Titre: Title:	Prediction of Aerodynamic Coefficients using Proper Orthogonal Decomposition
Auteur: Author:	Benoit Malouin
Date:	2010
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Malouin, B. (2010). Prediction of Aerodynamic Coefficients using Proper Orthogonal Decomposition [Mémoire de maîtrise, École Polytechnique de Montréal]. PolyPublie. https://publications.polymtl.ca/475/

UNIVERSITÉ DE MONTRÉAL

PREDICTION OF AERODYNAMIC COEFFICIENTS USING PROPER ORTHOGONAL DECOMPOSITION

BENOIT MALOUIN
DÉPARTEMENT DE GÉNIE MÉCANIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE MÉCANIQUE)
DÉCEMBRE 2010

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé:

PREDICTION OF AERODYNAMIC COEFFICIENTS USING PROPER ORTHOGONAL DECOMPOSITION

présenté par : MALOUIN Benoit

en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de :

M. PARASCHIVOIU Ion, Ph.D., président

M. TRÉPANIER Jean-Yves, Ph.D., membre et directeur de recherche

M. LAFOREST Marc, Ph.D., membre

À ma famille !

ACKNOWLEDGMENTS

Je voudrais tout d'abord et principalement remercier mes parents ainsi que mon frère qui m'ont toujours fortement encouragé à poursuivre mes études. Nul doute que si ce n'était d'eux, je ne serais pas rendu là. Merci !

Je remercie également mon directeur de recherche, le Professeur Jean-Yves Trépanier, pour m'avoir confié ce projet et, par le fait même, d'avoir eu confiance en moi. Je le remercie également pour l'aide qu'il m'a apportée ainsi que des idées de recherche qu'il a partagées avec moi.

Je remercie également Christophe Tribes, associé de recherche, Martin Gariépy et Sébastien Leclaire, étudiants au doctorat ainsi que Matthieu Melot, ancien étudiant à la maîtrise. Ils m'ont beaucoup apporté, autant du point de vu amical que professionnel. Leur point de vue judicieux et leurs bonnes idées m'ont, à plusieurs occasions, été très utiles. Merci également à Eddy Petro, associé de recherche, pour la génération du maillage que j'ai utilisé dans ce travail.

En dernier lieu, mais non le moindre, je remercie toute l'équipe de la chaire IDEA. Mon séjour à la maîtrise n'aurait pu être plus agréable.

RÉSUMÉ

Une méthode de décomposition orthogonale (POD) a été développée afin d'analyser l'écoulement autour d'un profil de type RAE2822. Le but de l'étude est de déterminer si l'utilisation de telles méthodes peut servir à l'analyse des dérivées de stabilité d'un avion.

Les POD utilisent des solutions numériques existantes, appelés "snapshots", afin de créer des fonctions propres. Le réarrangement de ces dernières par le biais de coefficients de pondération permet d'obtenir une solution interpolée correspondant à d'autres valeurs des paramètres d'intérêts que ceux utilisés pour la création des snapshots. Dans le cas présent, ces paramètres sont le nombre de Mach et l'angle d'attaque. Les snapshots ont été obtenus par l'entremise du logiciel commercial Fluent.

Deux méthodes ont été étudiées afin d'interpoler des solutions autant subsoniques que transsoniques. La première méthode utilisée consiste à dériver les fonctions propres à partir des snapshots les plus près de la solution désirée afin d'éviter les perturbations causées par des comportements trop différents. La seconde méthode consiste à obtenir deux simulations, une visqueuse et non-visqueuse, pour chaque snapshot. La différence est interpolée par POD et additionnée à une solution non-visqueuse préalablement calculée afin de prédire le comportement général de l'écoulement.

Ces méthodes ont été appliquées à la prédiction des coefficients de moment et de portance afin de connaître leur potentiel en stabilité et contrôle. Les résultats de chaque méthode sont comparés avec une solution numérique Navier-Stokes.

ABSTRACT

A POD method has been developped to analyse flow around an RAE2822 airfoil. The goal of the study is to determine if the use of such methods could be useful in the analyse of stability derivatives of an aircraft.

POD use existing numerical simulations, called snapshots, to create eigenfunctions. The eigenfunctions are then combined using weighting coefficients to create a new solution describing the same problem for different values of input parameters. In the present case, these parameters are the Mach number and the angle of attack. The snapshots were generated using FLUENT, a commercial CFD software.

Two methods were tested. The first method is to derive the eigenfunctions from the closest snapshots from the desired solution. The idea is to remove perturbations caused by different behaviours. The second method is to obtain two simulations, viscous and inviscid, for each snapshot. The difference is then interpolated by POD and added to an inviscid simulation previously calculated to predict the general behaviour of the flow.

These methods have been applied to the prediction of the moment and lift coefficients to determine their capabilities in stability and control. Results of each method are compared to CFD.

CONDENSÉ EN FRANÇAIS

0.1 Introduction

Les modèles d'ordre réduit sont des modèles qui représentent les propriétés générales d'un système complexe. À l'opposé, nous retrouvons les modèles de haute fidélité utilisant les méthodes de volumes finis ou d'éléments finis. Dans ces derniers, les équations d'états du système sont résolues.

Malgré la croissance de la puissance informatique, les modèles à ordre réduit sont toujours utiles pour les ingénieurs afin de leur permettre d'obtenir de rapides approximations des solutions. Ces modèles sont également utilisés en optimisation multi-disciplinaire car le temps requis par itération est beaucoup moins élevé qu'une simulation haute fidélité.

Le dernier grand défi en aéronautique est la prédiction des dérivées de stabilité des avions par des méthodes numériques [1]. Le présent travail consiste à déterminer si de tels modèles peuvent être utilisés à cette fin. L'avantage de modèles simplifiés est la réduction du temps de calcul. Une simulation haute fidélité sur un avion complet coûte très cher d'un point de vue informatique. C'est pourquoi ce projet consiste à étudier le potentiel de méthodes simplifiées afin de réaliser cette tâche. Dans le présent travail, l'écoulement autour d'un profil en deux dimensions est étudié. Si la méthode est concluante en deux dimensions, son usage pourra être étendu aux configurations plus complexes en trois dimensions.

La méthode choisie dans ce travail est la décomposition par projection orthogonale ou "Proper Orthogonal Decomposition" (POD). Cette méthode utilise des résultats de simulations haute fidélité appelés "snapshots" et effectue une décomposition en fonctions propres appelées "eigenfunctions". Ces dernières sont combinées entre elles par l'intermédiaire de coefficients de pondération afin de générer de nouvelles simulations à des paramètres d'entrée différents de ceux utilisés pour la génération des snapshots. Dans ce travail, les paramètres retenus sont le nombre de Mach ainsi que l'angle d'attaque.

0.2 Description du problème

Dans ce travail, l'écoulement autour d'un profil de type RAE2822 est étudié. Les snapshots sont crées en variant l'angle d'attaque entre -3 et 5 degrés par incréments de 1 degrés et en variant le nombre de Mach entre 0.3 et 0.8 par incréments de 0.1. Le RAE2822 a été choisi car c'est un profil transsonique qui représente bien le type de profil utilisé de nos jours sur les avions commerciaux. De plus, beaucoup de données sont disponibles sur ce profil, autant expérimentales que numériques. La plage des paramètres (α et $Mach$) a été choisie afin de représenter une enveloppe de vol complète d'un avion commercial.

Un défi a été de trouver un maillage permettant de capter tous les phénomènes présents sur toute la plage de simulations. De plus, une des méthodes étudiée requiert les résultats d'une simulation visqueuse et non-visqueuse sur un même maillage. Il a donc été impossible de générer un maillage possédant un $y+$ de 1 car ce dernier aurait été très difficile à converger en non-visqueux. Le modèle de turbulence de Spalart-Allmaras étant utilisé, il fallait éviter des valeurs de $y+$ entre 1 et 30. Un maillage adapté à une simulation non-visqueuse a donc été utilisé. Plusieurs cellules ont été ajoutées autour du profil afin de capter les chocs partout sur l'intrados et l'extrados.

0.3 Méthodes de Projections Orthogonales

0.3.1 Algorithme

Les principales étapes de la génération des fonctions propres sont résumées ci-dessous.

1. Génération des snapshots. M snapshots sont créés en variant le nombre de Mach et l'angle d'attaque.

2. La matrice de corrélation est construite comme suit :

$$R = U^T U \tag{1}$$

où U est une matrice de dimension $N \times M$ représentant la collection de snapshots. N étant le nombre de cellule du maillage. R est symmétrique.

3. Création des fonctions propres. Les vecteurs et valeurs propres de R doivent être calculées :

$$RV = \lambda V \tag{2}$$

Une fois λ et V connus, les fonctions propres se définissent comme suit :

$$\phi = UV \tag{3}$$

4. Les fonctions propres sont orthogonales entre elles et doivent être normalisées afin d'être orthonormées.

Une nouvelle solution se définit comme suit:

$$K(x, y) = \sum_{i=1}^{M} \eta_i \phi_K^i(x, y) \tag{4}$$

où les "η_i" sont appelés coefficients de pondération et K représente n'importe quelle variable d'état. Il est à noter que les étapes 2 à 4 doivent être répétées pour chaque variable d'état.

0.3.2 Interpolation

Dans ce travail, les snapshots sont reconstruits, ou exprimés en fonction des fonctions propres, en résolvant l'équation suivante pour η_i :

$$K^n(x, y) - \sum_{i=1}^{M} \eta_i^n \phi_K^i(x, y) = 0, \quad n = 1, .., M \tag{5}$$

pour les M snapshots. K représente n'importe quelle variable d'état et n représente le snapshot

n. Chaque snapshot étant reconstruit, les coefficients η_i sont connus pour toutes les valeurs des paramètres. La méthode utilisée interpole par l'intermédiaire d'une spline cubique afin de déterminer les coefficients d'une nouvelle solution pour un angle et un Mach donné. En d'autres mots, les coefficients sont fonctions de l'angle d'attaque et du nombre de Mach comme suit:

$$\eta_i = f_i(Angle, Mach) \tag{6}$$

où f_i est une spline cubique obtenues à partir des données $\eta_i^1,\ \eta_i^2, .., \eta_i^M$.

L'équation (5) se résout à l'aide d'un résoluteur aux moindres-carrés appelé Levenberg-Marquardt. Cette fonction tend à minimiser le résidu au sens des moindres carrés sur toutes les cellules. Cependant, la minimisation du résidu sur toutes les cellules du maillage correspond à une méthode nommée "Least-Square", une méthode plus efficace, nommée "Least-Square Collocation" utilise moins de cellules, par exemple uniquement celles près du profil, afin de calculer le résidu. Cette dernière méthode a été employée. Il a été démontré que l'utilisation de toutes les cellules à l'intérieur d'une bande d'environ une corde autour du profil était la solution la plus efficace. Le nombre de cellules où un résidu doit être évalué a grandemant diminué, tout comme le temps de calcul de reconstruction.

De plus, lors de la création des fonctions propres, les valeurs propres ont été ordonnées en ordre décroissant de valeur. Ce classement implique que les premières fonctions propres contiennent plus de dynamique du système que les fonctions subséquentes. Un compromis a dont été trouvé afin de minimiser le temps de calcul et maximiser la précision. Cet optimal se trouve en utilisant 28 des 54 fonctions propres disponibles. Le nombre d'inconnus que la fonction Levenberg-Marquardt doit trouver se trouve diminué de moitié ce qui augmente grandemant la vitesse de reconstruction des snapshots.

0.4 Méthodes étudiées

0.4.1 Séparation des snapshots

0.4.1.1 Description

Cette méthode étudie l'effet de l'utilisation des 4 et des 16 snapshots les plus près de la solution désirée afin de générer les fonctions propres. En faisant cela, les snapshots présentant des comportements trop différents, par exemple des chocs, seront exclus des fonctions propres et ne pollueront pas la solution finale.

0.4.1.2 Résultats

Cette étude a démontré que, en subsonique, l'utilisation des 16 snapshots les plus près donne de meilleurs résultats que l'utilisation de l'ensemble des 54 snapshots. Cependant, les résultats sont moins bons en transsonique. Une solution serait d'utiliser plus de snapshots dans cette plage où les effets non-linéaires sont prédominants. Une étude plus approfondie montre que les fontions propres ressemblent plus à la solution finale avec 16 snapshots qu'avec 54. Cela explique l'amélioration des résultats.

0.4.2 "Mapping"

0.4.2.1 Description

La seconde méthode consiste à faire un lien entre deux niveaux de fidélité, soit les résoluteurs Euler et Navier-Stokes. Pour chaque combinaison de nombre de Mach et d'angle d'attaque, deux simulations sont effectuées: une visqueuse et une non-visqueuse. La création d'une nouvelle solution requiert qu'une simulation non-visqueuse soit effectuée aux paramètres désirés. La différence entre les simulations Euler et Navier-Stokes est interpolée par POD et additionnée à la simulation non-visqueuse.

La génération d'une simulation non-visqueuse devrait permettre d'obtenir une assez bonne idée de la dynamique de l'écoulement ce qui aidera les POD à interpoler plus efficacement en présence de chocs. Il peut sembler coûteux d'effectuer une telle simulation a priori, mais lorsque des configurations d'avions complets seront étudiées, une simulation non-visqueuse coûtera beaucoup moins cher en temps de calcul qu'une simulation Navier-Stokes.

0.4.2.2 Résultats

Les résultats montrent que cette méthode améliore les résultats, mais seulement pour les angles d'attaques négatifs. Une étude plus approndie montre que cela est causé par le fait qu'il n'y a peu de différence entre la position du choc entre les deux simulations. Pour tous les autres cas, la méthode donne de moins bons résultats que les POD utilisés seuls, sans la méthode. La position du choc ainsi que le comportement général des simulations sont trop différents entre les deux niveaux de fidélité pour faire un lien.

0.5 Conclusion

En ce qui a trait aux dérivées de stabilité, toutes les méthodes étudiées sont aptes à prédire adéquatement le coeffcient C_{l_α}, mais uniquement les méthodes de base et de séparations à 16 snapshots sont aptes à prédire le coeffcient C_{m_α}. En subsonique, la méthode à 16 snapshots s'avère la meilleure.

Il est donc possible de conclure que la dérivation des fonctions propres à partir des snapshots les plus près est la meilleure option et devrait donc être utilisée à l'avenir. Cependant, afin d'adéquatement prédire l'écoulement en régime transsonique, plus de snapshots sont requis dans cette plage.

TABLE OF CONTENTS

DEDICATION iii

ACKNOWLEDGMENTS iv

RÉSUMÉ v

ABSTRACT vi

CONDENSÉ EN FRANÇAIS vii

0.1 Introduction vii

0.2 Description du problème viii

0.3 Méthodes de Projections Orthogonales viii

0.3.1 Algorithme viii

0.3.2 Interpolation ix

0.4 Méthodes étudiées xi

0.4.1 Séparation des snapshots xi

0.4.1.1 Description xi

0.4.1.2 Résultats xi

0.4.2 "Mapping" xi

0.4.2.1 Description xi

0.4.2.2 Résultats xii

0.5 Conclusion xii

TABLE OF CONTENTS xiii

LIST OF FIGURES xvi

LIST OF TABLES xviii

NOTATIONS AND SYMBOLS xix

INTRODUCTION 1

CHAPTER 1 DESCRIPTION OF THE PROBLEM 3
1.1 Equations 3
1.1.1 Navier-Stokes equations 3
1.1.2 Aerodynamic coefficients 4
1.2 Geometry 5
1.3 Mesh 5
1.4 Generation of snapshots 6
1.5 Validation 9

CHAPTER 2 LITERATURE REVIEW 11
2.1 Proper Orthogonal Decomposition 11
2.2 Improving Methods 12
2.2.1 Data splitting 12
2.2.2 Domain decomposition 12
2.2.3 Projection window and Domain sparsing 13
2.2.4 Extrapolation 14
2.2.5 Number of modes 14

CHAPTER 3 PROPER ORTHOGONAL DECOMPOSITION 15
3.1 Algorithm 15
3.2 Interpolation methods 17
3.2.1 Weighted residual methods 17
3.2.2 Interpolated POD 19
3.3 Parametric studies 20
3.3.1 Eigenfunctions 21
3.3.2 Projection window 22
3.3.3 Collocation 24
3.3.4 Reconstruction of snapshots 26
3.3.5 POD vs spline 28
3.3.6 Spacing effects 28
3.4 Results 31

CHAPTER 4 PROPOSED IMPROVEMENTS 34
4.1 Splitting 34
4.1.1 Description 34
4.1.2 Results 35
4.1.3 Discussion 38
4.2 Mapping 42
4.2.1 Description 42
4.2.2 Results 43
4.2.3 Discussion 46
4.3 Stability derivatives 50

CHAPTER 5 CONCLUSION 53

BIBLIOGRAPHY 54

LIST OF FIGURES

Figure 1.1 Geometrical relationship of differential lengths 5
Figure 1.2 Lattice of simulations 6
Figure 1.3 Zoom on airfoil 7
Figure 1.4 Zoom on leading edge 7
Figure 1.5 Computational domain 8
Figure 1.6 Airfoil C_p distribution 10
Figure 3.1 Variation of weighting coefficients in function of Mach number and angle of attack 20
Figure 3.2 Effects of the number of eigenfunctions used for reconstruction 21
Figure 3.3 Cells within a band thickness of 0.4 chord around the airfoil 23
Figure 3.4 Effects of the projection window's size used for reconstruction 23
Figure 3.5 Effects of the number of cells used for reconstruction 25
Figure 3.6 Effects of the number of cells used for reconstruction 25
Figure 3.7 Reconstruction at $Mach = 0.7$ and $\alpha = 4°$ 27
Figure 3.8 Reconstruction at $Mach = 0.4$ and $\alpha = 3°$ 27
Figure 3.9 Studied cases for the snapshots spacing (case $Mach = 0.55$ and $\alpha = 1.5$) 29
Figure 3.10 Spacing effects at $Mach = 0.65$ and $\alpha = 1.5°$ 30
Figure 3.11 $Mach = 0.45$ and $\alpha = 2.5°$ 31
Figure 3.12 $Mach = 0.75$ and $\alpha = 3.5°$ 32
Figure 4.1 Splitting schemes (case $Mach = 0.55$ and $\alpha = 1.5$) 35
Figure 4.2 Contours of pressure 37
Figure 4.3 Contours of pressure $Mach = 0.45$ and $\alpha = 1.5°$ 39
Figure 4.4 Contours of eigenfunctions of pressure $Mach = 0.45$ and $\alpha = 1.5°$ with 16 snapshots 40
Figure 4.5 Contours of eigenfunctions of pressure $Mach = 0.45$ and $\alpha = 1.5°$ with 54 snapshots 40
Figure 4.6 Contours of pressure $Mach = 0.55$ and $\alpha = 3.5°$ 40

Figure 4.7 Contours of eigenfunctions of pressure $Mach = 0.55$ and $\alpha = 3.5°$ with 4 snapshots 41
Figure 4.8 Contours of eigenfunctions of pressure $Mach = 0.55$ and $\alpha = 3.5°$ with 16 snapshots 41
Figure 4.9 Contours of pressure $Mach = 0.75$ and $\alpha = 2.5°$ 41
Figure 4.10 Contours of pressure $Mach = 0.45$ and $\alpha = 2.5°$ 45
Figure 4.11 Contours of pressure $Mach = 0.75$ and $\alpha = 2.5°$ 47
Figure 4.12 Shock position at $Mach = 0.8$ 47
Figure 4.13 Airfoil C_p distribution 48
Figure 4.14 Airfoil C_p distribution 48
Figure 4.15 Contours of eigenfunctions of the difference between inviscid and viscous 49
Figure 4.16 C_l vs α 51
Figure 4.17 C_m vs α 52

LIST OF TABLES

Table 1.1 Air properties 8

Table 1.2 Time comparison for the case $M = 0.73$ and $\alpha = 3^\circ$ 8

Table 1.3 CFD and Experimental results 9

Table 3.1 Least-Square error (in Pa) on pressure 26

Table 3.2 CFD and POD results 26

Table 3.3 Relative ($\%$) errors between POD and Spline for the pressure 28

Table 3.4 Difference ($\%$) between case 1 and case 2 29

Table 3.5 Least-square error on pressure (Pa) between POD and CFD 31

Table 3.6 Relative error ($\%$) on C_l between POD and CFD 32

Table 3.7 Relative error ($\%$) on C_m between POD and CFD 32

Table 4.1 Relative gap ($\%$) between errors on pressure compared to 54 snapshots . . 35

Table 4.2 Relative error ($\%$) between CFD and POD for C_l 36

Table 4.3 Relative error ($\%$) between CFD and POD for C_m 36

Table 4.4 Comparison of Time and Memory 38

Table 4.5 Relative gap ($\%$) between errors with and without mapping 44

Table 4.6 Relative error ($\%$) between CFD and POD for C_l 44

Table 4.7 Relative error ($\%$) between CFD and POD for C_m 45

NOTATIONS AND SYMBOLS

α:	Angle of attack
C_l:	Lift coefficient
C_m:	Moment coefficient
C_p:	Pressure coefficient
$C_{p,u}$:	Pressure coefficient on upper surface
$C_{p,l}$:	Pressure coefficient on lower surface
C_d:	Drag coefficient
R:	Correlation matrix
U:	Matrix of snapshots for any state variable
η_K:	Weighting coefficient for the state variable K
N:	Number of cells in the mesh
λ:	Eigenvalues
e:	Interal energy
$\vec{V}$:	Velocity vector
V:	Norm of velocity vector $\vec{V}$
V:	Eigenvectors
M:	Number of snapshots
ϕ:	Eigenfunctions
K:	Vector of length N asscociate to any state variable
E:	Least-square error
H:	Enthalpy
T:	Temperature
$<,>$:	Inner product

INTRODUCTION

Despite the incredible improvement in computer resources, the needs to generate accurate low-cost Reduced-Order Models (ROM) are always present in the conceptual design process. ROM are models which represent the overall properties of a whole system but with less complexity. These kind of models are opposed to high fidelity or full order models such as finite-volume in computational fluid dynamic (CFD) or finite-element in structure analysis which solve the state equations of the modeled phenomenon. Full order models have the advantage to be very accurate, but are expensive from a computer time point of view.

According to Salas [1], the last challenge in aeronautics is the prediction of stability derivatives with numerical methods. To evaluate these derivatives, full-order simulations of a full-aircraft configuration are needed. Such simulations are costly from a computational time point of view. It is interesting to evaluate if ROM could do the same work, but at a fraction of the computational time. This idea is tested in this work where the flow around an airfoil is studied. If the method gives good results in two dimensions, its use may be extended to more complex, three-dimensional configurations.

The ability of ROM to produce computationally inexpensive, but less accurate solutions, is exploited by engineers to help them to solve design problems and to have access to a good approximation of the final solution. ROM permits them to quickly analyze design solutions so they can converge a lot faster to only a few viable solutions. ROM are also widely used in Multi-Disciplinary Optimization (MDO) because the computational time needed per iteration is noticeably less than a full order simulation.

In this work, ROM are produced using Proper Orthogonal Decomposition (POD). POD is a tool to generate a series of functions that describe the dynamic of the whole system. These functions are called **eigenfunctions** and they are derived from existing, full-order, solutions called **snapshots**. POD produces a number of eigenfunctions equal to the number of snapshots provided. Each eigenfunction is considered a vector of length equal to the number of cells in the mesh. Eigenfunctions are then combined using weighting coefficients to produce a new solution describing the same prob-

lem for different values of input parameters. Methods to determine these weighting coefficients are discussed in this work.

The idea behind POD is to use a minimum set of existing solutions and interpolate between them to generate new ones. The snapshots are generated using either Computational Fluid Dynamics (CFD) or experimental data. In this work, they are generated using FLUENT, a commercial CFD software. Since the minimization of the total computational time is the main objective of using ROM, the set of snapshots must be as small as possible.

Over the years, many authors have proposed ways to improve the accuracy and efficiency of these POD based ROM. However, these methods are mostly focused on the minimization of computer time for the same accuracy. The goal of this project is to propose methods to improve these models, but with a focus on an improvement of the accuracy. Two methods are proposed here. The first is a method to map between two levels of fidelity, given by Euler and Navier-Stokes flow solver. The second proposes to use different snapshots subsets with regard to some flow characteristics such as shocks. The objectives of this work are to improve the prediction of the calculation of the aerodynamic coefficients of an airfoil and to determine if POD based ROM is a good tool for the stability derivatives calculation of a full-aircraft.

The first chapter describes the studied problem. It is followed by a literature review combined with an overview of the POD algorithm. The fourth chapter presents the proposed methods and the results and a conclusion is given in Chapter 5.

CHAPTER 1

DESCRIPTION OF THE PROBLEM

1.1 Equations

This section reviews the equations used in the present problem. The first part contains the Navier-Stokes equations governing the dynamic of the flow. The second part contains the equations used to derive the aerodynamic coefficients from the pressure coefficients on the airfoil.

1.1.1 Navier-Stokes equations

The steady state Navier-Stokes equations in conservative form are:

Continuity

$$\nabla \cdot (\rho \vec{V}) = 0 \tag{1.1}$$

x-Momentum

$$\nabla \cdot (\rho u \vec{V}) = -\frac{\partial p}{\partial x} + \frac{\partial}{\partial x}\left(\mu \frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(\mu \frac{\partial u}{\partial y}\right) \tag{1.2}$$

y-Momentum

$$\nabla \cdot (\rho v \vec{V}) = -\frac{\partial p}{\partial y} + \frac{\partial}{\partial x}\left(\mu \frac{\partial v}{\partial x}\right) + \frac{\partial}{\partial y}\left(\mu \frac{\partial v}{\partial y}\right) \tag{1.3}$$

Energy

$$\begin{aligned} \nabla \cdot \left[\rho\left(e + \frac{\mathbf{V}^2}{2}\vec{V}\right)\right] = {} & \frac{\partial}{\partial x}\left(k\frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(k\frac{\partial T}{\partial y}\right) - \frac{\partial (up)}{\partial x} - \frac{\partial (vp)}{\partial y} \\ & + \frac{\partial}{\partial x}\left[u\left(\mu\frac{\partial u}{\partial x}\right)\right] + \frac{\partial}{\partial y}\left[u\left(\mu\frac{\partial u}{\partial y}\right)\right] \\ & + \frac{\partial}{\partial x}\left[v\left(\mu\frac{\partial v}{\partial x}\right)\right] + \frac{\partial}{\partial y}\left[v\left(\mu\frac{\partial v}{\partial y}\right)\right] \end{aligned} \tag{1.4}$$

where $\mathbf{V} = \sqrt{u^2 + v^2}$ and $\vec{V}$ is the velocity vector.

1.1.2 Aerodynamic coefficients

$$C_p = \frac{p - p_\infty}{\frac{1}{2}\rho V_\infty^2} \tag{1.5}$$

$$C_n = \frac{1}{c}\int_0^c \left(C_{p,l} - C_{p,u}\right) dx \tag{1.6}$$

$$C_a = \frac{1}{c}\int_0^c \left(C_{p,u}\frac{dy_u}{dx} - C_{p,l}\frac{dy_l}{dx}\right) dx \tag{1.7}$$

$$C_l = C_n \cos\alpha - C_a \sin\alpha \tag{1.8}$$

$$C_d = C_n \sin\alpha + C_a \cos\alpha \tag{1.9}$$

$$\begin{aligned} C_m = \frac{1}{c^2}\Bigg[& \int_0^c \left(C_{p,u} - C_{p,l}\right)\left(x - 0.25c\right) dx + \\ & \int_0^c C_{p,u}\frac{dy_u}{dx} y_u dx - \int_0^c C_{p,l}\frac{dy_l}{dx} y_l dx \Bigg] \end{aligned} \tag{1.10}$$

where the quantity dy and dx are shown in Figure 1.1.

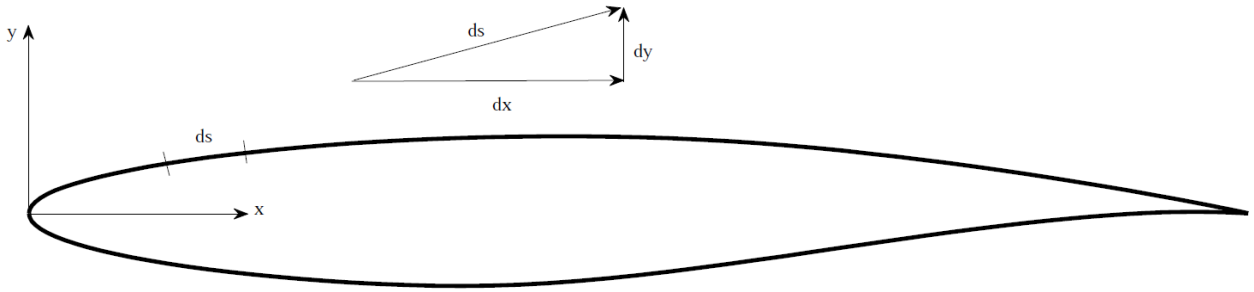
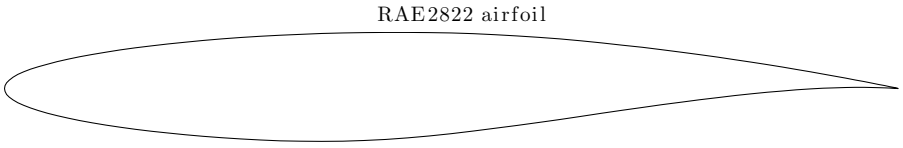


Figure 1.1 Geometrical relationship of differential lengths

1.2 Geometry

In this work, the flow around an RAE2822 airfoil is studied. Snapshots are generated by varying the angle of attack from -3 degrees to 5 degrees with 1 degree increment and by varying the Mach number from 0.3 to 0.8 with increments of 0.1. Figure 1.2 illustrates the lattice of simulations. The points represents the snapshots while the crosses represent where a solution is interpolated by POD. The RAE2822 airfoil has been chosen because it is a transonic airfoil like those used on aircraft nowadays. It has a cove on the lower surface and a flat upper surface. Furthermore, a lot of data are available for this geometry. Since the goal of the project is to analyze stability derivatives over the whole flight envelope, the range of Mach number and angle of attack was chosen in a way to represent this problematic. In this project, the goal is to calculate accurately the aerodynamic coefficients. The results are compared to CFD to measure the accuracy.



1.3 Mesh

Because flow characteristics are quite different over the span of simulations, a mesh suited for all simulations had to be generated. The mesh must be able to deal with the presence of shocks any-

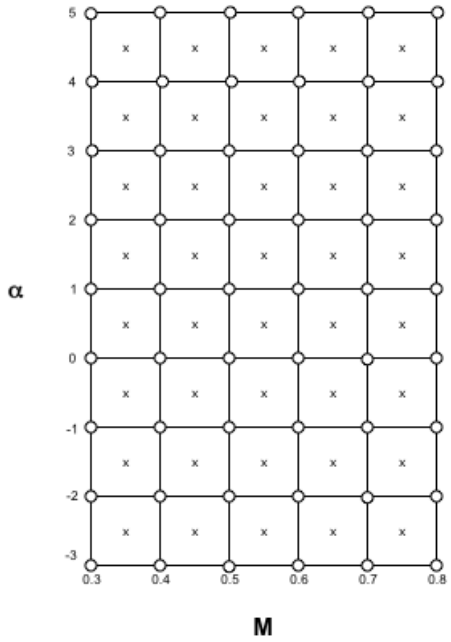


Figure 1.2 Lattice of simulations

where on the upper and lower surfaces. Furthermore, one of the proposed methods requires the same mesh to run Euler and Navier-Stokes simulations and that poses a problem. Since Spalart-Allmaras's turbulence model is used, the first-layer height must be chosen carefully to avoid values of $y+$ between 1 and 30. Moreover, an Euler simulation run with a mesh with $y+$ lower than 1 would be really difficult to converge. The solution is to take a mesh that could be used for an Euler simulation and add cells around the airfoil to be able to capture shocks anywhere. However, this mesh is not suited to predict the boundary layer. An error is therefore inevitable in the aerodynamic coefficients. The resulting mesh is shown in Figures 1.3-1.4. The overall computational domain is shown in Figure 1.5. The mesh contains 33565 cells and 537 cells are located around the airfoil.

1.4 Generation of snapshots

All of the 54 snapshots are generated using FLUENT, a commercial CFD software. The steady, implicit pressure based solver is used for all simulations. Second order discretization scheme is chosen for each state variable. The pressure based solver is chosen because of its flexibility. It can

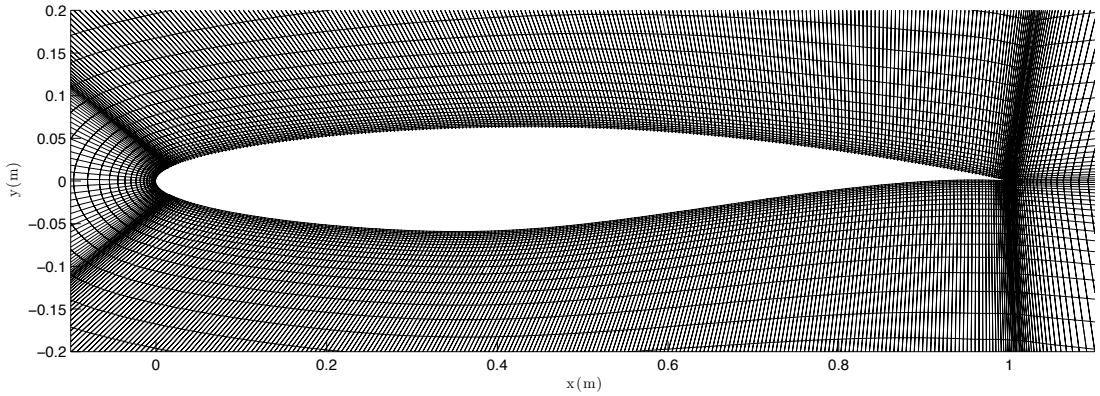


Figure 1.3 Zoom on airfoil

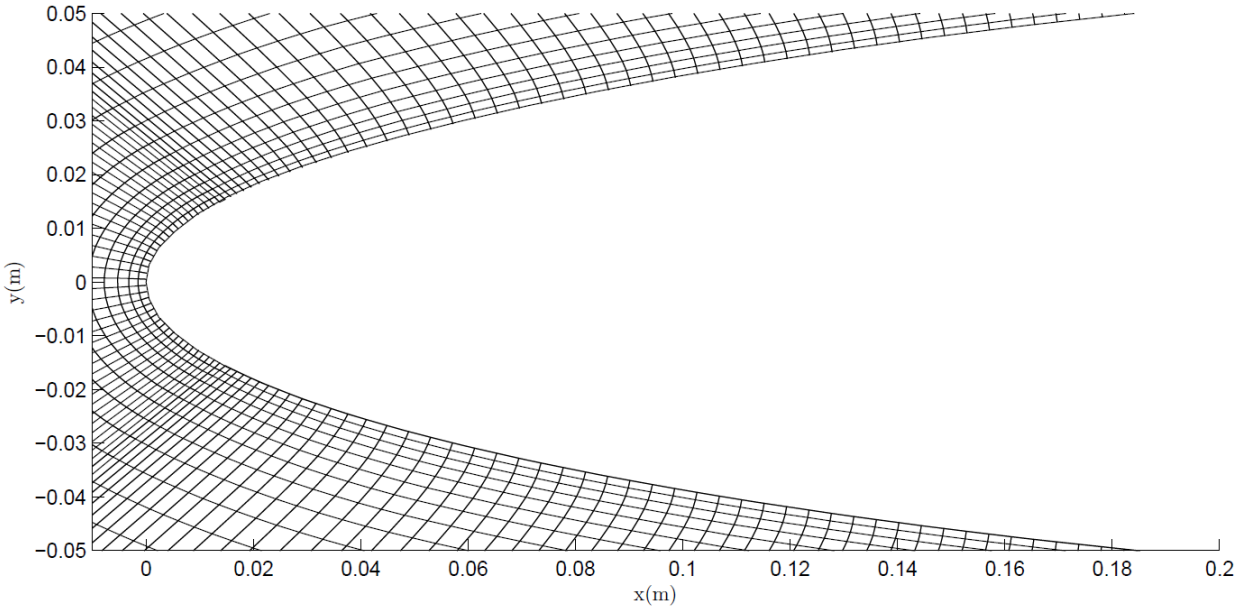


Figure 1.4 Zoom on leading edge

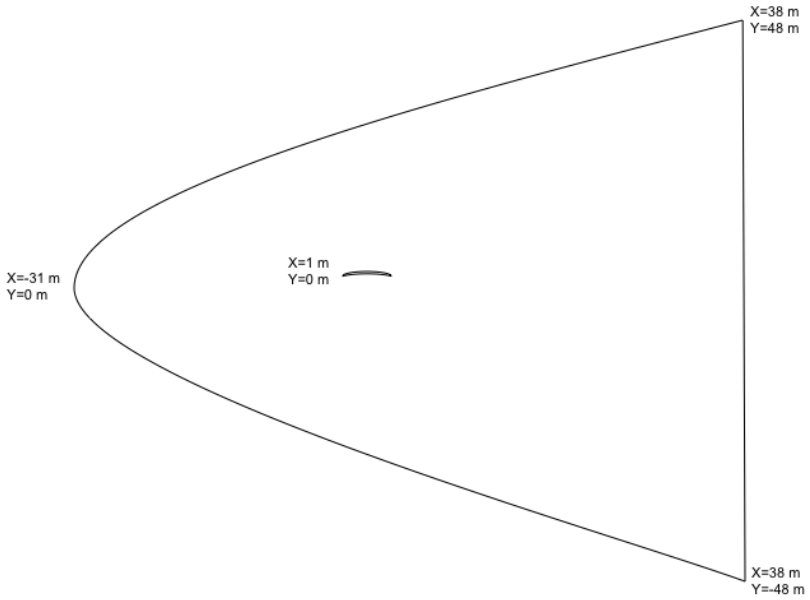


Figure 1.5 Computational domain

be used either for incompressible or compressible flow. Since the process to generate the snapshots is automated via journal files, using the same solver and parameters is more convenient. Spalart-Allmaras turbulence model is used for viscous simulations. A criterion of 10^{-10} for the continuity is used for the convergence. The Reynolds number varies with the Mach number. The air properties are constants and summarized in Table 1.1. Table 1.2 summarizes the time required for a simulation with regard to the solver.

Data has to be extracted from Fluent to supply the algorithm. To achieve this task a "User-Defined Function" has been implemented to access flow solutions. Flow data are extracted and written in a text file to be read by the POD algorithm.

Viscosity μ $(kg/m-s)$	**Temperature** (K)	**Pressure** P_∞ (Pa)	**Density** ρ (kg/m^3)
$1.7894 \times\ 10^-5$	293	101325	1.205

Table 1.1 Air properties

Solver	**Time** (s)
Euler	1000
Navier-Stokes	1360

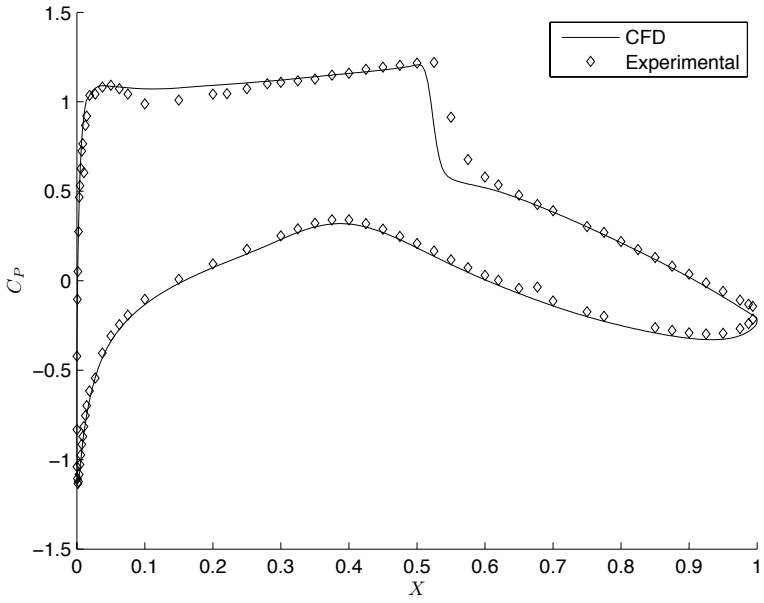
Table 1.2 Time comparison for the case $M = 0.73$ and $\alpha = 3^\circ$

1.5 Validation

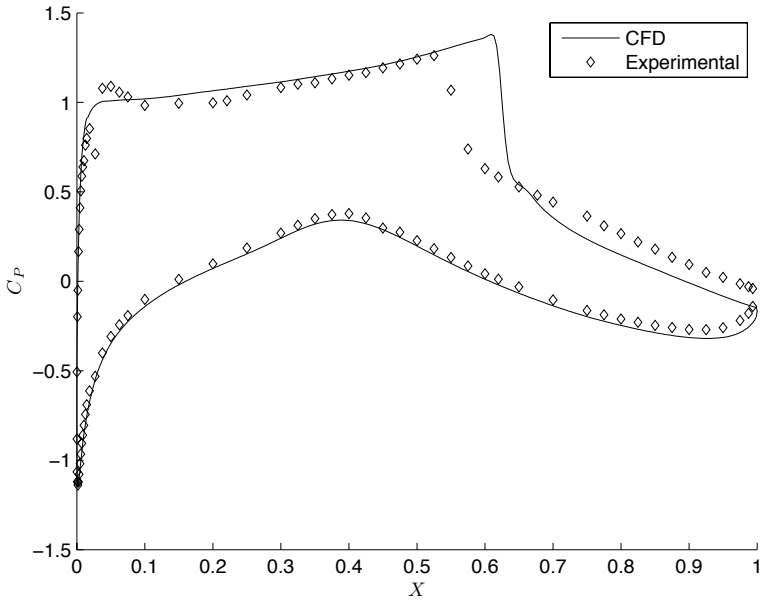
It is possible to compare CFD results to experimental results [2]. Two cases are tested here and they are summarized in Table 1.3. The differences are mainly due to the boundary layer that the mesh is not able to capture. However, Figure 1.6 shows that CFD results are very close to the experimental results and, hence, the mesh can be used for further analysis. The differences are mainly explained by the first layer height. It was chosen to avoid values of $y+$ between 1 and 30 and the consequence is that the boundary layer is not well predicted. The repercussion is that the thickness added by the viscosity is not exact so is the effective shape of the airfoil. This influences the values of the coefficients and the position of the shock. Improving the results would require a finer mesh, but it would be hard to use for inviscid simulations. We are forced to deal with this error. This is a disadvantage of the method.

	Experimental			**CFD (Navier-Stokes)**		
Parameters	C_l	C_d	C_m	C_l	C_d	C_m
$M = 0.725\ \alpha = 2.92°$	0.743	0.0127	0.095	0.7543	0.0131	0.092
$M = 0.750\ \alpha = 3.19°$	0.743	0.0242	0.106	0.8129	0.0279	0.1145

Table 1.3 CFD and Experimental results



(a) $\alpha = 2.92°$, Mach=0.725



(b) $\alpha = 3.19°$, Mach=0.75

Figure 1.6 Airfoil C_p distribution

CHAPTER 2

LITERATURE REVIEW

2.1 Proper Orthogonal Decomposition

Proper orthogonal decomposition, also known as principal component analysis or Karhunen-Loève expansion, comes from the field of statistics [3]. There are many applications of POD: the characterization of human faces [4, 5], the control of temperature in a room [6], the modeling of natural convection [7], the particle dispersion [8] and so on. POD was also applied to the fluid-structure interaction of a full aircraft configuration [9]. Another method, known as Gappy POD is used to deal with missing data or incomplete data set. For example, the complete velocity field could be reconstructed from surface pressure measurements [5, 10, 11]. The latter is also used in image processing.

Holmes et al. [12] used POD to produce ROM of turbulent flow. It has been used by Cazemier et al. [13] to analyse coherent structures of flow in a lid-driven cavity. POD has been widely used for analysing unsteady flows [6, 8, 9, 13–22], incompressible or subsonic steady flows [11, 23–25] and transonic or supersonic steady flows [10, 26–28]. However, before LeGresley et al. [27], all the studies considered the time as the varying parameter. Each snapshot represents the solution of the problem at a different time. They were the first to use design variables such as Mach number instead of time evolution as the varying parameter.

There are other methods to generate ROM. Burkardt et al. [25] used a Centroidal Voronoi Tessellation (CVT) and Lorente et al. [26] used the High-Order Singular Value Decomposition (HOSVD). However, we choose to work with POD because this method has shown good results over the years for this kind of problem. Furthermore, a lot of modifications and improvements have been developed for this method in fluid applications. These modifications are discussed in the next lines.

2.2 Improving Methods

Over the years, many researchers have proposed new developments to improve the accuracy of POD based ROM. The following subsections review these developments.

2.2.1 Data splitting

Cizmas et al. [21] proposed to split the snapshots in two parts. The first part contains snapshots within a Δt where unsteadiness is predominant while the second one contains the remaining snapshots where unsteadiness is less important. They applied this method to the two-phase flow in fluidized bed. Since the dynamic of the problem is quite different from a part to another, the eigenfunctions associated with each part will describe the problem in a better way than all the eigenfunctions combined together. Indeed, their results have shown that the reconstruction of a snapshot needs fewer eigenfunctions when snapshots are divided rather than combined. This principle can be transposed to steady problems such as transonic aerodynamics. Data could be splitted in parts relative to the presence or not of shocks. This strategy has been applied to our case and will be presented in section 4.1.

2.2.2 Domain decomposition

Lucia et al. [14] proposed to split the flow domain in multiple sections. The goal is to isolate the portion of the domain where POD have difficulties to interpolate. In their case, this was the shock part of the flow. They split the domain in three parts. The first one contains the area where a shock is expected, the second one is located behind the shock and the third one contains the remaining cells. Two approaches were used and compared. In the first one, POD is performed on the second and third parts whereas full-order simulation is performed in the first part where a shock is expected. The second approach was to use a ROM in the first part instead of a full-order simulation. The first approach was the better one and gives very accurate results for the 2-D blunt body problem. The biggest disadvantage of the domain decomposition method is that a complex treatment has to be made for internal boundaries. Internal boundaries separate the cells

that belong to one or another part created by the decomposition. Solution must be smoothened by overlapping the domain. The reader may consult Ref. [14] for further details. LeGresley et al. [27] used domain decomposition for airfoil analysis. They treated the internal boundaries by placing a series of sources (or sinks) of mass, momentum and energy to keep conservation through this interface. They showed good improvement for transonic flows while using domain decomposition. Their POD based ROM combined with domain decomposition was able to accurately predict the position of the shock by using only 5 snapshots to construct their eigenfunctions (Mach number varying from 0.3 to 0.7 with increments of 0.1). However, the treatment of the internal boundaries is difficult to implement in commercial software. This is why this method is not used in the present work.

2.2.3 Projection window and Domain sparsing

Alonso et al. [24] developed a POD based ROM of the backward facing step. They proposed to base their ROM on a portion of the computational domain, the projection window. In their case, they used the portion of the domain behind the step. This highly decreases the number of cells in the computation of the eigenfunctions which leads to a reduction in computer time. Furthermore, they have shown that this technique leads to errors of the same order of the ROM based on the whole domain.

Because this technique is promising, it has been applied to the problem at hand by considering only cells within a wall distance of two chords around an airfoil as the computational domain for POD. Section 3.3.2 tests this idea.

An other technique developed by Alonso et al. [24] and called domain sparsing consists in using a limited number of cells for the computation of the eigenfunctions within the projection window. They observed no loss of accuracy in the results but a noticeable reduction in computer time.

2.2.4 Extrapolation

POD based ROM are accurate if the interpolated solution is not too far from the snapshots. If a solution is needed outside the span of snapshots, the direct extrapolation will give poor results. Qamar et al. [29] have proposed to derive the eigenfunctions, extrapolate closely to the upper (or lower) limit snapshot and to re-derive the eigenfunctions while including the extrapolated solution as any other snapshots. This process is repeated until the desired value of the parameter of interest is reached. They applied this technique to an axisymmetric surface-mounted triangular protuberance in supersonic flow. This technique leads to a reduction of the error by 50%. However, this method is computationally expensive since eigenfunctions have to be derived at each iteration. Though, it is still less expensive than solving the partial differential equations.

2.2.5 Number of modes

The last way to improve the efficiency of POD based ROM is to choose wisely the number of modes - eigenfunctions - used to construct a new solution. This is an iterative procedure because there is no way to know in advance this number. POD modes are constructed in a way that the i^{th} mode always contains more information than the $(i+1)^{th}$ mode. So there is an optimal number of modes to be used. It depends on the problem, the degree of unsteadiness and the linearity of the response of the problem to a perturbation of a parameter. Section 3.3.1 studies that effect in more details.

The next section presents the algorithm that we have used to generate the POD.

CHAPTER 3

PROPER ORTHOGONAL DECOMPOSITION

This section is divided in three subsections. First, the method to compute the eigenfunctions is presented. Secondly, all the methods to generate new solutions from the eigenfunctions are described. The last part studies some parameters that influence the accuracy of POD.

3.1 Algorithm

The goal of the POD algorithm is to generate eigenfunctions from existing solutions. These solutions are generated by varying flow parameters, in our case, inlet Mach number and angle of attack.

The method to generate the eigenfunctions is based on the so-called **method of snapshots** developed by Sirovich [30]. The following steps to construct the POD eigenfunctions are described in Ref. [23]:

1. The initial step is to run M simulations or snapshots by varying the inlet Mach number and angle of attack.

2. The second step is to construct the correlation matrix R as follows:

$$R = U^T U \tag{3.1}$$

   where U is a matrix representing the collection of snapshots for any state variable of the flow simulation. This matrix has the dimensions $N \times M$ where N is the number of cells in the mesh and M is the number of snapshots. Note that R is a symetric matrix.

3. The third step is to calculate the eigenfunctions. To do so, eigenvalues λ and eigenvectors V

of R must be determined by solving

$$RV = \lambda V \tag{3.2}$$

Once λ and V are known, eigenvalues are sorted in descending order and M eigenfunctions can be constructed:

$$\phi = UV \tag{3.3}$$

4. Eigenfunctions have to be normalized as follows:

$$\phi^{Normalized} = \frac{\phi}{||\phi||} \tag{3.4}$$

and if every step was done correctly, they should be orthonormal:

$$\phi_i^T \phi_j = \delta_{ij} \tag{3.5}$$

where δ_{ij} is the Kronecker delta.

The next step is to construct a new solution from the eigenfunctions. This solution is defined as:

$$K(x,y) = \sum_{i=1}^{M} \eta_i \phi_K^i(x,y) \tag{3.6}$$

where η's are the weighting coefficients chosen to construct the new solution, K is any state variable and ϕ_K are the eigenfunctions associated with this state variable. These coefficients should be chosen in a way to minimize the error, where the error is calculated with the relation:

$$E = \sqrt{\frac{1}{N}\sum_{j=1}^{N}(K_j^{CFD} - K_j^{Reconstructed})^2} \tag{3.7}$$

A solution is thus a combination of eigenfunctions. By using 54 snapshots, one could use a max-

imum of 54 eigenfunctions to create a new solution. However, it is not necessary to use all of the eigenfunctions since they are sorted with regard to value of the eigenvalue of the correlation matrix. In other words, the k^{th} eigenfunction contains more information than the $(k+1)^{th}$. This is why this method is relatively inexpensive from a computer time point of view. From a number of unknowns equals to the number of cells in the mesh times the number of flow variables n; we pass to only $M \times n$ unknowns, where M is the number of eigenfunctions used. In our case, we pass from $33565 \times 6 = 201390$ to a maximum of $54 \times 6 = 324$ unknowns.

The next section describes the methods to determine the weighting coefficients η_i.

3.2 Interpolation methods

This section reviews the methods to reconstruct a solution from the eigenfunctions derived in the previous section. Two types of methods exist: weighted residual methods and interpolation methods. The first one is more accurate but more expensive in the sense that it needs a projection into the governing equation. The second method does not need such a projection, but is in general less accurate. However, this latter method is easier to implement. Both type of methods are presented in this section.

3.2.1 Weighted residual methods

Galerkin projection

A complete mathematical development was done for the Galerkin and direct projection and the reader may consult [17] for further details. They are commonly used to analyze unsteady flows [8, 13, 16–18]. The following lines present an overview of this method even though it has not been used in this work.

Let us consider Euler's equations in integral form:

$$\frac{d}{dt}\iint_{\Omega} A\,dx\,dy + \oint_{\partial\Omega}(Bdy - Cdx) = 0 \tag{3.8}$$

where A is the vector of conserved flow variables, B and C are the Euler flux vectors:

$$A = \begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ \rho H \end{Bmatrix} \quad B = \begin{Bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uH \end{Bmatrix} \quad C = \begin{Bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vH \end{Bmatrix}$$

Since we are interested in steady solutions, the first term is equal to zero, hence:

$$\oint_{\partial\Omega}(Bdy - Cdx) = R_i, \quad i = 1, .., N \tag{3.9}$$

where R_i is the residual and N is the number of cells in the mesh. The Galerkin projection is expressed as follows:

$$\langle R, \phi \rangle = 0 \tag{3.10}$$

where $\langle\ ,\ \rangle$ is the inner product and ϕ represents the eigenfunctions. The state variables contained in the terms B and C in Eq. (3.9) are replaced by the POD expansions like the one of Eq. (3.6). Equation (3.10) is in fact four coupled equations (Euler) that need to be solved for the weighting coefficients η_i. Since R is projected into the eigenfunctions ϕ, the weighting functions are the eigenfunctions. Once again, this method is not used in this work so the full development is omitted.

The projection in the governing equation can be done by using a solver. This idea has been tested. It is an optimization process in which the residual R from Eq. (3.9) is minimized. A new solution is constructed from different values of η_i supplied by the optimization algorithm. This solution is supplied in the solver which calculates the residual. The process goes until R is close to 0. This method is not feasible in our case because a commercial software (Fluent) is used. At each iteration Fluent has to initialize itself, set the problem, load the solution and calculate the residual.

The time per iteration was too high and running directly the full-order simulation in Fluent was faster. Furthermore, when dealing with commercial softwares, it is not possible to know exaclty how the residuals are calculated. Moreover, when iterating, Fluent does not write all variables to the output file. If one tries to supply the software with homemade file rather than Fluent built-in format, it would not work because these latter files contain more details about the solution. Finally, our conclusion is that this method could only be used with an in-house or open-source flow solver.

3.2.2 Interpolated POD

The next method, called Interpolated POD [28], is simpler in the sense that it does not need a projection into the governing equations. The idea is to reconstruct each snapshot in function of the eigenfunctions. So, each solution or snapshot is expressed in the following form:

$$K^n(x,y) - \sum_{i=1}^{M} \eta_i^n \phi_K^i(x,y) = 0 \tag{3.11}$$

where n corresponds to the n^{th} snapshot and K represents any state variable. To solve Eq. (3.11) for η_i, the Levenberg-Marquardt algoritm is used for non-linear least-square minimization. This equation is solved for each snapshots and for each state variable. In other words, this equation is solved $54 \times 6 = 324$ times.

Once each solution $K^n(x,y)$ is reconstructed, all coefficients η_i^n are determined. After that, the method interpolates between the coefficients to obtain new solutions. In our case, for each eigenfunction, the weighting coefficients η_i are functions of the flow parameters (Mach number and angle of attack) as depicted in Figure 3.1. Weighting coefficients are interpolated with a cubic spline.

All the algorithm is coded in Matlab. The *interp2* function which uses cubic spline is used to interpolate and the *lsqnonlin* function is used for the non-linear least-square (Levenberg-Marquardt)

minimization.

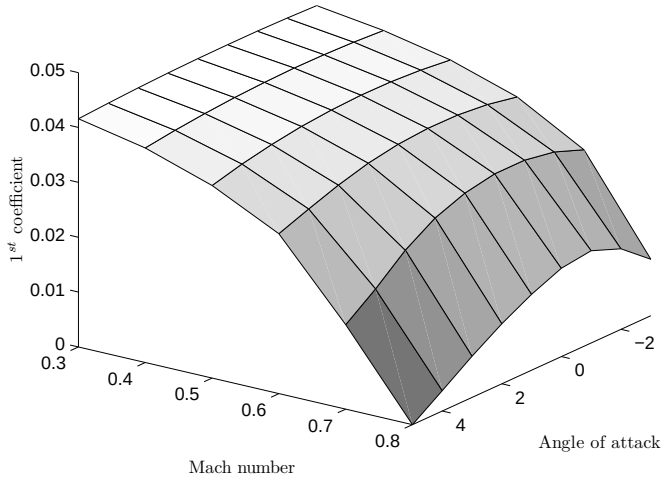


Figure 3.1 Variation of weighting coefficients in function of Mach number and angle of attack

Two methods are then available to reconstruct the snapshots, which are described below.

Least Square

This method, developed by Gauss and Legendre, consists in minimizing the residual R in a least square sense over all cells in the mesh.

Least Squares Collocation

This method is a hybrid between collocation and least square. The residual is evaluated at a number of cells varying from the number of eigenfunctions to the number of cells in the mesh. This method minimizes the residual R in a least square sense over all the selected cells. This method has been identified in the litterature as the best approach for the type of problem at hand. However, the number of cells required has to be determined.

3.3 Parametric studies

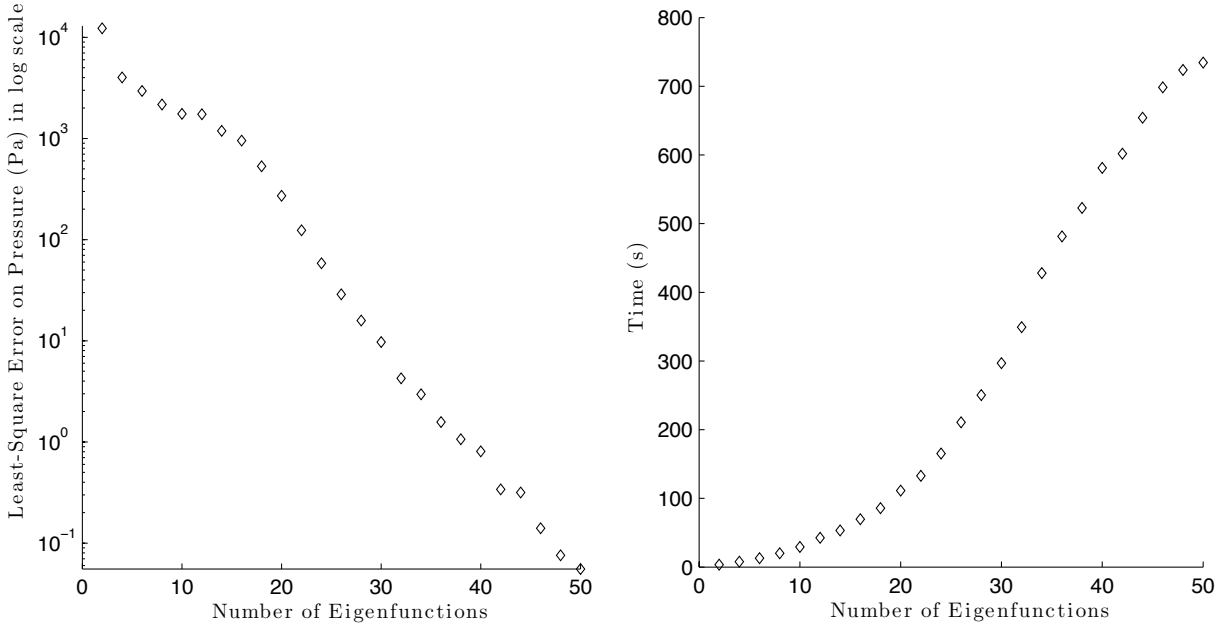
In this section, various variants of the basic POD are investigated to arrive at a fixed methodology.

3.3.1 Eigenfunctions

Like we said previously, the number of eigenfunctions used to reconstruct the snapshots may be varied. There is an optimal compromise between accuracy and time. Eq. (3.11) is solved for each snapshot and for each state variable. For a given snapshot, the number of unknowns is equal to the number of state variable times the number of eigenfunctions used. Using more eigenfunctions will give more accurate results, but the time to solve the equation will be higher. Furthermore, beyond a certain point, the error becomes asymptotic. The error is calculated with the relation:

$$E = \sqrt{\frac{1}{N}\sum_{j=1}^{N}(K_j^{CFD} - K_j^{Reconstructed})^2} \tag{3.12}$$

where N is the number of cells, K is any state variable and $K_j^{Reconstructed}$ is calculated with Eq. (3.11).



(a) Error vs number of eigenfunctions

(b) Time vs number of eigenfunctions

Figure 3.2 Effects of the number of eigenfunctions used for reconstruction

To study the effect of the number of eigenfunctions, all snapshots are reconstructed from a given number of eigenfunctions and the error is calculated for each reconstructed snapshot. Over all

snapshots, the one with the highest error is plotted in the graph presented in Figure 3.2(a). The least-square error must be of the magnitude $O(10^1\ Pa)$ for the pressure. This means that the sum of the least-square error over the 33565 cells of the mesh is below 0.01% of the atmospheric pressure (101 325 Pa). This level of precision is needed because a least-square error greater than 50 Pa leads to visible differences in the C_p distribution and in the pressure contours. The pressure has been chosen because the goal of the method is to predict the aerodynamic coefficients and they are calculated from the pressure coefficients.

It is possible to see in Figure 3.2(a) that around 28 eigenfunctions is enough. Indeed, using 28 eigenfunctions leads to a least-square error of the magnitude $O(10^1\ Pa)$ for the pressure. The criterion is satisfied with 28 eigenfunctions. Furthermore, Fig 3.2(b) shows that using 28 eigenfunctions leads to a reduction of about 70% in computational time compared to 54 snapshots.

The reconstruction time with 28 eigenfunctions is 400 seconds but the reconstruction is performed only once. After that, one could generate any new solution within the span of snapshots. Hence, the efficiency of the POD method is best when a large number of new solutions are requested. It takes 400 seconds plus 1 second to create a new solution and it takes 400 seconds plus i seconds to create i new solutions.

3.3.2 Projection window

As mentionned in Section 2.2.3, only a portion of the domain could be used to derive the eigenfunctions. This section studies the effect of the number of cells of the projection window for deriving the POD eigenfunctions. Cells not contained in that window are unused. The projection window is built by taking all cells within a given distance to the airfoil. An example is shown in Figure 3.3. In that case, all cells with a wall-distance equal or below 0.4 chord length are used to derive the eigenfunctions.

As seen on Figure 3.4(b), this method reduces computational time. The gain in time is not considerable as the window varies from 5 to 50 chord and more. It takes 20 chord length to the error to become below $10\ Pa$ and the time required is around 180 seconds. If all the cells are used, it takes

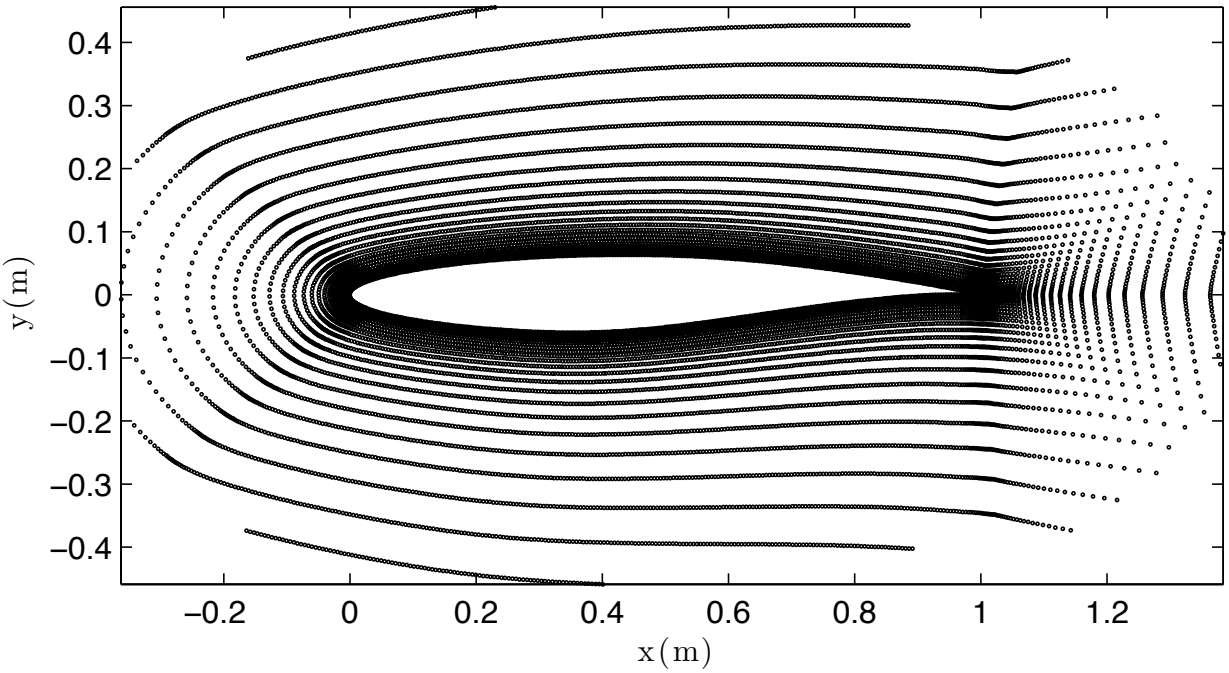
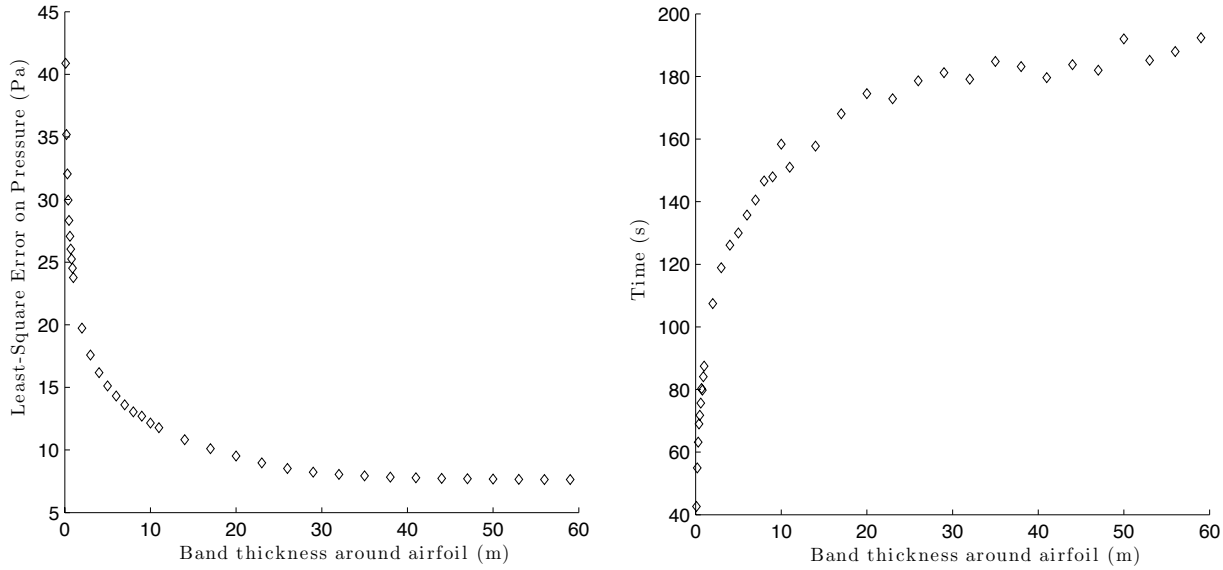


Figure 3.3 Cells within a band thickness of 0.4 chord around the airfoil



(a) Error vs band thickness

(b) Time vs band thickness

Figure 3.4 Effects of the projection window's size used for reconstruction

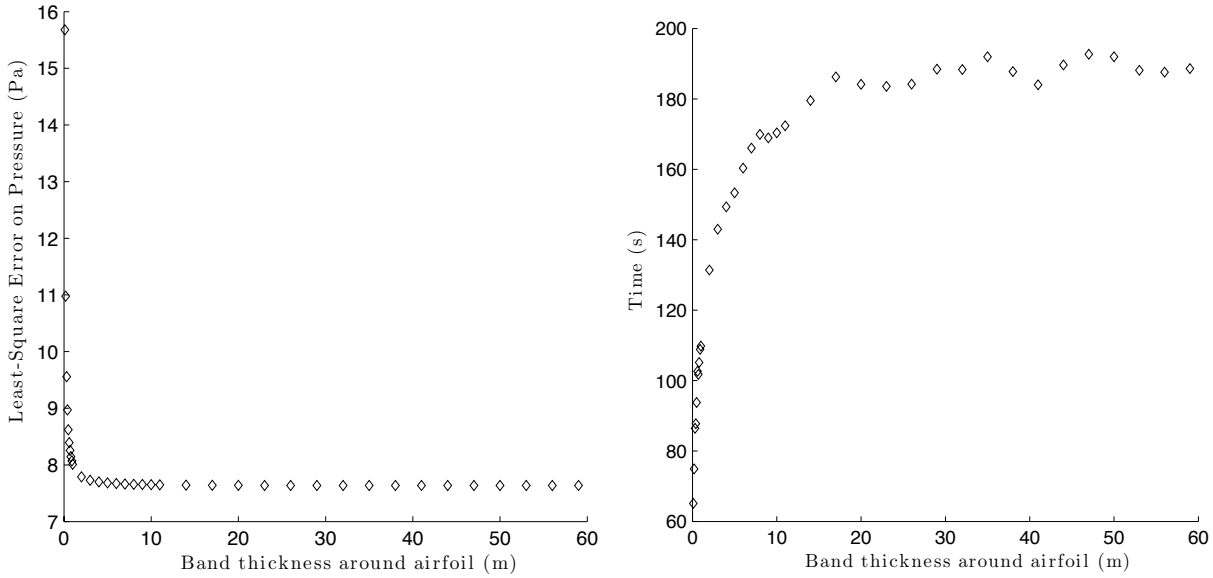
200 seconds. Because of that, all cells will be used for the derivation of eigenfunctions for the rest of this work. By doing so, the error is minimized and the time required is somewhat the same.

3.3.3 Collocation

In the previous sections, all cells used for the derivation of eigenfunctions were used for snapshots reconstruction. This method is called least square, as described in Section 3.2.2. However, another method, called least square collocation, is more efficient for non-linear problem characterized by transonic flows with shocks. This section studies the effect of the number of cells used by the Levenberg-Marquardt algorithm to obtain the η_i (see Eq. 3.11) on the reconstructed solutions. In other words, instead of minimizing the error in a least-square sense over all cells, only a fraction of the cells used to derive the eigenfunctions are considered. It is believed that cells close to the airfoil are more important but the optimal number of cells to use must be determined.

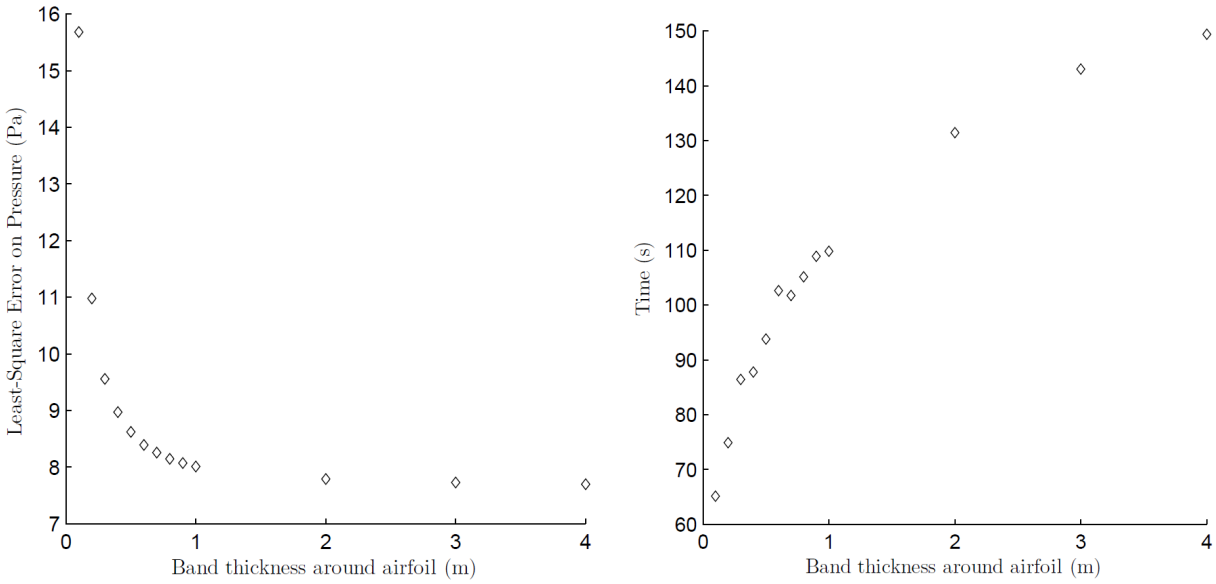
As said in section. 3.3.1 the error must be of the magnitude $O(10^1\ Pa)$ for the pressure to avoid visible differences in the C_p distribution and in the pressure contours. Figure 3.5 shows that using a band thickness below 5 chord length is less costly from a computer time point of view and provide the same level of accuracy. Figure 3.6(a) shows that the error is stabilized and below $10\ Pa$ when using a band thickness of 1 chord length.

Of course, to maximise the accuracy, using a band thickness of 4 chord length would be better, but the improvement is below 5%. Furthermore, the time required with 4 chord length is 150 seconds and around 100 seconds with 1 chord length. Using all cells within a distance of 1 chord length increases the error by 5% but reduces the computational time by 30 $\%$ compared to 4 chord length. Therefore, a band thickness of 1 chord length is used for snapshots reconstruction for the rest of this document.



(a) Error vs band thickness (b) Time vs band thickness

Figure 3.5 Effects of the number of cells used for reconstruction



(a) Error vs band thickness (b) Time vs band thickness

Figure 3.6 Effects of the number of cells used for reconstruction

3.3.4 Reconstruction of snapshots

The idea behind POD is that eigenfunctions must be able to reproduce any snapshots used to derive them. In our case, to improve efficiency, the number of eigenfunctions and the number of cells used for reconstruction was decreased. Before going any further, it is necessary to be sure that the error generated by the reconstruction is tolerable. Table 3.1 shows that using only cells within a distance of 1 chord length from the airfoil and by using 28 eigenfunctions permits to perfectly reconstruct the snapshots. Indeed, the error is less than 10 Pa for all snapshots, i.e. 0.1% relative error.

Figures 3.7-3.8 show two examples to demonstrate the reconstruction capabilities. The POD and CFD solutions are overlapping, hence, the snapshots are perfectly reconstructed.

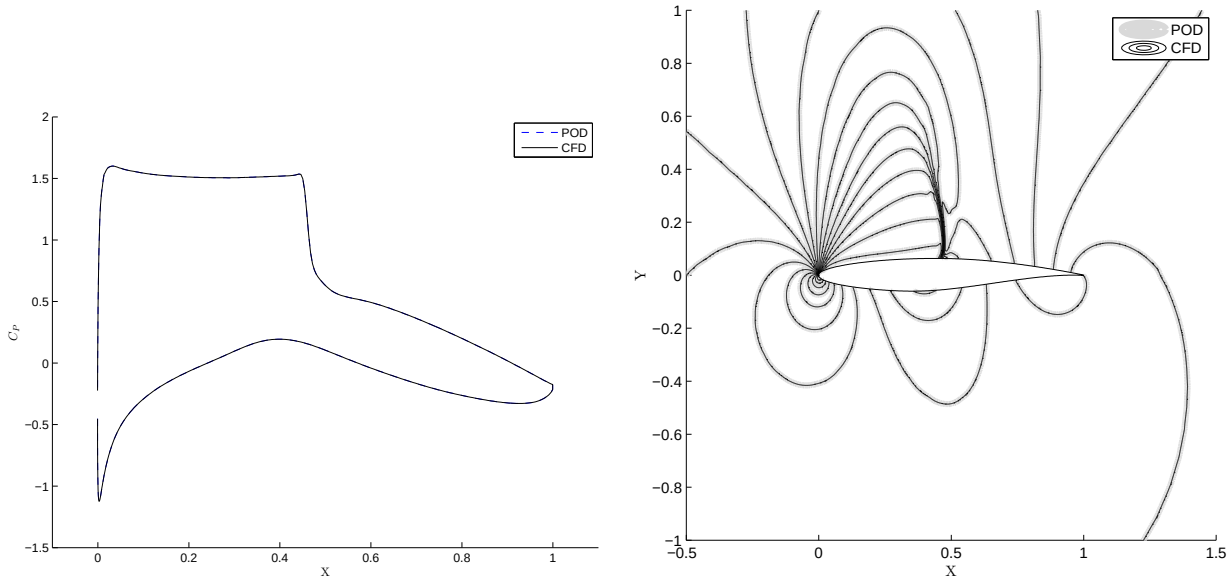
Even if POD and CFD solutions are overlapping, there is an error in the aerodynamics coefficients (see Table 3.2). This error is due to the fact that the coefficients are calculated by neglecting the viscosity. Only the pressure coefficients are used to calculate C_l and C_m. However, the errors are very small and could be neglected because they are not due to the POD.

Angle of attack \ Mach number	0.3	0.4	0.5	0.6	0.7	0.8
-3	1.842	1.829	2.826	2.390	1.062	0.002
-2	1.888	1.433	1.979	3.731	4.500	0.008
-1	2.373	1.907	1.861	2.434	4.391	0.001
0	2.133	1.523	1.799	2.582	3.933	0.007
1	1.448	1.176	1.714	2.870	4.779	0.005
2	1.165	1.783	1.838	4.435	1.492	0.005
3	2.109	2.230	2.588	0.743	0.024	0.002
4	3.248	1.455	4.732	0.041	0.012	0.003
5	4.347	7.637	0.245	0.056	0.014	0.002

Table 3.1 Least-Square error (in Pa) on pressure

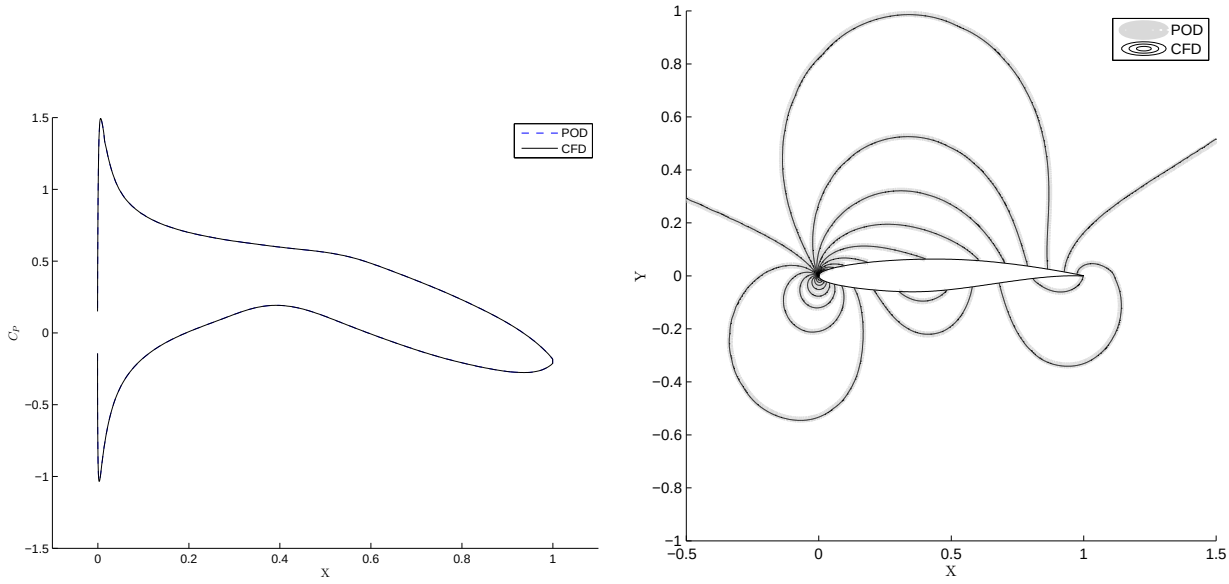
	POD		**CFD**		**Error**	
Parameters	C_l	C_m	C_l	C_m	C_l	C_m
$Mach = 0.7\ \alpha = 4°$	0.9878	0.0787	0.9885	0.0787	0.0007	0
$Mach = 0.4\ \alpha = 3°$	0.5837	0.0704	0.5841	0.0703	0.0004	0.0001

Table 3.2 CFD and POD results



(a) Airfoil C_p distribution (b) Contours of pressure

Figure 3.7 Reconstruction at $Mach = 0.7$ and $\alpha = 4^\circ$



(a) Airfoil C_p distribution (b) Contours of pressure

Figure 3.8 Reconstruction at $Mach = 0.4$ and $\alpha = 3^\circ$

3.3.5 POD vs spline

It is interesting to compare POD to another interpolation method, such as cubic spline. In the latter, an interpolation is performed over all cells in the mesh.

		Mach number				
		0.35	0.45	0.55	0.65	0.75
Angle of attack	-2.5	-0.12846	-0.03582	-0.02343	-0.04618	0.04177
	-1.5	-0.49462	-0.47072	-0.05580	-0.02808	-0.00994
	-0.5	-0.57078	-0.44568	-0.03941	-0.00478	-0.01554
	0.5	-0.20574	-0.21383	-0.04826	-0.01085	-0.00585
	1.5	-0.06319	-0.14372	0.00415	-0.01996	-0.00379
	2.5	-0.05913	-0.06041	-0.00786	-0.00440	0.00284
	3.5	-0.01286	0.00657	-0.00874	0.00161	-0.00129
	4.5	0.02333	0.01551	-0.01990	0.00004	-0.00003

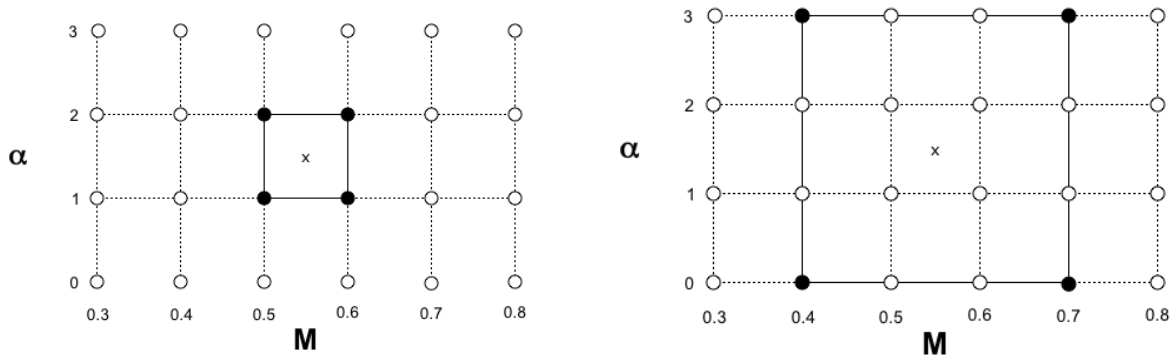
Table 3.3 Relative ($\%$) errors between POD and Spline for the pressure

Table 3.3 shows that an interpolation with POD gives the same accuracy as an interpolation with cubic spline. The reason is simple: in both case, cubic spline were used. For POD they served to interpolate between coefficients. When reconstructing the snapshots, POD provides a way to express the same thing but with less parameters. The advantage of POD is the computational time. It takes about 70 seconds to interpolate all of the 33565 cells of the mesh, but only about 1 second to interpolate between the 28 coefficients. The results show the accuracy, to within 0.5%, but with a reduction of the computer time of about 7000%. This is why POD is preferred over any cell-by-cell interpolation.

3.3.6 Spacing effects

This section studies the effect of the proximity between the snapshots and the interpolated solution. Two cases are tested and both of them uses four snapshots surrounding the interpolated solution. These cases are summarized by Figure 3.9.

These schemes are repeated in a way that the interpolated solution is always equi-distant from all four snapshots. Therefore, the only parameter that influcences the accuracy is the distance between the snapshots and the interpolated solution. For different values of Mach number and angle of



(a) Case 1 (b) Case 2

Figure 3.9 Studied cases for the snapshots spacing (case $Mach = 0.55$ and $\alpha = 1.5$)

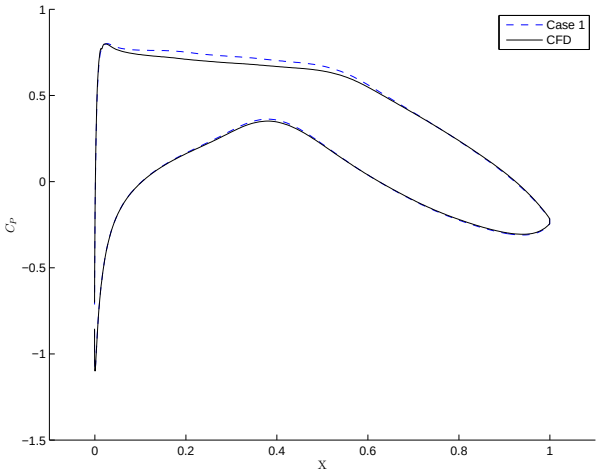
attack, summarized in Table 3.4, two interpolated solutions were created corresponding to the two presented schemes. For each pair of solutions, the relative difference between the least-square error is computed and listed in Table 3.4.

		Mach number		
		0.45	0.55	0.65
AOA	-1.5	876	986	1245
	1.5	884	1142	816
	3.5	933	581	128

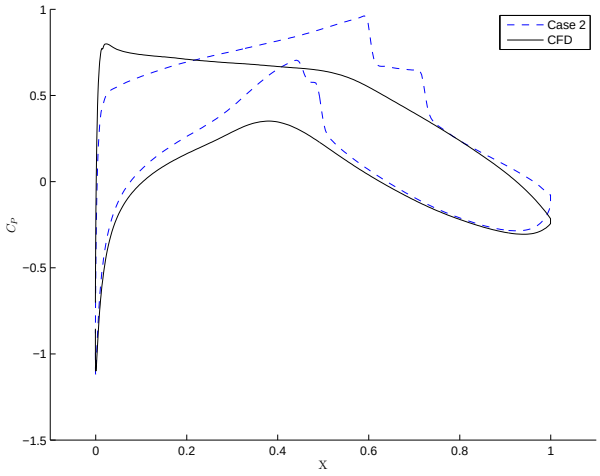
Table 3.4 Difference ($\%$) between case 1 and case 2

The proximity to a snapshot is a factor that highly influences accuracy of a predicted solution. Indeed, Table 3.4 shows that the least-square error was dramatically increased when using the second configuration. For example, the error at $Mach = 0.65$ and $\alpha = 1.5$ is increased by 816 $\%$. The conclusion of this is that using more snapshots in the same span of parameters is a good way to improve the accuracy and the precision of POD.

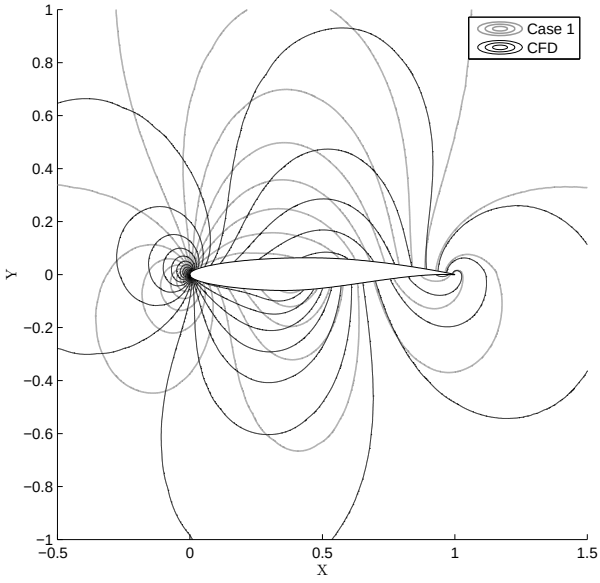
To reinforce this conclusion, we may look at Figure 3.10. The solution is better when using the closest snapshots. It is possible to see the influence of the snapshots by looking at Figures 3.10(b)-3.10(d). Some of the snapshots showed shocks ($Mach = 0.8$ and $\alpha = 0°, 3°$) and this behaviour has polluted the interpolated solution. It is not the case in Figures 3.10(a)-3.10(c) where all the snapshots used showed similar behaviours and a better interpolation has resulted.



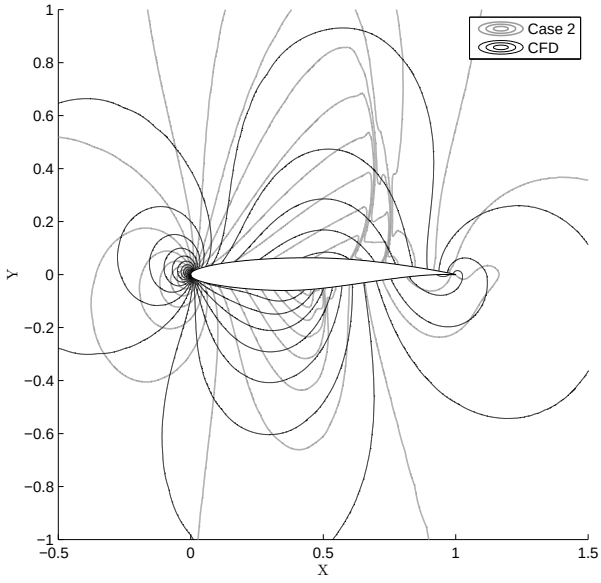
(a) C_p - Case 1



(b) C_p - Case 2



(c) Contours of pressure - Case 1

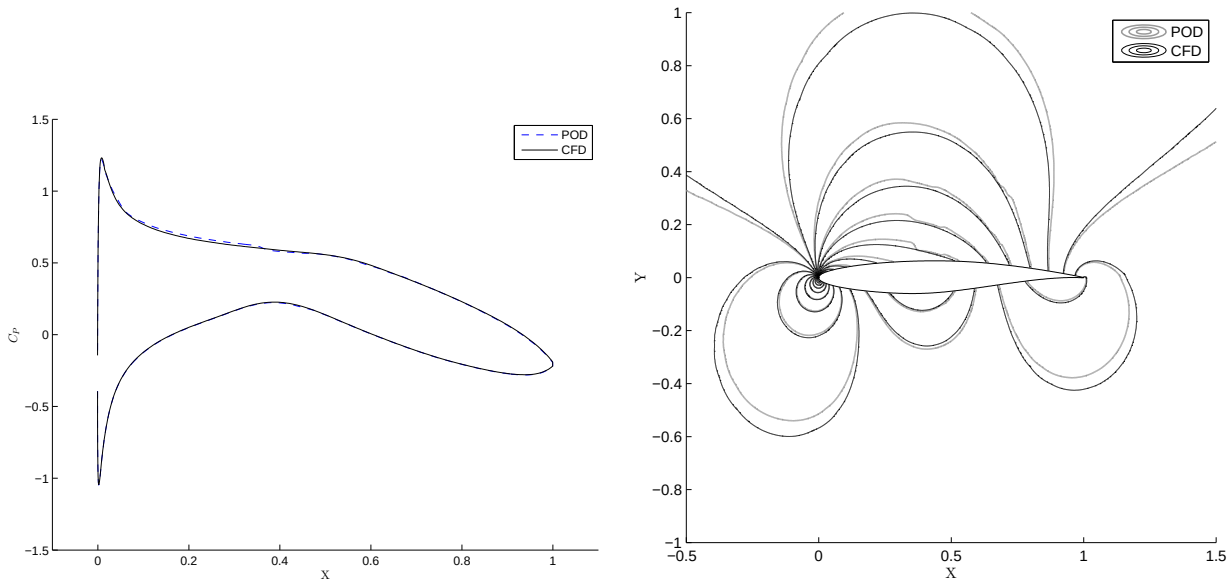


(d) Contours of pressure - Case 2

Figure 3.10 Spacing effects at $Mach = 0.65$ and $\alpha = 1.5°$

3.4 Results

This section illustrates the interpolation capabilities of the POD method alone. All the results were generated by using a band thickness of 1 chord length and by using 28 eigenfunctions.



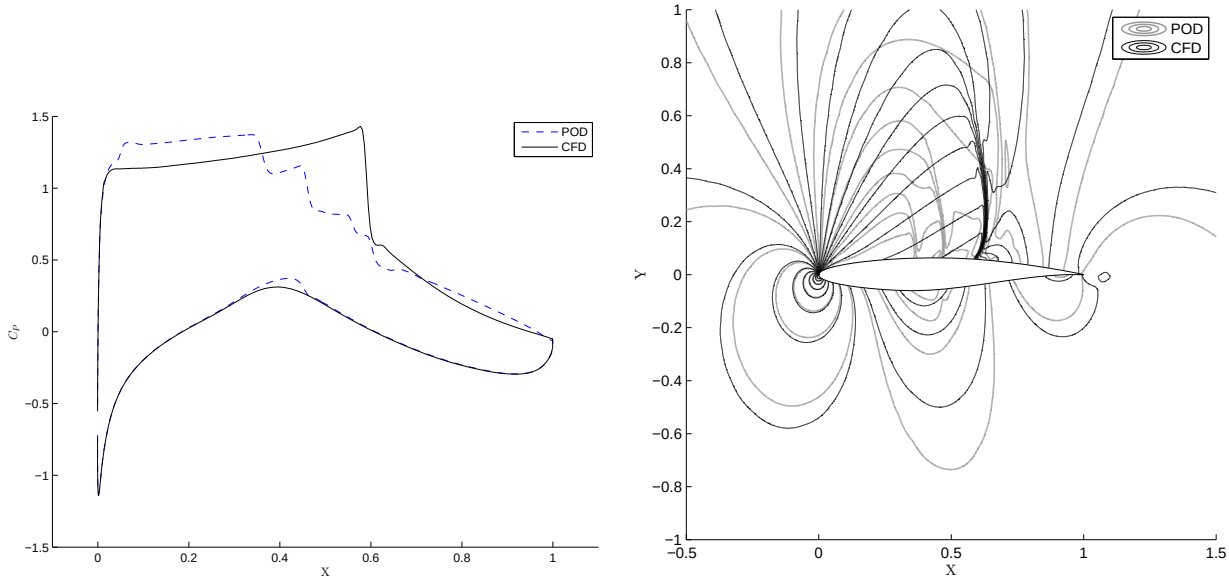
(a) C_p (b) Contours of pressure

Figure 3.11 $Mach = 0.45$ and $\alpha = 2.5°$

		Mach number				
		0.35	0.45	0.55	0.65	0.75
Angle of attack	-2.5	20	20	61	227	1920
	-1.5	18	18	53	196	1601
	-0.5	18	17	51	188	1276
	0.5	20	20	59	219	1812
	1.5	24	24	75	275	2642
	2.5	51	52	162	659	3547
	3.5	123	124	386	1660	3868
	4.5	258	265	865	3389	4297

Table 3.5 Least-square error on pressure (Pa) between POD and CFD

Table 3.5 shows the least-square error on the pressure. It is possible to see that a high angle of attack and Mach number increase the error. For example, the error at $Mach = 0.45$ and $\alpha = 2.5°$ is 52 Pascal and, at $Mach = 0.75$ and $\alpha = 3.5°$, the error is 3868 Pascal. Figures 3.11-3.12 reinforce this conclusion. We might see that a subsonic flow is well predicted which is clearly not the case when a shock is involved. It is the same conclusion for the aerodynamic coefficients.



(a) C_p (b) Contours of pressure

Figure 3.12 $Mach = 0.75$ and $\alpha = 3.5°$

		Mach number				
		0.35	0.45	0.55	0.65	0.75
Angle of attack	-2.5	1.52	1.08	1.89	4.15	15.20
	-1.5	0.57	0.51	0.49	2.42	31.98
	-0.5	0.07	0.09	0.14	0.15	1.24
	0.5	0.41	0.14	0.58	1.07	7.00
	-1.5	1.00	0.48	1.19	2.54	12.36
	2.5	1.56	0.82	1.80	4.18	13.07
	3.5	2.26	1.26	2.62	5.74	4.92
	4.5	2.60	1.45	2.68	4.50	4.17

Table 3.6 Relative error (%) on C_l between POD and CFD

		Mach number				
		0.35	0.45	0.55	0.65	0.75
Angle of attack	-2.5	0.95	0.56	1.02	2.52	6.46
	-1.5	0.23	0.18	0.32	0.85	2.74
	-0.5	0.50	0.20	0.39	0.97	3.16
	0.5	0.81	0.40	0.72	1.78	4.36
	1.5	0.67	0.31	0.58	1.37	2.41
	2.5	1.15	0.55	1.09	2.61	15.54
	3.5	0.84	0.38	0.79	1.11	20.81
	4.5	2.39	1.30	1.92	11.34	18.81

Table 3.7 Relative error (%) on C_m between POD and CFD

Table 3.6 shows that the error on C_l for the latter cases is 0.82% and 4.92% respectively. For the C_m, Table 3.7 shows that the error is 0.55% and 20.81% respectively.

This section has showed the forces and the weaknesses of POD. For some situations, the interpolation capabilities is enough, but it is far from being accurate when dealing with transonic flows. Enlightened by the previous results, it is possible to say that there is definitely room for improvement. The next section proposes two methods to improve accuracy of POD in transonic flows.

CHAPTER 4

PROPOSED IMPROVEMENTS

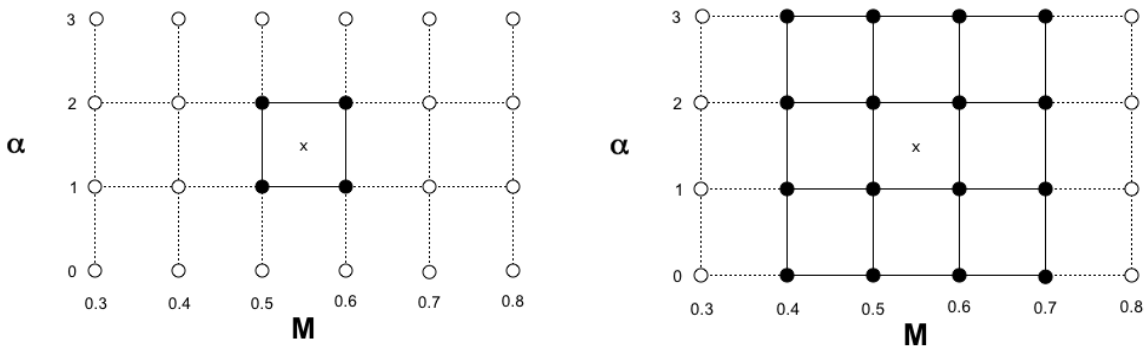
This section describes the proposed methods to improve accuracy of POD based ROM. Two methods are proposed. The first one is a splitting method where the goal is to separate different behaviours to facilitate interpolation. The second method, called "mapping", proposes a hybrid between POD and a full order simulation.

4.1 Splitting

4.1.1 Description

The splitting method follows the idea proposed by Cizmas et al. [21], but applied to steady aerodynamics. Instead of separating snapshots by their level of unsteadiness, the eigenfunctions are derived from the 4 and 16 closest neighbours to the desired parameters as shown in Figure 4.1. In that way, when interpolating a subsonic solution, the result will not be affected too much by solutions presenting strong shocks.

Every snapshot used to derive the eigenfunctions influences the final results. Of course, interpolating closer to a snapshot improves the results as demonstrated in Section 3.3.6. However, all discontinuities, represented by shocks, are altering the final result in a sense that these discontinuities are presents in the eigenfunctions. Since the set of eigenfunctions must be able to perfectly reconstruct any snapshot used to derive them, it is normal to find discontinuities in the eigenfunctions. By using only the snapshots close to the interpolated solution, the snapshots containing shocks or presenting behaviours that are different from the actual flow are removed.



(a) 4 Snapshots (b) 16 Snapshots

Figure 4.1 Splitting schemes (case $Mach = 0.55$ and $\alpha = 1.5$)

4.1.2 Results

The goal of this method is to investigate if a more local basis for the reconstruction could lead to improved results. For exemple, using shock-free snapshots to derive the POD eigenfunctions should improve the accuracy of interpolated subsonic flows. In that case, the "shock-free" eigenfunctions are not polluted by snapshots presenting a shock and vice-versa.

(a) 16 Snapshots

Angle of attack	Mach number 0.45	0.55	0.65
-1.5	-81	-74	34
-0.5	-81	-75	37
0.5	-81	-72	33
1.5	-80	-67	28
2.5	-86	-47	22
3.5	-75	-22	14

(b) 4 Snapshots

Angle of attack	Mach number 0.45	0.55	0.65
-1.5	184.9	59.8	1.5
-0.5	191.6	63.2	-0.3
0.5	176.6	53.9	0.4
1.5	161.5	45.5	10.4
2.5	48.4	-12.7	2.3
3.5	-20.5	-30.8	3.5

Table 4.1 Relative gap ($\%$) between errors on pressure compared to 54 snapshots

Negatives values in Table 4.1 means that the method has improved the accuracy. For example, when using 16 snapshots, the least-square error on pressure has decreased by 81% for the case $Mach = 0.45$ and $\alpha = -1.5°$, but the error has increased by 185% when using 4 snapshots. It is possible to notice a pattern for the case of 16 snapshots. In this case, accuracy has been improved for all simulations corresponding to a Mach number below or equal $Mach = 0.55$ for any angle of attack. However, the splitting did not improve results for higher Mach number. Taking only transonic snapshots or snapshots characterized by higher Mach number to derive eigenfunctions

(a) 16 Snapshots

Angle of attack \ Mach number	0.45	0.55	0.65
-1.5	**0.23**	**0.33**	2.67
-0.5	0.18	0.27	**0.11**
0.5	0.19	**0.34**	1.37
1.5	**0.20**	**0.40**	3.33
2.5	**0.21**	**0.50**	5.50
3.5	**0.16**	**0.89**	7.45

(b) 54 Snapshots

Angle of attack \ Mach number	0.45	0.55	0.65
-1.5	0.50	0.49	**2.42**
-0.5	**0.09**	**0.14**	0.15
0.5	**0.14**	0.58	**1.07**
1.5	0.48	1.19	**2.54**
2.5	0.82	1.80	4.18
3.5	1.26	2.62	5.74

(c) 4 Snapshots

Angle of attack \ Mach number	0.45	0.55	0.65
-1.5	0.53	0.69	4.52
-0.5	1.86	1.64	1.96
0.5	2.00	1.91	2.67
1.5	-2.06	2.04	3.12
2.5	2.11	2.15	**3.85**
3.5	2.14	2.00	**4.52**

Table 4.2 Relative error (%) between CFD and POD for C_l

(a) 16 Snapshots

Angle of attack \ Mach number	0.45	0.55	0.65
-1.5	**0.07**	**0.11**	1.11
-0.5	**0.06**	**0.15**	1.35
0.5	**0.06**	**0.17**	2.39
1.5	**0.05**	**0.15**	2.13
2.5	**0.03**	**0.27**	3.46
3.5	**0.18**	**0.24**	1.79

(b) 54 Snapshots

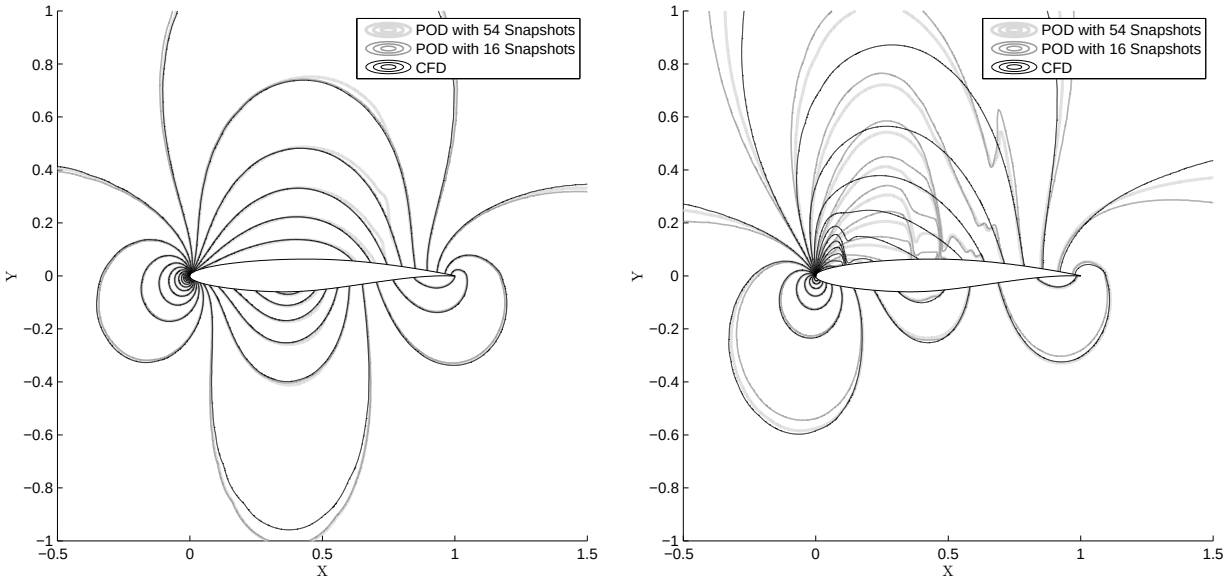
Angle of attack \ Mach number	0.45	0.55	0.65
-1.5	0.18	0.32	**0.85**
-0.5	0.20	0.39	**0.97**
0.5	0.40	0.72	**1.78**
1.5	0.31	0.58	**1.37**
2.5	0.55	1.09	2.61
3.5	0.38	0.78	**1.11**

(c) 4 Snapshots

Angle of attack \ Mach number	0.45	0.55	0.65
-1.5	2.04	1.82	2.02
-0.5	2.06	1.84	2.09
0.5	2.05	1.82	2.14
1.5	2.00	1.75	1.87
2.5	1.92	1.51	**0.30**
3.5	1.72	0.38	2.50

Table 4.3 Relative error (%) between CFD and POD for C_m

did not improve the interpolation in that region. Fig 4.2 reinforces this conclusion. Figure 4.2(a) shows that the splitting has improved the results for subsonic flows. Indeed, the contours are overlapping in a better way with 16 snapshots. Fig 4.2(b) illustrates another phenomenon: POD with or without splitting is not able to interpolate flow with higher Mach number. The accuracy is better with 54 snapshots, but the solution could not be used because the difference between POD and CFD is too high.



(a) $\alpha = 0.5°$, Mach=0.55 (b) $\alpha = 3.5°$, Mach=0.65

Figure 4.2 Contours of pressure

From Tables 4.2-4.3, it is possible to conclude that for lower Mach number, using 16 snapshots usually leads to better results and using 54 snapshots for higher Mach number is a good choice. As shown in Table 4.1, POD derived from 4 snapshots is not a good option because it rarely improves results and when it occurs, the improvement is very small. In tables 4.2-4.3, the values associated with the best method is bold. The same trend is noticeable in the aerodynamic coefficients: 16 snapshots gives better results in subsonic flows.

A good advantage when using all snapshots is that the construction of the eigenfunctions and the reconstruction of the snapshots has to be made only once and the user can interpolate anywhere. When using 16 and 4 snapshots, the computational time is 40 seconds and 6 seconds respectively compared to 400 seconds with 54 snapshots. However, the process has to be done many times because the span of simulation has decreased. In other words, the interpolated solution must al-

ways be surrounded by the snapshots. When using a limited number of snapshots the scheme (see Figure 4.1) has to be repeated at each interpolation to respect this condition. 40 new solutions were created, so the scheme must be repeated 40 times. Table 4.4 summarizes the time to create the 40 new solutions for each method. Furthermore, it is possible to save the eigenfunctions and weighting coefficients in text files in order to avoid reconstructing everything twice if the user wishes to interpolate two times in the same span. Table 4.4 summarizes the memory required to save the eigenfunctions for each methods. Using 54 snapshots is the more efficient solution. In summary, the total time and the total memory represent what each method needs to be able to interpolate anywhere.

Number of Snapshots	Time (s)	Memory (Mb)	Total Time (s)	Total Memory (Mb)
4	6	12	240	480
16	40	45	1600	1800
54	400	160	400	160

Table 4.4 Comparison of Time and Memory

4.1.3 Discussion

Figure 4.3 and Table 4.1 show that using 16 snapshots is the best choice for subsonic solutions. Analysing the eigenfunctions permits to understand this behaviour. It is possible to see, on Figure 4.4, that the eigenfunctions, based on 16 snapshots, are really similar to the solution. This is not the case in Figure 4.5, based on 54 snapshots. One can conclude that using only the portion of the snapshots close to these interpolated solution leads to eigenfunctions that represent flow tendencies more accurately. In these figures, only the first four eigenfunctions were plotted with regard to the value of their weighting coefficient. In each case, the four eigenfunctions are those who have the highest influence on the final solution. When using 54 snapshots, the eigenfunctions have to represent a wider spectrum of solutions. This is why the eigenfunctions are so different. Locally derived eigenfunctions lead to better accuracy because they are not polluted with solutions presenting different flow behaviours and characteristics. There is no shock at $Mach = 0.45$ and yet, the first eigenfunction depicted at Figure 4.5(a) is presenting a sharp pressure variation.

However, occasionally using even less snapshots gives better results as shown in Figure 4.6. Once

again, this could be explained by the behaviour of the eigenfunctions. On Figure 4.7 one can see that the eigenfunctions are quite similar to the solution which is not the case when using 16 snapshots like depicted in Figure 4.8. In the latter figure, eigenfunctions are less smooth. The reason is simple, when using the 16 closest snapshots, some of them contains shocks which perturbs the eigenfunctions. By using 4 snapshots, these perturbations are eliminated. However, 4 snapshots is not much and sometimes even 16 is not enough to capture the general flow behaviour. That is why, for transonic cases, 54 snapshots lead to the smallest error. This is illustrated in Figure 4.9. When it does not work well with 16 snapshots, the solution is not really improved with 54.

It is hard to interpolate shocks because they are non-linear phenomena and POD is a linear basis. This section has proved that locally derived POD eigenfunctions give better accuracy, but are more computationnaly and memory intensive. To be able to correctly interpolate in transonic regime, it is believed that more snapshots are required in that region where the sensibility to the variation of parameters is greater than it is in subsonic.

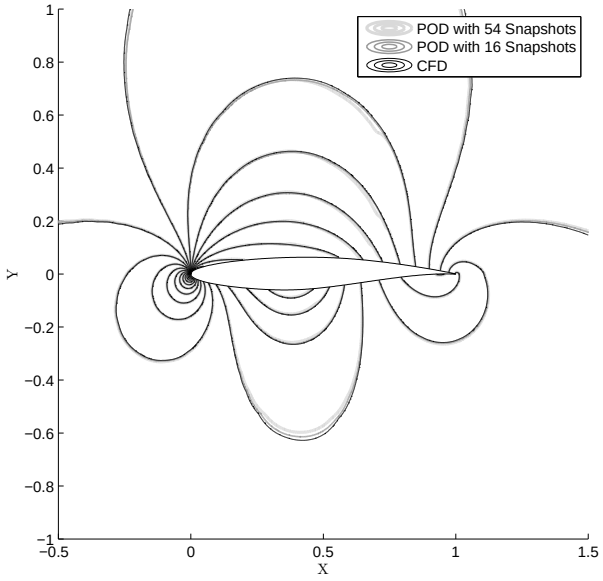
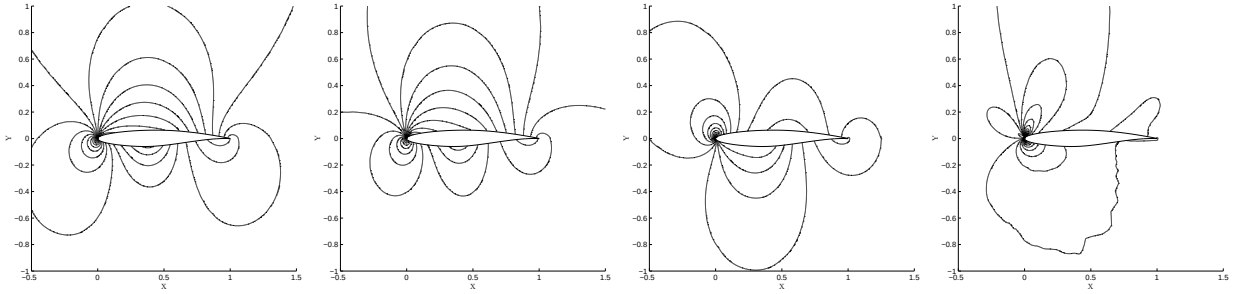
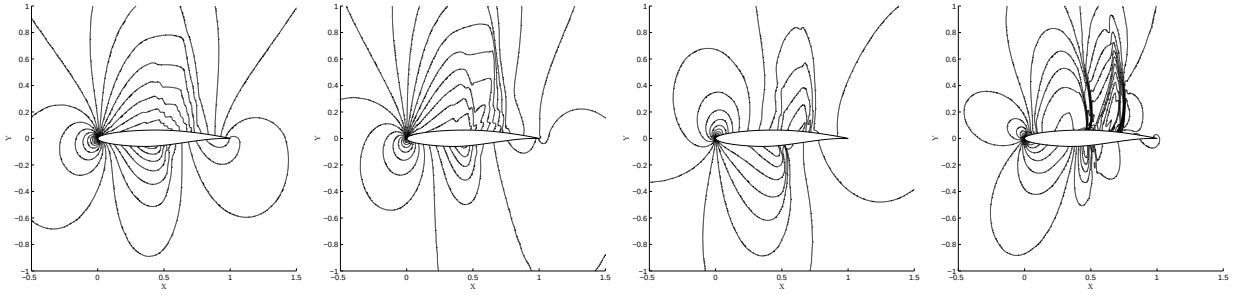


Figure 4.3 Contours of pressure $Mach = 0.45$ and $\alpha = 1.5^{\circ}$



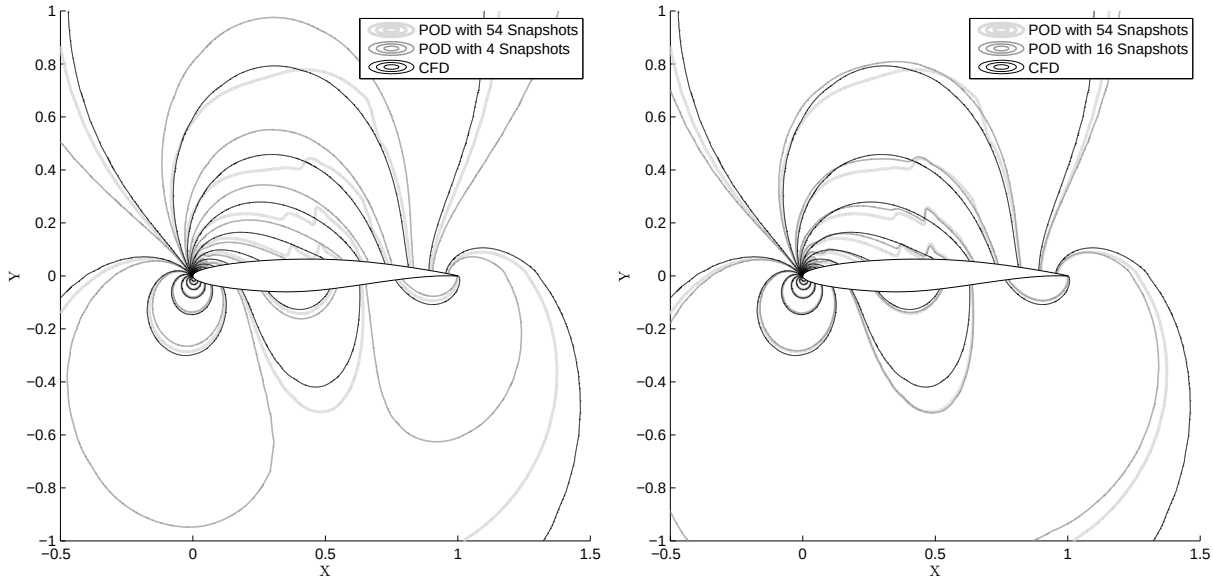
(a) ϕ_1 (b) ϕ_2 (c) ϕ_3 (d) ϕ_4

Figure 4.4 Contours of eigenfunctions of pressure $Mach = 0.45$ and $\alpha = 1.5°$ with 16 snapshots



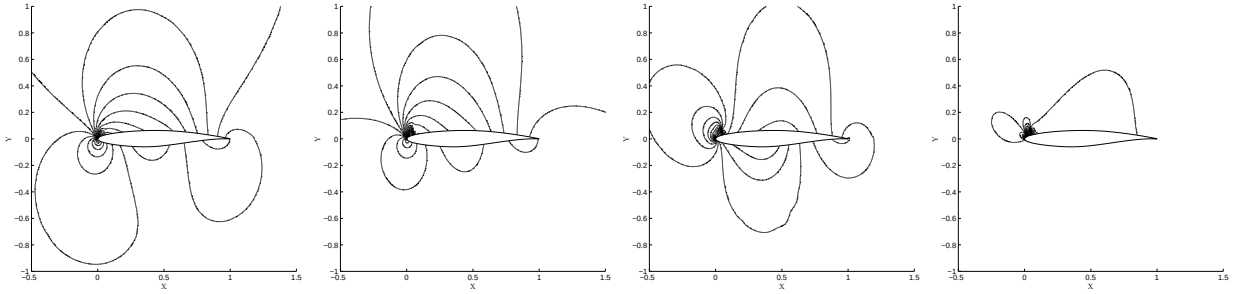
(a) ϕ_1 (b) ϕ_2 (c) ϕ_3 (d) ϕ_5

Figure 4.5 Contours of eigenfunctions of pressure $Mach = 0.45$ and $\alpha = 1.5°$ with 54 snapshots



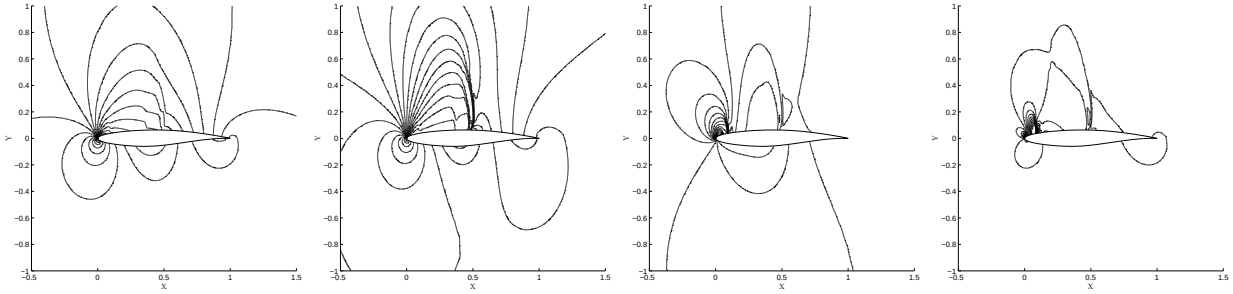
(a) 4 Snapshots (b) 16 Snapshots

Figure 4.6 Contours of pressure $Mach = 0.55$ and $\alpha = 3.5°$



(a) ϕ_1 (b) ϕ_2 (c) ϕ_3 (d) ϕ_4

Figure 4.7 Contours of eigenfunctions of pressure $Mach = 0.55$ and $\alpha = 3.5°$ with 4 snapshots



(a) ϕ_1 (b) ϕ_2 (c) ϕ_3 (d) ϕ_7

Figure 4.8 Contours of eigenfunctions of pressure $Mach = 0.55$ and $\alpha = 3.5°$ with 16 snapshots

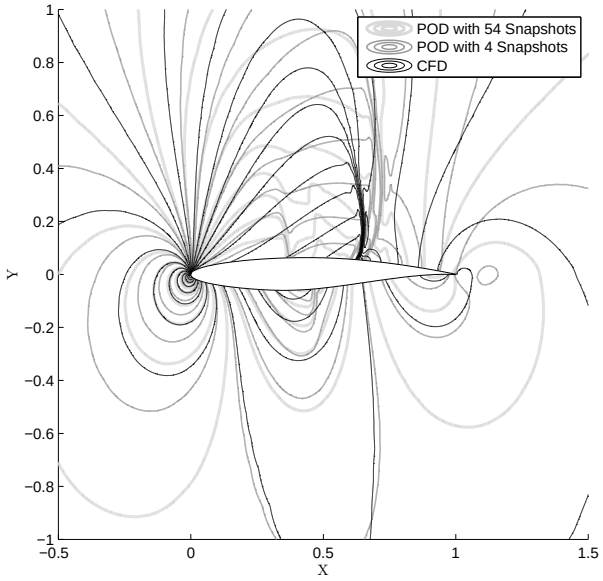


Figure 4.9 Contours of pressure $Mach = 0.75$ and $\alpha = 2.5°$

4.2 Mapping

The previous sections have shown that the most problematic phenomena in transonic aerodynamics is the shock. The shock is very difficult to interpolate for a linear basis like POD. The present section describes an attempt to introduce a non-linear effect in the reconstruction with the goal to improve shock resolution.

4.2.1 Description

This method proposes a way to make a link between Euler and Navier-Stokes simulations. Considering the same geometry with the same mesh, for each snapshot (i.e. different value of Mach number and angle of attack), 2 simulations are run. For each state variable (density, velocity, etc) the difference between both simulations is computed:

$$\begin{Bmatrix} \rho \\ u \\ v \\ p \\ T \\ H \end{Bmatrix}^{Viscous} - \begin{Bmatrix} \rho \\ u \\ v \\ p \\ T \\ H \end{Bmatrix}^{Inviscid} = \begin{Bmatrix} \Delta\rho \\ \Delta u \\ \Delta v \\ \Delta p \\ \Delta T \\ \Delta H \end{Bmatrix} \tag{4.1}$$

Instead of using POD to interpolate between 2 viscous simulations, POD is used to interpolate between 2 "Δ". The result is added to an Euler simulation to generate the "viscous" interpolation:

$$\begin{Bmatrix} \rho \\ u \\ v \\ p \\ T \\ H \end{Bmatrix}^{Viscous}_{Interpolated} = \begin{Bmatrix} \rho \\ u \\ v \\ p \\ T \\ H \end{Bmatrix}^{Inviscid}_{CFD} + \begin{Bmatrix} \Delta\rho \\ \Delta u \\ \Delta v \\ \Delta p \\ \Delta T \\ \Delta H \end{Bmatrix}_{POD} \tag{4.2}$$

The major issue with POD is that it is a linear basis. Hence, as shown in previous sections, POD is not well suited for non-linear problems like flows with shocks. The idea behind the mapping method is to take into account some non-linearities by solving an Euler flow on the same mesh. By doing so, a lot of flow informations are gathered from the inviscid simulation. It is expected that the mapping will yield to a good approximation of CFD when a shock is involved. Of course, the need of calculating an inviscid simulation is time consuming, but it is less costly than a viscous simulation. Furthermore, when dealing with full aircraft configuration the computer time gap between an inviscid and a viscous simulation will become even higher and the method will gain in performance. Of course, it has to be previously tested on a simple configuration, like here, on a two dimensional RAE2822 airfoil.

4.2.2 Results

The results are presented in Tables 4.5 to 4.7. Table 4.5 shows that an improvement is noticeable, when the angle of attack is -2.5° or -1.5°. Indeed, in these cases, the error is lower with the mapping. Figures 4.13 illustrates this conclusion and shows how the shock is well predicted. However, this method has worsened the results for any other value of the angle of attack. From an aerodynamic coefficient point of view, Tables 4.6-4.7 show that the coefficients are more accurately predicted with the POD method alone than combined with the mapping. Again, values in bold indicate which method gives the best results. Even for low Mach number, this method does not improve the results like depicted in Figure 4.10.

Angle of attack \ Mach number	0.35	0.45	0.55	0.65	0.75
-2.5	-3	-5	-7	-7	-24
-1.5	-26	-25	-24	-24	-34
-0.5	4	7	9	8	25
0.5	36	37	38	38	62
1.5	62	57	55	62	70
2.5	51	49	47	50	46
3.5	8	7	19	30	70
4.5	3	-1	16	17	113

Table 4.5 Relative gap (%) between errors with and without mapping

(a) POD

Angle of attack \ Mach number	0.35	0.45	0.55	0.65	0.75
-2.5	1.52	1.08	1.89	4.15	15.20
-1.5	**0.57**	**0.51**	**0.49**	**2.42**	31.98
-0.5	**0.07**	**0.09**	**0.14**	**0.15**	**1.24**
0.5	**0.41**	**0.14**	**0.58**	**1.07**	7.00
1.5	**1.00**	0.48	1.19	2.54	12.36
2.5	1.56	0.82	1.80	4.18	13.07
3.5	2.26	1.26	2.62	5.74	4.92
4.5	**2.60**	1.45	**2.68**	4.50	**4.17**

(b) POD with Mapping

Angle of attack \ Mach number	0.35	0.45	0.55	0.65	0.75
-2.5	**0.76**	**0.36**	**0.71**	**1.77**	**6.51**
-1.5	1.87	0.69	2.30	4.91	**20.88**
-0.5	0.68	0.36	0.74	1.56	3.81
0.5	0.83	0.37	0.84	1.74	**6.24**
1.5	1.03	**0.46**	**1.01**	**2.13**	**9.57**
2.5	**0.73**	**0.36**	**0.89**	**1.79**	**10.36**
3.5	**1.36**	**0.82**	**0.78**	**2.70**	**3.70**
4.5	4.14	**0.33**	5.24	**3.32**	-4.47

Table 4.6 Relative error (%) between CFD and POD for C_l

(a) POD

Angle of attack \ Mach number	0.35	0.45	0.55	0.65	0.75
-2.5	0.95	0.56	1.02	2.52	6.46
-1.5	**0.23**	**0.18**	**0.32**	**0.85**	**2.74**
-0.5	**0.50**	**0.20**	**0.39**	**0.97**	**3.16**
0.5	**0.81**	**0.40**	**0.72**	**1.78**	**4.36**
1.5	**0.67**	**0.31**	**0.58**	**1.37**	**2.41**
2.5	**1.15**	**0.55**	**1.09**	**2.61**	**15.54**
3.5	**0.84**	**0.38**	**0.79**	**1.11**	20.81
4.5	**2.39**	**1.30**	**1.92**	11.34	18.81

(b) POD with Mapping

Angle of attack \ Mach number	0.35	0.45	0.55	0.65	0.75
-2.5	**0.40**	**0.34**	**0.52**	**1.21**	**3.40**
-1.5	0.70	0.44	0.83	1.90	6.72
-0.5	0.98	0.69	1.13	2.75	9.74
0.5	1.87	1.18	2.05	5.00	15.84
1.5	3.28	2.00	3.51	8.59	20.49
2.5	4.31	2.61	5.08	12.10	18.74
3.5	5.04	3.51	4.28	9.94	**8.63**
4.5	9.51	1.88	12.76	**6.93**	**1.57**

Table 4.7 Relative error (%) between CFD and POD for C_m

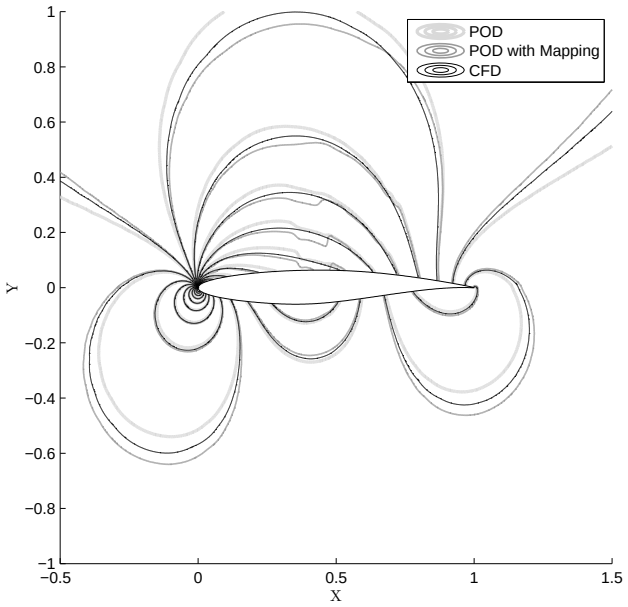


Figure 4.10 Contours of pressure $Mach = 0.45$ and $\alpha = 2.5°$

4.2.3 Discussion

The position and strength of the shock are well captured when the angle of attack is equal to -2.5° or -1.5° because there is a very small difference between the position of the shock captured with an Euler solver compared to Navier-Stokes solver. Figure 4.14 demonstrates that fact. However, for any other value of the angle of attack, there is a difference in the shock position because of the boundary layer. This difference is shown in Figure 4.12. The boundary layer creates an additional displacement thickness δ^* on the upper surface which changes the effective shape of the airfoil. This alteration ultimately modifies the position of the shock. The point behind running an Euler was to gather information about the shock position and strength, but it becomes useless since the effective shape is varying too much. As a result, even with this method, POD eigenfunctions are not able to override the non-linearity caused by the shock. Looking at Figure 4.12 allows us to understand why. There is a huge difference between the behaviour of the flow when viscosity is involved. When using an Euler solver to gather the shock's position, we are presuming that the more the angle of attack is high, the closer to the trailing edge the shock will appear on the airfoil and it is not the case. The behaviors are too different.

It is possible to take a look at the eigenfunctions in Figure 4.15. They are very irregular and that behaviour is visible in the interpolated solution (Figures 4.10, 4.11 and 4.13). These irregularities lead to non-physical solutions like depicted in Figure 4.11.

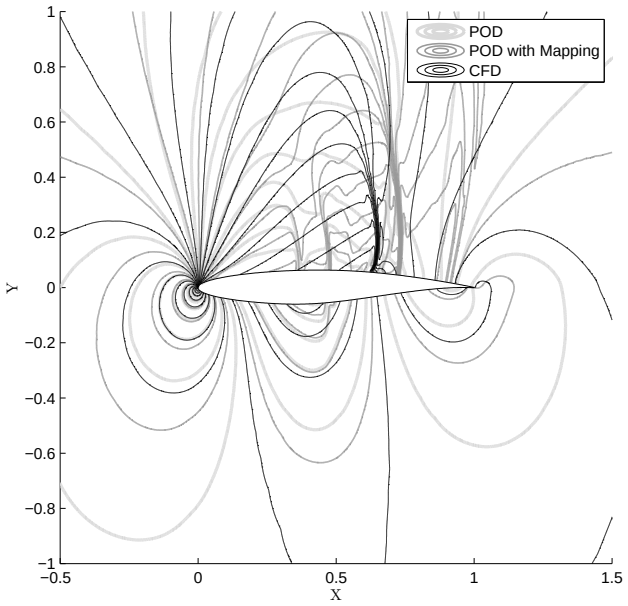


Figure 4.11 Contours of pressure $Mach = 0.75$ and $\alpha = 2.5°$

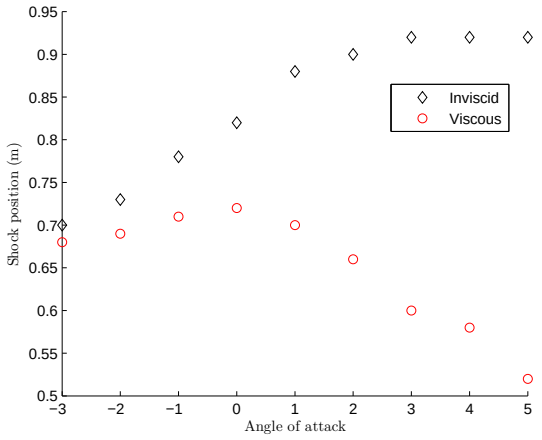
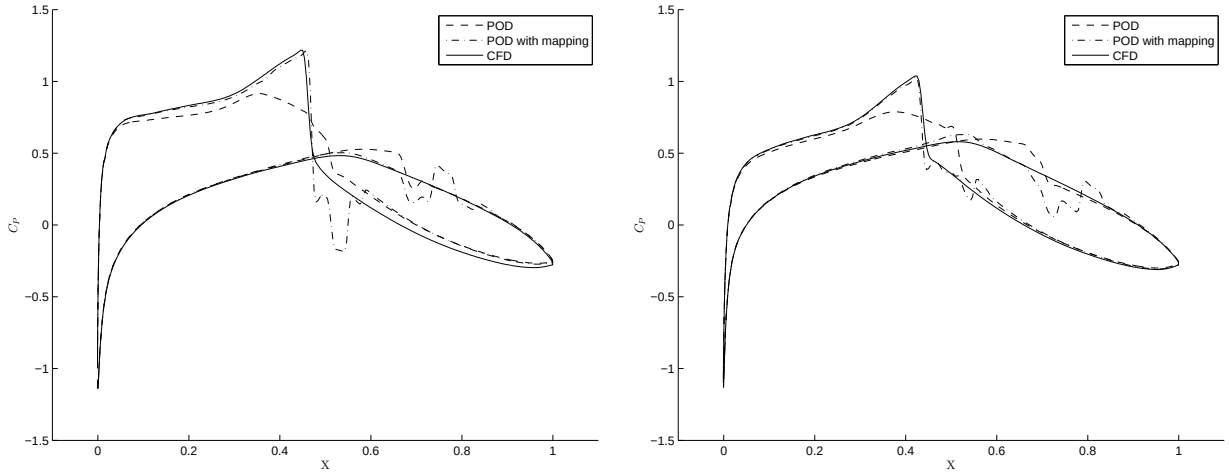
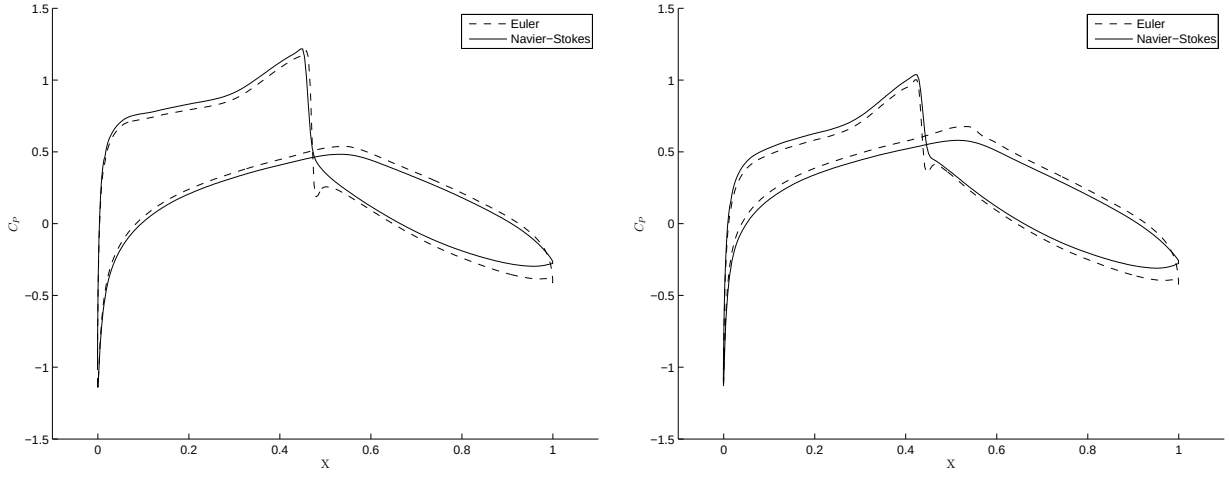


Figure 4.12 Shock position at $Mach = 0.8$



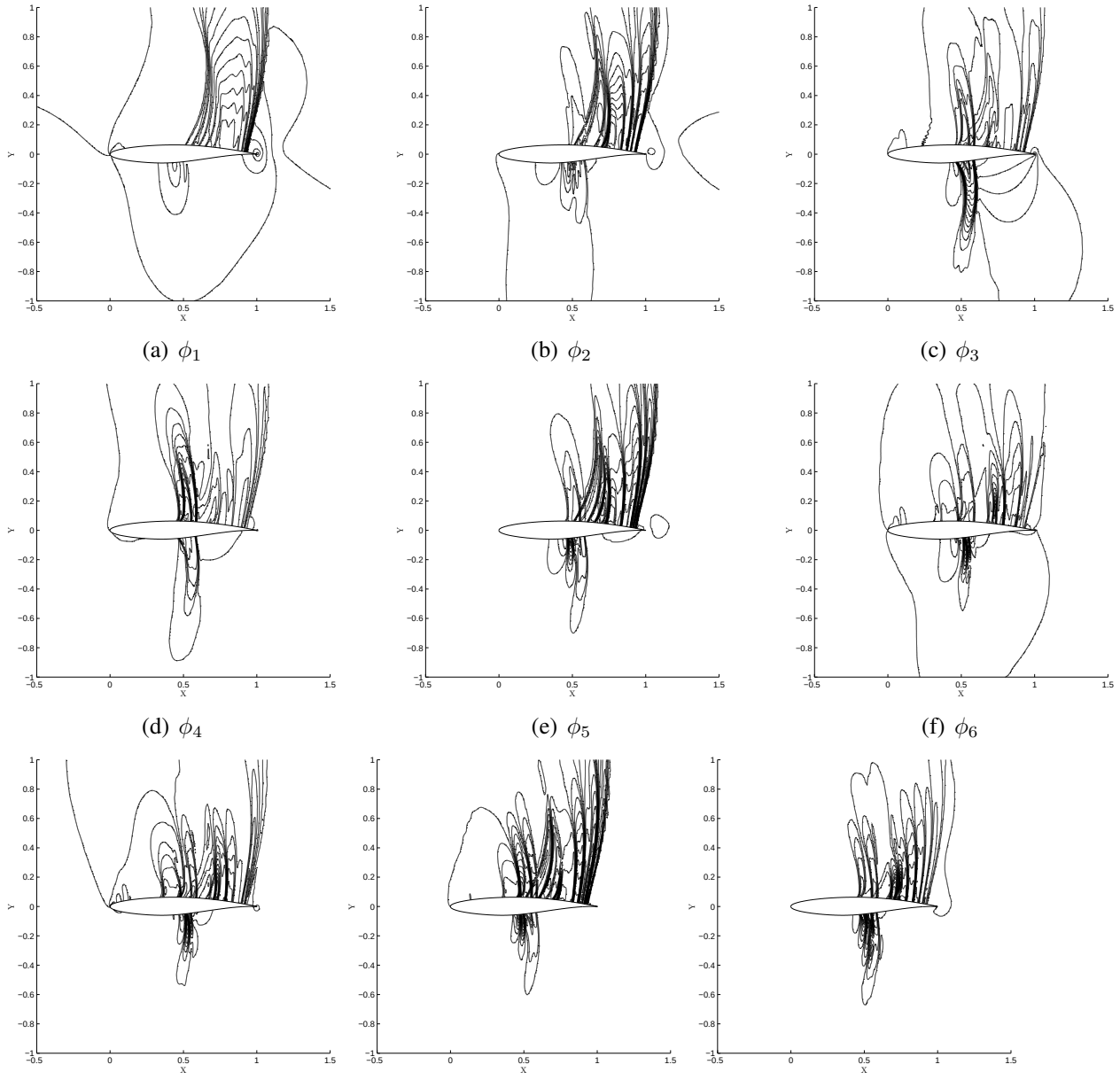
(a) $\alpha = -2.5°$, Mach=0.75 (b) $\alpha = -1.5°$, Mach=0.75

Figure 4.13 Airfoil C_p distribution



(a) $\alpha = -2.5°$, Mach=0.75 (b) $\alpha = -1.5°$, Mach=0.75

Figure 4.14 Airfoil C_p distribution



(a) ϕ_1 (b) ϕ_2 (c) ϕ_3

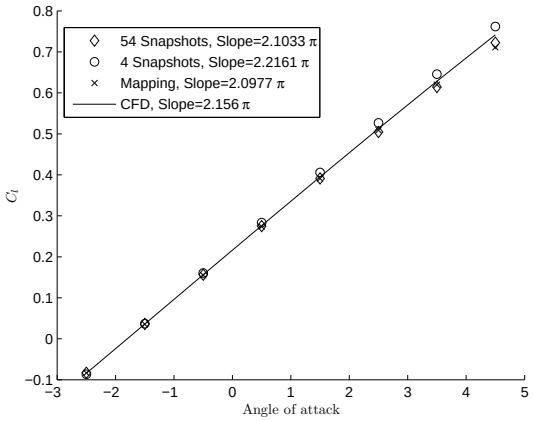
(d) ϕ_4 (e) ϕ_5 (f) ϕ_6

(g) ϕ_7 (h) ϕ_8 (i) ϕ_9

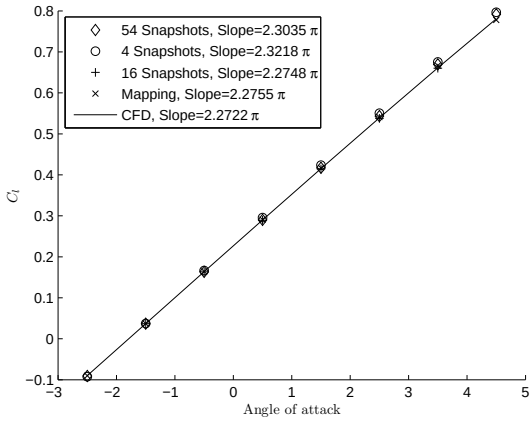
Figure 4.15 Contours of eigenfunctions of the difference between inviscid and viscous

4.3 Stability derivatives

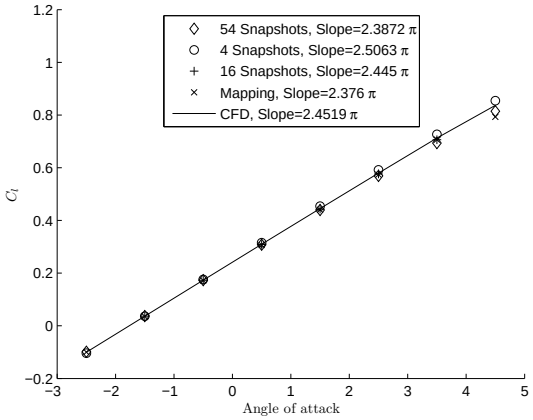
In order to analyze the potential application of POD to the estimation of aircraft stability derivatives, it is important to plot lift and moment coefficients polars. They are represented in Figures 4.16 and 4.17 respectively. It is possible to conclude that any of the methods presented in this work could be use to get good values of C_{l_α}. However, the situation is very different for C_{m_α}. The best methods are the splitting with 16 snapshots or the basic POD derived from all 54 snapshots. The other methods do not get it right and none of them are good in transonic regime. It is believed that using more snapshots in the transonic region will improve results. The mapping method is the worse of all. The idea to gather information about the flow and the shock was interesting, but another method should be used to do so. It could be, for example, using a Navier-Stokes solver on a coarser grid, but the use of an inviscid solver is not a good solution.



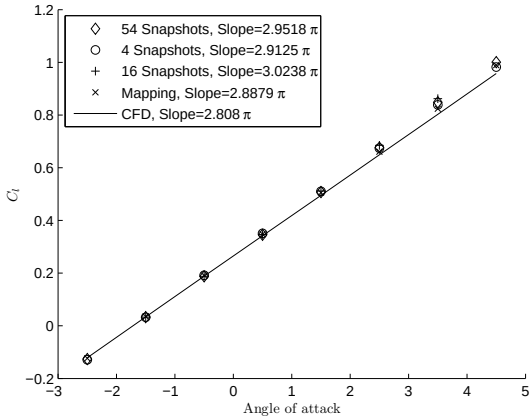
(a) M0.35



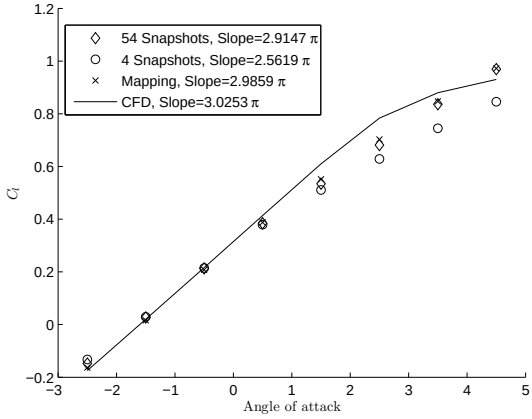
(b) M0.45



(c) M0.55

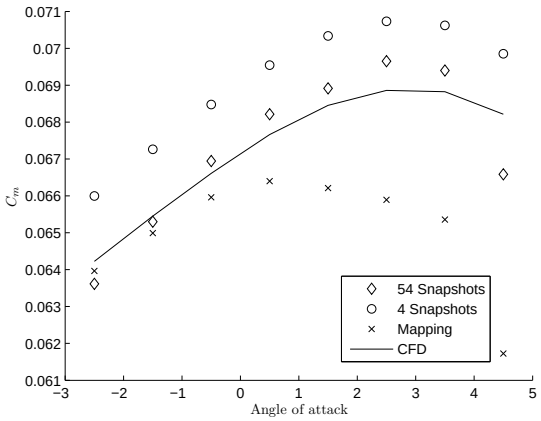


(d) M0.65

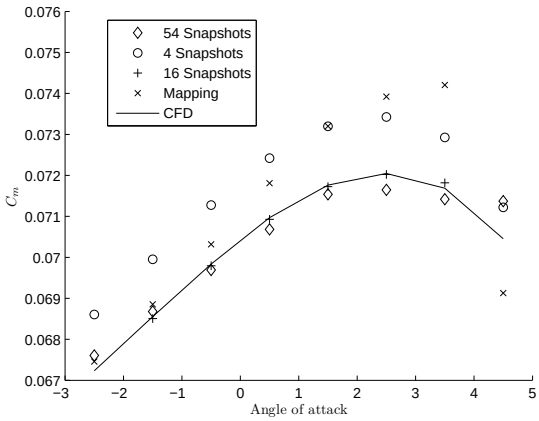


(e) M0.75

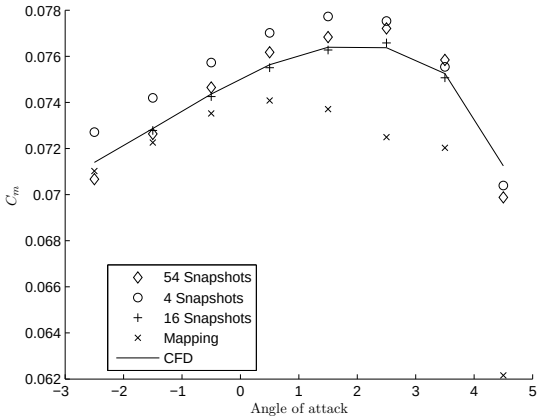
Figure 4.16 C_l vs α



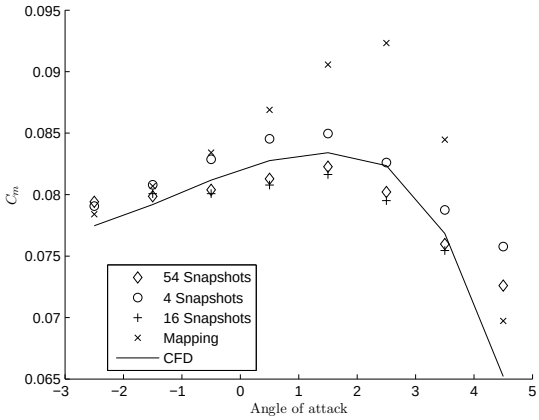
(a) M0.35



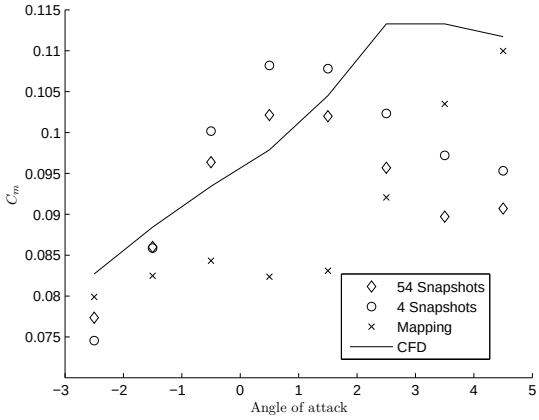
(b) M0.45



(c) M0.55



(d) M0.65



(e) M0.75

Figure 4.17 C_m vs α

CHAPTER 5

CONCLUSION

An algorithm to generate reduced-order models based on proper orthogonal decomposition has been used to study the flow around an RAE2822 airfoil. The goal of the project was to investigate the potential of the method to predict stability derivatives for steady flows. We also investigated ways to improve the representation accuracy of such reduced-order models.

Methods proposed in the literature by several authors were applied to the present problem. These methods are the projection window, the least-square collocation and the use of cubic splines to interpolate weighting coefficients. Furthermore, analysis were performed regarding the number of eigenfunctions and the number of cells to use for snapshots reconstruction. The application on the present problem showed that using 28 eigenfunctions and using all cells within a band thickness of 1 chord length represents the best compromise between the representation error and CPU time for reconstruction.

Two methods were proposed to improve accuracy of POD: splitting and mapping. The first one attempts to isolate different flow behaviours while the second one attempts to improve accuracy when dealing with shocks. Results showed that the splitting methods improved accuracy but only for subsonic flow. Using only "shock-snapshots" to interpolate transonic did not improve the quality of interpolated solutions. The second method was able to correctly capture the shock, but only for a negative angle of attack. Furthermore, the latter method leads to the worst prediction of lift and moment coefficients. A deeper analysis showed that these results were not caused by the method itself but by the fact that the effective shape of the extrados of the airfoil was altered by the boundary layer which causes the shock's position to be different in inviscid compared to viscous situations.

The splitting method improved accuracy of the model but only for subsonic flows. In spite of the bad results achieved by the mapping method, more development could still be done. Indeed, the point behind this method is to gather information about the shock position to help POD. Techniques

replacing the inviscid simulation by something else could be used. For example, one could run a viscous simulation on a coarser grid and interpolate results to the finer grid. Furthermore, other methods, similar to POD, exist such as the High-Order Singular Value Decomposition (HOSVD). It would be interesting to investigate these.

Finally, all methods were able to give good approximations of the C_{l_α} coefficients, but only the splitting with 16 snapshots or the POD with 54 snapshots were able to accurately predict the C_{m_α} coefficients. We may conclude that locally derived POD eigenfunctions are the best way to use POD. However, usage in transonic regime requires more snapshots.

BIBLIOGRAPHY

[1] M. Salas, "Digital Flight: The Last CFD Aeronautical Grand Challenge", *Journal of Scientific Computing*, vol. 28, pp. 479–505, 2006, 10.1007/s10915-006-9087-7.

[2] PH Cook, MA McDonald and MCP Firmin, "'Aerofoil RAE 2822- Pressure Distributions, and Boundary Layer and Wake Measurements,' AR-138, AGARD, 1979, pp. A 6. 1-A 6. 77".

[3] K. Fukunaga, *Introduction to statistical pattern recognition*, Academic, New York, 1972.

[4] L. Sirovich and M. Kirby, "Low-dimensional procedure for the characterization of human faces", *Journal of the Optical Society of America A*, vol. 4, n. 3, pp. 519–524, 1987.

[5] R. Everson and L. Sirovich, "Karhunen–Loeve procedure for gappy data", *Journal of the Optical Society of America A*, vol. 12, n. 8, pp. 1657–1664, 1995.

[6] A. Sempey, C. Inard, C. Ghiaus and C. Allery, "Reduced order model for air temperature control in indoor spaces".

[7] H. Ly and H. Tran, "Modeling and control of physical processes using proper orthogonal decomposition", 1999.

[8] C. Allery, C. Beghein 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, n. 8, pp. 907–920, 2005.

[9] T. Lieu, C. Farhat and M. Lesoinne, "Reduced-order fluid/structure modeling of a complete aircraft configuration", *Computer Methods in Applied Mechanics and Engineering*, vol. 195, n. 41-43, pp. 5730–5742, 2006.

[10] T. Bui-Thanh, *Proper Orthogonal Decomposition Extensions and Their Applications in Steady Aerodynamics*, PhD thesis, 2003.

[11] K. Lee, S. Rallabhandi and D. Mavris, "Aerodynamic Data Reconstruction via Probabilistic

Principal Component Analysis".

[12] P. Holmes, J. Lumley and G. Berkooz, *Turbulence, coherent structures, dynamical systems and symmetry*, Cambridge Univ Pr, 1998.

[13] W. Cazemier, R. Verstappen and A. Veldman, "Proper orthogonal decomposition and low-dimensional models for driven cavity flows", *Phys. Fluids*, vol. 10, pp. 1685, 1998.

[14] D. Lucia, P. King and P. Beran, "Reduced order modeling of a two-dimensional flow with moving shocks", *Computers and Fluids*, vol. 32, n. 7, pp. 917–938, 2003.

[15] B. Epureanu, "A parametric analysis of reduced order models of viscous flows in turbomachinery", *Journal of Fluids and Structures*, vol. 17, n. 7, pp. 971–982, 2003.

[16] H. Park and M. Lee, "An efficient method of solving the Navier-Stokes equations for flow control", *International Journal for Numerical Methods in Engineering*, vol. 41, n. 6, pp. 1133–1151, 1998.

[17] D. Lucia and P. Beran, "Projection methods for reduced order models of compressible flows", *Journal of Computational Physics*, vol. 188, n. 1, pp. 252–280, 2003.

[18] C. Rowley, T. Colonius and R. Murray, "Model reduction for compressible flows using POD and Galerkin projection", *Physica D: Nonlinear Phenomena*, vol. 189, n. 1-2, pp. 115–129, 2004.

[19] K. Hall, J. Thomas and E. Dowell, "Proper orthogonal decomposition technique for transonic unsteady aerodynamic flows", *AIAA Journal*, vol. 38, n. 10, pp. 1853–1862, 2000.

[20] 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, n. 2, pp. 292–311, Feb 2010.

[21] P. Cizmas, B. Richardson, T. Brenner, T. O'Brien and R. Breault, "Acceleration techniques for reduced-order models based on proper orthogonal decomposition", *Journal of Computational Physics*, vol. 227, n. 16, pp. 7791–7812, 2008.

[22] T. Bui-Thanh, K. Willcox and O. Ghattas, "Parametric reduced-order models for probabilistic analysis of unsteady aerodynamic applications", *AIAA Journal*, vol. 46, n. 10, pp. 2520–2529, 2008.

[23] P. LeGresley and J. Alonso, "Airfoil Design Optimization Using Reduced Order Models Based on Proper Orthogonal Decomposition", *AIAA Journal*.

[24] D. Alonso, A. Velazquez and J. Vega, "A method to generate computationally efficient reduced order models", *Computer Methods in Applied Mechanics and Engineering*, 2009.

[25] J. Burkardt, M. Gunzburger and H. Lee, "POD and CVT-based reduced-order modeling of Navier-Stokes flows", *Computer Methods in Applied Mechanics and Engineering*, vol. 196, n. 1-3, pp. 337–355, 2006.

[26] L. Lorente, J. Vega and A. Velazquez, "Generation of Aerodynamic Databases Using High-Order Singular Value Decomposition", *J. Aircraft*, vol. 45, n. 5, 2008.

[27] P. LeGresley, *Application of proper orthogonal decomposition (POD) to design decomposition methods*, PhD thesis, 2005.

[28] T. Bui-Thanh, M. Damodaran and K. Willcox, "Proper orthogonal decomposition extensions for parametric applications in transonic aerodynamics", *AIAA paper*, vol. 4213, pp. 2003, 2003.

[29] A. Qamar and S. Sanghi, "Steady supersonic flow-field predictions using proper orthogonal decomposition technique", *Computers and Fluids*, vol. 38, n. 6, pp. 1218–1231, 2009.

[30] L. Sirovich, "Turbulence and the dynamics of coherent structures. I- Coherent structures. II- Symmetries and transformations. III- Dynamics and scaling", *Quarterly of applied mathematics*, vol. 45, pp. 561–571, 1987.