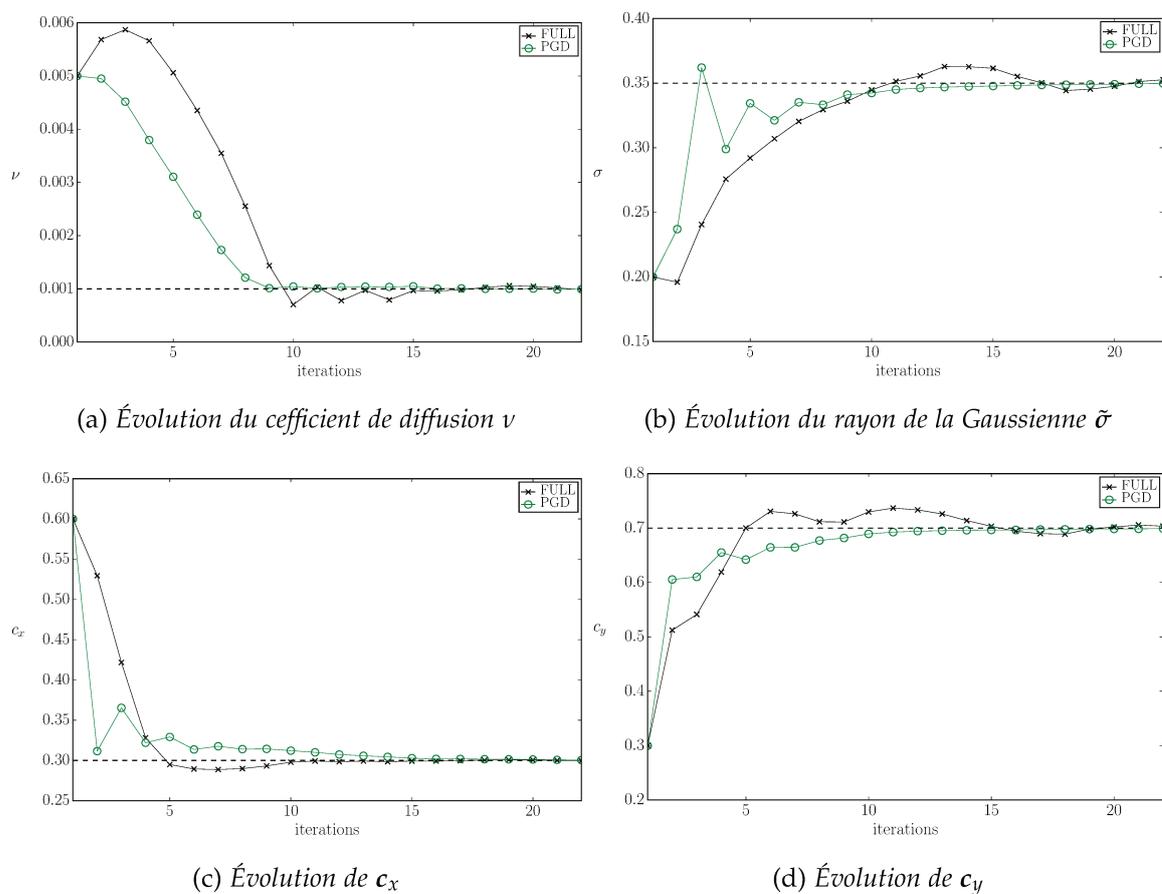


FIGURE 5.11 – Évolution des paramètres de contrôles par rapport aux itérations de l'algorithme de contrôle.

Le terme associé aux forces extérieures \mathbf{f} s'écrit

$$\mathbf{f}(x, y) = \frac{\sqrt{2}}{2} \begin{pmatrix} c_y - y \\ x - c_x \end{pmatrix} \mathbf{G}(x, y)$$

\mathbf{G} est la fonction Gaussienne de rayon $\tilde{\sigma}$, centrée au point $\theta = (c_x, c_y)$ donnée par

$$\mathbf{G}(x, y) = e^{-\frac{(x - c_x)^2 + (y - c_y)^2}{2\tilde{\sigma}^2}}$$

La condition initiale \mathbf{u}_0 est définie comme un champ de vecteurs à divergence nulle qui tourne dans le sens opposé du champs de vecteurs du terme source, soit

$$\mathbf{u}_0(x, y) = \begin{pmatrix} y - 0.5 \\ 0.5 - x \end{pmatrix}$$

On définit la fonctionnelle objectif comme suit

$$\mathcal{J}(\mathbf{u}, \theta) = \frac{1}{2} \int_0^T \int_{\Omega} |\mathbf{u} - \hat{\mathbf{u}}|^2 dx dt + \frac{1}{2} \int_{\Omega} |\mathbf{u}(T) - \hat{\mathbf{u}}(T)|^2 dx + \frac{\kappa}{2} |\theta|^2 \quad (5.18)$$

où κ est un paramètre de régularisation et $\hat{\mathbf{u}}$ une solution cible. L'objectif est donc de contrôler les equations de Burgers en agissant sur le couple de paramètres (c_x, c_y) et d'atteindre les performances définies par la fonctionnelle objectif (5.18). Pour ce cas d'étude, on choisit $d = 1$, $T = 6$, $\nu = 10^{-3}$ et $\tilde{\sigma} = 0.15$. L'espace des solutions admissibles $K \subset \Omega$ est donné par

$$K = \{(c_x, c_y) \in \mathbb{R}^2 \mid 0.25 < c_x < 0.75; 0.25 < c_y < 0.85\} \quad (5.19)$$

b) Écriture du problème de contrôle optimal réduit

Supposons que $\mathbf{u}(\theta)$ et $\hat{\mathbf{u}}$ sont approximées dans deux sous espaces vectoriels engendrés respectivement par deux bases POD $\boldsymbol{\varphi} = \{\varphi^1, \dots, \varphi^q\}$ et $\hat{\boldsymbol{\varphi}} = \{\hat{\varphi}^1, \dots, \hat{\varphi}^{\hat{q}}\}$, soit :

$$\mathbf{u}(t, x, \theta) \approx \sum_{k=1}^q \alpha^k(t, \theta) \varphi^k(x) \quad \hat{\mathbf{u}}(x, t) \approx \sum_{j=1}^{\hat{q}} \hat{\alpha}^j(t) \hat{\varphi}^j(x) \quad (5.20)$$

• Le modèle d'ordre réduit issu de la projection de Galerkin de l'équation (5.17) sur la base POD $\boldsymbol{\varphi}$ s'écrit

$$\begin{cases} \sum_{j=1}^q \mathbf{M}_{ij} \frac{d\alpha^j}{dt} + \nu \sum_{j=1}^q \mathbf{R}_{ij} \alpha^j + \sum_{j,k=1}^q \mathbf{C}_{ijk} \alpha^j \alpha^k = \mathbf{F}_i, & \forall i \in 1, \dots, q \\ \sum_{j=1}^q \mathbf{M}_{ij} \alpha^j(0) = \langle \mathbf{u}_0, \varphi^i \rangle_{L^2(\Omega)}, & \forall i \in 1, \dots, q \end{cases} \quad (5.21)$$

où $F^{(u)}$ et les matrices réduites \mathbf{M} , \mathbf{R} et \mathbf{C} sont obtenus comme suit

$$\begin{aligned} \mathbf{M}_{ij}^{(u)} &= \int_{\Omega} \varphi_u^i \varphi_u^j dx & \mathbf{R}_{ij}^{(u)} &= \int_{\Omega} \nabla \varphi_u^i : \nabla \varphi_u^j dx \\ \mathbf{C}_{ijk}^{(u)} &= \int_{\Omega} (\varphi_u^j \cdot \nabla) \varphi_u^k \cdot \varphi_u^i dx & \mathbf{F}_i^{(u)} &= \int_{\Omega} \mathbf{f} \varphi_u^i dx \end{aligned}$$

- Le problème adjoint réduit associé aux équations d'état réduites 5.21 est donné par

$$\begin{cases} \sum_{i=1}^q M_{ij} \frac{d\beta^i}{dt} - \nu \sum_{i=1}^q R_{ij} \beta^i - \sum_{i,k=1}^q (C_{ijk} + C_{ikj}) \alpha^k \beta^i = \sum_{i=1}^q I_{ji} \alpha^i - \sum_{i=1}^r L_{ji} \hat{\alpha}^i, & \forall j \in 1, \dots, q \\ \sum_{i=1}^q M_{ij} \beta^i(T) = \sum_{i=1}^q L_{ji} \hat{\alpha}^i(T) - \sum_{i=1}^{\hat{q}} I_{ji} \alpha^i(T), & \forall j \in 1, \dots, q \end{cases} \quad (5.22)$$

où I_{kj} sont les coefficients de la fonctionnelle coût réduite obtenus en injectant les approximations (5.20) dans la fonctionnelle objectif (5.18). Ces coefficients sont obtenus par les identités suivantes

$$I_{kj} = \int_{\Omega_c} \varphi^k \varphi^j dx \quad L_{kl} = \int_{\Omega_c} \varphi^k \hat{\varphi}^l dx$$

- Les directions de descente suivant les variables de contrôle c_x, c_y sont finalement calculées par les relations

$$d_{c_x} = -\frac{\sqrt{2}}{2} \sum_{j=1}^q \int_0^T \beta^j dt \int_{\Omega} \varphi^j \frac{x - c_x}{\tilde{\sigma}^2} \begin{pmatrix} c_y - y \\ x - c_x - 1 \end{pmatrix} \mathbf{G}(x, y) dx - \kappa c_x$$

$$d_{c_y} = -\frac{\sqrt{2}}{2} \sum_{j=1}^q \int_0^T \beta^j dt \int_{\Omega} \varphi^j \frac{x - c_y}{\tilde{\sigma}^2} \begin{pmatrix} c_y - y + 1 \\ x - c_x \end{pmatrix} \mathbf{G}(x, y) dx - \kappa c_y$$

c) Application

Partant d'un paramètre initial $(c_x^{init}, c_y^{init}) = (0.45, 0.75)$, le but est d'appliquer la stratégie du contrôle optimal réduit (Algo. 16) afin de retrouver le paramètre cible $(\hat{\theta}_x, \hat{\theta}_y) = (0.63, 0.52)$ pour lequel \hat{u} vérifie l'équation (5.17). Les lignes de courant des solutions initiale et cible sont tracées en trois instants $t = T/3, t = 2T/3$ et T sur la figure 5.12. Cette figure montre que la zone de recirculation initialement centrée au point $(c_x, c_y) = (0.5, 0.5)$ forcée par la condition initiale, est transportée au cours du temps à l'endroit où les forces extérieures sont appliquées.

L'étape offline du contrôle optimal réduit en utilisant les méthodes ITSGM et MPS exige de supposer un échantillonnage de paramètres de contrôle. Pour ce cas d'étude, les l'échantillonnage des paramètres est considérés sur la grille construite en choisissant les valeurs 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8 pour c_x et les mêmes point pour c_y . Le modèle complet est ensuite résolu pour chaque couple de paramètres (c_x, c_y) en utilisant la méthode des éléments finis (avec le pas de temps $\delta t = 0.002$) et les solutions en 500 instants uniformément répartis sur $]0, T]$ sont stockées. En utilisant ces données les bases POD pour chaque paramètre ainsi que la base MPS/POD sont créées. Comme pour l'exemple précédent du contrôle de l'équation de réaction-diffusion, seules les bases associées aux 4 paramètres voisins seront considérées dans l'interpolation ITSGM. Les figures 5.13 et 5.14 montrent l'évolution de la fonctionnelle et des paramètres de contrôle en fonction des itérations de

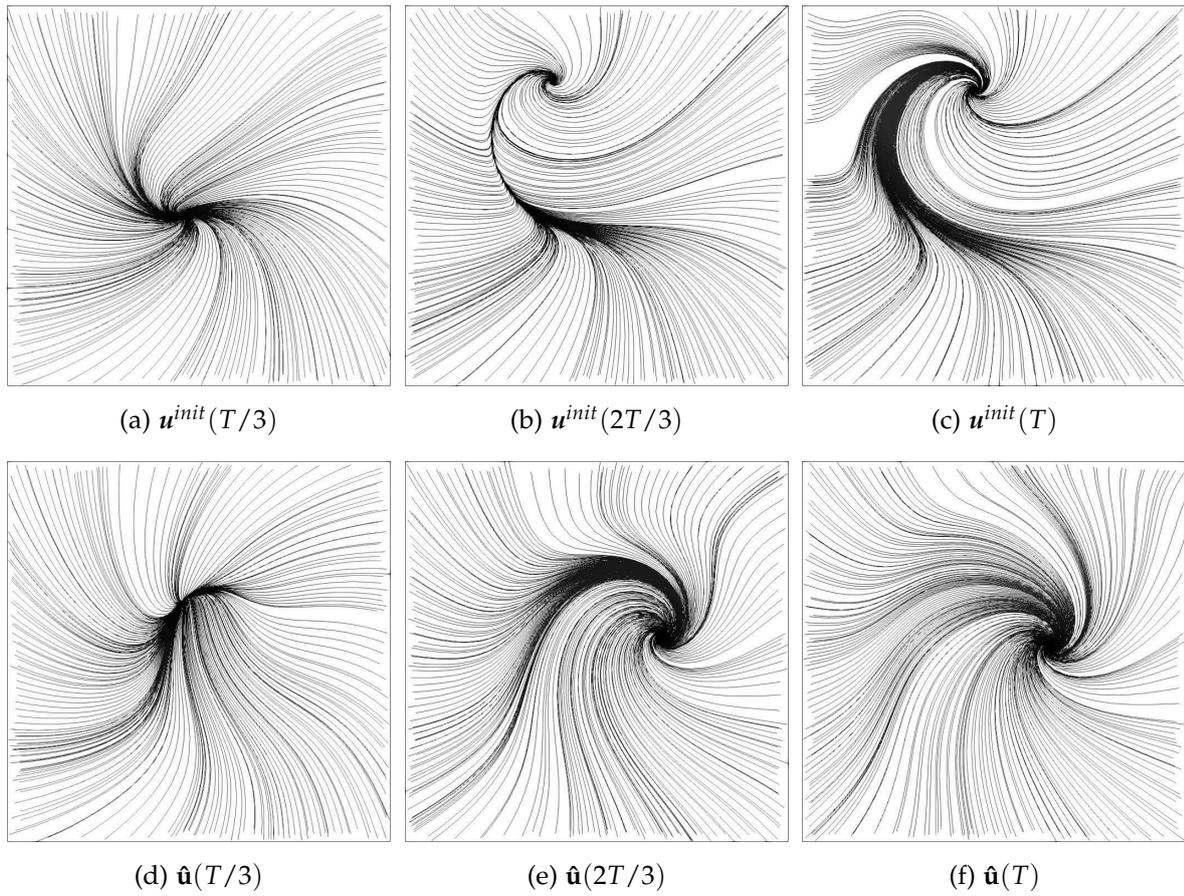


FIGURE 5.12 – Lignes de courant de la solution cible associée au paramètre $(\hat{\theta}_x, \hat{\theta}_y) = (0.63, 0.52)$ (dessous) et de la solution d'état associée au paramètre first-guess du contrôle optimal $(\mathbf{c}_x^{init}, \mathbf{c}_y^{init}) = (0.45, 0.75)$ (dessus).

l'algorithme de contrôle. Les erreurs relatives à l'issue du contrôle sont répertoriées dans le tableau 5.4. On constate que cette erreur est de 5.45% en utilisant la MPS, 5.17% en utilisant la ITSGM et 5.22% en utilisant la PGD.

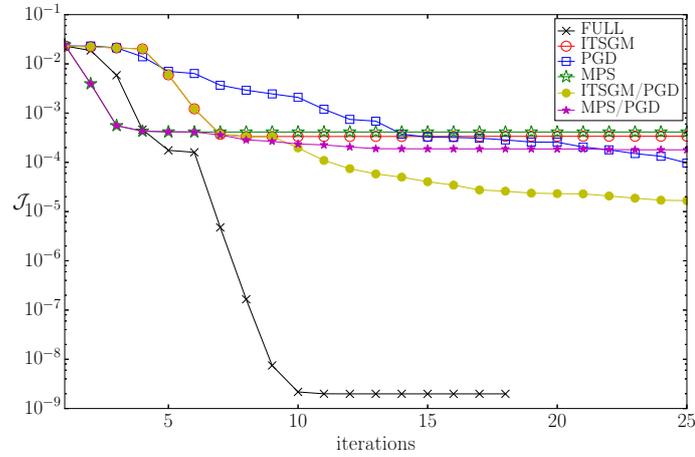
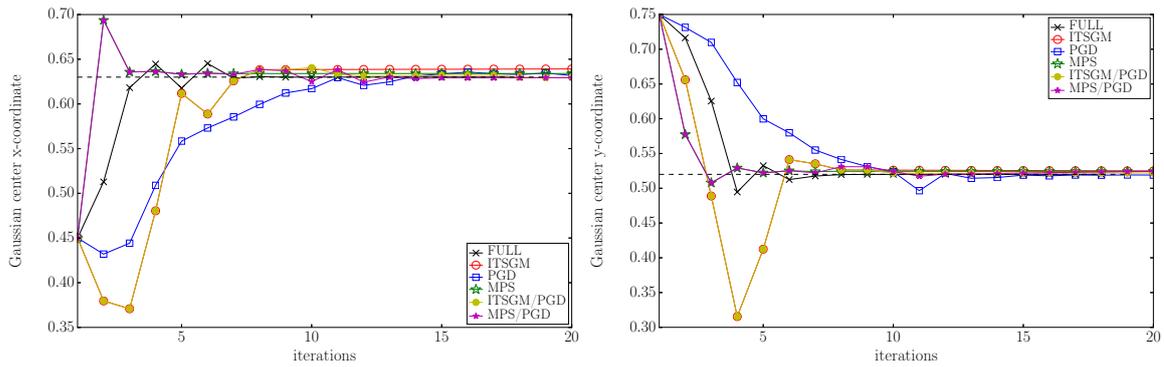


FIGURE 5.13 – Évolution de la fonctionnelle \mathcal{J} .



(a) évolution du paramètre c_x

(b) évolution du paramètre c_y

FIGURE 5.14 – Évolution des paramètres de contrôles par rapport aux itérations de l'algorithme de contrôle.

Method/iteration	$\mathcal{T}^{full} / \mathcal{T}_{online}^{method}$	$\theta^{(\infty)}$	$\mathcal{J}(\theta^{(\infty)})$	$\varepsilon_{\%}^T$
Full (iter. 11)	1	(0.630, 0.520)	1.98×10^{-9}	$10^{-4}\%$
ITSGM (iter. 10)	351	(0.638, 0.526)	3.3×10^{-4}	5.17%
MPS (iter. 6)	754	(0.633, 0.524)	4.1×10^{-4}	5.45%
PGD (iter. 15)	93	(0.635, 0.517)	3.2×10^{-4}	5.22%
ITSGM/PGD (iter. 15)	90	(0.631, 0.521)	3.3×10^{-5}	1.53%
MPS/PGD (iter. 15)	89	(0.629, 0.522)	1.8×10^{-4}	3.90%

TABLE 5.4 – Rapport de temps CPU entre le contrôle optimal classique et le contrôle optimal réduit avec phase d'adaptation, paramètre de contrôle convergé $\theta^{(\infty)}$, valeurs des fonctionnelles et pourcentages de d'erreurs de reconstruction entre la solution cible \hat{u} et la solution $u(\theta^{(\infty)})$ obtenues à l'issue de l'algorithme de contrôle.

La précision maximale des méthodes MPS, ITSGM et PGD dans l'algorithme de contrôle optimal est atteinte respectivement aux itérations 6, 10 et 15. Le contrôle réduit

avec la MPS est le plus rapide, suivi par le contrôle réduit par la ITSGM et la PGD. Le temps CPU estimé pour le contrôle optimal réduit représente 0.13% en utilisant la MPS, 0.28% en utilisant la ITSGM et 1% en utilisant la PGD, du temps nécessaire en utilisant l'approche de contrôle optimale classique. Afin d'améliorer les résultats obtenus par la MPS et la ITSGM à l'issue du contrôle, un couplage avec la PGD est mis en œuvre à partir du moment où la fonctionnelle objectif stagne (6^{ème} itération pour la MPS et 10^{ème} itération pour la ITSGM). En observant les résultats de la fonctionnelle sur la figure 5.13, il est clair que la PGD apporte une amélioration sur la solution en forçant la fonctionnelle à décroître davantage. Un comportement différent sur la fonctionnelle du contrôle optimal réduit en utilisant la méthode PGD et les variantes PGD/ITSGM et PGD/MPS est observé.

5.2.4 Application au contrôle de l'équation de Navier-Stokes

a) Position du problème

Dans cette partie, on s'intéresse au contrôle optimal de l'écoulement autour d'un cylindre pour des nombres de Reynolds $90 \leq Re \leq 450$. On rappelle (voir section 4.5) que le problème est gouverné par les équations de Navier-Stokes suivantes

$$\left\{ \begin{array}{ll} \partial_t \mathbf{u} - \frac{1}{Re} \Delta \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} + \nabla p = 0 & \text{dans } \Omega \times [0, T[\\ \nabla \cdot \mathbf{u} = 0 & \text{dans } \Omega \times [0, T[\\ \mathbf{u} = \mathbf{u}_\infty & \text{sur } \Gamma_{in} \times [0, T[\\ \mathbf{u} = 0 & \text{sur } \Gamma_{cylindre} \times [0, T[\\ u_2 = 0 & \text{sur } \Gamma_{noslip} \times [0, T[\\ \partial_n u_1 = 0 & \text{sur } \Gamma_{noslip} \times [0, T[\\ -\frac{1}{Re} \partial_n \mathbf{u} + p \mathbf{n} = 0 & \text{sur } \Gamma_{out} \times [0, T[\\ \mathbf{u}(0) = \mathbf{u}_0 & \text{dans } \Omega \end{array} \right. \quad (5.23)$$

Considérons une fonctionnelle objectif $\mathcal{J}(\mathbf{u}, Re)$ définie de manière à atteindre des performances données. Le but du problème de contrôle optimal est de chercher le nombre de Reynolds Re_{cible} pour lequel la fonctionnelle objectif \mathcal{J} est minimisée et tel que \mathbf{u} soit solution de l'équation de Navier-Stokes (5.23). L'objectif est d'appliquer la méthode ITSGM/ROM dans la stratégie du contrôle optimal réduit adaptatif décrite par l'algorithme 16. On rappelle que les variables de vitesse et de pression $\bar{\mathbf{u}}$ et \bar{p} sont décomposées comme suit

$$\left\{ \begin{array}{l} \mathbf{u}(t, x) = \bar{\mathbf{u}}(x) + \tilde{\mathbf{u}}(t, x) \\ p(t, x) = \bar{p}(x) + \tilde{p}(t, x) \\ \bar{\mathbf{u}} = \mathbf{u}_\infty, \quad \text{sur } \Gamma_{in} \end{array} \right.$$

où $\bar{\mathbf{u}}$ et \bar{p} sont respectivement la vitesse et la pression moyennes calculées en considérant la moyenne des snapshots pour tous les nombres de Reynolds d'apprentissage $Re =$

$\{90, 120, 150, 180, \dots, 450\}$. Le champs fluctuant de vitesse $\tilde{\mathbf{u}}$ et la pression fluctuante \tilde{p} sont par la suite approximés dans deux sous-espaces vectoriels de dimension finie q_u et q_p , engendrés respectivement par deux bases réduites $\boldsymbol{\varphi}_u = \{\varphi_u^1, \dots, \varphi_u^{q_u}\}$ et $\boldsymbol{\varphi}_p = \{\varphi_p^1, \dots, \varphi_p^{q_p}\}$, soit :

$$\tilde{\mathbf{u}} \approx \sum_{i=1}^{q_u} \alpha_u^i \varphi_u^i \quad \tilde{p} \approx \sum_{i=1}^{q_p} \alpha_p^i \varphi_p^i \quad (5.24)$$

En injectant ces expressions dans l'équation (5.23), puis en annulant la projection de Galerkin du résidu sur les fonctions de base φ_u^i et sur le gradient des fonctions de base φ_p^l , le modèle réduit s'écrit

$$\begin{cases} \sum_{j=1}^{q_u} \mathbf{M}_{ij}^{(u)} \frac{d\alpha_u^j}{dt} + \sum_{j=1}^{q_u} \left[\frac{1}{Re} \mathbf{R}_{ij}^{(u)} + \overline{\mathbf{C}}_{ij}^{(u)} \right] \alpha_u^j + \sum_{j=1}^{q_u} \sum_{k=1}^{q_u} \mathbf{C}_{ijk}^{(u)} \alpha_u^j \alpha_u^k + \sum_{l=1}^{q_p} \mathbf{K}_{il}^{(u)} \alpha_p^l = \tilde{\mathbf{F}}_i^{(u)}, \\ \sum_{j=1}^{q_u} \mathbf{M}_{mj}^{(p)} \frac{d\alpha_u^j}{dt} + \sum_{j=1}^{q_u} \left[\frac{1}{Re} \mathbf{R}_{mj}^{(p)} + \overline{\mathbf{C}}_{mj}^{(p)} \right] \alpha_u^j + \sum_{j=1}^{q_u} \sum_{k=1}^{q_u} \mathbf{C}_{mjk}^{(p)} \alpha_u^j \alpha_u^k + \sum_{l=1}^{q_p} \mathbf{K}_{ml}^{(p)} \alpha_p^l = \tilde{\mathbf{F}}_m^{(p)} \\ \sum_{j=1}^{q_u} \mathbf{M}_{ij}^{(u)} \alpha_u^j(0) = \int_{\Omega} \mathbf{u}_0 \varphi_u^i dx \\ \forall i = 1, \dots, q_u, \quad \forall m = 1, \dots, q_p \end{cases} \quad (5.25)$$

où

$$\begin{aligned} \mathbf{M}_{ij}^{(u)} &= \int_{\Omega} \varphi_u^j \varphi_u^i dx & \mathbf{R}_{ij}^{(u)} &= \int_{\Omega} \nabla \varphi_u^j : \nabla \varphi_u^i dx & \mathbf{K}_{il}^{(u)} &= \int_{\Gamma} \varphi_p^l \varphi_u^i \cdot \mathbf{n} d\sigma \\ \overline{\mathbf{C}}_{ij}^{(u)} &= \int_{\Omega} (\bar{\mathbf{u}} \cdot \nabla) \varphi_u^j \cdot \varphi_u^i dx + \int_{\Omega} (\varphi_u^j \cdot \nabla) \bar{\mathbf{u}} \cdot \varphi_u^i dx \\ \mathbf{C}_{ijk}^{(u)} &= \int_{\Omega} (\varphi_u^j \cdot \nabla) \varphi_u^k \cdot \varphi_u^i dx & \tilde{\mathbf{F}}_i^{(u)} &= \int_{\Omega} \left(\frac{1}{Re} \Delta \bar{\mathbf{u}} - \bar{\mathbf{u}} \cdot \nabla \bar{\mathbf{u}} - \nabla \bar{p} \right) \varphi_u^i dx \\ \mathbf{M}_{mj}^{(p)} &= \int_{\Omega} \varphi_p^j \nabla \varphi_p^m dx & \mathbf{R}_{mj}^{(p)} &= - \int_{\Omega} \Delta \varphi_p^j : \nabla \varphi_p^m dx & \mathbf{K}_{ml}^{(p)} &= \int_{\Omega} \nabla \varphi_p^l \cdot \nabla \varphi_p^m dx \\ \overline{\mathbf{C}}_{mj}^{(p)} &= \int_{\Omega} (\bar{\mathbf{u}} \cdot \nabla) \varphi_p^j \cdot \nabla \varphi_p^m dx + \int_{\Omega} (\varphi_p^j \cdot \nabla) \bar{\mathbf{u}} \cdot \nabla \varphi_p^m dx \\ \mathbf{C}_{mjk}^{(p)} &= \int_{\Omega} (\varphi_p^j \cdot \nabla) \varphi_p^k \cdot \nabla \varphi_p^m dx & \tilde{\mathbf{F}}_m^{(p)} &= \int_{\Omega} \left(\frac{1}{Re} \Delta \bar{\mathbf{u}} - \bar{\mathbf{u}} \cdot \nabla \bar{\mathbf{u}} - \nabla \bar{p} \right) \nabla \varphi_p^m dx \end{aligned}$$

Le problème réduit a été rappelé ici car les différents coefficients vont intervenir dans l'expression du problème adjoint et l'expression du gradient de la fonctionnelle.

b) Test 1 : contrôle optimal pour atteindre une vitesse cible

• Soit $\hat{\mathbf{u}}$ un champ de vitesse cible tel que $\hat{\mathbf{u}}$ est approximée dans le sous espace vectoriel de dimension \hat{q} engendré par les \hat{q} premiers modes de la base POD associée. On écrit donc

$$\hat{\mathbf{u}}(x, t) \approx \sum_{j=1}^{\hat{q}} \hat{\alpha}^j(t) \hat{\varphi}^j(x)$$

Dans cette partie, on souhaite minimiser par rapport au nombre de Reynolds la fonctionnelle objectif suivante

$$\mathcal{J}(\mathbf{u}, Re) = \frac{1}{2} \int_0^T \int_{\Omega} |\mathbf{u} - \hat{\mathbf{u}}|^2 dx dt + \frac{1}{2} \int_{\Omega} |\mathbf{u}(T) - \hat{\mathbf{u}}(T)|^2 dx + \frac{\kappa}{2} Re^2 \quad (5.26)$$

- Le calcul des directions de descente est effectué en utilisant les équations adjointes obtenues par la méthode des multiplicateurs de Lagrange. Les équations adjointes réduites s'écrivent dans ce cas

$$\left\{ \begin{array}{l} \sum_{i=1}^{q_u} M_{ij}^{(u)} \frac{d\beta_u^i}{dt} - \sum_{i=1}^{q_u} \left[\frac{1}{Re} R_{ij}^{(u)} + \bar{C}_{ij}^{(u)} \right] \beta_u^i - \sum_{i=1}^{q_u} \sum_{k=1}^{q_u} \left[C_{ijk}^{(u)} + C_{ikj}^{(u)} \right] \alpha^k \beta_u^i + \sum_{m=1}^{q_u} M_{mj}^{(p)} \frac{d\beta_p^m}{dt} \\ - \sum_{m=1}^{q_u} \left[\frac{1}{Re} R_{mj}^{(p)} + \bar{C}_{mj}^{(p)} \right] \beta_p^m - \sum_{m=1}^{q_p} \sum_{k=1}^{q_u} \left[C_{mjk}^{(p)} + C_{mkj}^{(p)} \right] \alpha^k \beta_p^m = \sum_{i=1}^q I_{ji} \alpha^i - \sum_{i=1}^r L_{ji} \hat{\alpha}^i \\ \sum_{i=1}^{q_u} K_{il}^{(u)} \beta_u^i + \sum_{m=1}^{q_p} K_{ml}^{(u)} \beta_p^m = 0 \\ \sum_{j=1}^{q_u} M_{ij}^{(u)} \beta_u^i(T) + \sum_{m=1}^{q_p} M_{ml}^{(p)} \beta_p^m(T) = \sum_{i=1}^q L_{ji} \hat{\alpha}^i(T) - \sum_{i=1}^{\hat{q}} I_{ji} \alpha^i(T) \\ \forall j = 1, \dots, q_u, \quad \forall l = 1, \dots, q_p \end{array} \right. \quad (5.27)$$

où

$$I_{kj} = \int_{\Omega} \varphi^k \varphi^j dx \quad L_{kl} = \int_{\Omega} \varphi^k \hat{\varphi}^l dx$$

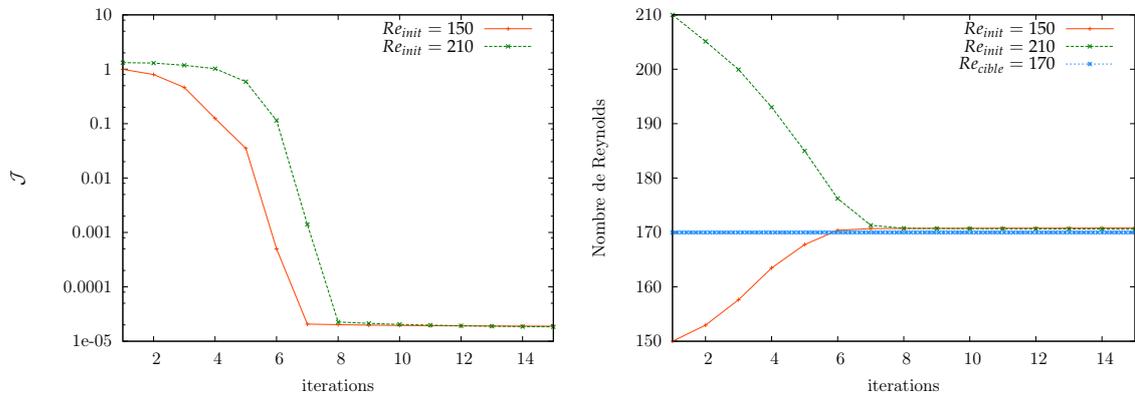
En utilisant les équations adjointes réduites, la direction de descente est calculée comme suit

$$d_{Re} = \frac{1}{Re^2} \sum_{i=1}^q \sum_{j=1}^q R_{ij} \int_0^T \alpha^i \beta^j dt - \frac{1}{Re^2} \sum_{i=1}^q \int_{\Omega} \Delta \bar{u} \varphi_u^i dx \int_0^T \beta^j dt - \kappa Re \quad (5.28)$$

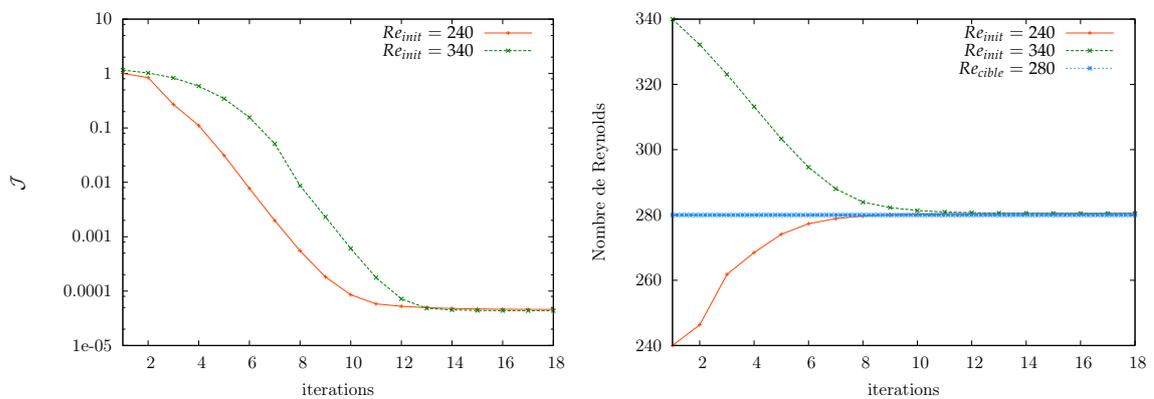
- Dans la suite, on considère deux champs de vitesse cibles \hat{u} construits pour $Re_{cible} = 170$ puis $Re_{cible} = 280$. Pour chaque \hat{u} deux conditions de démarrage de l'algorithme de contrôle réduit ont été considérées. Dans ce cas, le modèle réduit obtenu par l'adaptation des bases réduites en utilisant les méthodes d'interpolation classiques (Lagrange et RBF), l'approche MPS ou la méthode PGD ne parvient pas à reproduire la dynamique temporelle du problème pour un nombre de Reynolds cible. Par conséquent, ces méthodes échouent dans le contrôle optimal réduit. La méthode ITSGM par contre permet une bonne prédiction de l'écoulement et une convergence rapide de l'algorithme de contrôle réduit. Ainsi, pour $Re_{cible} = 170$, en partant de $Re_{init} = 150$ ou de $Re_{init} = 210$, l'algorithme de contrôle réduit converge respectivement en 8 et 10 itérations (voir figure 5.15 et tableau 5.5). De même, pour $Re_{cible} = 280$, en partant de $Re_{init} = 240$ et $Re_{init} = 340$ l'algorithme de contrôle réduit converge respectivement en 12 et 14 itérations (voir figure 5.16 et tableau 5.5).

Re_{init}	itération	Temps CPU	Re_{∞}	$\mathcal{J}(Re_{\infty})$
150	8	1 min 48 sec	170.73	2.01×10^{-5}
210	10	2 min 10 sec	170.71	2.04×10^{-5}

TABLE 5.5 – Temps CPU du contrôle optimal réduit avec phase d'adaptation ITSGM, paramètre de contrôle convergé Re_{∞} et valeur de la fonctionnelle à l'issue de l'algorithme de contrôle pour $Re_{cible} = 170$.

FIGURE 5.15 – Évolution de la fonctionnelle et du nombre de Reynolds pour $Re_{cible} = 170$.

Re_{init}	itération	Temps CPU	Re_{∞}	$\mathcal{J}(Re_{\infty})$
240	12	2.7 min 24 sec	280.41	5.25×10^{-5}
340	14	3.2 min 12 sec	280.52	4.51×10^{-5}

TABLE 5.6 – Temps CPU du contrôle optimal réduit avec phase d'adaptation ITSGM, paramètre de contrôle convergé Re_{∞} et valeur de la fonctionnelle à l'issue de l'algorithme de contrôle pour $Re_{cible} = 280$.FIGURE 5.16 – Évolution de la fonctionnelle et du nombre de Reynolds pour $Re_{cible} = 280$.

c) Test 2 : contrôle optimal pour atteindre un coefficient $C_{L,rms}$ cible

- On rappelle que le coefficient de portance adimensionnel C_L est donné par

$$C_L = 2 \int_{\Gamma_{cylindre}} \left(\frac{1}{Re} (\nabla \mathbf{u} + \nabla^T \mathbf{u}) - p \mathbf{I} \right) \cdot \mathbf{n} d\sigma \mathbf{e}_y$$

Dans cet exemple, la fonctionnelle objectif d'intérêt est définie par

$$\mathcal{J}(\mathbf{u}, p, Re) = \frac{1}{2} (C_{L,rms} - \hat{C}_{L,rms})^2 + \frac{\kappa}{2} Re^2 \quad (5.29)$$

telle que

$$C_{L,rms} = \sqrt{\frac{1}{T} \int_0^T C_L^2 dt} \quad (5.30)$$

En utilisant les représentations réduites de $\tilde{\mathbf{u}}$ et \tilde{p} (5.24), le coefficient C_L peut être approximé par

$$C_L \approx C'_L = \left(\frac{1}{Re} \sum_{j=1}^{q_u} \mathbf{Z}_L^j \alpha_u^j - \sum_{l=1}^{q_p} \mathbf{W}_L^l \alpha_p^l \right) + \frac{1}{Re} \bar{\mathbf{Z}}_L - \bar{\mathbf{W}}_L \quad (5.31)$$

où

$$\begin{aligned} \mathbf{Z}_L^j &= \int_{\Gamma_{cylindre}} (\nabla \phi_u^j + \nabla^T \phi_u^j) \cdot \mathbf{n} d\sigma \mathbf{e}_y & \mathbf{W}_L^l &= \int_{\Gamma_{cylindre}} \phi_p^l \mathbf{I} \cdot \mathbf{n} d\sigma \mathbf{e}_y \\ \bar{\mathbf{Z}}_L &= \int_{\Gamma_{cylindre}} (\nabla \bar{\mathbf{u}} + \nabla^T \bar{\mathbf{u}}) \cdot \mathbf{n} d\sigma \mathbf{e}_y & \bar{\mathbf{W}}_L &= \int_{\Gamma_{cylindre}} \bar{p} \mathbf{I} \cdot \mathbf{n} d\sigma \mathbf{e}_y \end{aligned}$$

d'où

$$C_{L,rms} \approx C'_{L,rms} = \sqrt{\frac{1}{T} \int_0^T C'_L{}^2 dt} \quad (5.32)$$

La fonctionnelle objectif réduite s'écrit donc

$$\mathcal{J}_\varphi(\boldsymbol{\alpha}_u, \boldsymbol{\alpha}_p, Re) = \frac{1}{2} (C'_{L,rms} - \hat{C}_{L,rms})^2 + \frac{\kappa}{2} Re^2 \quad (5.33)$$

- Le problème adjoint réduit associé au problème de contrôle optimal étudié est donné par les équations

$$\begin{cases} \sum_{i=1}^{q_u} \mathbf{M}_{ij}^{(u)} \frac{d\beta_u^i}{dt} - \sum_{i=1}^{q_u} \left[\frac{1}{Re} \mathbf{R}_{ij}^{(u)} + \bar{\mathbf{C}}_{ij}^{(u)} \right] \beta_u^i - \sum_{i=1}^{q_u} \sum_{k=1}^{q_u} \left[\mathbf{C}_{ijk}^{(u)} + \mathbf{C}_{ikj}^{(u)} \right] \alpha^k \beta_u^i + \sum_{m=1}^{q_u} \mathbf{M}_{mj}^{(p)} \frac{d\beta_p^m}{dt} \\ - \sum_{m=1}^{q_u} \left[\frac{1}{Re} \mathbf{R}_{mj}^{(p)} + \bar{\mathbf{C}}_{mj}^{(p)} \right] \beta_p^m - \sum_{m=1}^{q_p} \sum_{k=1}^{q_u} \left[\mathbf{C}_{mjk}^{(p)} + \mathbf{C}_{mkj}^{(p)} \right] \alpha_u^k \beta_p^m = \frac{C'_L (C'_{L,rms} - \hat{C}_{L,rms})}{T Re C'_{L,rms}} \mathbf{Z}_L^j \\ \sum_{i=1}^{q_u} \mathbf{K}_{il}^{(u)} \beta_u^i + \sum_{m=1}^{q_p} \mathbf{K}_{ml}^{(u)} \beta_p^m = - \frac{C'_L (C'_{L,rms} - \hat{C}_{L,rms})}{T C'_{L,rms}} \mathbf{W}_L^l \\ \sum_{j=1}^{q_u} \mathbf{M}_{ij}^{(u)} \beta_u^j(T) + \sum_{m=1}^{q_p} \mathbf{M}_{ml}^{(p)} \beta_p^m(T) = 0 \\ \forall j = 1, \dots, q_u, \quad \forall l = 1, \dots, q_p \end{cases} \quad (5.34)$$

- En utilisant ces équations, les directions de descente sont finalement calculées par les relations suivantes

$$\begin{aligned}
d_{Re} = & \frac{C'_L(C'_{L,rms} - \hat{C}_{L,rms})}{TRe^2 C'_{L,rms}} \left(\sum_{j=1}^{q_u} \frac{\mathbf{z}_L^j}{T} \int_0^T \alpha^j dt + \bar{\mathbf{z}}_L \right) + \frac{1}{Re^2} \sum_{i=1}^{q_u} \sum_{j=1}^{q_u} \mathbf{R}_{ij}^{(u)} \int_0^T \alpha^i \beta^j dt \\
& + \frac{1}{Re^2} \sum_{l=1}^{q_u} \sum_{j=1}^{q_p} \mathbf{R}_{lj}^{(p)} \int_0^T \alpha^l \beta_p^j dt + \frac{1}{Re^2} \sum_{i=1}^{q_u} \int_{\Omega} \Delta \bar{\mathbf{u}} \varphi_u^i dx \int_0^T \beta^i dt \\
& + \frac{1}{Re^2} \sum_{l=1}^{q_u} \int_{\Omega} \Delta \bar{\mathbf{u}} \nabla \varphi_p^l dx \int_0^T \beta_p^l dt - \kappa Re
\end{aligned}$$

- L'objectif est de tester encore une fois la robustesse du contrôle optimal réduit en utilisant la méthode ITSGM. Deux coefficients de portance cibles $\hat{C}_{L,rms}$ correspondants à $Re_{cible} = 130$ et $Re_{cible} = 190$ ont été considérés. Pour chaque $\hat{C}_{L,rms}$, deux conditions de démarrage de l'algorithme de contrôle réduit ont été choisies. Pour $Re_{cible} = 130$, en partant de $Re_{init} = 90$ ou de $Re_{init} = 220$, l'algorithme de contrôle réduit converge respectivement en 13 et 18 itérations (voir figure 5.17 et tableau 5.7). De même, pour $Re_{cible} = 190$, en partant de $Re_{init} = 140$ et $Re_{init} = 230$ l'algorithme de contrôle réduit converge respectivement en 17 et 15 itérations (voir figure 5.18 et tableau 5.8).

Re_{init}	itération	Temps CPU	Re_{∞}	$\mathcal{J}(Re_{\infty})$
90	13	5 min 24 sec	129.01	3.63×10^{-7}
220	18	6 min 42 sec	129.76	1.17×10^{-7}

TABLE 5.7 – Temps CPU du contrôle optimal réduit avec phase d'adaptation ITSGM, paramètre de contrôle convergé Re_{∞} et valeur de la fonctionnelle à l'issue de l'algorithme de contrôle pour $Re_{cible} = 130$.

Re_{init}	itération	Temps CPU	Re_{∞}	$\mathcal{J}(Re_{\infty})$
140	17	5 min 12 sec	189.35	4.97×10^{-7}
230	15	9 min 12 sec	190.19	4.31×10^{-7}

TABLE 5.8 – Temps CPU du contrôle optimal réduit avec phase d'adaptation ITSGM, paramètre de contrôle convergé Re_{∞} et valeur de la fonctionnelle à l'issue de l'algorithme de contrôle pour $Re_{cible} = 190$.

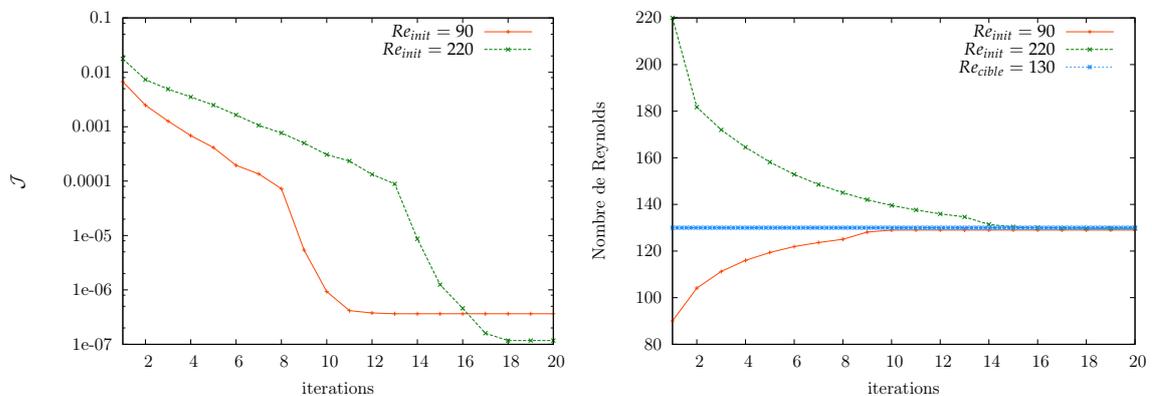


FIGURE 5.17 – Évolution de la fonctionnelle et du nombre de Reynolds pour $Re_{cible} = 130$.

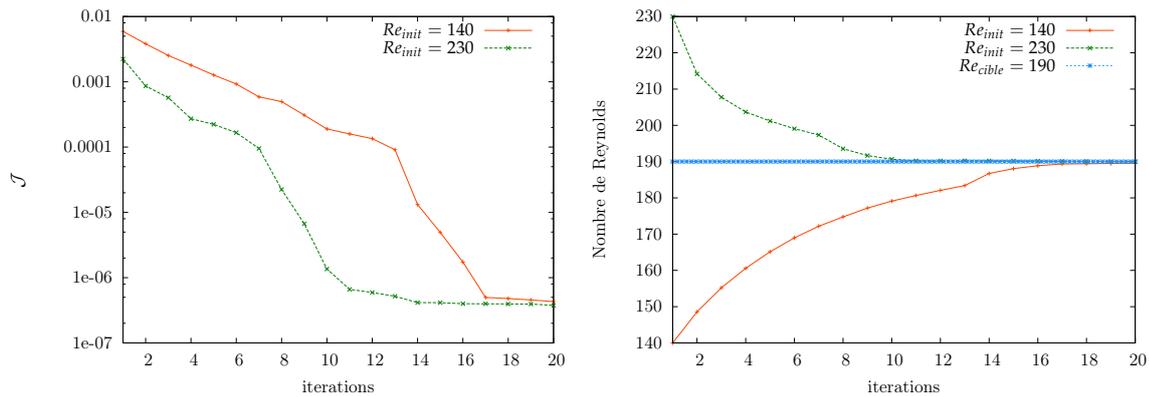


FIGURE 5.18 – Évolution de la fonctionnelle et du nombre de Reynolds pour $Re_{cible} = 190$.

Les résultats obtenus par l'approche du contrôle réduit en utilisant la méthode ITSGM dans la phase d'adaptation de bases sont donc satisfaisants et permettent des temps de calcul réduit. En effet, un temps réel de 5min en moyenne est suffisant pour atteindre un contrôle optimal d'une précision satisfaisante.

5.3 CONTRÔLE OPTIMAL RÉDUIT EN UTILISANT LES ALGORITHMES GÉNÉTIQUES

5.3.1 Principe des algorithmes génétiques

L'algorithme de contrôle optimal réduit basé sur l'équation adjointe nécessite, pour le calcul des directions de descente, d'avoir recours aux équations du phénomène étudié. Cette approche est donc intrusive et ne peut donc pas être couplée avec les méthodes NIMR et HNIMR qui sont par construction non intrusives afin de pouvoir appliquer ces méthodes. Il est donc nécessaire d'utiliser des approches de contrôle optimal alternatives à caractère non intrusif afin de pouvoir appliquer ces méthodes. Une approche possible consiste à utiliser les algorithmes génétiques formellement introduit dans les années 70 par John Holland à l'université de Michigan [156]. Les algorithmes génétiques sont des algorithmes d'optimisation stochastiques fondés sur les mécanismes de la sélection naturelle et de la génétique. Leur principe est simple. On part d'une population initiale arbitrairement choisie, dont les membres sont des individus qui représentent des solutions potentielles du problème d'optimisation que l'on souhaite résoudre. La population initiale est définie comme un ensemble de N individus. Chaque individu est représenté par un ou plusieurs chromosomes et chaque chromosome comprend un nombre de gènes égal aux nombre de variables à optimiser. Pour simplifier, on se restreint au cas où les individus sont eux mêmes les chromosomes. L'algorithme génétique consiste à sélectionner des sous ensembles (généralement des paires) d'individus, appelés parents, pour les combiner et produire de nouveaux individus appelés enfants. Les règles de combinaison pour donner des enfants sont basées sur la notion génétique de croisement qui consiste

à interchanger les gènes des individus, et sur des opérations occasionnelles (appelées mutations) se traduisant par un changement de gènes de façon aléatoire. Les enfants issus de l'accouplement des parents, et qui passent un test de "survie" (appelé fitness), sont alors disponibles pour être choisis en tant que parents de la prochaine génération. Ces opérateurs génétiques simples permettent de faire évoluer les solutions vers la solution optimale en ne "gardant que les meilleures" individus. Ce cycle est recommencé jusqu'à ce que l'on trouve une solution satisfaisante.

Le principe des algorithmes génétiques est résumé sur la figure 5.19. Les aspects les plus importants lors de l'utilisation des algorithmes génétiques sont donc :

- la définition de la fonction fitness,
- la définition de la représentation génétique,
- la définition des opérateurs génétiques.

Ces différents points sont décrits dans les paragraphes suivants.

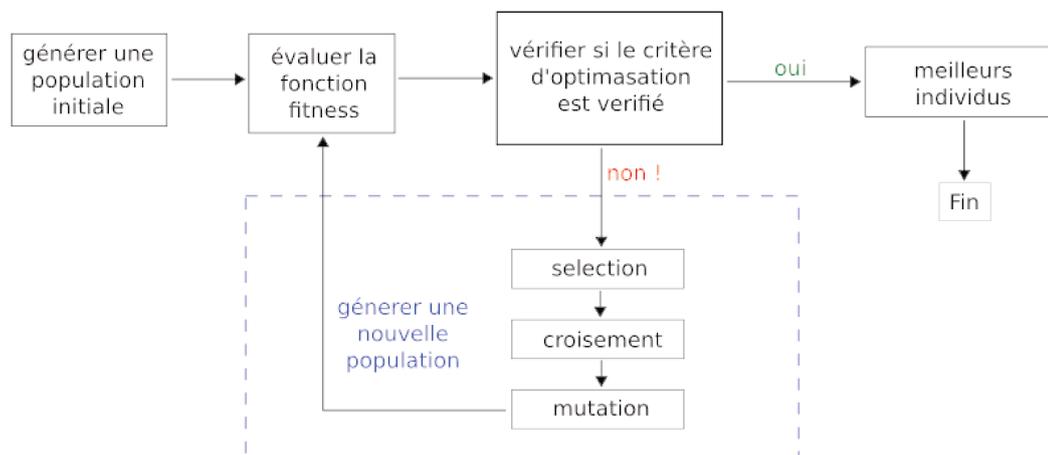


FIGURE 5.19 – Schématisation des étapes de l'algorithme génétique.

5.3.2 Représentation génétique

La mise en œuvre d'un algorithme génétique nécessite tout d'abord de définir une structure génétique pour les individus qui formeront la population initiale. La procédure directe pour coder un algorithme génétique qui évolue en fonction de plusieurs paramètres consiste à construire les individus sous forme de tableaux formés par concaténation des paramètres d'optimisation. Ce codage est appelé codage réel et apporte une simplicité et flexibilité à l'algorithme génétique. Une autre approche pour coder l'algorithme génétique est de représenter chaque paramètre d'optimisation par un nombre binaire (séquence de bits). Ces nombres binaires sont par la suite concaténés l'un après l'autre pour former l'individu. Chaque séquence du vecteur total représente un gène, et la valeur de chaque gène est un allèle. Par exemple, les composantes d'un vecteur θ composé de trois

paramètres θ_1 , θ_2 et θ_3 codés sur 5 bits s'écrivent :

$$\theta_1 = \{01101\} \quad \theta_2 = \{10110\} \quad \theta_3 = \{01001\}$$

et la séquence totale serait $\bar{\theta} = \{01101|10110|01001\}$.

5.3.3 Opérateurs génétiques

L'algorithme génétique (figure 5.19) commence à partir d'une population initiale, dont les individus (solutions candidates) sont généralement générés au hasard dans l'espace des solutions admissibles. Les opérations génétiques : sélection, croisement et mutation, sont répétées jusqu'à atteindre une solution ou pour un nombre de générations donné. Ces opérations génétiques sont décrites ci-après.

a) Croisement

Le croisement permet la recombinaison des chromosomes (figure 5.20) en échangeant des parties de deux génotypes. Trois opérations de croisement sont illustrées sur la figure 5.20. Classiquement, les croisements se font avec deux parents et donnent naissance à deux enfants. Dans le premier cas, on tire aléatoirement une position d'un gène dans deux individus parents, et on échange les deux sous chaînes de chacun des chromosomes. Dans le deuxième cas, deux points de croisement aléatoires sont considérés. Le chromosome enfant 1 hérite de la partie milieu du chromosome parent 2 et des extrémités du chromosome parent 1, et le chromosome enfant 2 hérite de la partie milieu du chromosome parent 1 et des extrémités du chromosome parent 2. Le croisement multi-point est généralisé dans le troisième cas où la recombinaison à n points des gènes est effectuée. Il est aussi possible dans certains cas d'effectuer le croisement en faisant la moyenne des valeurs d'un certain nombre de gènes, tout en gardant les autres gènes inchangés.

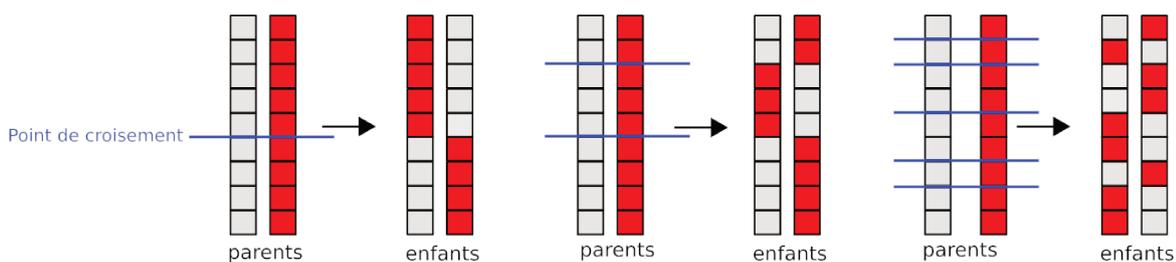


FIGURE 5.20 – Schématisation des opérations de croisement uniformes. Dans ce cas, chaque enfant hérite de 50% des gènes du parent 1 et de 50% des gènes du parent 2.

b) Mutation

La mutation est utilisée pour préserver la diversité génétique de la population et pour empêcher les individus de devenir trop similaires en introduisant de petits changements dans leur génotype. Cette diversification permet d'éviter les minima locaux. L'opérateur de mutation consiste à tirer aléatoirement une position dans le chromosome et à remplacer

le gène qui lui correspond. Dans le cas du codage binaire, les valeurs des bits sélectionnés au hasard sont basculés, et pour l'encodage de nombres réels la modification peut être effectuée en remplaçant (avec une probabilité donnée) la valeur d'un gène avec un nombre tiré d'une distribution normale.

c) Évaluation de la fonction fitness

Après l'application des opérateurs génétiques, une nouvelle génération d'individus est créée. Pour chaque individu de la population actuelle la fonction "fitness" est évaluée. L'algorithme de sélection utilisé ici suppose que les valeurs élevées de la fonction "fitness" sont directement liées aux meilleures solutions. Dans ce manuscrit, la fonction "fitness" correspond à l'inverse de la fonctionnelle objectif à minimiser.

d) Sélection

La dernière étape est la sélection des individus les plus aptes à d'autres modifications. Nous rappelons ci-après quelques algorithmes de sélection [157, 158] :

- Sélection par la roulette :

Pour chaque individu, la valeur de la fonction "fitness" est normalisée en la divisant par la somme des valeurs de fitness de tous les individus de la population, de sorte que la somme des valeurs de fitness pour la population totale soit égale à 1. La part de la roulette (figure 5.21) correspondante à chaque individu est donc proportionnelle à la valeur normalisée de sa fonction fitness. Dans l'étape qui suit, un nombre réel aléatoire entre 0 et 1 est choisi, et l'individu dont la valeur de fitness normalisée cumulée est supérieure à cette valeur aléatoire sera sélectionné pour la reproduction. Le nombre de rotations de la roulette est égal à la taille de la population. Comme on peut le voir à partir de la façon dont la roue est divisée, chaque fois que la roue s'arrête, il y a une forte probabilité que les individus ayant une fitness élevée soient sélectionnés pour l'accouplement dans la prochaine génération.

- Sélection par troncature :

La sélection par troncature est la stratégie de sélection la plus simple. Cette sélection consiste à retenir une partie de la population qui contient uniquement les individus les plus performants. Ces individus sont dupliqués jusqu'à atteindre la taille de population initiale. Par exemple, nous pourrions choisir les 25% des individus les plus aptes parmi une population de 100 individus. Dans ce cas, nous créerions quatre copies de 25 candidats les plus aptes afin de maintenir une population de 100 personnes. Il s'agit d'une stratégie de sélection facile à mettre en œuvre, mais elle peut entraîner une convergence prématurée lorsque des candidats moins aptes sont éliminés sans avoir la possibilité d'évoluer vers quelque chose de mieux. Néanmoins, la sélection par troncature s'avère être une stratégie efficace pour certains problèmes.

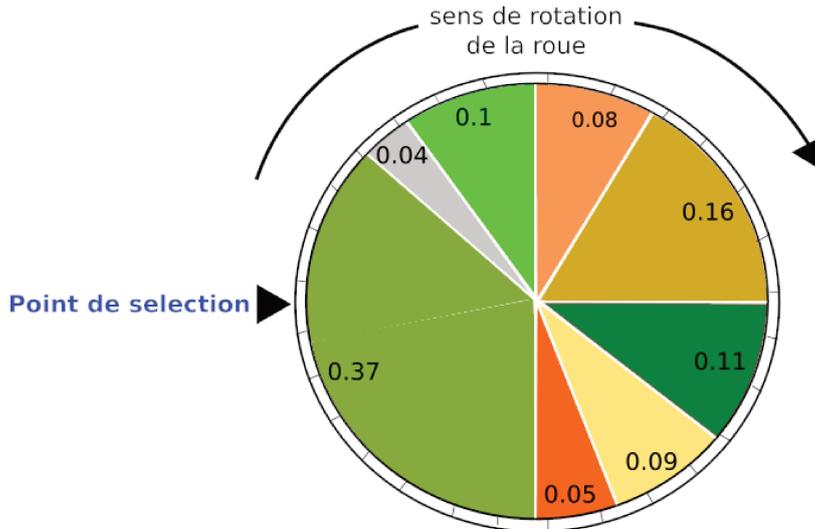


FIGURE 5.21 – Illustration de l'opération de sélection par la roulette.

- Sélection par rangement :

Les individus sont triés en fonction de la valeur de la fonction "fitness" et une probabilité est accordée à chaque individu selon sa position (l'individu au dernier rang est le meilleur), soit :

$$P_i = \frac{1}{N} \left(\eta^- + (\eta^+ - \eta^-) \frac{i-1}{N-1} \right), \quad \text{pour } i = 1 \dots, N.$$

où $\frac{\eta^-}{N}$ est la probabilité de sélection de l'individu le plus fragile et $\frac{\eta^+}{N}$ la probabilité de sélection de l'individu le plus fort. Comme la population est constituée d'un nombre d'individu fixe, les conditions $\eta^+ = 2 - \eta^-$ et $\eta^- \geq 0$ doivent être vérifiées.

- Sélection par tournoi :

Un groupe de k individus sont choisis dans la population et le meilleur individu est sélectionné. Ce processus est répété jusqu'à avoir le nombre d'individu suffisant pour la nouvelle reproduction.

Dans l'algorithme génétique que nous utiliserons par la suite, l'étape de sélection est effectuée en utilisant la technique de la roulette.

5.3.4 Application au contrôle de l'écoulement d'un fluide autour d'un cylindre

Dans cette partie, l'algorithme génétique décrit précédemment, couplé avec les méthodes de réduction de modèle NIMR et HNIMR est appliqué pour contrôler l'écoulement autour d'un cylindre (même configuration qu'à la section 4.5).

a) Test 1 : contrôle optimal pour atteindre une vitesse cible

Soit la fonctionnelle objectif légèrement modifiée suivante

$$\mathcal{J}(\mathbf{u}, Re) = \int_{T_1}^{T_2} \int_{\Omega} |\mathbf{u}(t, x, Re) - \hat{\mathbf{u}}(t, x)|^2 dx dt$$

où \mathbf{u} est la vitesse solution des équations de Navier-Stokes (5.23), $\hat{\mathbf{u}}$ une vitesse cible donnée et $[T_1, T_2]$ l'intervalle temporelle³ sur lequel les bases POD de l'échantillonnage ont été calculées. On rappelle que nous cherchons le nombre de Reynolds optimal qui minimise la fonctionnelle ci-dessus sous la contrainte des équations de Navier-Stokes régissant le problème d'écoulement autour d'un cylindre.

Comme les méthodes NIMR et HNIMR peuvent donner une solution approximée déphasée de la solution de référence, nous introduisons une variable d'optimisation supplémentaire δ caractérisant le déphasage dans le problème d'optimisation sous contrainte. La fonctionnelle à minimiser devient donc

$$\mathcal{J}(\mathbf{u}, Re, \delta) = \int_{T_1}^{T_2} \int_{\Omega} |\mathbf{u}(t + \delta, x, Re) - \hat{\mathbf{u}}(t, x)|^2 dx dt$$

Dans l'algorithme génétique, les méthodes NIMR et HNIMR sont utilisées pour évaluer la fonction "fitness" qui correspond dans notre cas à l'inverse de la fonctionnelle objectif. Une population initiale de taille 100 a été aléatoirement générée telle que le déphasage δ et le nombre de Reynolds Re varient dans l'espace des contrôles admissibles K donné par

$$K = \{(Re, \delta) \in \mathbb{R}_+ \times \mathbb{R}, 100 \leq Re \leq 300 \quad \text{et} \quad -0.3 \leq \delta \leq 0.3\}$$

L'évolution des populations par rapport aux itérations de l'algorithme génétique est représentée sur les figures 5.22, 5.23 et 5.24 pour les nombres de Reynolds cibles $Re_{cible} = 130$, $Re_{cible} = 195$ et $Re_{cible} = 260$. Les points en bleu représentent la population initiale et les points en rouge sont les individus sélectionnés à chaque génération. On constate que le contrôle optimal réduit en utilisant l'algorithme génétique couplé avec la méthode NIMR ou sa variante HNIMR permet bien de retrouver une solution optimale. Ces approches permettent de réduire drastiquement le temps de calcul CPU en comparaison avec le temps CPU nécessaire pour l'approche du contrôle optimal usuelle. En effet avec le même nombre de génération (9 générations) les équations de de Navier-Stokes et leurs équations adjointes doivent être résolus 900 fois (temps estimé à plusieurs jours sur 1 CPU). Le temps de calcul nécessaire pour évaluer une population de taille 100 en utilisant la méthode NIMR demande uniquement 4 minutes. Comme l'algorithme est convergé en 9 générations, le temps CPU est seulement d'une trentaine de minutes pour la NIMR et d'une trentaine de secondes pour la HNIMR. Pour la même population, ce temps de calcul est réduit à 4 secondes en utilisant la version hyper réduite HNIMR. Les résultats de 9^{ème} génération de l'algorithme génétique couplé avec la méthode HNIMR sont répertoriés dans le tableau 5.9. Les paramètres de contrôle obtenues sont en bon accord avec

3. On rappelle que l'intervalle correspond à 8 périodes du détachement tourbillonnaire de l'écoulement.

les paramètres recherchés et les temps de calcul sont drastiquement réduits. Ces résultats sont satisfaisants et prouvent la robustesse des méthodes de réduction de modèle non intrusive NIMR et HNIMR dans l'algorithme génétique. Comme les méthodes NIMR et HNIMR donnent des résultats équivalents et que la méthode HNIMR est plus performante en temps de calcul, les résultats de l'exemple ci après seront traités uniquement avec la méthode HNIMR.

Méthode	Re_{cible}	Temps CPU	(Re, δ)	$\mathcal{J}(Re, \delta)$
HNIMR	130	38 sec	(130.89, 0.14)	8.9×10^{-4}
	195	37 sec	(193.65, 0.12)	7.19×10^{-4}
	260	38 sec	(262.38, -0.04)	1.02×10^{-4}

TABLE 5.9 – Temps CPU, paramètre de contrôle et valeur de la fonctionnelle associés à l'individu dominant de la génération 9 de l'algorithme génétique couplé avec la méthode HNIMR.

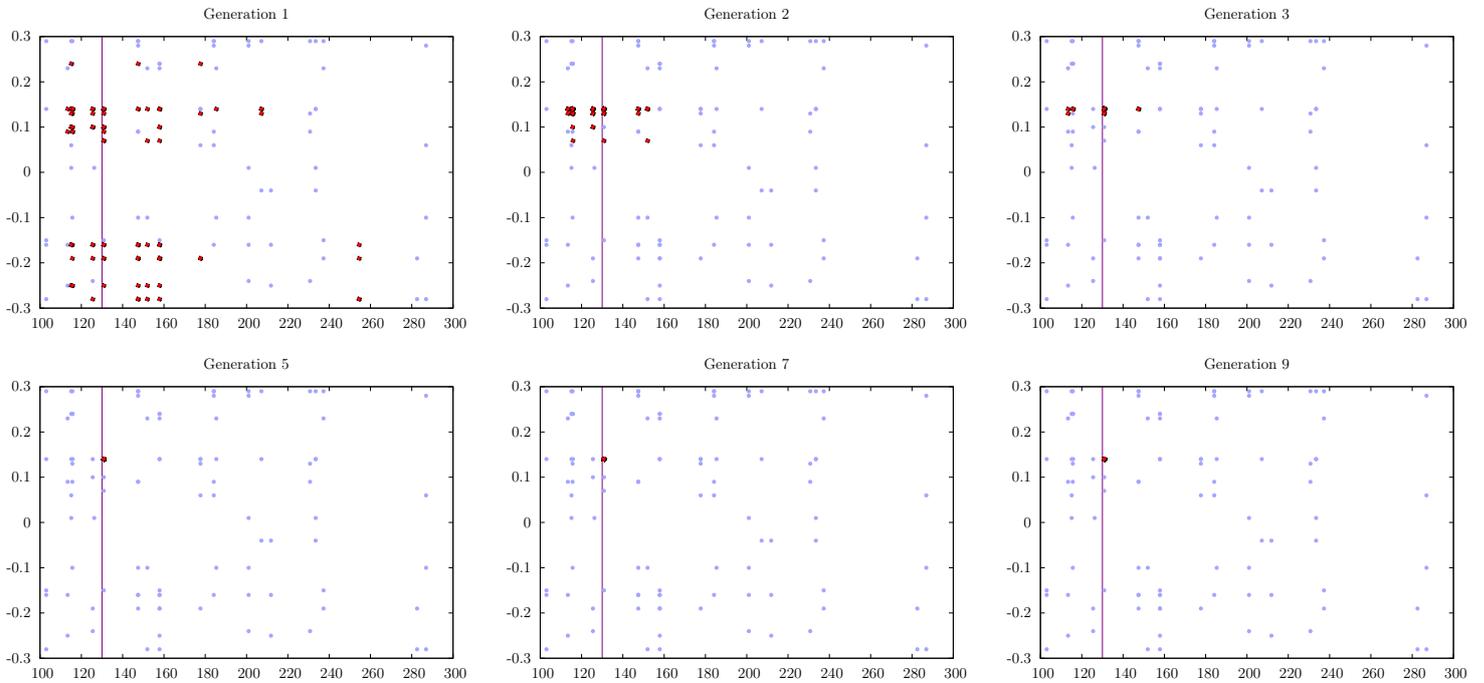


FIGURE 5.22 – Évolution des générations pour $Re_{cible} = 130$.

b) Test 2 : contrôle optimal pour atteindre un coefficient $C_{L,rms}$ cible

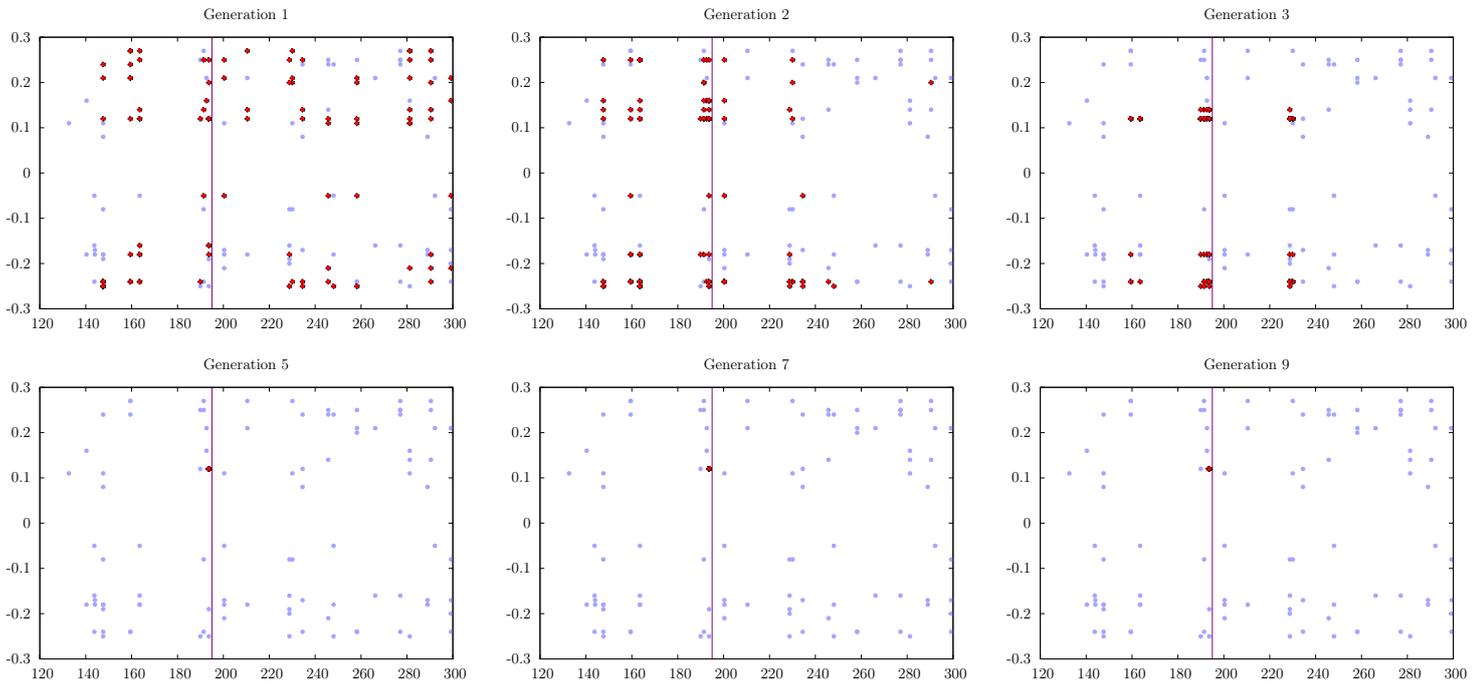
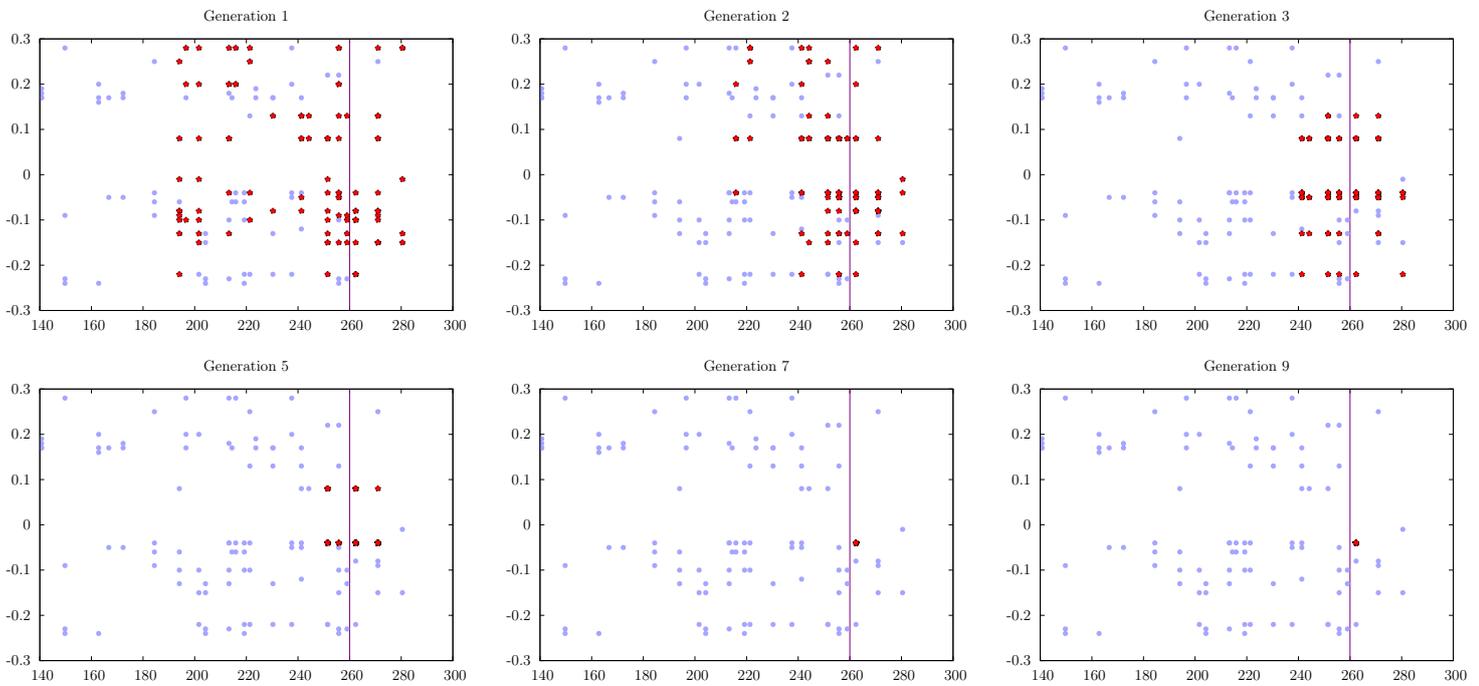
La fonctionnelle objectif étudiée ici est donnée par

$$\mathcal{J}(\mathbf{u}, p, Re) = (C_{L,rms} - \hat{C}_{L,rms})^2 \quad (5.35)$$

telle que

$$C_{L,rms} = \sqrt{\frac{1}{T_2 - T_1} \int_{T_1}^{T_2} C_L dt}$$

On s'intéresse au problème de minimisation de la fonctionnelle (5.35) dont l'objectif est de trouver le nombre de Reynolds qui permet au coefficient $C_{L,rms}$ d'atteindre la valeur

FIGURE 5.23 – Évolution des générations pour $Re_{cible} = 195$.FIGURE 5.24 – Évolution des générations pour $Re_{cible} = 260$.

cible $\hat{C}_{L,rms}$. L'algorithme génétique couplé avec la méthode HNIMR est donc appliquée pour résoudre ce problème. Une population initiale de taille 60 a été aléatoirement générée dans l'espace des contrôle admissibles $100 \leq Re \leq 300$. Par la suite, trois coefficients de portance cibles $\hat{C}_{L,rms}$ correspondant à $Re_{cible} = 135$, $Re_{cible} = 165$ et 225 ont été considérés. L'évolution de la fonctionnelle moyenne pour ces trois cas est représentée sur les figures (5.25), (5.26) et (5.27). On constate que le contrôle optimal réduit en utilisant l'algorithme génétique couplé avec les méthodes de réduction de modèle non intrusives (NIMR et HNIMR) permet de retrouver une bonne approximation du nombre de Reynolds recherché. Les résultats issus de la 7^{ème} génération de l'algorithme génétique couplé avec la méthode HNIMR sont répertoriés dans le tableau 5.10. La fonctionnelle moyenne correspond à la somme des valeurs de la fonctionnelle de tous les individus présents dans la génération divisé par la taille de la population. Le nombre de Reynolds tracé sur ces figures est le nombre de Reynolds ayant la plus grande occurrence dans la génération. Ces résultats montrent la robustesse de la procédure du contrôle optimal réduit en utilisant l'algorithme génétique couplé avec les méthode NIMR ou HNIMR .

Méthode	Re_{cible}	Temps CPU	Re	$\mathcal{J}(Re)$
HNIMR	135	17 sec	135.78	1.3×10^{-3}
	165	17 sec	162.44	2.2×10^{-3}
	225	17 sec	222.23	1.9×10^{-3}

TABLE 5.10 – Temps CPU, paramètre de contrôle et valeur de la fonctionnelle associés à l'individu dominant de la génération 7 de l'algorithme génétique couplé avec la méthode HNIMR.

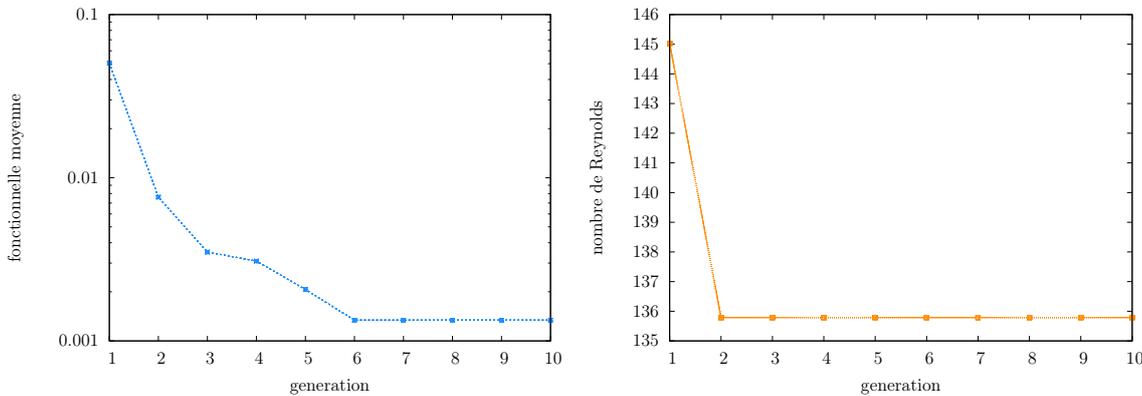


FIGURE 5.25 – Évolution de la fonctionnelle moyenne et du nombre de Reynolds en fonction des générations pour $Re_{cible} = 135$. Le nombre représenté ici correspond celui qui domine dans chaque génération.

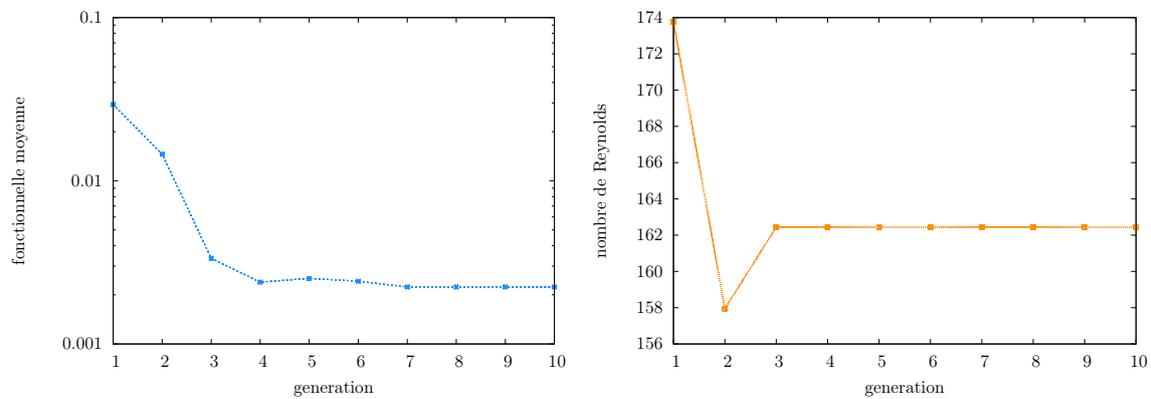


FIGURE 5.26 – Évolution de la fonctionnelle moyenne et du nombre de Reynolds en fonction des générations pour $Re_{cible} = 165$. Le nombre représenté ici correspond celui qui domine dans chaque génération.

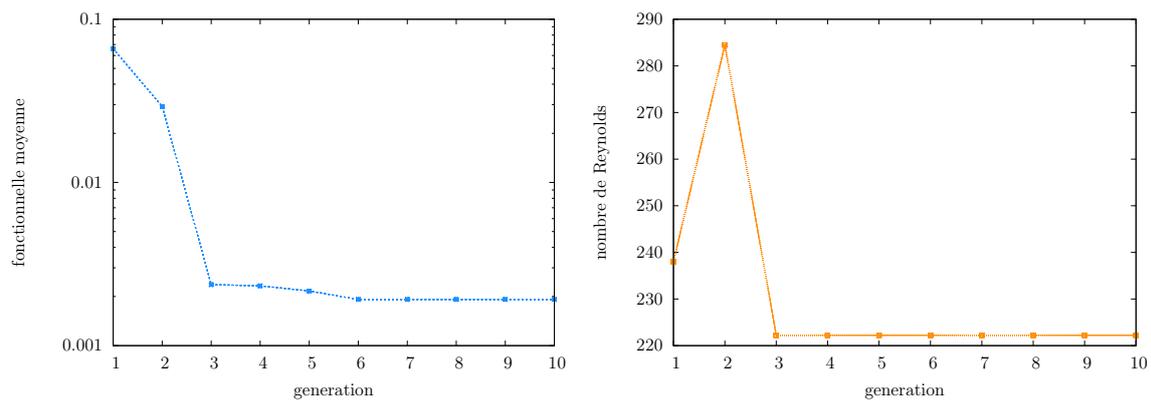


FIGURE 5.27 – Évolution de la fonctionnelle moyenne et du nombre de Reynolds en fonction des générations pour $Re_{cible} = 225$. Le nombre représenté ici correspond celui qui domine dans chaque génération.

5.4 CONCLUSION DU CHAPITRE

Le but de la première partie de ce chapitre était d'illustrer la fiabilité et les gains potentiels des méthodes d'adaptation de bases ITSGM (Interpolation on Tangent Subspace of the Grassmann Manifold) et PGD (Proper Generalized Decomposition) dans l'algorithme de contrôle optimal réduit basé sur les équations adjointes. Ces approches ont été testées sur les problèmes de contrôle optimal des équations de réaction-diffusion et de Burgers. Ces deux approches se sont avérées efficaces que ce soit en termes de précision ou en temps de calcul CPU. En effet, le pourcentage d'erreurs entre les solutions obtenues à l'issue de l'algorithme de contrôle optimal réduit et les solutions réelles varient entre 0.2% et 5% quel que soit le cas étudié. Le temps CPU en comparaison avec le contrôle optimal complet, est quant à lui divisé par 85 dans le cas le plus défavorable et par 350 dans le cas le plus favorable. La méthode ITSGM a également été utilisée pour contrôler l'écoulement autour d'un cylindre pour des nombres de Reynolds $90 \leq Re \leq 450$. Là encore l'approche a permis de retrouver avec une bonne précision les paramètres de contrôle cibles en quelques minutes.

La deuxième partie de ce chapitre a été dédiée à l'étude du potentiel des méthodes de réduction de modèles non intrusives NIMR et HNIMR dans le contrôle optimal réduit en utilisant les algorithmes génétiques. Cette approche a été appliquée sur le problème de contrôle optimal de l'écoulement autour d'un cylindre. L'algorithme génétique couplé avec la méthode HNIMR a permis d'obtenir les solutions cibles avec une bonne précision en moins de 40 secondes. Le choix de la taille de la population initiale, le codage des chromosomes et les opérations génétiques sont des étapes cruciales pour la génération d'un algorithme génétique robuste. L'algorithme génétique utilisé ici est basique. L'utilisation d'un algorithme génétique évolué donnerait donc la possibilité de choisir des populations de petite taille et permettrait par conséquent de diminuer davantage les temps de calcul en espérant atteindre un contrôle optimal quasi-réel.

CONCLUSION GÉNÉRALE

La résolution des problèmes de contrôle optimal nécessite des ressources de calcul et des capacités de stockage très élevées. Pour s'affranchir de ces contraintes, il est possible d'utiliser les méthodes de réduction de modèles comme la POD (Proper Orthogonal Decomposition). L'inconvénient de cette approche est que la base POD n'est valable que pour des paramètres situés dans un voisinage proche des paramètres pour lesquels elle a été construite. Par conséquent, en contrôle optimal, cette base peut ne pas être représentative de tous les paramètres qui seront proposés par l'algorithme de contrôle. Dans le but de s'affranchir de cet handicap, nous avons donc proposé une méthodologie de contrôle optimal réduit en utilisant des modèles réduits adaptatifs obtenus à l'aide des méthodes d'adaptation de bases réduites ITSGM (Interpolation on Tangent Subspace of Grassman Manifold) et PGD (Proper Generalized Decomposition). Nous avons également développé deux méthodes de réduction de modèles non intrusives appelées NIMR (Non Intrusive Model Reduction) et HNIMR (Hyper Non Intrusive Model Reduction). Ces méthodes ont été couplées avec un algorithme génétique pour résoudre rapidement un problème de contrôle optimal.

- Le Chapitre 1 a permis de poser les bases de cette étude, à savoir la définition du problème de contrôle optimal qui sera traité par la suite. Le problème initial d'optimisation sous contraintes a été reformulé sous la forme d'un problème d'optimisation sans contrainte, à l'aide de la méthode des multiplicateurs de Lagrange. Sa résolution numérique se fait par un algorithme de descente de type gradient. Cette approche a été illustrée sur le problème de contrôle optimal d'un écoulement gouverné par les équations de Navier-Stokes. Comme mentionné précédemment, cette stratégie de contrôle est très coûteuse en temps de calcul et en stockage mémoire et ne peut être employée avec des modèles CFD. Il est donc nécessaire de la coupler avec des modèles d'ordre réduit.
- Le Chapitre 2 a été consacré aux techniques de réduction de modèles et à leur utilisation en contrôle optimal réduit. Dans la première partie, quelques méthodes de réduction de modèles ont été rappelées avec un focus sur la POD discrète. C'est cette approche qui a été choisie, par la suite, pour la construction des bases réduites. Le modèle d'ordre réduit a été obtenu par projection de Galerkin des équations d'origine sur la base POD. Dans la deuxième partie du chapitre, le couplage du contrôle optimal avec les modèles réduits a été décrit puis illustré sur l'exemple du contrôle optimal des équations de réaction-diffusion. Deux approches de contrôle réduit ont été étudiées. La première approche appelée "Reduce Then Differentiate" consiste à formuler le problème de contrôle

optimal réduit, à partir de l'approximation POD de la solution du problème d'état, et à dériver les conditions nécessaires d'optimalités associées. La deuxième approche appelée "Differentiate Then Reduce", consiste quant à elle à supposer que les variables d'état et adjointes peuvent être représentées dans deux sous-espaces vectoriels de dimension finie distincts. Les problèmes d'état et adjoint de l'algorithme de contrôle optimal complet sont alors remplacés par leur modèle réduit respectif. L'étude de ces deux approches a révélé que l'approche "Reduce Then Differentiate" est plus précise que l'approche "Differentiate Then Reduce". De plus, elle est plus facile à mettre en œuvre. En effet, l'échantillonnage nécessaire à la construction de la base réduite de l'adjoint peut être très coûteux et très délicat à créer. C'est donc l'approche "état-état" qui a été utilisée par la suite. Bien que l'approche "état-état" soit souvent efficace, il a été souligné que l'utilisation d'une base POD construite pour un paramètre donné peut entraîner une non convergence de l'algorithme de contrôle optimal réduit lorsque les paramètres sortent de la zone de validité de la base POD; d'où la nécessité d'avoir recours à des modèles réduits adaptatifs.

- Le Chapitre 3 a été consacré à l'introduction de deux méthodes d'adaptation de bases réduites. La première méthode d'adaptation est fondée sur l'outil d'interpolation, et la deuxième sur l'outil d'enrichissement. La première, notée ITSGM (Interpolation on Tangent Subspace of Grassman Manifold), consiste à construire un échantillonnage de bases POD pour différents paramètres de contrôle; puis à interpoler ces bases, en passant sur l'espace tangent de la variété de Grassman, pour obtenir la base correspondant à un nouveau paramètre. Contrairement aux méthodes d'interpolation usuelles (Lagrange, RBF, spline ...) la méthode d'interpolation ITSGM est fondée sur des résultats de géométrie différentielle qui permettent de donner un sens au problème d'interpolation des bases POD. La seconde consiste à enrichir en utilisant la méthode PGD, la base associée à un paramètre donné pour qu'elle soit valide pour un nouveau paramètre.

- Le Chapitre 4 a été voué au développement des méthodes de réduction de modèle non intrusives basées sur l'interpolation ITSGM. L'approche proposée, appelée NIMR, consiste à interpoler à la fois les bases spatiales et temporelles obtenues par POD, en utilisant la méthode ITSGM; puis à appliquer un processus de calibration aux bases interpolées afin d'obtenir des bases ayant la structure des bases POD. Le problème de calibration de ces bases revient à résoudre un problème d'optimisation sous contraintes dont la solution est analytiquement déterminée. Une variante de cette méthode notée sous l'acronyme HNIMR, qui permet de réduire drastiquement les temps de calcul de la méthode NIMR, a également été proposée. Ces méthodes ont été appliquées et validées sur le problème de l'écoulement d'un fluide autour d'un cylindre lorsque le nombre de Reynolds change. Il s'est avéré que les résultats obtenus par les méthodes NIMR ou HNIMR sont en accord avec les résultats du modèle complet (erreur moyenne relative inférieure à 1.5% et 6% respectivement pour la vitesse et la pression) et permettent une réduction drastique du temps de calcul CPU. En effet, l'interpolation de bases par la méthode NIMR se fait en moins de 2 secondes et en moins de 0.05 secondes par l'approche HNIMR. Les performances de ces approches ont également été comparées à celles

de l'approche ITSGM/ROM qui correspond au modèle réduit obtenu par projection de Galerkin des équations de Navier Stokes sur la base réduite interpolée par la méthode ITSGM. L'approche ITSGM/ROM s'est avérée plus précise (erreur moyenne relative inférieure à 0.5% et 2% respectivement pour la vitesse et la pression) que les méthodes non intrusives mais moins rapides (la méthode HNIMR est 170 fois plus rapide).

- Le Chapitre 5 a été dédié à la mise en œuvre du contrôle optimal réduit adaptatif basé sur l'équation adjointe ou sur un algorithme génétique. Dans la première partie de ce chapitre, un algorithme de contrôle optimal réduit avec adaptation de bases réduites par les méthodes ITSGM et PGD a été proposé. Cet algorithme consiste à résoudre une séquence de problèmes d'optimisation de taille réduite entrecoupée par la phase d'adaptation de la base réduite. Le potentiel de cette approche a été illustré sur le problème de contrôle optimal des équations de réaction-diffusion et de Burgers. Ces deux approches se sont avérées efficaces que ce soit en termes de précision ou en temps de calcul CPU. En effet, le pourcentage d'erreurs entre les solutions obtenues à l'issue de l'algorithme de contrôle optimal réduit et les solutions réelles varient entre 0.2% et 5% quel que soit le cas étudié. Le temps CPU en comparaison avec le contrôle optimal complet, est quant à lui divisé par 85 dans le cas le plus défavorable et par 350 dans le cas le plus favorable. La méthode ITSGM a également été utilisée pour contrôler l'écoulement autour d'un cylindre pour des nombres de Reynolds $90 \leq Re \leq 450$. Là encore, l'approche a permis de retrouver avec une bonne précision les paramètres de contrôle cibles. La deuxième partie du chapitre a été consacrée au contrôle optimal réduit en utilisant les méthodes de réduction de modèles non intrusives NIMR et HNIMR couplés avec un algorithme génétique. Le problème du contrôle optimal de l'écoulement autour d'un cylindre a été étudié et les résultats ont montré l'efficacité de cette approche. En effet, l'algorithme génétique couplé avec la méthode HNIMR a permis d'obtenir les solutions avec une bonne précision en moins de 40 secondes.

- En perspective de ce travail, nous proposons les travaux suivants :

- Améliorer l'étape d'échantillonnage des paramètres pour lesquels les bases POD sont calculés dans la phase offline. Cet échantillonnage pourrait notamment se faire en utilisant des grilles de Smolyak [159].
- Appliquer les méthodes NIMR et HNIMR à des problèmes numériques plus complexes (écoulements turbulents) et à des données expérimentales.
- Utiliser des algorithmes génétiques plus évolués afin de pouvoir considérer des populations de taille petite et par conséquent réduire davantage les temps de calcul.
- Appliquer d'autres techniques récentes d'interpolation Grassmannienne dans la méthode NIMR [160].

BIBLIOGRAPHIE

- [1] M. R. Hestenes, "Multiplier and gradient methods," *Journal of Optimization Theory and Applications*, vol. 4, pp. 303–320, Nov 1969.
- [2] M. J. D. Powell, "A method for nonlinear constraints in minimization problems," in *Optimization* (R. Fletcher, ed.), pp. 283–298, New York : Academic Press, 1969.
- [3] F. Abergel and R. Temam, "On some control problems in fluid mechanics," *Theoretical and Computational Fluid Dynamics*, vol. 1, 1990.
- [4] A. Iollo, M. Ferlauto, and L. Zannetti, "An aerodynamic optimization method based on the inverse problem adjoint equations," *Journal of Computational Physics*, vol. 173, 2001.
- [5] N. K. Yamaleev, B. Diskin, and E. J. Nielsen, "Local-in-time adjoint-based method for design optimization of unsteady flows," *Journal of Computational Physics*, vol. 229, 2010.
- [6] D. Papadimitriou and K. Giannakoglou, "A continuous adjoint method with objective function derivatives based on boundary integrals, for inviscid and viscous flows," *Computers and Fluids*, vol. 36, 2007.
- [7] J. Céa, *Optimisation Theorie et Algorithmes (Optimization Theory and Algorithms)*. Dunod, 1971.
- [8] J.-C. Culioli, *Introduction à l'Optimisation*. Ellipses, 1994.
- [9] F. Bonnans, J.-C. Gilbert, C. Lemaréchal, and C. Sagastizábal, *Optimisation Numérique : Aspects théoriques et pratiques (Mathématiques et Applications)*. Springer, 1997.
- [10] F. Bonnans and A. Shapiro, *Perturbation Analysis of Optimization Problems*. Springer Series in Operations Research, Springer-Verlag New York, 1 ed., 2000.
- [11] G. Allaire, *Analyse numérique et optimisation : Une introduction à la modélisation mathématique et à la simulation numérique*. 2005.
- [12] P. Colmez, *Eléments d'analyse et d'algèbre (et de théorie des nombres)*. Ecole polytechnique, 2009.
- [13] H. Brezis, *Functional Analysis, Sobolev Spaces and Partial Differential Equations*. Springer, 2011.
- [14] A. R. Conn, K. Scheinberg, and L. N. Vicente, *Introduction to Derivative-Free Optimization*. MOS-SIAM Series on Optimization, 2009.

- [15] J.-B. Hiriart-Urruty and C. Lemarechal, *Convex Analysis and Minimization Algorithms I : Fundamentals*, vol. 305. Springer-Verlag, 1993.
- [16] R. G. Michel Fortin, *Augmented Lagrangian methods : applications to the numerical solution of boundary-value problems*. Studies in mathematics and its applications 15, North-Holland, 1983.
- [17] I. Kazufumi and K. Karl, *Lagrange Multiplier Approach to Variational Problems and Applications*. Philadelphia, PA, USA : Society for Industrial and Applied Mathematics, 2008.
- [18] D. P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*. ACADEMIC PRESS, INC, 1982.
- [19] M. Gunzburger, *Perspectives in Flow Control and Optimization*. Society for Industrial and Applied Mathematics, 2002.
- [20] P. G. Ciarlet, *Linear and Nonlinear Functional Analysis with Applications*. SIAM-Society for Industrial and Applied Mathematics, 2013.
- [21] Y. Liang, W. Lin, H. Lee, S. Lim, K. Lee, and H. sun, "Proper orthogonal decomposition and its applications – part ii : Model reduction for MEMS dynamical analysis," *Journal of Sound and Vibration*, vol. 256, 2002.
- [22] J. P. Thomas, E. H. Dowell, and K. C. Hall, "Three-dimensional transonic aeroelasticity using Proper orthogonal decomposition-based reduced-order models," *Journal of Aircraft*, vol. 40, 05 2003.
- [23] D. J. Lucia, P. S. Beran, and W. A. Silva, "Reduced-order modeling : new approaches for computational physics," *Progress in Aerospace Sciences*, vol. 40, 2004.
- [24] S. Park, J.-J. Lee, C.-B. Yun, and D. J. Inman, "Electro-mechanical impedance-based wireless structural health monitoring using PCA-Data compression and k-means Clustering Algorithms," *Journal of Intelligent Material Systems and Structures*, vol. 19, 05 2007.
- [25] A. AL-Dmour and K. Mohammad, "Active control of flexible structures using principal component analysis in the time domain," *Journal of Sound and Vibration*, vol. 253, 2002.
- [26] V. Buljak and G. Maier, "Proper orthogonal decomposition and radial basis functions in material characterization based on instrumented indentation," *Engineering Structures*, vol. 33, 2011.
- [27] P. De Boe and J.-C. Golinval, "Principal component analysis of a piezosensor array for damage localization," *Structural Health Monitoring*, vol. 2, 6 2003.
- [28] U. Galvanetto, C. Surace, and A. Tassotti, "Structural damage detection based on Proper orthogonal decomposition : Experimental verification," *AIAA Journal*, vol. 46, 07 2008.

- [29] C. Shane and R. Jha, "Proper orthogonal decomposition based algorithm for detecting damage location and severity in composite beams," *Mechanical Systems and Signal Processing*, vol. 25, 2011.
- [30] S. Han and B. Feeny, "Application of Proper orthogonal decomposition to structural vibration analysis," *Mechanical Systems and Signal Processing*, vol. 17, 2003.
- [31] M. A. Singer and W. H. Green, "Using adaptive Proper orthogonal decomposition to solve the reaction–diffusion equation," *Applied Numerical Mathematics*, vol. 59, 2009.
- [32] J. L. Lumley, "The Structure of Inhomogeneous Turbulent Flows," in *Atmospheric turbulence and radio propagation* (A. M. Yaglom and V. I. Tatarski, eds.), pp. 166–178, Moscow : Nauka, 1967.
- [33] N. Aubry, P. Holmes, J. L. Lumley, and E. Stone, "The dynamics of coherent structures in the wall region of a turbulent boundary layer," *Journal of Fluid Mechanics*, vol. 192, p. 115–173, 1988.
- [34] W. Cazemier, R. W. C. P. Verstappen, and A. E. P. Veldman, "Proper orthogonal decomposition and low-dimensional models for driven cavity flows," *Physics of Fluids*, vol. 10, no. 7, pp. 1685–1699, 1998.
- [35] A. Tallet, C. Allery, and C. Leblond, "Optimal flow control using a POD based Reduced-Order Model," *Numerical Heat Transfer, Part B*, vol. 170, 2016.
- [36] P. Druault and C. Chaillou, "Use of Proper orthogonal decomposition for reconstructing the 3D in-cylinder mean-flow field from PIV data," *Comptes Rendus Mécanique*, vol. 335, no. 1, pp. 42 – 47, 2007.
- [37] C. Allery, C. Béghein, and A. Hamdouni, "Applying Proper orthogonal decomposition to the computation of particle dispersion in a two-dimensional ventilated cavity," *Communications in Nonlinear Science and Numerical Simulation*, vol. 10, pp. 907–920, 2005.
- [38] N. Verdon, C. Allery, D. Ryckelnyck, and A. Hamdaoui, "An adaptive ROM approach for solving transfer equations," *Revue européenne de mécanique numérique*, vol. 15, pp. 589–605, 06 2006.
- [39] C. Allery, C. Béghein, and A. Hamdouni, "On investigation of particle dispersion by a POD approach," *International Applied Mechanics*, vol. 44, pp. 110–119, 01 2008.
- [40] C. Béghein, C. Allery, M. Waławczyk, and J. Pozorski, "Application of POD-based dynamical systems to dispersion and deposition of particles in turbulent channel flow," *International Journal of Multiphase Flow*, vol. 58, pp. 97–113, 1 2014.
- [41] M. Pomarede, A. Hamdouni, E. Liberge, E. Longatte, and J.-F. Sigrist, "Tackling FSI simulation for five problems in tube bundle systems with POD approach," vol. 4, pp. 337–343, ASME 2011 Pressure Vessels and Piping Conference, 2011.
- [42] M. Pomarede, E. Liberge, A. Hamdouni, E. Longatte, and J.-F. Sigrist, "Numerical study of fluid-structure interactions in tube bundles with multiphase-POD reduced-

- order approach," vol. 2, pp. 137–, 11th Biennial Conference on Engineering Systems Design and Analysis, 2012.
- [43] S. Sanghi and N. Aubry, "Mode interaction models for near-wall turbulence," *Journal of Fluid Mechanics*, vol. 247, 2 1993.
- [44] G. Berkooz, P. Holmes, and J. L. Lumley, "Intermittent dynamics in simple models of the turbulent wall layer," *Journal of Fluid Mechanics*, vol. 230, 9 1991.
- [45] G. Berkooz, P. Holmes, and J. L. Lumley, "On the relation between low-dimensional models and the dynamics of coherent structures in the turbulent wall layer," *Theoretical and Computational Fluid Dynamics*, vol. 4, pp. 255–269, 09 1993.
- [46] A. Glezer, Z. Kadioglu, and A. J. Pearlstein, "Development of an extended Proper orthogonal decomposition and its application to a time periodically forced plane mixing layer," *Physics of Fluids A Fluid Dynamics*, vol. 1, 1989.
- [47] S. Ciliberto, F. Francini, and F. Simonelli, "Real time measurements of optical disuniformity fields," *Optics Communications*, vol. 54, 1985.
- [48] S. Ciliberto and B. Nicolaenko, "Estimating the number of degrees of freedom in spatially extended systems," *EPL (Europhysics Letters)*, vol. 14, 02 1991.
- [49] R. E. A. Arndt, D. F. Long, and M. N. Glauser, "The Proper orthogonal decomposition of pressure fluctuations surrounding a turbulent jet," *Journal of Fluid Mechanics*, vol. 340, 6 1997.
- [50] C. E. Tinney, M. N. Glauser, and L. S. Ukeiley, "Low-dimensional characteristics of a transonic jet. part 1. Proper orthogonal decomposition," *Journal of Fluid Mechanics*, vol. 612, 10 2008.
- [51] S. Hong and F. Payne, "[american institute of aeronautics and astronautics 19th aiaa, fluid dynamics, plasma dynamics, and lasers conference - honolulu,hi,u.s.a. (08 june 1987 - 10 june 1987)] 19th aiaa, fluid dynamics, plasma dynamics, and lasers conference - development of large-eddy interaction model for inhomogeneous turbulent flows," 1987.
- [52] P. Moin and R. D. Moser, "Characteristic-eddy decomposition of turbulence in a channel," *Journal of Fluid Mechanics*, vol. 200, 3 1989.
- [53] L. Sirovich and J. Rodriguez, "Coherent structures and chaos : A model problem," *Physics Letters A*, vol. 120, pp. 211–214, 1987.
- [54] L. Sirovich, M. Kirby, and M. Winter, "An eigenfunction approach to large scale transitional structures in jet flow," 1990.
- [55] L. Sirovich and H. Park, "Turbulent thermal convection in a finite domain : Part i. theory," *Physics of Fluids A Fluid Dynamics*, vol. 2, 09 1990.
- [56] E. Merzari, H. Ninokata, A. Mahmood, and M. Rohde, "Proper orthogonal decomposition of the flow in geometries containing a narrow gap," *Theoretical and Computational Fluid Dynamics*, vol. 23, 09 2009.

- [57] S. Glegg and W. Devenport, "Proper orthogonal decomposition of turbulent flows for aeroacoustic and hydroacoustic applications," *Journal of Sound and Vibration*, vol. 239, 2001.
- [58] B. O'Donnell and B. Helenbrook, "Proper orthogonal decomposition and incompressible flow : An application to particle modeling," *Computers and Fluids*, vol. 36, 2007.
- [59] B. Moore, "Principal component analysis in linear systems : controllability, observability and model reduction," *IEEE Trans. Autom. Contr.*, pp. 17–32, 1981.
- [60] P. J. Schmid, "Dynamic mode decomposition of numerical and experimental data," *Journal of Fluid Mechanics*, vol. 656, pp. 5–28, 8 2010.
- [61] J. Burkardt, M. Gunzburger, and H.-C. Lee, "Centroidal voronoi tessellation-based reduced-order modeling of complex systems," *SIAM Journal on Scientific Computing*, vol. 28, pp. 459–484, 01 2006.
- [62] C. Kenney and G. Hewer, "Necessary and sufficient conditions for balancing unstable systems," 1987.
- [63] K. Glover, "All optimal Hankel-norm approximation of linear multivariable systems and their L^∞ -error bounds," *Int. J. Control*, pp. 1115–1193, 1984.
- [64] D. Enns, "Model reduction with balanced realizations : An error bound and a frequency weighted generalization," in *IEEE Conference on Decision and Control - Las Vegas, Nevada, USA*, pp. 127–132, 1984.
- [65] L. Sirovich, "Turbulence and the dynamics of coherent structures : Part I, II and III," *Quarterly of Applied Mathematics*, pp. 461–590, 1987.
- [66] S. Puntanen, *Projection Matrices, Generalized Inverse Matrices, and Singular Value Decomposition by Haruo Yanai, Kei Takeuchi, Yoshio Takane*, vol. 79. 2011.
- [67] M. Azaïez, F. B. Belgacem, and T. C. Rebollo, "Recursive POD expansion for reaction-diffusion equation," *Advanced Modeling and Simulation in Engineering Sciences*, vol. 3, pp. 3–, 12 2016.
- [68] G. Tissot, L. Cordier, N. Benard, and B. R. Noack, "Model reduction using dynamic mode decomposition," *Comptes Rendus Mécanique*, vol. 342, pp. 410–416, 06 2014.
- [69] T. W. Muld, G. Efraimsson, and D. S. Henningson, "Flow structures around a high-speed train extracted using Proper orthogonal decomposition and dynamic mode decomposition," *Computers and Fluids*, vol. 57, pp. 87–97, 2012.
- [70] M. Grilli, P. J. Schmid, S. Hickel, and N. A. Adams, "Analysis of unsteady behaviour in shockwave turbulent boundary layer interaction," *Journal of Fluid Mechanics*, vol. 700, pp. 16–28, 6 2012.
- [71] A. Seena and H. J. Sung, "Dynamic mode decomposition of turbulent cavity flows for self-sustained oscillations," *International Journal of Heat and Fluid Flow*, vol. 32, pp. 1098–1110, 2011.

- [72] C. W. Rowley, I. Mezic, S. Bagheri, P. Schlatter, and D. S. Hennigson, "Spectral analysis of nonlinear flows," *Journal of Fluid Mechanics*, vol. 641, pp. 115–, 12 2009.
- [73] P. J. Schmid, "Application of the dynamic mode decomposition to experimental data," *Experiments in Fluids*, vol. 50, pp. 1123–1130, 04 2011.
- [74] A. Quarteroni and G. Rozza, "Numerical solution of parametrized Navier-Stokes equations by reduced basis methods," *Numerical Methods for Partial Differential Equations*, vol. 23, pp. 923–948, 2007.
- [75] G. Rozza, "Reduced basis methods for Stokes equations in domains with non-affine parameter dependence," *Computing and Visualization in Science*, vol. 12, pp. 23–35, 01 2009.
- [76] D. Xiao, F. Fang, A. Buchan, C. Pain, I. Navon, J. Du, and G. Hu, "Non-linear model reduction for the Navier-Stokes equations using residual deim method," *Journal of Computational Physics*, vol. 263, pp. 1–18, 04 2014.
- [77] M. Barrault, Y. Maday, N. C. Nguyen, and A. T. Patera, "An 'empirical interpolation' method : application to efficient reduced-basis discretization of partial differential equations," *Comptes Rendus Mathematique*, vol. 339, pp. 667–672, 2004.
- [78] B. R. Noack, P. Papas, and P. A. Monkewitz, "The need for a pressure-term representation in empirical galerkin models of incompressible shear flows," *Journal of Fluid Mechanics*, vol. 523, pp. 339–365, 1 2005.
- [79] M. Bergmann, C.-H. Bruneau, and A. Iollo, "Enablers for robust POD models," *Journal of Computational Physics*, vol. 228, pp. 516–538, 2009.
- [80] A. J. Chorin, "The numerical solution of the Navier-Stokes equations for an incompressible fluid," *Bulletin of the American Mathematical Society*, vol. 73, pp. 928–932, 11 1967.
- [81] R. Temam, "Sur l'approximation de la solution des équations de Navier-Stokes par la méthode des pas fractionnaires (i)," *Archive for Rational Mechanics and Analysis*, vol. 32, pp. 135–153, 2 1969.
- [82] I. Akhtar, A. H. Nayfeh, and C. J. Ribbens, "On the stability and extension of reduced-order galerkin models in incompressible flows," *Theoretical and Computational Fluid Dynamics*, vol. 23, 07 2009.
- [83] G. Stabile, S. Hijazi, A. Mola, S. Lorenzi, and G. Rozza, "POD-galerkin reduced order methods for cfd using finite volume discretisation : vortex shedding around a circular cylinder," *Communications in Applied and Industrial Mathematics*, vol. 8, 12 2017.
- [84] G. Stabile and G. Rozza, "Finite volume POD-galerkin stabilised reduced order methods for the parametrised incompressible Navier-Stokes equations," *Computers and Fluids*, 2 2018.

- [85] C. Leblond, C. Allery, and C. Inard, "An optimal projection method for the reduced-order modeling of incompressible flows," *Computer Methods in Applied Mechanics and Engineering*, vol. 200, pp. 2507–2527, 2011.
- [86] A. Tallet, C. Allery, C. Leblond, and E. Liberge, "A minimum residual projection to build coupled velocity–pressure POD-rom for incompressible Navier-Stokes equations," *Communications in Nonlinear Science and Numerical Simulation*, vol. 22, pp. 909–932, 05 2015.
- [87] B. Podvin and J. Lumley, "A low-dimensional approach for the minimal flow unit," *Journal of Fluid Mechanics*, vol. 362, pp. 121–155, 5 1998.
- [88] L. Ukeiley, L. Cordier, R. Manceau, J. Delville, M. Glauser, and J. P. Bonnet, "Examination of large-scale structures in a turbulent plane mixing layer. part 2. dynamical systems model," *Journal of Fluid Mechanics*, vol. 441, 8 2001.
- [89] S. Sirisup and G. Karniadakis, "A spectral viscosity method for correcting the long-term behavior of POD models," *Journal of Computational Physics*, vol. 194, pp. 92–116, 2004.
- [90] D. Rempfer, *Kohärenten structuren und chaos beim laminar-turbulenten Grenzschichtumschlag*. Thèse de doctorat, 1998.
- [91] A. Iollo, A. Dervieux, J.-A. Désidéri, and S. Lanteri, "Two stable POD-based approximations to the Navier–Stokes equations," *Computing and Visualization in Science*, vol. 3, pp. 61–66, 05 2000.
- [92] A. Iollo, S. Lanteri, and J.-A. Désidéri, "Stability properties of POD–galerkin approximations for the compressible Navier–Stokes equations," *Theoretical and Computational Fluid Dynamics*, vol. 13, pp. 377–396, 03 2000.
- [93] T. Akman, "Local improvements to reduced order approximations of optimal control problems governed by diffusion convection reaction equation," *Computers and Mathematics with Applications*, vol. 70, pp. 104–131, 2015.
- [94] H. P. Langtangen and A. Logg, *Solving PDEs in Python : The FEniCS Tutorial I*. Springer Publishing Company, Incorporated, 1st ed., 2017.
- [95] E. Liberge and A. Hamdouni, "Reduced order modelling method via Proper orthogonal decomposition (POD) for flow around an oscillating cylinder," *Journal of Fluids and Structures*, vol. 26, no. 2, pp. 292 – 311, 2010.
- [96] M. Bergmann, L. Cordier, and J.-P. Brancher, *Drag Minimization of the Cylinder Wake by Trust-Region Proper orthogonal decomposition*, pp. 309–324. Berlin, Heidelberg : Springer Berlin Heidelberg, 2007.
- [97] M. J. Mifsud, S. T. Shaw, and D. G. MacManus, "A high-fidelity low-cost aerodynamic model using Proper orthogonal decomposition," *International Journal for Numerical Methods in Fluids*, vol. 63, 2010.
- [98] D. Amsallem and C. Farhat, "An interpolation method for adapting reduced-order models and application to aeroelasticity," *AIAA Journal*, pp. 1803–1813, 2008.

- [99] D. Amsallem, J. Cortial, K. Carlberg, and C. Farhat, "A method for interpolating on manifolds structural dynamics reduced-order models," *International Journal for Numerical Methods in Engineering*, vol. 80, 2009.
- [100] D. Shepard, "[acm press the 1968 23rd acm national conference - not known (1968.-..)] proceedings of the 1968 23rd acm national conference on - - a two-dimensional interpolation function for irregularly-spaced data," pp. 517–524, 1968.
- [101] A. Edelman, T. A. Arias, and S. T. Smith, "The geometry of algorithms with orthogonality constraints," *SIAM Journal on Matrix Analysis and Applications*, vol. 20, pp. 303–353, 01 1998.
- [102] R. Wald, "General relativity," *The University of Chicago Press*, 1984.
- [103] N. Boumal and P.-A. Absil, "Low-rank matrix completion via preconditioned optimization on the Grassmann manifold," *Linear Algebra and its Applications*, vol. 475, pp. 200–239, 06 2015.
- [104] T. Connie, M. K. O. Goh, and A. B. J. Teoh, "A Grassmannian approach to address view change problem in gait recognition," *IEEE Transactions on Cybernetics*, pp. 1–14, 2016.
- [105] A. Srivastava and E. Klassen, "Bayesian and geometric subspace tracking," *Advances in Applied Probability*, vol. 36, pp. 43–56, 03 2004.
- [106] H. Cetingul and R. Vidal, "Intrinsic mean shift for clustering on Stiefel and Grassmann manifolds," 2009.
- [107] M. T. Harandi, C. Sanderson, S. Shirazi, and B. C. Lovell, "Graph embedding discriminant analysis on Grassmannian manifolds for improved image set matching," in *CVPR 2011*, pp. 2705–2712, June 2011.
- [108] M. Harandi, R. Hartley, M. Salzmann, and J. Trumf, *Dictionary Learning on Grassmann Manifolds*, pp. 145–172. Cham : Springer International Publishing, 2016.
- [109] A. Absil, R. Mahony, and R. Sepulchre, "Riemann geometry of Grassmann manifolds with a view on algorithmic computation," *Acta Applicandae Mathematicae*, vol. 80, Issue 2, p. 199–220, 2004.
- [110] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*. Princeton, NJ, USA : Princeton University Press, 2007.
- [111] Y. Chikuse, *Statistics on Special Manifolds*, vol. 174. Springer, 2002.
- [112] J. Hamm and D. D. Lee, "Grassmann discriminant analysis : A unifying view on subspace-based learning," in *Proceedings of the 25th International Conference on Machine Learning, ICML '08*, (New York, NY, USA), pp. 376–383, ACM, 2008.
- [113] M. T. Harandi, M. Salzmann, S. Jayasumana, R. Hartley, and H. Li, *Expanding the Family of Grassmannian Kernels : An Embedding Perspective*, pp. 408–423. Cham : Springer International Publishing, 2014.

- [114] B. Wang, Y. Hu, J. Gao, Y. Sun, and B. Yin, *Low Rank Representation on Grassmann Manifolds*, pp. 81–96. Cham : Springer International Publishing, 2015.
- [115] E. Begelfor and M. Werman, “Affine invariance revisited,” vol. 2, pp. pp. 2087 – 2094, 2006.
- [116] I. U. Rahman, I. Drori, V. C. Stodden, D. L. Donoho, and P. Schroder, “Multiscale representations for manifold-valued data,” *Multiscale Modeling and Simulation*, vol. 4, pp. 1201–1232, 01 2005.
- [117] Q. Rentmeesters, P. A. Absil, P. V. Dooren, K. Gallivan, and A. Srivastava, “An efficient particle filtering technique on the grassmann manifold,” in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 3838–3841, March 2010.
- [118] P. Ladevèze and A. Nouy, “On a multiscale computational strategy with time and space homogenization for structural mechanics,” *Computer Methods in Applied Mechanics and Engineering*, vol. 192, no. 28, pp. 3061 – 3087, 2003. Multiscale Computational Mechanics for Materials and Structures.
- [119] P. Ladeveze, *Nonlinear computational structural mechanics : New approaches and non-incremental methods of calculation*. Springer (New York), 1999.
- [120] P. Ladevèze, J.-C. Passieux, and D. Néron, “The latin multiscale computational method and the proper generalized decomposition,” *Computer Methods in Applied Mechanics and Engineering*, vol. 199, no. 21, pp. 1287 – 1296, 2010. Multiscale Models and Mathematical Aspects in Solid and Fluid Mechanics.
- [121] A. Nouy, “generalized spectral decomposition technique to solve a class of linear stochastic partial differential equations.,” *Computer Methods in Applied Mechanics and Engineering, Elsevier*, vol. 196 (45-48), pp. 521–4537, 2007.
- [122] A. Nouy and O. P. L. Maitre, “Generalized spectral decomposition for stochastic nonlinear problems,” *Journal of Computational Physics*, vol. 228, pp. 202 – 235, 2009.
- [123] A. Ammar, B. Mokdad, F. Chinesta, and R. Keunings, “A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modeling of complex fluids,” *Journal of Non-Newtonian Fluid Mechanics*, vol. 139, no. 3, pp. 153 – 176, 2006.
- [124] B. Mokdad, E. Pruliere, A. Ammar, and F. Chinesta, “On the simulation of kinetic theory models of complex fluids using the fokker-planck approach,” *Applied Rheology*, vol. 17, no. 2, pp. 1–14, 2007.
- [125] E. Pruliere, F. Chinesta, and A. Ammar, “On the deterministic solution of multidimensional parametric models using the proper generalized decomposition,” *Mathematics and Computers in Simulation*, vol. 81, no. 4, pp. 791–810, 2010.
- [126] A. Dumon, C. Allery, and A. Ammar, “Proper general decomposition (PGD) for the resolution of Navier-Stokes equations,” *Journal of Computational Physics*, vol. 230, no. 4, pp. 1387 – 1407, 2011.

- [127] A. Dumon, C. Allery, and A. Ammar, "Proper generalized decomposition method for incompressible flows in stream-vorticity formulation," *European Journal of Computational Mechanics*, vol. 19, no. 5-7, pp. 591–617, 2011.
- [128] A. Dumon, C. Allery, and A. Ammar, "Proper generalized decomposition method for incompressible Navier-Stokes equations with a spectral discretization," *Applied Mathematics and Computation*, vol. 219, no. 15, pp. 8145 – 8162, 2013.
- [129] A. Dumon, C. Allery, and A. Ammar, "Simulation of heat and mass transport in a square lid-driven cavity with proper generalized decomposition (pgd)," *Numerical Heat Transfer, Part B : Fundamentals*, vol. 63, no. 1, pp. 18–43, 2013.
- [130] C. Leblond and C. Allery, "A priori space-time separated representation for the reduced order modeling of low reynolds number flows," *Computer Methods in Applied Mechanics and Engineering*, vol. 274, pp. 264 – 288, 2014.
- [131] F. Chinesta, A. Ammar, and P. Joyot, "The nanometric and micrometric scales of the structure and mechanics of materials revisited : An introduction to he challenges of fully deterministic numerical descriptions," *International Journal for Multiscale Computational Engineering*, vol. 6/3, pp. 191–213, 2008.
- [132] B. Bognet, F. Bordeu, F. Chinesta, A. Leygue, and A. Poitou, "Advanced simulation of models defined in plate geometries : 3D solutions with 2d computational complexity," *Computer Methods in Applied Mechanics and Engineering*, vol. 201-204, pp. 1–12, 2012.
- [133] E. Giner, B. Bognet, J. J. Ródenas, A. Leygue, F. J. Fuenmayor, and F. Chinesta, "The proper generalized decomposition (pgd) as a numerical procedure to solve 3D cracked plates in linear elastic fracture mechanics," *International Journal of Solids and Structures*, vol. 50, no. 10, pp. 1710 – 1720, 2013.
- [134] B. Bognet, A. Leygue, and F. Chinesta, "Separated representations of 3D elastic solutions in shell geometries," *Advanced Modeling and Simulation in Engineering Sciences*, vol. 1, p. 4, Jan 2014.
- [135] C. Ghnatios, F. Masson, A. Huerta, A. Leygue, C. E., and F. Chinesta, "Proper generalized decomposition based dynamic data-driven control of thermal processes," *Computer Methods in Applied Mechanics and Engineering*, vol. 213-216, no. 2, pp. 29–41, 2012.
- [136] F. Chinesta, A. Leygue, F. Bordeu, J. V. Aguado, E. Cueto, D. Gonzalez, I. Alfaro, A. Ammar, and A. Huerta, "Pgd-based computational vademecum for efficient design, optimization and control," *Archives of Computational Methods in Engineering*, vol. 20, pp. 31–59, Mar 2013.
- [137] D. Xiao, F. Fang, C. Pain, and I. Navon, "A parameterized non-intrusive reduced order model and error analysis for general time-dependent nonlinear partial differential equations and its applications," *Computer Methods in Applied Mechanics and Engineering*, vol. 317, pp. 868 – 889, 2017.

- [138] D. Xiao, P. Yang, F. Fang, J. Xiang, C. Pain, and I. Navon, "Non-intrusive reduced order modelling of fluid–structure interactions," *Computer Methods in Applied Mechanics and Engineering*, vol. 303, 05 2016.
- [139] V. Shinde, E. Longatte, F. Baj, Y. Hoarau, and M. Braza, "A galerkin-free model reduction approach for the Navier-Stokes equations," *Journal of Computational Physics*, vol. 309, pp. 148 – 163, 2016.
- [140] M. Joyner, "Comparison of two techniques for implementing the Proper orthogonal decomposition method in damage detection problems," *Mathematical and Computer Modelling*, vol. 40, 2004.
- [141] L. Fiabane, M. Gohlke, and O. Cadot, "Characterization of flow contributions to drag and lift of a circular cylinder using a volume expression of the fluid force," *European Journal of Mechanics - B/Fluids*, vol. 30, 2011.
- [142] W. R. Graham, J. Peraire, and K. Y. Tang, "Optimal control of vortex shedding using low-order models. part ii—model-based control," *International Journal for Numerical Methods in Engineering*, vol. 44, 1999.
- [143] M. Bergmann, L. Cordier, and J.-P. Brancher, "Optimal rotary control of the cylinder wake using proper orthogonal decomposition reduced-order model," *Physics of Fluids*, vol. 17, no. 9, pp. 97–101, 2005.
- [144] S. S. Ravindran, "Reduced-order adaptive controllers for fluid flows using POD," *Journal of Scientific Computing*, vol. 15, 12 2000.
- [145] S. Ravindran, "Control of flow separation over a forward-facing step by model reduction," *Computer Methods in Applied Mechanics and Engineering*, vol. 191, 2002.
- [146] S. S. Ravindran, "Reduced-order controllers for control of flow past an airfoil," *International Journal for Numerical Methods in Fluids*, vol. 50, 2006.
- [147] S. Ravindran, "Optimal boundary feedback flow stabilization by model reduction," *Computer Methods in Applied Mechanics and Engineering*, vol. 196, 2007.
- [148] A. Tallet, C. Allery, and C. Leblond, "Optimal flow control using a POD-based reduced-order model," *Numerical Heat Transfer Part B Fundamentals*, pp. 1–24, 06 2016.
- [149] J. Atwell and B. King, "Proper orthogonal decomposition for reduced basis feedback controllers for parabolic equations," *Mathematical and Computer Modelling*, vol. 33, 2001.
- [150] K. Kunisch and S. Volkwein, "Control of the burgers equation by a reduced-order approach using Proper orthogonal decomposition," *Journal of Optimization Theory and Applications*, vol. 102, 08 1999.
- [151] K. Kunisch and L. Xie, "POD-based feedback control of the burgers equation by solving the evolutionary hjb equation," *Computers and Mathematics with Applications*, vol. 49, 2005.

- [152] F. Leibfritz and S. Volkwein, "Reduced order output feedback control design for pde systems using Proper orthogonal decomposition and nonlinear semidefinite programming," *Linear Algebra and its Applications*, vol. 415, 2006.
- [153] L. T. B. Anshul Agarwal, "A trust-region framework for constrained optimization using reduced order modeling," *Optimization and Engineering*, vol. 14, 03 2013.
- [154] M. Bergmann and L. Cordier, "Optimal control of the cylinder wake in the laminar regime by trust-region methods and POD reduced-order models," *Journal of Computational Physics*, vol. 227, pp. 7813–7840, 2008.
- [155] M. Oulghelou and C. Allery, "A fast and robust sub-optimal control approach using reduced order model adaptation techniques," *Applied Mathematics and Computation*, vol. 333, pp. 416 – 434, 2018.
- [156] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI : University of Michigan Press, 1975. second edition, 1992.
- [157] D. E. Goldberg and K. Deb, "A comparative analysis of selection schemes used in genetic algorithms," vol. 1 of *Foundations of Genetic Algorithms*, pp. 69 – 93, Elsevier, 1991.
- [158] T. Weise, "Global optimization algorithms - theory and application," vol. 1, 2009.
- [159] P. Gajda, "Smolyak's algorithm for weighted l^1 -approximation of multivariate functions with bounded mixed derivatives over \mathbb{R}^d ," *Numerical Algorithms*, vol. 40, 12 2005.
- [160] M. M. Rolando, *Thèse : Interpolation sur les variétés Grassmanniennes et applications à la réduction de modèles en mécanique*. Université de La Rochelle, 2018.