Reduced-Order Modeling of Unsteady Aerodynamics Across Multiple Mach Regimes

by

Torstens Skujins

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Aerospace Engineering)
in the University of Michigan
2013

Doctoral Committee:

Professor Carlos E.S. Cesnik, Chair
Professor Peretz P. Friedmann
Walter A. Silva, NASA Langley Research Center
Associate Professor Yin Lu (Julie) Young

To my family: Mom, Dad, Janis, and Riss

ACKNOWLEDGMENTS

Funds for the Michigan-AFRL Collaborative Center in Control Science (MAC-CCS) were made available from the Air Force Research Laboratory/Air Vehicles Directorate grant number FA 8650-07-2-3744 (program manager Michael Bolender). Funds were also made available through NASA award NNX08AB32A (program managers Don Soloway and Jorge Bardina) along with computer resources from the NASA Advanced Supercomputing Division.

It would not have been possible for me to complete my work here at the University of Michigan without the guidance and support of numerous people, and for that I am very grateful. I would first like to thank my family: my parents, Ojars and Margaret; my brother, Janis; and my sister-in-law, Riss, for all of their love and support over the years. And I would especially like to thank my parents for, among a countless multitude of other things, taking me to the U.S. Air Force Museum and Dayton Air Show numerous times while growing up, helping to spark my interest in airplanes.

I want to thank my adviser, Prof. Carlos Cesnik, for all of his guidance throughout my years at Michigan. He allowed me the freedom to pursue my own research interests while ensuring that I kept the overall goals of the research project in mind, and he has always been very helpful and supportive of me and my work. Working with him has made a significant positive impact on my skills and abilities as a researcher. I am also very appreciative of my other committee members, Prof. Peretz Friedmann and Prof. Yin Lu (Julie) Young from the University of Michigan and Dr. Walter Silva from NASA Langley Research Center, for their valuable feedback on my work.

I would also like to thank all the other professors from Michigan who helped me along the way, either through classroom instruction or more research-specific discussions. The aerospace engineering faculty has shown a great willingness to help out students who come across issues in class or research work. In addition to the faculty, I am very appreciative of the hard work of the aerospace engineering staff members. I would specifically like to thank Denise Phelps for being an enthusiastic helper and invaluable resource for any question or issue that may arise for a graduate student and Dave McLean for his willing help with all of the various computer issues that have popped up during my time here. Additionally, I am greatly appreciative of the Michigan CAEN Advanced Computing (CAC) staff, whose combination of response time, helpfulness, and ability to solve computing issues is nothing short of phenomenal.

A number of people outside of the University of Michigan have also provided me with valuable assistance. I would like to thank Dr. Silva for being patient and willing to help with the wide variety of CFL3D issues that I faced, as well as for the various research suggestions he has provided. Dr. Robert Bartels and Dr. Robert Biedron, both also from NASA Langley, provided valuable CFL3D help as well. I also want to thank the research collaborators from the Air Force Research Laboratory, Dr. Michael Bolender, Dr. David Doman, and Dr. Michael Oppenheimer, for their guidance throughout my graduate career and especially for the two summers I spent at Wright-Patterson Air Force Base. I am very appreciative of the aforementioned research funding from AFRL and NASA, which paved the way for me to conduct my research here at Michigan.

My fellow graduate students in the aerospace engineering department have been a great source of support for me. I would like to thank Nate Falkiewicz and Scott Stapleton for their friendship and support over the years, going through many shared grad school experiences together, from working on homework problem sets to studying for prelims, after arriving at Michigan the same year. My labmates in the Active Aeroelasticity and Structures Research Laboratory have been very helpful and supportive: Dr. Nate Falkiewicz, Dr. Weihua Su, Kalyan Nadella, Scott Frendreis, Devesh Kumar, Ben Hallissy, Matt Dillsaver, Matt Obenchain, Jessica Jones, Dr. Ken Salas, Dr. Jiwon Mok, and Dr. Satish Chimakurthi. Additionally, I want to thank fellow MACCCS students Dr. Sean Torrez, Derek Dalle, and Dr. Matt Fotia for their support throughout my time here.

During my time in Ann Arbor, I have been blessed to be surrounded by caring and supportive people. I am thankful for the friendship and support of those in the Graduate Christian Fellowship group. Throughout the various Bible studies, group meetings, parties, and other activities over the years, I have been fortunate enough to form quality friendships in an outstanding community of people. I would also like to thank my housemates of the past several years, Andrew Crow, Brittan Farmer, Mark Gordon, and Mike Nelson, for making the Nixon House a great and never-boring place to live. Finally, I would like to give a special thanks to George and Mary Lindquist for cheerfully opening up their home and feeding me and a large group of people each and every Tuesday evening throughout the school year; Tuesday dinners will not be the same after I leave Ann Arbor.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

ABSTRACT

Reduced-Order Modeling of Unsteady Aerodynamics Across Multiple Mach Regimes

by

Torstens Skujins

Chair: Carlos E.S. Cesnik

The accurate prediction of unsteady aerodynamic loads is of utmost importance in an aeroelastic simulation framework. Inaccurate prediction of these loads may result in inaccurate control design and evaluation, which, in a worst-case scenario, could cause loss of control of the vehicle. In addition to accuracy, these simulations require that the aerodynamic calculations be computationally efficient, so this often eliminates the use of full-order computational fluid dynamics (CFD) simulations, which can be quite computationally-intensive. Reduced-order models (ROMs) offer a solution to these competing demands of accuracy and efficiency by extracting pertinent data from a limited number of full-order CFD simulations and using that data to construct computationally-efficient models that retain a high amount of the accuracy of the full-order solution while running orders of magnitude faster computationally.

This dissertation focuses on the development of a reduced-order modeling methodology for unsteady aerodynamics based on linear convolution combined with a nonlinear correction factor. Rather than being limited to a specific Mach regime, the ROM formulation is general enough such that it can be applied over a wide range of Mach regimes, from subsonic to hypersonic flight. The correction factor term allows the ROM to be accurate over a range of vehicle elastic modal deformation amplitudes as well as flight conditions representing off-design conditions. This generality is important because it permits a single form of the equations for aerodynamic loads to be used throughout all simulations in a controls framework, further increasing the efficiency. The evaluation of the ROM is accomplished through the comparison of ROM results with full-order CFD simulations for test-case geometries in the subsonic, transonic, and super/hypersonic regimes. Additionally, methods for ROM construction are explored, including the development of a simplified aerodynamic model in the transonic regime for use in aiding ROM construction. Overall, good agreement is obtained between the ROM and CFD results, generally improving as Mach number increases. The potential of the ROM is illustrated by following a single example case from low subsonic up through supersonic flight, thus demonstrating the usefulness of the approach over a wide range of conditions.

Chapter 1

Introduction

1.1 Background and Motivation

Among the most important aspects in the development of aerospace vehicles are the analysis of the vehicle's controllability and the development of appropriate control laws. When considered in the vehicle design phase, the design and evaluation of control laws allows for modifications to be made on the vehicle in this preliminary stage, saving significant time and expense compared to modifications made after a vehicle prototype has already been constructed. An important goal for conducting this type of control analysis is to have the vehicle simulation run in as close to real-time as possible, requiring efficient computational methods for each of the potentially large number of components of the analysis. One example of such a controls framework is the hypersonic vehicle simulation developed by Bolender and Doman,[1] which includes components describing unsteady aerodynamics,[2] vehicle flexibility,[3] and propulsion on a two-dimensional hypersonic vehicle. The motivation for the overall project for which the research for this thesis is a part of was the desire to take the simplified, two-dimensional models of this framework and extend them to more accurate, higher-fidelity, three-dimensional models while retaining the computational efficiency of the original model. Hypersonic vehicles are generally slender, flexible structures,[4] thus the modeling of the heating, aerodynamics, and other dynamics due to elastic modal deformations are of significant importance.[5] This thesis considers the unsteady aerodynamic model, analyzing the aerodynamic effects due to the vehicle's flexibility.

The calculation of unsteady aerodynamic loads can be achieved through a variety of methods with varying levels of fidelity. At the high end of the fidelity scale are computational fluid dynamics (CFD) simulations. These simulations are accurate, but the tradeoff for this accuracy is potentially large computational cost. CFD simulations can be very time-consuming to run, and large simulations may take on the order of days or even weeks to complete on multiple computer processors. At the low end of the fidelity scale are simplified models, which are very efficient computationally. However, the tradeoff for this computational efficiency is a loss of accuracy when compared to the high-fidelity models. Examples of lower fidelity models are piston theory[6] for hypersonic applications, transonic small disturbance codes[7] for transonic flow applications, and panel codes[8] for subsonic applications. Note that the hypersonic vehicle model in Ref. 1 is based on piston theory.

Next, consider the requirements for the aerodynamic model used in a control evaluation and simulation framework. When developing control laws and algorithms, the user needs to conduct numerous simulations in order to capture the overall controllability of the vehicle. If the full-order, high-fidelity CFD solutions mentioned above are used for this purpose, the turnaround time for controls simulation will be very long. In practice, the amount of time required will greatly exceed the amount of time allotted for the analysis, though the analysis itself may be accurate. An additional item to consider is the robustness of the CFD code, since certain types of inputs, such as very large grid deformations and/or velocities, may cause the code to crash. If the simplified models are used, the time requirement is not an issue. However, the loss of accuracy may adversely affect the control simulation results. When applied to the actual vehicle, the inaccuracy of the aerodynamic loads may result in non-ideal control laws, which in turn can result in loss of control of the vehicle. Thus, the aerodynamic model in a controls framework needs to combine the accuracy of a high-fidelity model with the computational efficiency of a low-fidelity model.

Reduced-order models (ROMs) fulfill this seemingly contradictory set of requirements of high accuracy and low computational expense. ROMs work by extracting data from a limited number of high-fidelity computations and then using that data to construct models which run in a computationally efficient manner. In general, the high-fidelity models only need to be run up-front, prior to model construction. Then, once the model has been constructed, the high-fidelity models are no longer needed, and the ROM runs with computational efficiency often times on the same order of magnitude as low-fidelity analysis while retaining much of the accuracy of high-fidelity analysis.

In addition to computational efficiency, another important aspect to consider about a particular model is the flight condition range for which it is applicable. For example, piston theory, to be discussed in Chapter 4, is meant to be used in the high supersonic and hypersonic flight regime. Applying it to, for example, the transonic regime would be inappropriate. In a controls framework, it is desired to have a single aerodynamic form to use throughout all flight conditions under consideration. Otherwise, if the specific aerodynamic model must be changed pending the specific flight conditions, this inevitably would add additional computational time to the simulation. Moreover, the use of a single representation avoids the inherent errors at the applicability boundaries of the various methods when one method needs to be switched out for a different one. Thus, a unified aerodynamic model permits the use of a single mathematical form for the aerodynamic loads across all possible flight conditions in the aeroelastic equations within a controls simulation framework.

The requirement for a single mathematical representation is important because flight vehicles will often pass through multiple Mach regimes and altitudes throughout the course of a single flight. Consider the schematic of a hypersonic vehicle flight

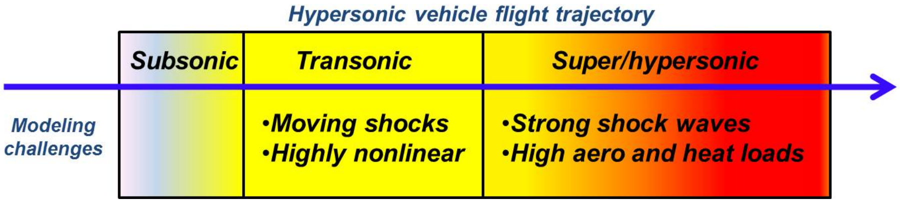


Figure 1.1: Sample hypersonic vehicle flight regimes

trajectory shown in Fig. 1.1. Each stage of the flight presents unique aerodynamic modeling challenges. Just after takeoff, the vehicle will be in the subsonic regime. Then, it will accelerate through the transonic regime, which is defined as the flight regime containing mixed sub- and supersonic flow, leading to a nonlinear flow field. This nonlinear environment results in phenomena such as the transonic dip seen in flutter analyses and a large increase in drag. As the vehicle continues to increase in speed, it will enter the super- and then hypersonic flight regimes. The hypersonic regime, defined as when $M > 5$, is characterized by strong shocks combined with high aerodynamic and heating loads, as well as other phenomena such as gas disassociation and shock/boundary layer interactions. As a result of these considerations, any single aerodynamic model that will be used in a controls simulation framework over the entire flight of a hypersonic vehicle will need to be adaptable to these and other phenomena inherent to the flight regimes of interest.

1.2 Literature Review

Aeroelasticity, which is the study of the interaction between fluids and structures when a feedback mechanism exists between the fluid and the elastic deformation, has been the subject of intensive study and has been treated in a number of textbooks, including Refs. 9–11, as well as review papers such as Refs. 12–14. It is important to characterize the aeroelastic behavior of flight vehicles in order to avoid potentially destructive phenomena such as divergence and flutter that can lead to catastrophic structural failures. Aeroelasticity is also of importance when looking at control design and simulation, as unmodeled aeroelastic effects can have a detrimental result on the vehicle controllability. Hypersonic vehicles are no exception to this, and thus hypersonic vehicle controls frameworks, in addition to the one mentioned above, have been an active area of research. Among the earliest simulation frameworks was that of Chavez and Schmidt,[15] which was a two-dimensional representation where the various components were modeled by computationally-efficient, relatively simple models that permitted the results to be in the form of analytical expressions. Parker et al.[16] used approximate curve fits of high-fidelity models of various components to produce a framework specifically tailored for control design.

A rich literature exists for computational methods of unsteady aerodynamics. General reviews of the methods can be found in Ref. 17 for application to aeroelasticity and in Refs. 18 and 19 for specific application to the transonic Mach regime. The methods themselves span a spectrum of complexity. One of the simplest approximations is linear two-dimensional incompressible unsteady potential flow. Among the advantages of using these equations is the fact that direct analytical solutions can often be found. Examples of such solutions are Theodorsen's theory[20] and the finite state aerodynamics of Peters et al.[21] For three-dimensional unsteady linear potential flow, unsteady vortex lattice methods have been developed. An example of a code using such a method is PMARC,[8] which has been employed for a variety of applications, including the modeling of unsteady aerodynamics using the Aerodynamic Impulse Response method.[22,23] Since these methods are linear, they are inherently limited to flight conditions that can be approximated as linear. On the other hand, the transonic regime in particular is highly nonlinear in nature, and thus alternative methods are needed. Solving the transonic small disturbance equations, in which the governing equations are written in terms of a velocity potential, is commonly used in this regime. Batina[24] developed an approximate factorization method to solve the equations which was then implemented into the CAP-TSD transonic unsteady aerodynamic code.[7] This code has been used for a range of applications, including aeroelastic computations of a scale F-16 model[7] as well as wing-flutter calculations on a swept wing.[25]

At the other end of the computational aeroelasticty spectrum are the solutions to the inviscid Euler equations and the full-order Navier-Stokes equations, accounting for viscosity in the flow. Reference 26 provides descriptions and examples of an array of CFD methods and applications for aeroelastic computations. Numerous solvers have been developed for these equations, but for this thesis, the flow solver CFL3D,[27] developed at NASA Langley, is utilized. Capable of solving the Euler and Navier-Stokes equations, the results of this code have been successfully compared with wind tunnel data at NASA.[28] While computationally expensive, these codes are a good basis from which to obtain reduced-order models.

The literature also contains a number of unsteady aerodynamic analysis techniques specifically aimed at the hypersonic regime. Surveys of some of these methods, including piston theory and Newtonian impact theory, among others, can be found in Refs. 29 and 30. Additionally, Scott and Pototzky[31] developed a method to calculate the aerodynamics for hypersonic flutter analyses by computing two steady CFD solutions per vibratory mode shape.

A number of different types of reduced-order modeling methodologies have been developed and presented in the literature; a general overview of several of these can be found in Refs. 32 and 33. Each of these methods involves conducting a finite number of CFD simulations up-front and then using the extracted data for model construction. The first method is proper orthogonal decomposition (POD).[34,35] POD uses a set of snapshots from CFD simulations in order to construct basis vectors which best represent the flow field. Initially, the number of variables in the problem is total number of gridpoints in the mesh multiplied by the number of flow field variables considered. The POD methodology reduces the number of variables down to the number of basis functions multiplied by the number of flow field variables. In most cases, the number of basis functions is much smaller than the number of grid points, thus significantly reducing the order of the problem. The next type of ROMs are those that utilize the fitting of coefficients, computed through the use of CFD simulations, in order to calculate flow field quantities. An example of this type is the autoregressive moving average (ARMA) model, which has been used to calculate the aerodynamic coefficients[36] and generalized aerodynamic forces[37] due to an airfoil gust response as well as for the basis of a system identification approach for computational aeroelastic analysis.[38] Another type of ROM methodology is surrogate modeling. Rather than repeatedly evaluating a function for a specific variable of interest through the use of individual, computationally-expensive CFD simulations, a surrogate, or representative, model of the function is constructed from a comparatively small number of function evaluations. Then, any future time the function needs to be evaluated, the corresponding value can just be picked off of the representative model in a computationally cheap manner rather than having to conduct further expensive full-order function evaluations. Examples of the applications of surrogate modeling include Glaz et al.,[39–41] who evaluate the unsteady aerodynamics on a 2-D rotating airfoil as well as the dynamic stall effects of a helicopter rotor blade, and Da Ronch et al.,[42] who generate kriging models for aerodynamic lookup tables for a number of different aircraft configurations. Finally, convolution and Volterra series-based methods[43,44] have been used to model unsteady aerodynamics as well. Convolution methods take advantage of Duhamel's integral to combine a system's unit step or impulse response with some arbitrary motion to calculate a quantity of interest for a linear system. The Volterra series[45,46] is the nonlinear analog of convolution with the addition of higher-order nonlinear terms. Early efforts to apply the Volterra series to aerodynamic problems include Baumann et. al,[47] who used Volterra series to compute aircraft flying quality parameters.

The ROM methodology presented in this thesis combines linear convolution with a nonlinear correction factor. During ROM simulations, the correction factor function itself is computed with the help of a representative surrogate model. This combination of methodologies is chosen for two reasons. First, convolution-type methods have previously been shown to be an effective unsteady aerodynamics modeling tool across a variety of flight conditions. Second, the nonlinear correction factor allows one to account for geometric and flight conditions away from those for which the model is constructed.

The idea to use convolution-type equations for unsteady aerodynamics is by no means a recently-developed one. Researchers have used indicial functions, which describe the responses of a system to a step-type of input, as a tool to investigate various aerodynamic problems since at least the 1920's, when Wagner[48] used indicial functions to calculate the lift on a two-dimensional thin airfoil. In 1940, Jones[49] used indicial functions to investigate the gust responses of finite aspect ratio wings. Other early studies include Heaslet and Lomax,[50] who used indicial functions to look at the response of a two-dimensional supersonic airfoil due to gusts as well as changes in angle of attack, and Tobak,[51] who investigated the unsteady aerodynamics during short period oscillations of both tailless and tailled aircraft configurations. Later on, Tobak[52] looked at indicial functions as a possible tool for the investigation of aerodynamic bifurcation phenomena. These studies relied on analytical expressions for the quantities of interest and thus could be analyzed in a continuous-time sense for the relatively simple configurations to which they were applied. However, computational simulations require quantities in a discrete-time domain, with values calculated at specific discrete instances in time. Early efforts of such numerical simulation of indicial responses include Beam and Warming,[53] who investigated the use of a third-order finite difference scheme to solve the Euler equations, and Ballhaus and Goorjian,[54] who used the indicial functions for transonic flutter analysis. In order to transform the Volterra equations to discrete time, Clancy and Rugh[55] looked at the identification of Volterra series kernels for discrete-time polynomial systems. This extension of the Volterra and convolution integrals to discrete time paved the way for its further implementation into CFD codes and simulations.

In his Ph.D. thesis, Silva[43] developed a method to calculate the unsteady aerodynamic loads on a transonic airfoil using first- and second-order Volterra series kernels obtained from CFD simulations. Then, a framework was developed to transform the step/impulse responses into a state-space reduced-order model of the system, in which a certain output is a function of current and past system state variables and system inputs; from this state-space representation, flutter results could be obtained.[56] In this framework, each elastic mode shape was considered separately, so separate simulations needed to be conducted for each individual mode shape. This issue can become problematic when the number of mode shapes increases. To remedy this problem, Silva[57] developed a method for the construction of the state-space system which permits the excitation of multiple mode shapes in the course of a single CFD simulation, thus increasing the computational efficiency by decoupling the number of CFD simulations from the number of mode shapes under consideration. Kim[58] and Kim et al.[59] have also researched methods to reduce the number of CFD simulations through the use of a single composite input comprised of multiple system inputs for state-space system construction. Gaitonde and Jones[60] developed a method for computing a continuous ROM based on CFD impulse responses, and Allen et al.[61] validated the model against full-order solutions for flutter boundary predictions for a two-dimensional airfoil/control surface combination. Singh and Baeder[62] computed indicial responses to changes in angle of attack for a three-dimensional wing and then used the solutions to build numerical databases.

Other important aspects of convolution/Volterra-type ROMs have received considerable attention as well. Lind et al.[63] used Volterra series to model the otherwise unmodeled nonlinear dynamics consisting of the difference between linear models and measured flight test data. Munteanu et al.[64] researched the use of a Volterra-based ROM on situations having structural as well as aerodynamic nonlinearities. Raveh investigated the use of step versus impulse-type of inputs into the ROM methodology[65] and then conducted a flutter analysis using the step-type of inputs.[66] Balajewicz et al.[67] developed a ROM methodology incorporating multi-input Volterra series to more efficiently model systems with multiple degrees of freedom, such as an airfoil undergoing pitch and plunge motions. In another paper, the same authors[68] developed a method to reduce the computational cost of Volterra series analysis by only considering the terms on the diagonals of the kernels. One potential limiting factor in Volterra analysis is the dimensionality of kernels when more higher-order terms are added. Kurdila and Prazenica[69] addressed this by looking at approximating the higher-order kernels with wavelet representations. In another paper, the same authors[70] extended this formulation to piecewise-polynomial multiwavelets having, among other desirable properties, closed-form solutions. Khawar et al.[71] took this approach, meant for single-input/single-output systems, and developed a method for use on multi-input/multi-output aeroelastic systems. Volterra analysis has also been used in attempts to control and suppress flutter. Marzocca et al.[72] used a closed-loop feedback system incorporating Volterra analysis in order delay flutter onset and make the flutter boundary a less destructive one. Others have further explored the general applicability of convolution/Volterra ROMs to certain flow conditions and/or configurations. Ghoreyshi et al.[73] investigated the agreement of Volterra-type ROMs with CFD results for a variety of different maneuvers for two- and three-dimensional geometry test cases. Lisandrin et al.[74] analyzed the applicability of linear models to perturbations of a transonic flow field. Finally, efforts to blend more than one type of ROM methodology have been pursued. Lucia and Beran created a hybrid Volterra series-POD ROM methodology which permitted Volterra-type analyses on subsonic[75] and supersonic[76] flow fields by first using POD to reduce the number of flow field unknowns to a relatively small number of fluid modes and then performing Volterra analysis on those modes rather than all variable values at each grid point location throughout the flow field.

In this thesis, a nonlinear correction factor is added to the convolution. Corrections of varying types have been applied to a host of reduced-order modeling methods in the literature as well. Crowell and McNamara[77] used an unsteady piston theory correction to a steady surrogate-type model in the hypersonic regime in order to account for unsteadiness. R. Silva et al.[78] employed a pressure correction to linear aerodynamic models in the transonic regime through the modification of the downwash vector. Thomas et al.[79] used a static/dynamic correction for a ROM methodology based on eigenvectors and POD.

A major issue when creating convolution/Volterra-type ROMs, as well as all ROMs in general, is the parameter range of applicability. For example, could the same ROM used at Mach 0.3 also be used at Mach 0.9? If an entirely different ROM must be constructed each time the Mach number or other parameter in the problem is modified, it would lead to a dramatic increase in ROM computational time, perhaps rendering the ROM methodology infeasible under certain circumstances. Thus, research has been conducted to remedy this problem for convolution/Volterra type of ROMs. Lind and Baldelli[80] used Volterra kernels from wind tunnel test data to compute state-space systems at several different flight conditions, to which a model was then fitted to permit the calculation of quantities of interest away from the specific flight conditions of the test data. Prazenica et al.[81] extrapolated Volterra kernels calculated using flight test data at certain flight conditions to other, different flight conditions, resulting in a ROM valid for varying parameters. Omran and Newman[82] used interpolation of Volterra series submodels constructed at specific flight conditions to build global piecewise Volterra kernels valid over a much larger range of flight conditions. In another work, the same authors[83] used a nonlinear parameter-varying approach consisting of local Volterra models in order to be able to account for strong nonlinearities over multiple aircraft flight regions. Other efforts have investigated the numerical integration of the state-space system itself. Silva[84] created a ROM valid over a range of velocities by changing the numerical integration time step.

1.3 Thesis Overview

A significant limitation of Volterra/convolution ROMs is the size of the step/impulse inputs which can be given to the CFD code. These limitations often stem from the induced grid velocities corresponding to a specific input. Since the induced grid velocities are computed as the change in grid position from one time step to the next divided by the time step, large grid deformations (corresponding, for example, to a large modal input amplitude) result in large velocities. At some point, the grid velocities become too large, and the code will crash. Thus, the size of the step/impulse input able to be given to a code is inherently limited. This becomes a problem when the modal amplitudes of interest in a particular problem are much larger than the maximum allowable step/impulse amplitudes, and the convolution/Volterra results based on the much smaller amplitudes may not be good predictors of the actual response at the larger amplitudes. To solve this issue, this thesis introduces a nonlinear correction factor designed to take these large modal amplitudes into account without the numerical issues arising from large step or impulse inputs. Moreover, the above ROM efforts have tended to focus on a limited range of flight conditions, or in the case of the varying-parameter ROMs, one Mach number regime. As detailed above, since a controls framework requires an aerodynamic model to be valid across a potentially wide range of parameters, the ROM methodology in this thesis is developed to be applicable across a range of Mach regimes, from subsonic flow up to hypersonic flow.

This thesis follows the development and testing of this ROM. Chapter 2 describes the methodology itself and gives details about the ROM construction procedure. Chapter 3 describes the application of the ROM to the hypersonic regime, while Chapter 4 describes the application to the transonic regime. Chapter 5 follows a single example test case from the subsonic regime up through the supersonic Mach regime, evaluating the accuracy of the method in each regime and showing how the method could be used in practice. Finally, Chapter 6 details the relevant conclusions and contributions of this thesis.

Chapter 2

ROM Methodology

This chapter presents the overall reduced-order modeling framework. The linear portion is found through the use of linear convolution, while the nonlinear portion is calculated using a correction factor term. The theory and calculation processes for both of these are given, as is an overview of the CFD code used in this dissertation to find those terms. Also, standard error metrics used to assess the accuracy of the ROM as compared with full CFD simulations are detailed. Finally, a general error estimation procedure to gain *a priori* knowledge of the error expected to be incurred through the use of the ROM is developed.

2.1 ROM Overview

The reduced-order modeling methodology presented in this dissertation combines linear convolution with a nonlinear correction factor. The unsteadiness of the flow is captured through the use of linear convolution, which has been shown to be an effective modeling tool for linear unsteady aerodynamics.[43,65] However, the main drawbacks of a pure convolution ROM stem from the fact that unsteady aerodynamics are in general nonlinear in nature. As described subsequently, the convolution ROM is based upon the aerodynamic response to a step input of a certain magnitude applied to the geometric configuration under consideration. When the actual input magnitude in a particular simulation increases beyond that used in the ROM development, the ROM accuracy begins to erode due to various nonlinearities. Moreover, the convolution ROM may not be valid for flight conditions away from those around which the model is constructed. To address these issues, a nonlinear correction factor has been introduced to this convolution ROM by calculating the response of the system at larger amplitudes and a range of Mach conditions. Thus, rather than being geometry, amplitude, or Mach number-dependent, the general mathematical form of the model does not place any inherent limitations on configurations, input size, or flow conditions for which it is applicable.

In general, the nonlinear corrected ROM response y_{corr} can be written as

$$y_{corr} = f_c y_{conv} \tag{2.1}$$

where f_c is the correction factor and y_{conv} is the linear convolution response.

Figure 2.1 shows the schematic of the overall ROM framework. To begin, the inputs to the model are the structural mode shapes of the geometry as well as the modal deformations at each time step throughout the simulation. The structural mode shapes are used in CFD simulations for both modal step inputs and correction factor calculation; the details of these runs will be described subsequently. However, since these structural mode shapes are known *a priori*, and the CFD runs only require knowledge of these structural mode shapes and not the per-iteration modal amplitudes, all CFD runs can be conducted up front, prior to model construction. Thus, once completed, the CFD calculations are taken out of the ROM loop, leaving linear convolution and nonlinear correction factor application as the two in-the-loop ROM features. The calculation of these two items takes orders of magnitude less than the full CFD simulations, thus making the ROM very computationally efficient. Finally, the outputs are the time-accurate force and moment coefficients or the generalized forces.

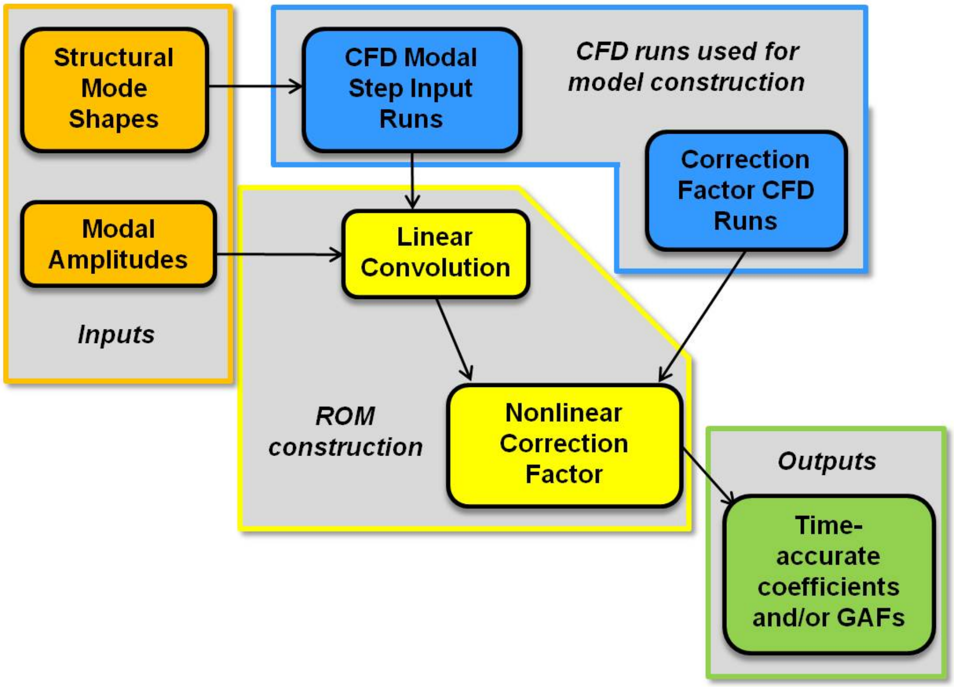

Structural Mode Shapes
Modal Amplitudes
Inputs
CFD Modal Step Input Runs
CFD runs used for model construction
Correction Factor CFD Runs
Linear Convolution
Nonlinear Correction Factor
ROM construction
Outputs
Time-accurate coefficients and/or GAFs

Figure 2.1: Overall ROM framework

2.2 Linear Convolution

The response of a linear system to an arbitrary input at time t can be found if the response of the system to a unit step $(H\left(t\right))$ or unit impulse $(h\left(t\right))$ function is known. In the continuous time form, the impulse and step input functions are written as follows, for a case in which the step/impulse is applied at time t_0:

$$\begin{array}{ll} \textbf{\textit{Impulse}} & \textbf{\textit{Unit Step}} \\ u_{imp.}\left(t-t_0\right)=\infty,\ t=t_0 & u_{step}\left(t-t_0\right)=0,\ t<t_0 \\ u_{imp.}\left(t-t_0\right)=0,\ t\neq t_0 & u_{step}\left(t-t_0\right)=1,\ t\geq t_0 \end{array} \tag{2.2}$$

The response $y(t)$ due to an arbitrary input $f(t)$ is found through the use of convolution:[43,85]

$$y\left(t\right)=f\left(0\right)H\left(t\right)+\int_0^t\frac{df}{dt}\left(\tau\right)H\left(t-\tau\right)d\tau \tag{2.3}$$

The unit impulse is the derivative of the unit step, so integration by parts yields

$$y\left(t\right)=f\left(t\right)H\left(0\right)+\int_{0}^{t}f\left(\tau\right)h\left(t-\tau\right)d\tau \tag{2.4}$$

Equations 2.3 and 2.4 are the two forms of Duhamel's integral. In general, convolution can be thought of as a summation of scaled and shifted step/impulse responses.

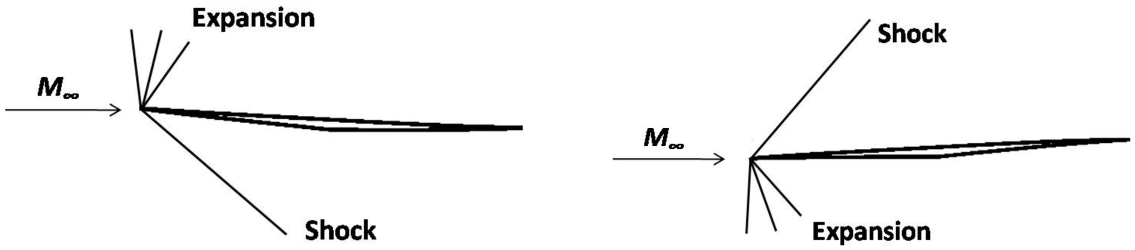
Rather than using the continuous-time form of Duhamel's integral, the application of the convolution integral to a CFD code requires its discrete form.[43] The definitions are slightly different for the discrete case, where the input values are only defined for specified points in time. Thus, the impulse/step inputs are given as follows, with the input occurring at time step 1:

$$\begin{array}{ll} \textbf{\textit{Impulse}} & \textbf{\textit{Unit Step}} \\ u_{imp.}\left[n\right]=\frac{1}{\Delta t},\ n=1 & u_{step}\left[n\right]=0,\ n<0 \\ u_{imp.}\left[n\right]=0,\ n\neq 1 & u_{step}\left[n\right]=1,\ n\geq 1 \end{array} \tag{2.5}$$

where the square brackets denote a value at a specified integer time step n. Equation 2.5 leads to the two forms of the discrete convolution integral:[66]

$$\begin{array}{ll} \textbf{\textit{Impulse:}} & y\left[n\right]=h\left[0\right]+\sum_{k=0}^{n}f\left[n\right]h\left[n-k\right]\Delta t \\ & \\ \textbf{\textit{Step:}} & y\left[n\right]=f\left[0\right]H\left[n\right]+\sum_{k=0}^{n}\left(u\left[n\right]-u\left[n-1\right]\right)H\left[n-k\right] \end{array} \tag{2.6}$$

In this work, the step input is chosen over the impulse input for use in the convolution integral due to both ease of implementation into the CFD code and the quality of the response found. This improvement of results using the step over impulse input was already noted by Raveh.[65] Also, note that the system inputs considered here are modal deformations. Thus, whatever modal deformation is used for the step input is considered to have a scaled value of 1. Because of this, for the rest of this dissertation,



(a) Positive step

(b) Negative step

Figure 2.2: Modal step inputs

all modal input values will be given in multiples of the step input, which makes the step value the unit value.

The first step in calculating the linear convolution ROM (y_{conv} from Eq. 2.1) is to calculate the responses to both a positive and negative step input for each mode being considered in a particular problem. To illustrate this, consider the sample asymmetric 2-D half-diamond airfoil geometry shown in Fig. 2.2 encountering hypersonic flow. In general, the responses, such as the force and moment coefficients, to positive and negative step inputs will be neither the same nor exactly the opposite of each other, as shown by the various shock/expansion fan systems in the figure. Thus, it is important to find the response to a step input in each of the positive and negative directions.

After the positive and negative step responses have been found, the next task is to use Eq. 2.6 to find the linear response to some arbitrary modal input. Note that, prior to implementing Eq. 2.6, the steady-state value of the particular quantity is subtracted from the entire step response, and this value is added back in at the end of the calculation process. The linear response is calculated two separate ways, once using the response to the positive step and once using the response of the negative step. Suppose that the positive step causes an increase in some quantity Q (Q may be c_l, c_d, etc.), and a negative step causes a decrease in the same quantity. The positive response would be expected to be valid for situations when Q is greater than the undeformed configuration value and vice-versa for the negative response.

Next, the ratio r_{pn} between the maximum value of Q in the response calculated through implementation of Eq. 2.6 using the positive step and the response calculated using the negative step is found:

$$r_{pn} = \frac{\max\left(\text{positive convolution}\right)}{\max\left(\text{negative convolution}\right)} \tag{2.7}$$

The final linear convolution ROM can thus be calculated in one of two ways, based on either the positive or negative step response. First, consider the positive response. Whenever Q is above the undeformed value, the ROM will consist of the positive step response as is. However, when Q drops below the undeformed value, the positive step response will be divided by r_{pn}. The second method is to use the negative step response when Q is below the undeformed value and multiply by r_{pn} when Q is above it. When compared, the results of both methods are effectively equivalent. To illustrate this process, the positive step convolution, negative step convolution, and final linear ROM for a sample case using the geometry in Fig. 2.2 are shown in Fig. 2.3. The final mathematical expression for this linear ROM y_{conv} is shown as follows,

$$\begin{aligned} y_{conv}\left[n\right] &= Q_0 + y\left[n\right] \qquad && Q \geq Q_0 \\ y_{conv}\left[n\right] &= Q_0 + r_{pn} y\left[n\right] && Q < Q_0 \end{aligned} \tag{2.8}$$

where $y\left[n\right]$ is the response value found using Eq. 2.6, and Q_0 is the steady-state, undeformed configuration value of the quantity Q of interest.

2.3 Correction Factor

In this form, the ROM will only work for flight conditions and input amplitudes near to those used for the step input, as the responses do not in general scale linearly with oscillation amplitude. Consider again the sample 2-D half-diamond airfoil shown in Fig. 2.2 in hypersonic flight undergoing oscillations of the first bending mode with a

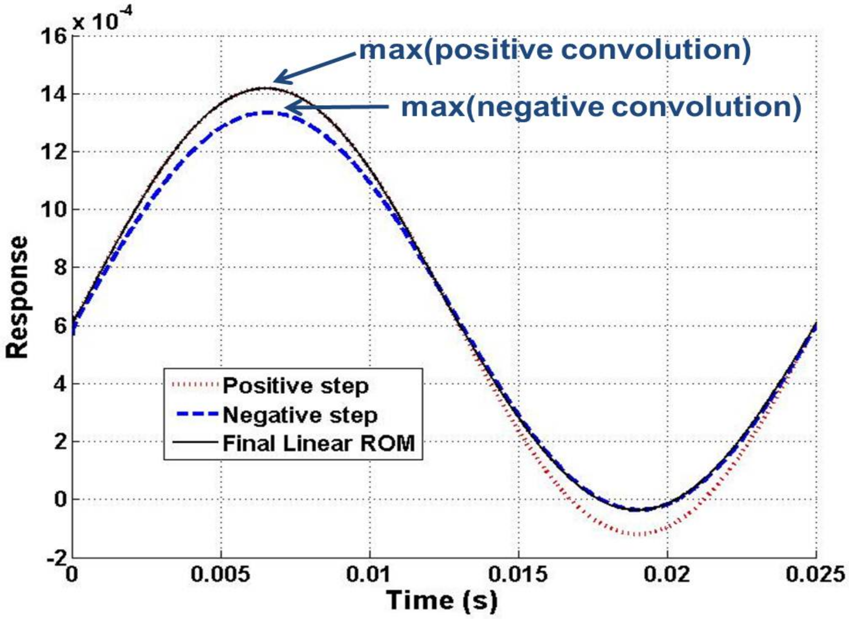


Figure 2.3: Linear convolution responses

maximum amplitude 40 times larger than the one used for the original identification of the step response. Figure 2.4 shows a sample drag coefficient comparison between the linear ROM y_{conv} and direct CFD result for this case. The response y_{conv} in this case is a qualitatively very poor representation of the actual CFD results, demonstrating the necessity of a nonlinear correction.

To obtain the nonlinear corrected ROM response y_{corr}, a correction factor f_c is introduced. This quantity is defined as the ratio between the steady linear (y_{lin}) and nonlinear (y_{nonlin}) responses of a certain configuration due to the modal deformations and flight conditions at a particular instant in time, that is,

$$f_c \equiv \frac{y_{nonlin}}{y_{lin}} \tag{2.9}$$

In computing the correction factor value, the first challenge is to calculate y_{nonlin}. The first step in doing so is to identify the input values of interest, including the Mach number and modal amplitudes, and apply these inputs to each mode shape

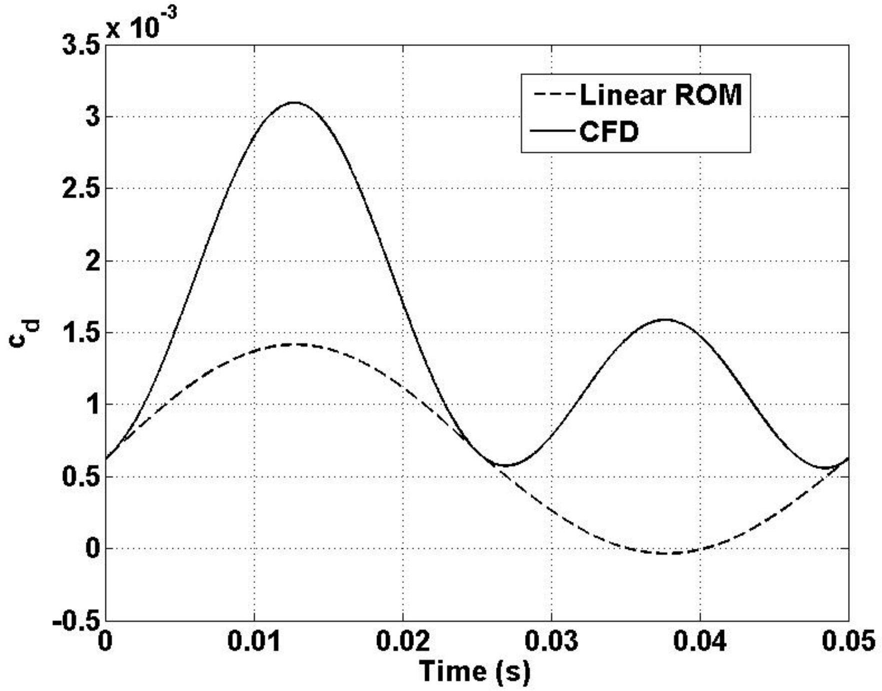


Figure 2.4: Linear ROM vs. CFD at large oscillation amplitude

simultaneously. Then, the response to these inputs is allowed to converge to some steady value, and, after the subtraction of the steady-state value, this is taken to be y_{nonlin}. Next, y_{lin} is found by first determining the final, steady response value for each mode after a step input for that particular mode has been applied individually. Then, each one of these values is multiplied by the respective modal amplitude of interest to find the individual linear modal responses. The value y_{lin} is then computed by summing these individual modal responses using superposition. On the other hand, y_{nonlin} is found by applying a composite input of multiple modal amplitudes simultaneously and then finding the final response. These calculation procedures for y_{nonlin} and y_{lin} are illustrated in Figs. 2.5 and 2.6, respectively. y_{nonlin} and y_{lin} differ from y_{corr} (nonlinear ROM response) and y_{conv} (linear convolution ROM response), respectively, in that y_{corr} and y_{conv} involve the convolution integral in their calculation. Thus, the modal velocities, which enter the convolution integral as the time derivative of the arbitrary modal inputs, are included in the calculation of y_{conv} and y_{corr} but not in the calculation of y_{nonlin} and y_{lin}.

For a purely linear system, the correction factor value will be 1. In certain situations, the individual responses used for y_{lin} calculations will sum to be equal to or very close to zero, resulting in a y_{lin} value approaching zero and hence an f_c value approaching infinity. For these situations, the definition is modified by the addition of an offset term δ:

$$f_c = \frac{y_{nonlin} + \delta}{y_{lin} + \delta} \tag{2.10}$$

Note that δ is placed in the numerator as well as the denominator such that a linear system will still have a correction factor value of 1.

With this correction factor definition in place, the corrected ROM value y_{corr} is calculated by:

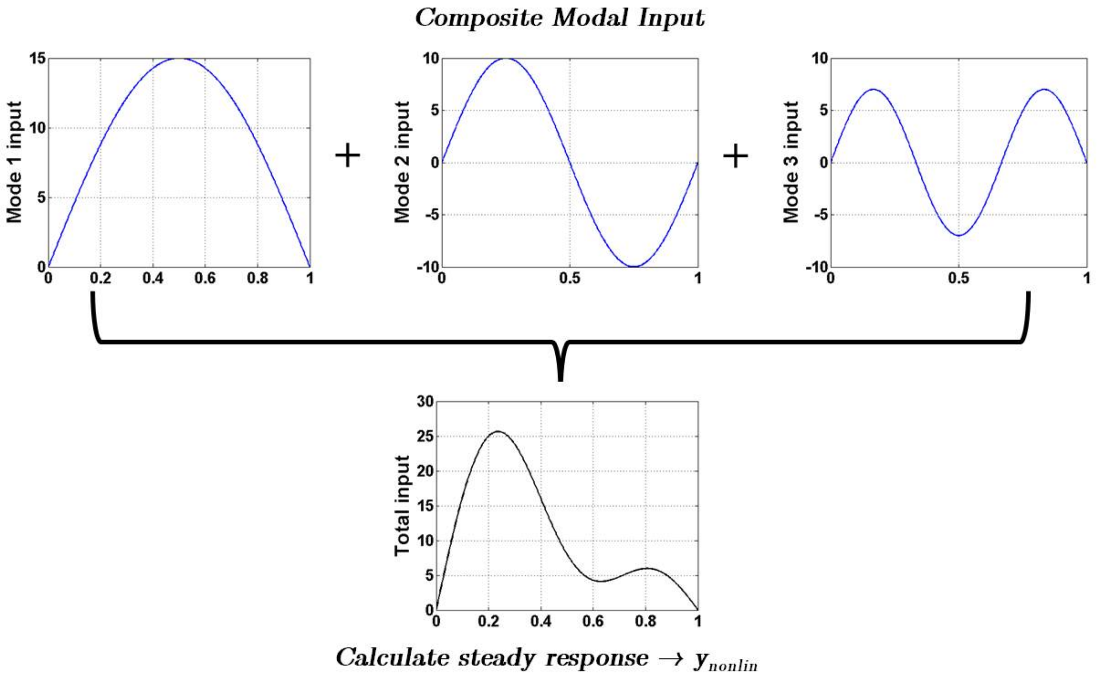


Figure 2.5: Schematic for the calculation of y_{nonlin}

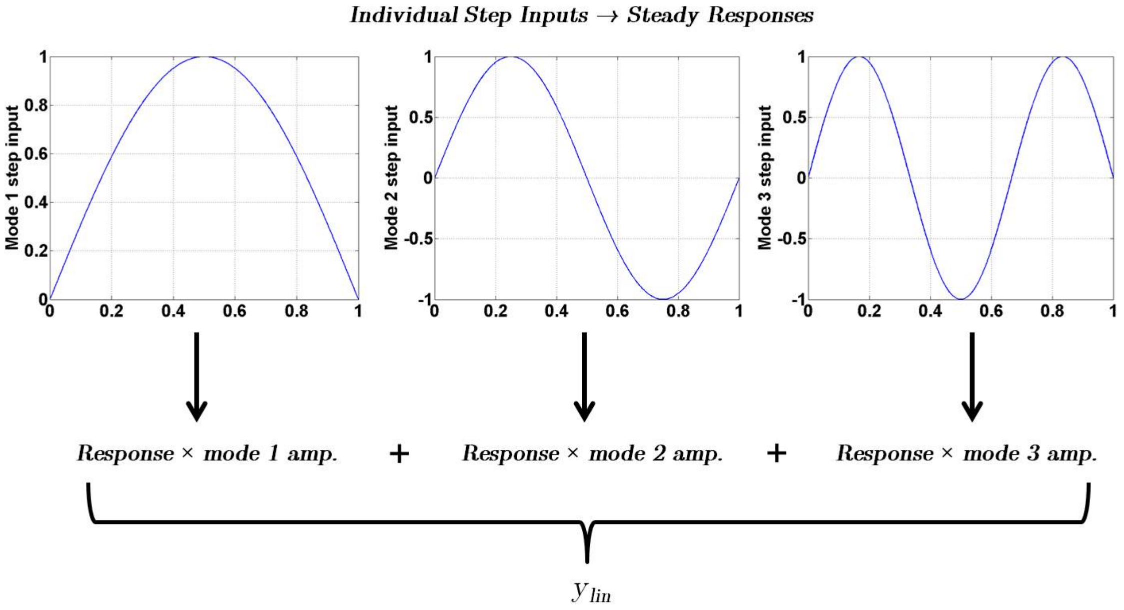


Figure 2.6: Superposition of responses in the calculation of y_{lin}

$$y_{corr} = (f_c)\, y_{conv} = \left(\frac{y_{nonlin}}{y_{lin}}\right) y_{conv} \tag{2.11}$$

This leads to the basic correction factor assumption, that the ratio of the steady response values at a particular time step t will be equal to the ratio of unsteady response values at that particular time step, namely:

$$\left.\frac{y_{nonlin}}{y_{lin}}\right|_t = \left.\frac{y_{corr}}{y_{conv}}\right|_t \tag{2.12}$$

The errors between the ROM and full-order CFD simulation results will characterize how valid this assumption is throughout various flight regimes, as conditions resulting in larger ROM errors will show likely areas where this assumption breaks down.

Now that the correction factor has been defined, the challenge is to find its value over the entire parameter space being considered, which in this work consists of modal amplitudes and Mach number. Using CFD to directly calculate f_c at every point of interest would be prohibitive in terms of computational cost. To solve this problem, consider the difference between computer experiments and actual physical experiments. Each time a physical experiment is repeated, the result will not be exactly the same as the time before due to measurement and other inherent random errors. However, a computer experiment will give the exact same result each time it is completed, and thus each response value in a certain parameter space would be expected to be an exact value of the response quantity.

2.3.1 Kriging

Kriging[86] is a methodology that takes advantage of this lack of random error to create a representation of the response function based on the results at a certain number of sampling points. The kriging function predictor $\hat{y}\left(X\right)$ is a combination of a regression model and a random process $Z\left(X\right)$, which are the first and second terms, respectively, in the following equation:[87]

$$\hat{y}\left(X\right)=\sum_{j=1}^{k}\beta_j f_j\left(X\right)+Z\left(X\right) \tag{2.13}$$

For the regression model, $f_{j,j=1..k}\left(x\right)$ are the set of k regression functions, and β_j are the set of regression parameters. The random process has a mean zero and covariance of $\sigma^2\mathcal{R}$, where σ is the process variance and $\mathcal{R}$ is the correlation model. The goal of the kriging method is to minimize the mean squared error φ of the predictor $\hat{y}$ over the parameter space, which is found by the following equation, where $E[\ ]$ denotes the covariance of a particular quantity:[87]

$$\varphi\left(x\right)=E\left[\left(\hat{y}\left(x\right)-y\left(x\right)\right)^2\right] \tag{2.14}$$

If a linear predictor is assumed over the parameter space for the value of $\hat{y}\left(x\right)$, this quantity can be expressed as[87]

$$\hat{y}\left(x\right) = c^{\mathrm{T}} Y \tag{2.15}$$

where Y are the outputs from sampled values. The kriging methodology finds the best linear unbiased predictor c^{T} by minimizing the mean squared error prediction. The result of this minimization procedure is that the predictor can be written as[88]

$$\hat{\mathbf{y}}\left(\mathbf{x}\right) = \mathbf{f}^{\mathrm{T}}\left(\mathbf{x}\right)\hat{\beta} + \mathbf{r}^{\mathrm{T}}\left(\mathbf{x}\right)\mathbf{R}^{-1}\left(\mathbf{Y} - \mathbf{F}\hat{\beta}\right) \tag{2.16}$$

where $\mathbf{F}$ is the vector of f_j at the sampling points, $\mathbf{R}$ is the correlation function matrix, $\mathbf{r}\left(x\right)$ is the correlation between an unknown point x and the known sampling points, and $\hat{\beta}$ is the least squares predictor given by[88]

$$\hat{\beta} = \left(\mathbf{F}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{F}\right)^{-1}\mathbf{F}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{Y} \tag{2.17}$$

In this research, the kriging methodology is implemented using the MATLAB Design and Analysis of Computer Experiments (DACE) toolbox's built-in functions.[87]

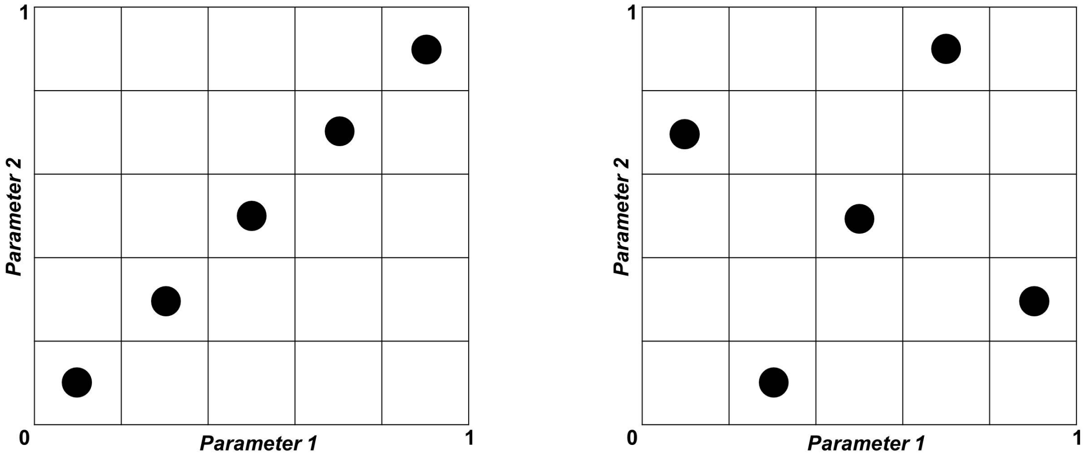
2.3.2 Latin Hypercube Sampling

In order to obtain the values to use for kriging surface construction, one must select appropriate sampling points within the parameter space. As the number of parameters increases in a particular problem, it becomes more difficult to conduct simulations pairing every value of one parameter with every value of all other parameters. For example, consider a problem with five separate parameters. Suppose that the range of each parameter is broken into ten intervals. In order to test each parameter value with all other parameter values, 10^5 trials would need to be conducted, which in many applications, including CFD simulations, is generally not feasible. Thus, it is important to be able to smartly sample the parameter space such that the behavior of the response function is known, but the overall number of trials to be run is not prohibitively high.

For this purpose, Latin hypercube sampling (LHS) is employed.[89] LHS works by first dividing up the range for each parameter into a user-defined number of intervals. Then, one sampling point is placed in each of the intervals for each parameter. Consider Fig. 2.7, which shows a sample parameter space consisting of two separate parameters divided into the intervals shown. Both Figs. 2.7(a) and 2.7(b) are examples of a potential Latin hypercube sampling configuration. However, in a strictly qualitative sense, it is obvious that Fig. 2.7(b) does a better job of "smearing" the sampling points more evenly throughout the parameter space. Thus, using any Latin hypercube design is not enough to guarantee a good sampling distribution, as it is important in many cases to sample points as evenly as possible. Because of this, LHS is furthered by the concept of orthogonal or nearly orthogonal Latin hypercube sampling,[90] which works to minimize the correlation among the various vectors of sampling point parameter values. In this research, the sampling point values are obtained through MATLAB's built-in `lhsdesign` command. Anywhere from $10,000$ to $100,000$ iterations of the command are run, and the sampling points used are selected from all the iterations using either the `maximin` option for the points with maximum minimum distance from each other in the parameter space or the `correlation` option for the vectors of input quantities to have the minimum correlation with each other.

2.3.3 Correction Factor CFD Runs

Much of the potential difficulty in calculating the correction factor at a certain point lies in the ability to calculate the quantity y_{nonlin} at that point. The quantity y_{lin} is calculated from various quantities obtained from the individual modal step inputs, so nothing new needs to be calculated for this term at each individual sampling point. However, for y_{nonlin}, the response quantity at a certain sampling point is calculated by inputting all modal input values simultaneously, requiring an individual CFD run



(a) Non-evenly spread LHS points

(b) More evenly-spread LHS points

Figure 2.7: Example LHS sampling points

at an individual sampling point. For some situations, these modal deformations are significantly larger than those used for the step inputs. This becomes a problem when the modal inputs become so large that the CFD code cannot input the deformations as step inputs without numerical issues arising in the obtained solution due to the resulting very large grid velocities (in many cases, the code will crash due to the large inputs). This problem can be solved by considering the fact that y_{nonlin} only relies on the final, steady response value after the desired inputs have been given. Thus, it does not matter if the inputs were given as steps or by gradually increasing the amplitude up to the final value. Because of this, to find y_{nonlin} for large amplitudes, the modal amplitudes are sinusoidally increased up to the final value and then leveled off. Once the response has reached a steady state, that value is used for the quantity y_{nonlin}. A sample response, including the labeling of the value to be used for y_{nonlin}, is shown in Fig. 2.8 an example of this process.

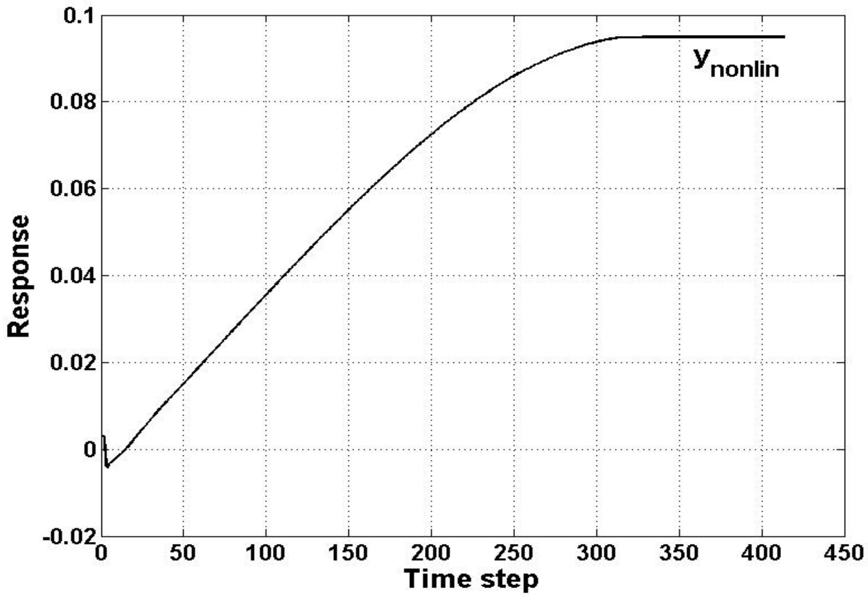


Figure 2.8: CFD response for finding y_{nonlin}

2.4 ROM Testing

The accuracy of the ROM methodology is analyzed by comparison of ROM results with computational results obtained from full CFD simulations. For these comparison test cases, sinusoidal modal inputs are given to the various modes under consideration. The CFD code used in this study is CFL3Dv6, developed at NASA Langley.[27] The code is capable of solving the Euler/Navier-Stokes equations for both steady and unsteady flows on two and three-dimensional structured grids and has mesh deformation capability. For the unsteady simulations, the CFD code inputs are modal deformations, in the form of step inputs or sinusoidal inputs, depending on the type of run being conducted. Grid velocities are derived from these modal inputs. For example, if a step input of amplitude a is given to a particular mode at time step n_0 and Δt is the time step being used, the grid velocities $\dot{\eta}$ are calculated as:[27,91]

$$\begin{aligned} \dot{\eta}\left[n\right] &= \tfrac{a}{\Delta t},\ n = n_0 \\ \dot{\eta}\left[n\right] &= 0,\ n \neq n_0 \end{aligned} \tag{2.18}$$

The response quantities tracked in this dissertation are the lift, drag, and moment coefficients, though the the generalized aerodynamic forces (GAFs) could also be chosen. All of these quantities are directly output by the CFL3D code.

2.5 Error Metrics

Two separate error metrics are used to judge the accuracy of the ROM compared with the CFD. The first metric, the L_1 error, is characterized by finding the mean absolute difference between the ROM and CFD results at each time step; it is normalized by the range spanned by the CFD results. For a simulation over N time steps, it yields:

$$L_1 \text{ error} = \frac{\frac{1}{N}\sum_{i=1}^{N}\left(\left|y_{ROM,i} - y_{CFD,i}\right|\right)}{\max\left(y_{CFD}\right) - \min\left(y_{CFD}\right)} \times 100\% \tag{2.19}$$

where $y_{ROM,i}$ and $y_{CFD,i}$ are the respective ROM and CFD response values found at time step i, and the denominator represents the difference between the maximum and minimum values found over all time steps of the ROM response.

The second error metric is the L_∞ error, defined as

$$L_\infty \text{ error} = \frac{\max\left(\left|y_{ROM} - y_{CFD}\right|\right)}{\max\left(y_{CFD}\right) - \min\left(y_{CFD}\right)} \times 100\% \tag{2.20}$$

Rather than the mean value of the difference over all time steps, the L_∞ error finds the maximum ROM-CFD difference over all time steps and normalizes this quantity by the same range as in the L_1 error.

2.6 ROM Error Estimation

When implementing this or any other ROM methodology, it is important to have some sense of the magnitude of error expected to be incurred through the use of the ROM. Ideally, if the user has a sense beforehand as to how much error they would accept, the ROM should be able to be constructed accordingly in order to fit within these error tolerance parameters. When considering the ROM's overall error, two separate areas need to be considered. The first is the error of the kriging surface compared to the function it is modeling, in this case the correction factor function over the parameter space. Among the important issues faced when constructing the ROM is deciding on how many sampling points are needed for the correction factor kriging surface. Too few points would result in an inaccurate representation of the function and thus loss of accuracy of the ROM in general. However, using more points than necessary would result in unneeded computational expense. Thus, the first part of the error estimation focuses on finding the optimal number of sampling points to use. One method which assesses the error of kriging surfaces is the Efficient Global Optimization (EGO) algorithm.[92, 93] The purpose of this algorithm is to find global maxima and minima on surfaces; this is accomplished by placing points at locations of maximum expected improvement and uncertainty on the surface. The method presented here is similar to the EGO algorithm except for the fact that the purpose is to simply minimize the error on the surface, not to find the specific location of extrema. Thus, the addition of sampling points is based solely on the mean squared error of a location on the surface, not the likelihood of a new extremum being found at a certain location.

The second area of ROM error analysis is the error of the function when compared with the truth model, considered in this research to be the CFD results. Even if the kriging surface matched the intended function exactly, the methodology would still result in some error. The second part of the error estimation focuses on quantifying this error.

2.6.1 Error of Kriging Surface Compared to Function

Due to the high computational expense of CFD simulations and thus kriging surface sample point calculations, it is important to know the number of points and location within the parameter space of each point before beginning model construction. Because of this, it is not feasible to use CFD itself to determine these items. Instead, some sort of simplified, computationally inexpensive models must be used. For example, when considering a hypersonic test case, piston theory has been chosen as a simplified model; specifics of those tests and results are presented in a subsequent chapter.

Figure 2.9 shows a diagram of this error analysis methodology. To start with, an initial number of sampling points within the parameter space is selected using Latin hypercube sampling. At each sampling point, a simplified model is used to calculate the final coefficient values instead of CFD. Then, using Eq. 2.9, these simplified model values are combined with CFD step responses to find the correction factor values at each of the points. Note that CFD is used for the step responses due to the relatively low computational cost involved, as the step response in general does not need to be at the same flight conditions as the sampling point in the parameter space, and thus only a small number of step responses will need to be calculated. Next, a kriging surface is constructed with the available sampling point data, and the maximum mean squared error (MSE) is calculated at points throughout the surface. This error s^2 at location x^* in the parameter space is found as follows:[92]

$$s^2\left(x^*\right) = \sigma^2 \left[1 - \mathbf{r}'\mathbf{R}^{-1}\mathbf{r} + \frac{\left(1 - \mathbf{1}'\mathbf{R}^{-1}\mathbf{1}\right)^2}{\mathbf{1}'\mathbf{R}^{-1}\mathbf{1}}\right] \tag{2.21}$$

In Eq. 2.21, σ is the surface's variance, $\mathbf{r}$ is a column of the correlation matrix $\mathbf{R}$, and $\mathbf{1}$ is a column vector of ones. See Refs. 87, 94, and 92 for the derivation and a

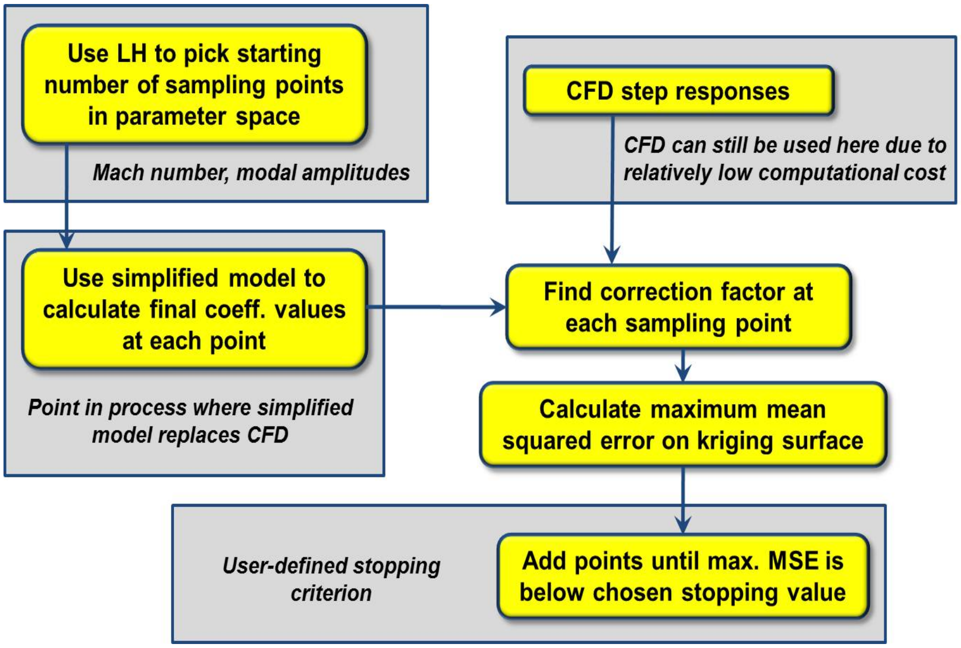

Use LH to pick starting number of sampling points in parameter space
Mach number, modal amplitudes
CFD step responses
CFD can still be used here due to relatively low computational cost
Use simplified model to calculate final coeff. values at each point
Point in process where simplified model replaces CFD
Find correction factor at each sampling point
Calculate maximum mean squared error on kriging surface
User-defined stopping criterion
Add points until max. MSE is below chosen stopping value

Figure 2.9: Error analysis: comparison of Kriging surface to function

detailed explanation. In this work, this error calculation is obtained from a built-in MATLAB subroutine.

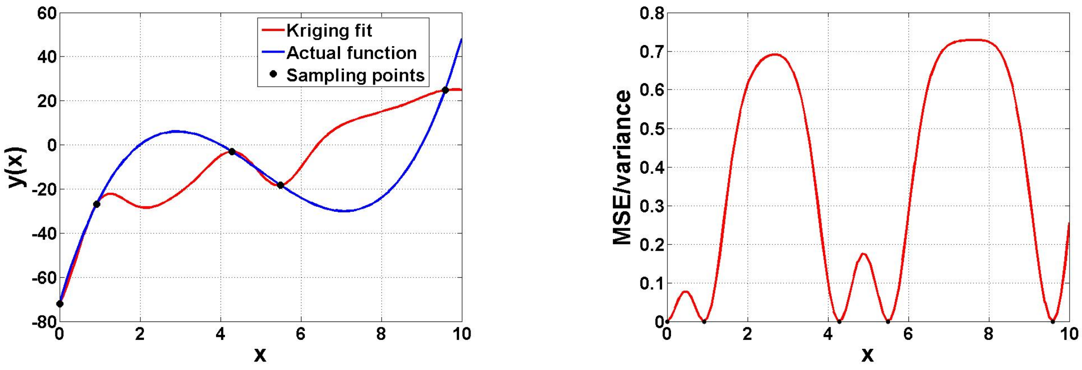
A new sampling point is then added at the location of the maximum MSE. The process is repeated until the error has fallen below the designated stopping criterion. A benefit of this methodology is that the stopping criterion can be input by the user and is quantitative rather than qualitative.

For an example, consider Figs. 2.10 and 2.11, which show a simple graphical example of this process. Five sampling points are at first selected to model the sample function $y = (x-2)(x-4)(x-9)$. The error criterion for this case is defined in terms of the surface's variance,

$$\max\left(s^2\right) < 0.01\sigma^2 \tag{2.22}$$

where $\max(s^2)$ is the maximum MSE on the surface.

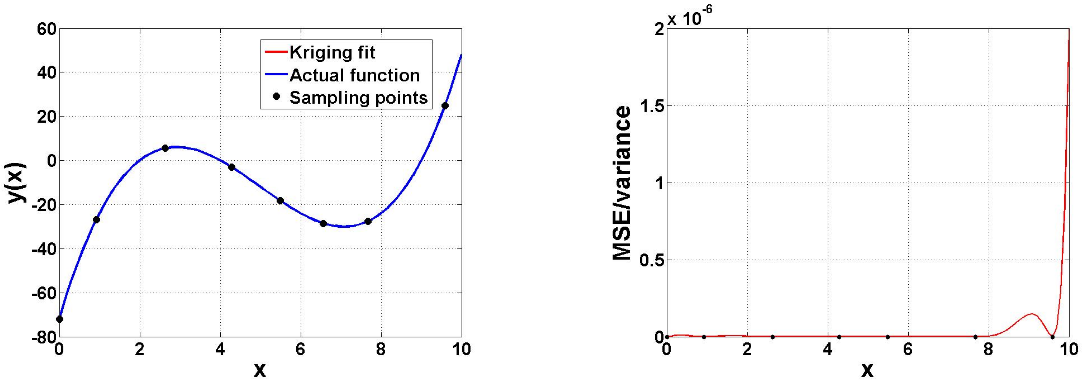
The corresponding kriging fit and MSE plot are shown in Fig. 2.10. Then, the



(a) Kriging fit

(b) Mean squared error

Figure 2.10: Kriging fit and MSE with initial sampling points



(a) Kriging fit

(b) Mean squared error

Figure 2.11: Kriging fit and MSE, error criterion satisfied

above process is applied, and the end result is shown in Fig. 2.11. Three more sampling points are added, and the function and kriging fit are indistinguishable in the plot.

2.6.2 Error of Function Compared to Truth Model

Once the kriging surface has been constructed in such a way that it matches up well with the intended function, it is necessary to evaluate how well the function itself represents the truth model. Fig. 2.12 shows the overall process that has been implemented. As before, simplified models are utilized due to low computational

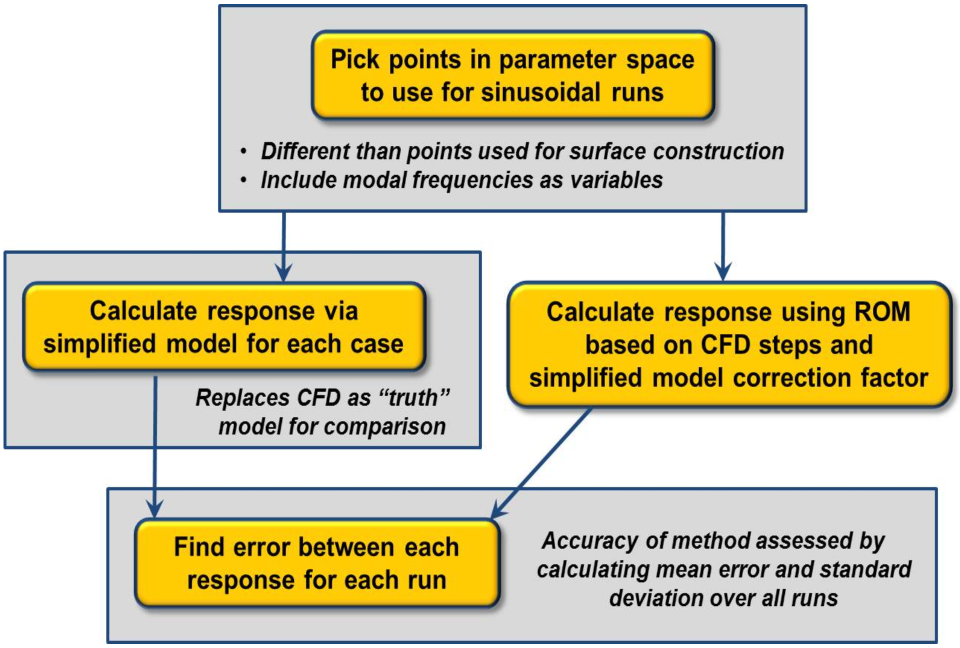

Pick points in parameter space to use for sinusoidal runs
• Different than points used for surface construction
• Include modal frequencies as variables
Calculate response via simplified model for each case
Replaces CFD as "truth" model for comparison
Calculate response using ROM based on CFD steps and simplified model correction factor
Find error between each response for each run
Accuracy of method assessed by calculating mean error and standard deviation over all runs

Figure 2.12: Error analysis: comparison of function to truth model

expense.

The overall error is investigated by comparing ROM and truth model results over a large sample of test cases throughout the parameter space. To begin with, Latin hypercube sampling is used to pick points at which to run sinusoidal test cases. For these cases, modal oscillation frequencies, though not included in the ROM construction parameter space, are included as variables here in order to investigate the ROM's accuracy as oscillation frequencies increase. These test points are in general different than the points used for kriging surface construction. At each test point, the sinusoidal input response is calculated in two different ways: once using the ROM based on a CFD step response and correction factor calculated using a simplified model and once using only the simplified model for the entire response calculation, without any ROM methodology employed. The straight simplified model result here replaces the CFD model as the "truth" model for comparison. Finally, the error of ROM as compared to simplified model results is found for each run. The ROM methodology's accuracy is assessed by calculating the mean and standard deviation of the errors over all runs.

The goal of this methodology is not to give an exact error that is expected to be incurred but rather a general picture of the error. Because the low-order simplified models are used here, the end result of this error methodology would not be a statement of some specific maximum error value that would be seen over the entire parameter space. Rather, the result would be estimates of errors throughout various areas of the parameter space; this is due to the simplified models' inherent approximations and potential to break down as model complexity is increased or certain flight conditions are changed.

2.7 ROM Application

The next chapter highlights the results and unique issues faced by the implementation of the ROM in the hypersonic flight regime; subsequent chapters detail the ROM's applicability to the transonic and subsonic regimes. The error assessment techniques described previously are employed to evaluate the ROM's accuracy over the resulting wide range of flight conditions.

Chapter 3

Applicability to Hypersonic Regime

This chapter presents the application of the reduced-order modeling methodology to the hypersonic regime. In doing so, the main geometry considered is a two-dimensional half-diamond airfoil model. Several different types of tests are conducted on this geometry by giving sinusoidal inputs to one or more of the mode shapes. First, single-modal oscillation tests are performed to evaluate the ROM as one particular variable, either oscillation amplitude or oscillation frequency, is increased. These tests are repeated with the linear step responses found at varying Mach numbers to assess how the ROM performs at conditions away from those at which it is constructed. Then, simulations with multiple modes of oscillation are conducted, and the errors over the parameter space being considered are characterized. Finally, the error assessment methodology from Chapter 2 is applied to the hypersonic regime, and the approximation of using the piston theory simplified aerodynamic model is evaluated.

3.1 Hypersonic Problem Setup

Flight in the hypersonic regime is characterized by strong nonlinear shocks and resulting large forces and moments on hypersonic vehicles. These vehicles themselves are highly coupled systems, with various components affecting each other that would not be expected to do so on a subsonic or transonic vehicle. For example, consider a

Figure 3.1: NASA X-43 (image from NASA.gov)

hypersonic vehicle with an underslung scramjet engine, such as the rendition of the NASA X-43 shown in Fig. 3.1. Rather than compressor fan blades, a scramjet relies on the incoming shock wave from the front of the vehicle for air compression into the engine. Thus, any change in the shock (due to angle of attack change, elastic modal oscillation, etc.) will directly affect the pressure of the air into the engine and hence the thrust produced by the engine. This changing thrust, due to the underslung nature of the engine, will in turn affect the pitching moment on the vehicle, which affects the position of the front of the vehicle, the shock, and so on. Due to these tight coupling interactions, it is vital to accurately predict the aerodynamic loads when simulating the vehicle's flight and conducting vehicle control evaluation. Inaccurate control algorithms, potentially resulting from inaccurate aerodynamic loads, may result in loss of control of the vehicle.

The basic geometry on which tests are conducted is a two-dimensional, 2.5% thick half-diamond airfoil with a flat top surface and length of 1.6 meters, which is not intended to be representative of any specific airfoil or vehicle configuration. This particular configuration is chosen due to its relative geometric simplicity and asymmetric nature. Geometric simplicity is desired for relatively efficient CFD computations, and asymmetry is desired in order to obtain different magnitudes for positive and negative step responses and hence not look at a specialized case of a symmetric airfoil. The CFD grid, shown in Fig. 3.2 (zoomed in on the airfoil) is a 548 × 674 structured grid with points concentrated more closely near the airfoil surface and was constructed

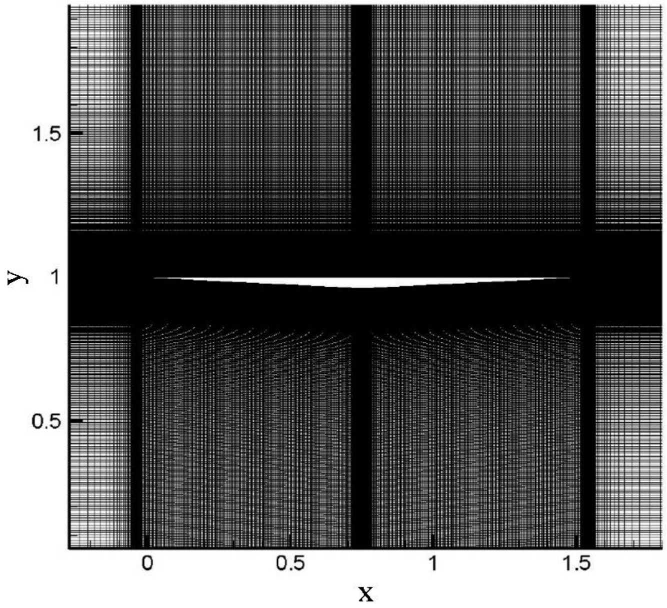


Figure 3.2: 2-D half-diamond airfoil CFD grid

using the mesh generator ICEM CFD from ANSYS.[95] The first mode step response obtained is virtually indistinguishable to that from a more refined grid of 644 × 866 points. All CFD solutions obtained are Euler solutions.

In general, some fundamental deformation modes of the elastic structure must be used when creating the unsteady aerodynamic ROM. Typically, those fundamental modes are elastic mode shapes of the structure, and they would come from the solution of the structural dynamics part of the problem. To simulate those in this research, three chordwise mode shapes are assumed. Like the geometry itself, the mode shapes assumed here do not correspond to any specific configuration. Figure 3.3 shows a plot of the centerline displacements of these mode shapes; the amplitudes shown correspond to those used for the step inputs.

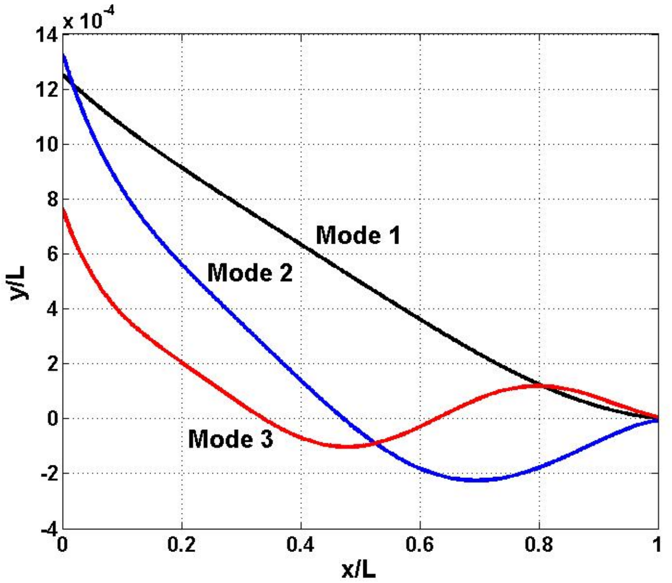


Figure 3.3: Mode shapes

3.2 ROM Assessment

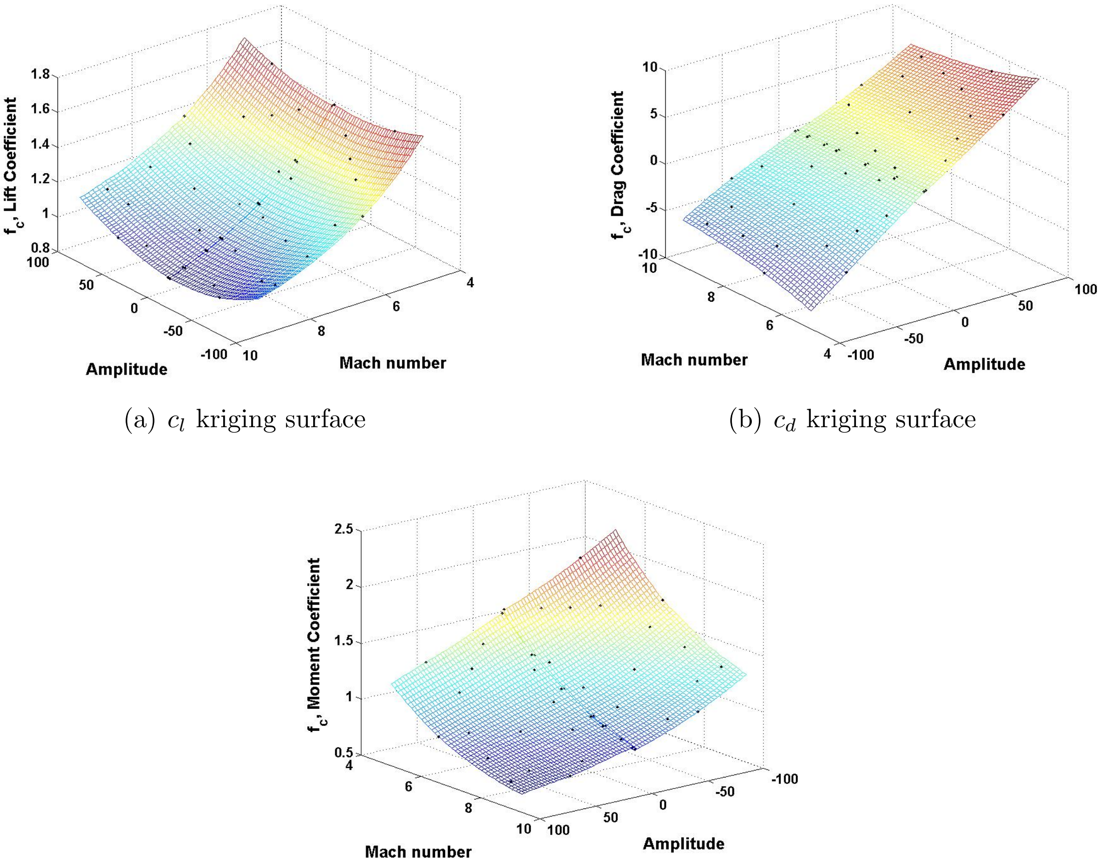
In order to test the ROM methodology in the hypersonic regime, solutions are obtained from several different sets of CFD simulation test cases. The first set consists of simulations with only a single mode of oscillation, which can be further broken down into subsets of amplitude tests and frequency tests. For the amplitude tests, sinusoidal inputs of varying amplitude are given to one structural mode shape while the frequency of oscillation, Mach number, and all other variables remain constant. These tests give insight into the improvement of ROM results due to the application of the correction factor by comparing corrected and uncorrected ROM results at increasing oscillation amplitudes. For the frequency tests, sinusoidal inputs of varying oscillation frequency are given to a single structural mode shape while the oscillation amplitude, Mach number, and other parameters remain constant. Investigations into the ROM's errors with increasing frequency are presented. Additionally, the choice of step response Mach number, M_{step}, to use for ROM construction is considered. Namely, if the step response used to calculate the uncorrected ROM quantity y_{conv} is changed from one found at Mach 8 to, for example, Mach 5, how will that affect the ROM-CFD agreement? Results are given showing ROM errors using $M_{step} = M_{sim}$ (Mach number of the simulation) as well as $M_{step} \neq M_{sim}$.

The next set of results are those from simulations with multiple-modal oscillations, in which sinusoidal inputs are given to each of the structural mode shapes being considered in the problem. The error assessment methodology described in Chapter 2 is applied to the half-diamond airfoil geometry to give an indication of the overall error as well as the number of parameter space sampling points necessary for ROM construction. Then, the correction factor values at the sampling points obtained through the error estimation methodology are calculated using CFD, and the corresponding ROM is compared with full-order CFD simulations.

3.2.1 2-D Half-Diamond Airfoil Single Mode Results

The first portion of ROM testing on the 2-D half-diamond airfoil consists of constructing the ROM for single-modal oscillations of each of the first and third mode shapes. Two separate mode shapes are considered here in order to see if various result trends hold for more than just the one specific mode shape currently under consideration. For these single mode cases, a total of 17 initial sampling points within the parameter space, consisting in each case of the single-modal amplitude as well as Mach number, are selected based on a spreadsheet from Sanchez[a], which finds the optimal nearly-orthogonal Latin hypercube sampling points for a given parameter space. Based on the number of input variables in the parameter space, the number of output sampling points is pre-determined, and the algorithm computes the input variable values at each of these sampling points such that each set of input values is as close to orthogonal with the others as possible. Then, after the sampling point data are collected, kriging surfaces for the lift, drag, and moment coefficient correc-

[a] Sanchez, S. M., "NOLH designs spreadsheet," 2005. Available online via http://diana.cs.nps.navy.mil/SeedLab, Last accessed 06/22/2010.



(a) c_l kriging surface

(b) c_d kriging surface

(c) c_m kriging surface

Figure 3.4: Mode 1 kriging surfaces (δ=0)

tion factors are calculated and shown in Fig. 3.4. Note that separate surfaces are calculated for positive and negative modal amplitudes and that additional sampling points are added near the zero-amplitude boundaries of the two surfaces in order to improve the matching of the two surfaces at this boundary. In a purely qualitative sense, these surfaces are smooth and lack large, sharp undulations, suggesting that the correction factor function is relatively smooth over the parameter space. This is desirable since a smooth function will in general require fewer sampling points than one with many undulations. For the multi-modal ROMs considered in this chapter, an offset δ value of 100 is used.

Table 3.1: **Amplitude test parameters**

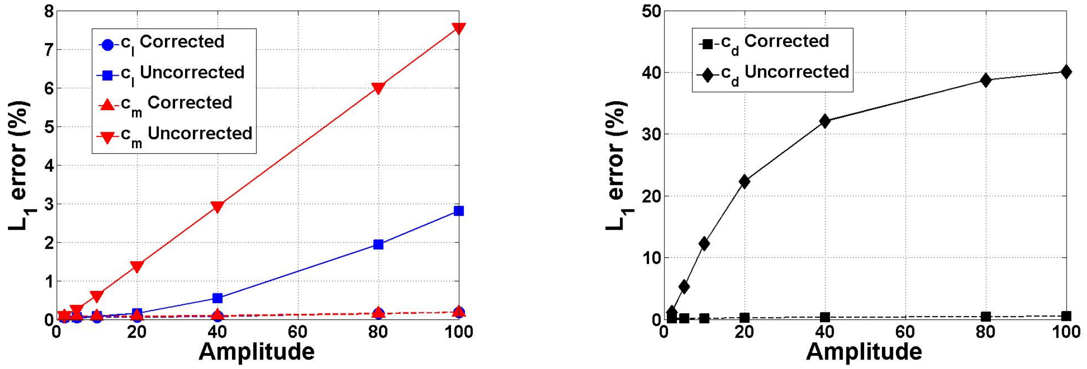
Mode	Mach number	Frequency	Max. amplitude	M_{step}	k
1	8	125.7 rad/s	100	8	0.04
3	8	251.4 rad/s	40	8	0.08

Single Mode Amplitude Tests

Table 3.1 shows the various parameters used in the amplitude tests for this half-diamond geometry, one set of tests each for the first and third mode shapes. For these tests, the Mach number and oscillation frequency remain constant over each run. Note that, for all cases presented in this section, $M_{step} = M_{sim}$ and that M_{step} is used in the calculation of both y_{conv} as well as y_{lin}. Figure 3.5 shows the errors for both the corrected ROM (y_{corr}) and uncorrected ROM (y_{conv}) for the first mode over a range of amplitudes from 1 to 100. For each of the three coefficients, the errors for corrected ROM remain small, on the order of 1%, while the uncorrected ROM errors continually increase. The largest increase is seen for the drag coefficient (Fig. 3.5(b)), which has errors of around 40% at an amplitude of 100. Figure 3.6 shows qualitative comparisons between the corrected ROM, uncorrected ROM, and CFD results for the lift and drag coefficients for the case with amplitude 40 from Fig. 3.5. As can be seen in the plots, the uncorrected ROM mispredicts the amplitude of the moment coefficient response and very badly misses the peaks of the drag response, while the corrected ROM results are virtually indistinguishable from the CFD results. The same general trends hold for the third mode, as shown in Fig. 3.7.

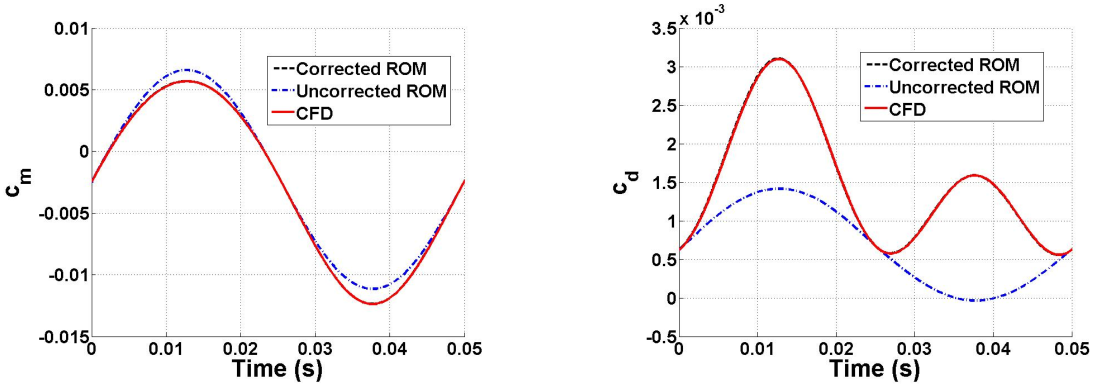
Single Mode Frequency Tests

The next set of tests, the parameters for which are shown in Table 3.2, consists of cases with constant Mach number and oscillation amplitude but varying oscillation frequency. As before, the step Mach number is equal to the simulation Mach num-



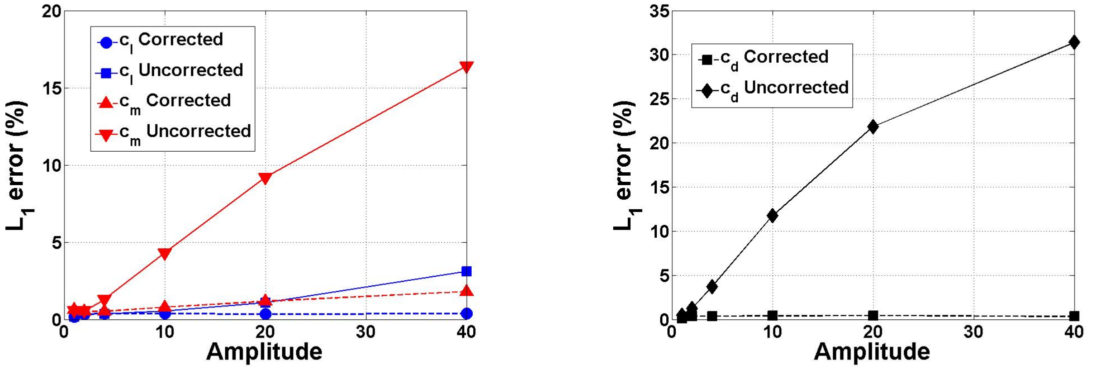
(a) c_l and c_m errors (b) c_d error

Figure 3.5: ROM errors, mode 1 amplitude tests, 2-D half-diamond airfoil



(a) c_m comparison (b) c_d comparison

Figure 3.6: ROM-CFD comparisons, mode 1, amplitude 40



(a) c_l and c_m errors (b) c_d error

Figure 3.7: ROM errors, mode 3 amplitude tests, 2-D half-diamond airfoil

Table 3.2: **Frequency test parameters**

Mode	Mach number	Amplitude	Max. Reduced Frequency	M_{step}
1	8	40	0.7	8
3	8	10	0.7	8

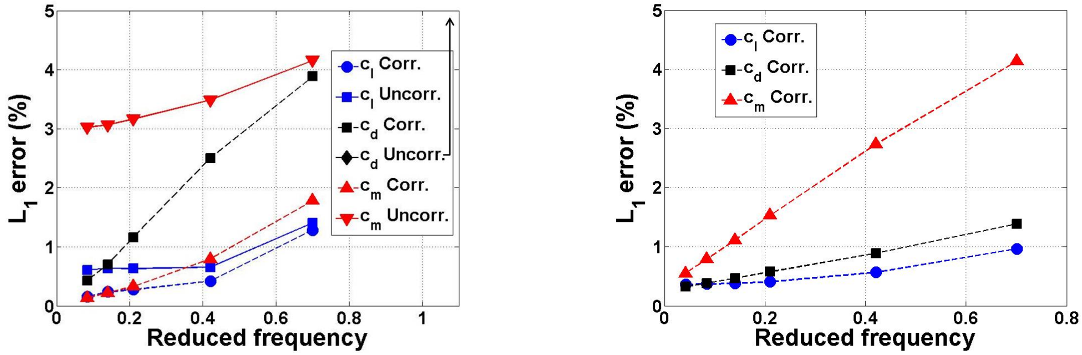
ber. Given the steady nature of the correction factor calculation, it is necessary to characterize the ROM's errors as frequency, and hence unsteadiness, increases. For the test cases considered here, the frequency is plotted as the reduced frequency k, defined as,

$$k = \frac{b\omega}{U_\infty} \tag{3.1}$$

where b is the airfoil half-chord, ω is the oscillation frequency in rad/s, and U_∞ is the freestream velocity. Larger reduced frequency values correspond to a larger degree of unsteadiness in the flow. Due to the inherently large values of U_∞, hypersonic flow tends to be characterized by relatively low k values. The highest reduced frequency values in these tests are just around 0.7, which correspond to a dimensional oscillation frequency of just under 2100 rad/s.

The lift, drag, and moment coefficient results for each of modes 1 and 3 are shown in Fig. 3.8. In general, the errors do increase with increased oscillation frequency, though they remain relatively small over the range tested, under 5% for all data points.

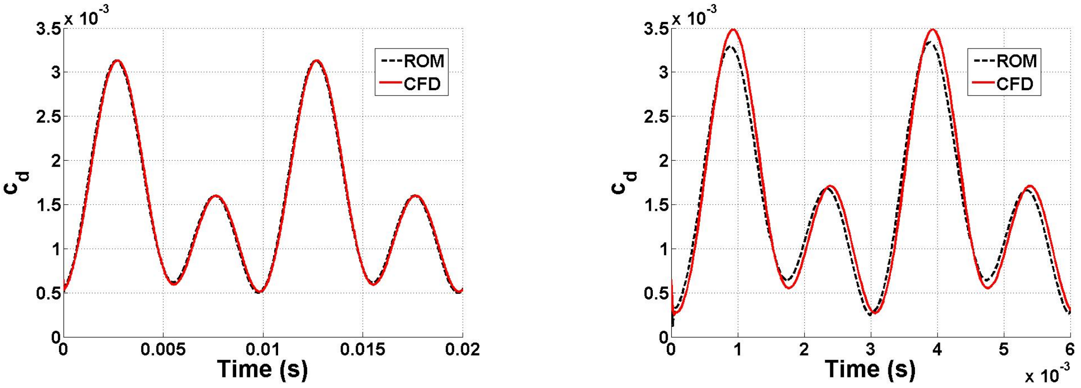
To investigate how this increasing error is being manifested in the ROM-CFD comparisons, consider Fig. 3.9, which shows the direct ROM-CFD drag coefficient comparisons showing two cycles of two cases with first mode oscillation reduced frequencies of 0.21 ($\omega = 628$ rad/s) and 0.70 ($\omega = 2094$ rad/s). In Fig. 3.9(a), the agreement is qualitatively very good. However, for the increased oscillation frequency of Fig. 3.9(b), two features are noticed. First, a slight response amplitude discrepancy



(a) Mode 1

(b) Mode 3

Figure 3.8: ROM errors, frequency tests, 2-D half-diamond airfoil



(a) k=0.21

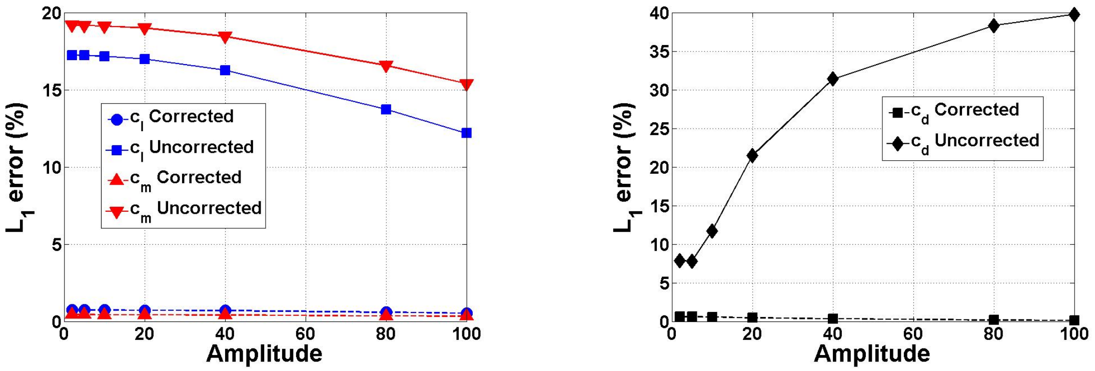
(b) k=0.70

Figure 3.9: ROM-CFD comparisons with increasing oscillation frequency, 2-D half-diamond airfoil

is seen, especially at the largest response peak. Second, a phase shift has developed between the two responses. These features are likely due to the increased unsteadiness of the flow inherent at higher oscillation frequencies.

Variation of M_{step}

For each of the preceding sets of results, the Mach number of the simulations has been equal to the Mach number of the step response upon which the ROM is based. However, since the ROM is desired to be valid over a range of Mach numbers, it is



(a) c_l and c_m errors

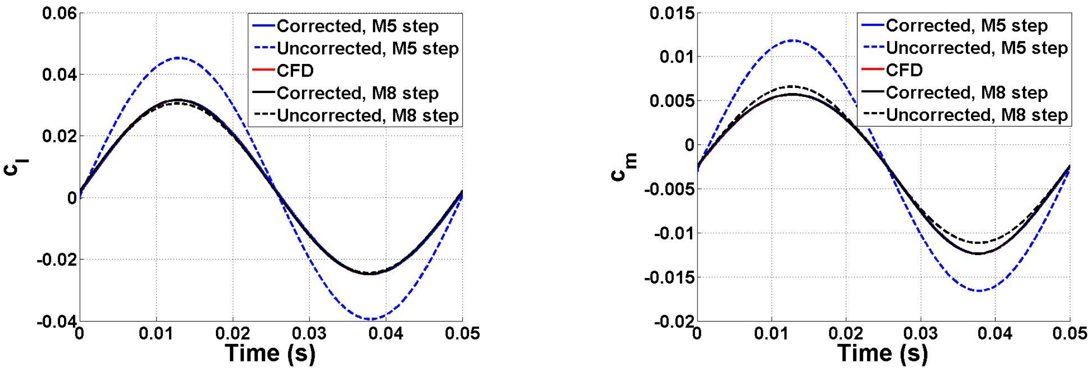
(b) c_d error

Figure 3.10: ROM errors, mode 1 amplitude tests, step response at Mach 5

necessary to investigate how the errors change as M_{step} moves away from the Mach number of the specific simulation under consideration. First, consider again Fig. 3.5, which shows errors over a range of oscillation amplitudes. Now, for these same simulations, Fig. 3.10 shows the error results for a ROM constructed with $M_{step} = 5$ rather than the Mach 8 simulation Mach number. As can be seen, the lift and moment errors for the uncorrected ROM are significantly higher than those found with $M_{step} = 8$; all uncorrected ROM errors are over 10%. The uncorrected drag errors, already high for the Mach 8 step response ROM, are slightly higher as well. However, for the corrected ROMs, the errors remain small in the same manner as the Mach 8 step response ROM.

To visualize what is causing these higher errors, consider Fig. 3.11, which shows the lift and moment coefficient results for an amplitude of 40, the same simulation parameters as used in Fig. 3.6. For the uncorrected ROM based on $M_{step} = 5$, a relatively large amplitude discrepancy can be seen, which is the major error source for that case. Note that the CFD and corrected ROM results (both $M_{step} = 5$ and $M_{step} = 8$) are virtually indistinguishable in the plots.

Next, consider the frequency tests shown in the preceding section. Figure 3.12 shows the results from the same simulations as in Fig. 3.8(a) but for ROMs con-



(a) c_l comparison

(b) c_m comparison

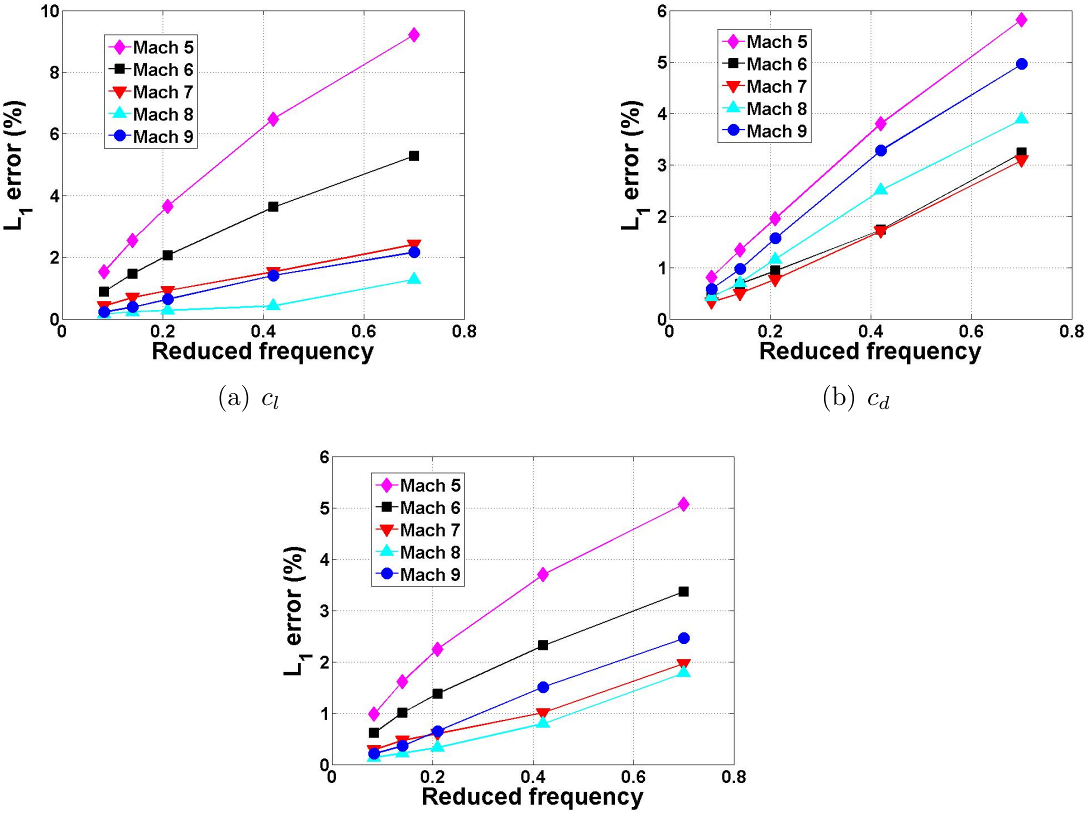
Figure 3.11: ROM comparisons, mode 1, amplitude 40

structed with varying M_{step}. For the lift and moment coefficients, the ROM with $M_{step} = M_{sim} = 8$ has the least amount of error for each of the frequencies tested. Also for these cases, the errors increase as the step moves further away from the simulation Mach number, with the Mach 5 step response ROM having the highest amount of error. However, for the drag coefficient, this pattern does not hold, as the $M_{step} = 7$ error is the lowest, followed by the $M_{step} = 6$ error; the $M_{step} = 8$ error is slightly higher than those two. For all of these cases, except for $M_{step} = 5$ and the highest frequency for the lift coefficient with $M_{step} = 6$, the errors remain under 5%.

These results show that, while the ROM errors do increase as M_{step} moves away from M_{sim}, they do remain relatively low throughout a range of Mach numbers. For example, for the lift coefficient, when $M_{step} = 6$ is used to model a simulation at Mach 8, the maximum change in L_1 error observed over a range of frequencies is around 4 percentage points when compared to a ROM with $M_{step} = 8$.

3.2.2 Multi-Modal Oscillation Results

Next, the ROM methodology is applied to a situation considering the oscillations of the first three elastic mode shapes. The first step in the process is to select the sampling points to be used in the ROM correction factor kriging surface construction.



(a) c_l

(b) c_d

(c) c_m

Figure 3.12: ROM errors, frequency tests, varying M_{step}

To do so, the error estimation method presented in Chapter 2 is utilized, the first item for which is to select an appropriate simplified model to assist in the calculations. For this purpose, piston theory[6,96] has been chosen. Piston theory is a simplified method for calculating unsteady pressures on a supersonic body which uses the approximation that a planar slab of fluid initially perpendicular to the flow direction will remain that way as it passes over a body. The normal velocity of the body surface may cause the slab to expand or compress as it travels down the surface, resulting in a changing pressure. Using the piston analogy, the pressure $p(x,t)$ on a point of the surface can be found by:[29]

$$p\left(x,t\right) = p_\infty \left(1 + \frac{\gamma - 1}{2} \frac{v_n}{a_\infty}\right)^{\frac{2\gamma}{\gamma-1}} \tag{3.2}$$

In Eq. 3.2, p_∞ is the freestream pressure, γ is the ratio of specific heats, v_n is the velocity of the surface normal to the flow direction, and a_∞ is the freestream speed of sound. Taking a third-order binomial expansion of the above expression, the third-order piston theory pressure at a certain location on the surface of the body is found as follows:

$$p\left(x,t\right) = p_\infty + \gamma p_\infty \left[\frac{v_n}{a_\infty} + \frac{\gamma+1}{4}\left(\frac{v_n}{a_\infty}\right)^2 + \frac{\gamma+1}{12}\left(\frac{v_n}{a_\infty}\right)^3\right] \tag{3.3}$$

Note that piston theory breaks down when the normal velocity of the surface approaches the speed of sound as well as in areas where curvature introduces three-dimensional effects, as in a flow moving down a cylindrical body.

With this simplified model in place, the sampling point determination process is shown in Fig. 3.13, which illustrates the specific application of the process presented in Fig. 2.9 in Chapter 2. In this case, the stopping criterion for the addition of sampling points is defined in terms of the ratio $r_{s\sigma}$ of kriging surface's variance σ to the mean squared error s^2 and is written as:

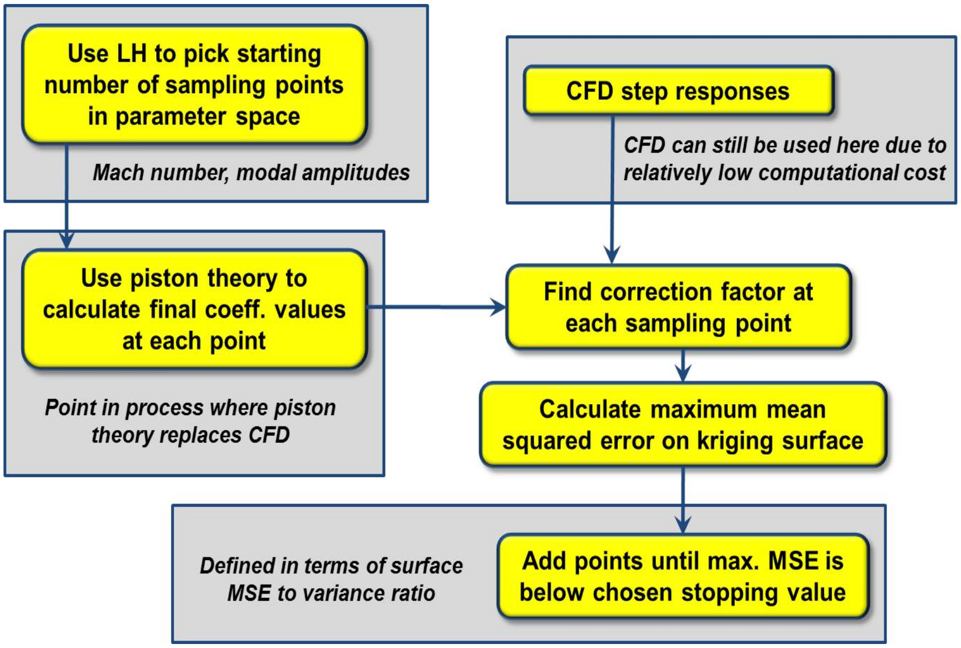

Use LH to pick starting number of sampling points in parameter space
Mach number, modal amplitudes
CFD step responses
CFD can still be used here due to relatively low computational cost
Use piston theory to calculate final coeff. values at each point
Point in process where piston theory replaces CFD
Find correction factor at each sampling point
Calculate maximum mean squared error on kriging surface
Defined in terms of surface MSE to variance ratio
Add points until max. MSE is below chosen stopping value

Figure 3.13: Hypersonic regime sampling point determination process

$$r_{s\sigma} = \frac{s^2}{\sigma} < 0.01 \tag{3.4}$$

The choice of the stopping value of 0.01 is relatively arbitrary in this case, but it is quantitative and can be user-defined for a particular case depending on the constraints of a specific problem. Because the `lhsdesign` command in MATLAB utilizes randomly generated points within the Latin hypercube design space (here consisting of three modes and Mach number), slightly different values for the number of points are found each time. To investigate this variation, the sampling point selection methodology is repeated 100 times; over these tests, the mean number of sampling points to reach the stopping criterion is 98 with a standard deviation of 13. For testing purposes on the 2-D half-diamond airfoil, a set of 88 sampling points meeting this criterion is selected for kriging surface construction.

Now that the kriging surface sampling points have been determined, ROM results are generated and compared with full-order CFD simulations. These test cases consist

Table 3.3: **Parameter ranges for half-diamond airfoil test cases**

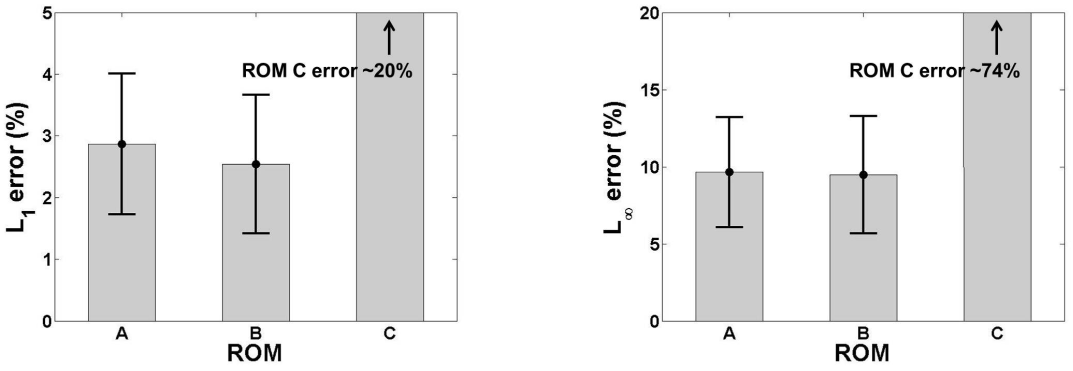
Parameter	**Min**	**Max**
M	5	9
d_1	-60	60
d_2	-30	30
d_3	-25	25
ω_{1-3} (rad/s)	100	$1,000$
k_{1-3}	0.03	0.54

Table 3.4: **Half-diamond airfoil ROM variations**

ROM	N_{samp}	f_c **calc.**	**Comment**
A	73	CFD/kriging	Sampling points placed at location of max. MSE
B	88	Piston/kriging	Samp. points mostly at same locations as ROM A
C	N/A	N/A	Linear ROM y_{conv}

of a set of 45 runs with sinusoidal oscillations given to each of the three mode shapes under consideration. The parameter ranges for these runs are shown in Table 3.3. The CFD results are compared to three separate ROMs, which are described in Table 3.4; note that, in the table, N_{samp} denotes the number of correction factor sampling points used for the particular ROM. For ROM A, the 73 sampling points are calculated using CFD simulations, while the sampling points in ROM B are computed using piston theory. ROM C is the linear, uncorrected ROM response y_{conv}. Note that ROM A consists of fewer sampling points than ROM B due to the fact that, for some of the sampling points, the CFD code ran into numerical issues caused by relatively large modal deformations. Also, all step responses here are computed at Mach 8.

Figure 3.14 shows the mean and standard deviations over all 45 test cases for both the L_1 and L_∞ drag coefficient errors. The first noticeable feature of the plots is the very large errors for the uncorrected ROM C. This shows that the drag coefficient results are nonlinear, and superposition cannot be used in this situation. The mean errors for both ROMs A and B are much smaller, under 3% for the L_1 error, showing



(a) L_1

(b) L_∞

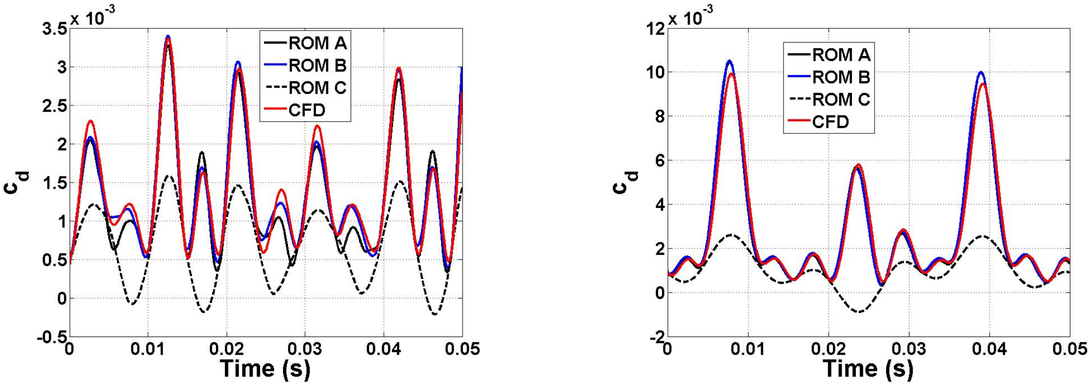
Figure 3.14: c_d error results over 45 test cases

Table 3.5: **Test case parameters and errors, drag**

Test	M	d_1	d_2	d_3	ω_1	ω_2	ω_3	ROM A	ROM B	ROM C
						(rad/s)			c_d errors: L_1/L_∞	
1	8.75	36.3	11.5	-24.7	648.0	854.9	850.2	5.52/18.4	3.52/11.4	24.5/64.4
2	6.71	-41.8	19.7	-2.00	602.2	206.0	822.4	2.86/11.3	2.96/12.7	18.6/77.7

that both of these ROMs agree well with the CFD data. One unexpected result is that, for each of the two error norms plotted, the piston theory correction factor ROM results (ROM B) show slightly lesser error than the CFD correction factor ROM results (ROM A). This could be caused by the fact that the piston theory correction factor kriging surface contains 15 more points than the CFD correction factor kriging surface due to the CFD code numerical issues mentioned above, though it is important to note that the overall errors for both of the ROMs in this case are small. For a more qualitative comparison, Fig. 3.15 shows the ROM-CFD comparisons for the run with maximum L_1 error (Fig. 3.15(a)) as well as a run with an L_1 close to the mean value (Fig. 3.15(b)). The parameters for these two specific runs are found in Table 3.5.

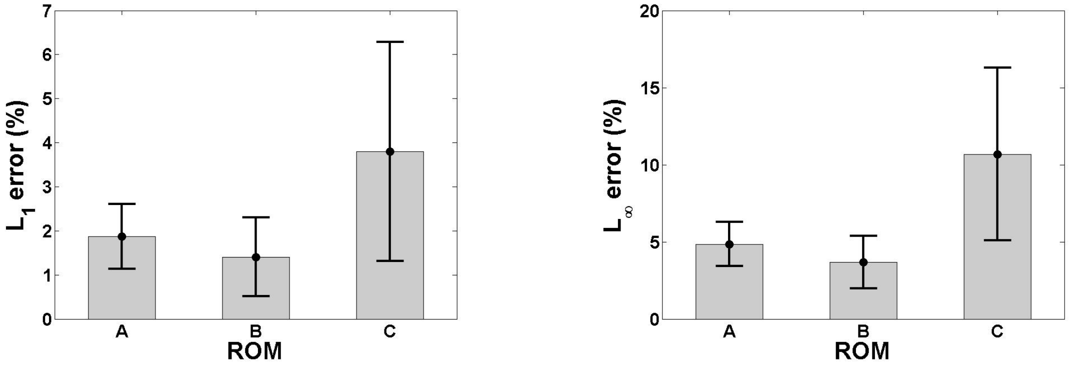
For the lift coefficient, the mean and standard deviations over all test cases for each of the two error norms are shown in Fig. 3.16. The first item to note is that, while the linear ROM still has the largest errors of all ROMs, the errors here for



(a) Test 1: max L_1 error

(b) Test 2: mean L_1 error

Figure 3.15: ROM-CFD sample comparisons, c_d



(a) L_1

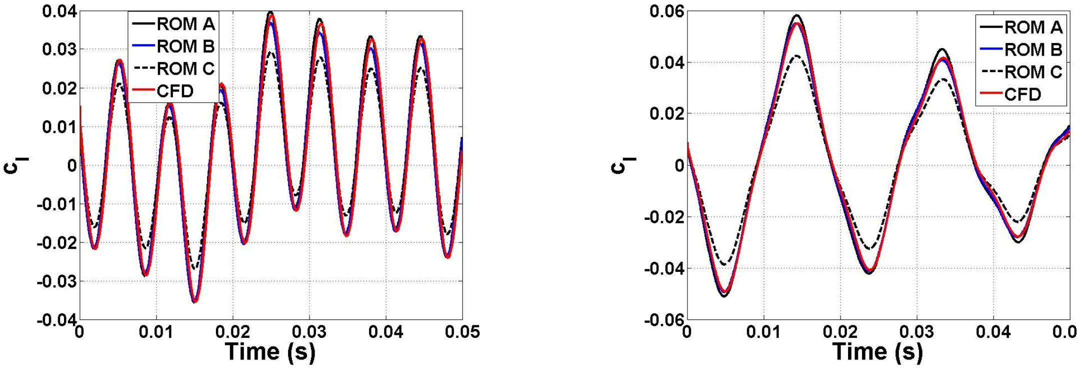
(b) L_∞

Figure 3.16: c_l error results over 45 test cases

ROM C are much smaller than those for the drag coefficient. Overall, each of the correction factor ROMs matches well with the CFD results, as demonstrated by the mean errors of under 2% for both ROMs A and B. For a qualitative comparison, consider Fig. 3.17, which shows the specific test with the highest L_1 error as well as a test with an error close to the mean value. The parameters for these test cases are displayed in Table 3.6.

Table 3.6: **Test case parameters and errors, lift**

Test	M	d_1	d_2	d_3	ω_1	ω_2	ω_3	ROM A	ROM B	ROM C
						(rad/s)			c_l errors: L_1/L_∞	
3	6.10	-26.8	4.81	-14.9	956.3	313.0	143.8	3.57/7.21	3.44/7.19	5.72/13.1
4	6.38	-39.0	-13.1	7.32	324.0	374.0	973.3	1.83/3.67	1.26/2.55	4.41/12.2



(a) Test 3: max L_1 error

(b) Test 4: mean L_1 error

Figure 3.17: ROM-CFD sample comparisons, c_l

3.3 Conclusion

The following are the relevant conclusions which can be deduced from this chapter:

- For the test cases with a single mode of oscillation, the corrected ROM works well to model the unsteady aerodynamics. The errors remain small as oscillation amplitude increases, and while the errors increase along with oscillation frequency, they remained relatively small over the parameter range considered. When compared to the linear ROM results, the addition of the correction factor shows the greatest overall improvement for the drag coefficient values.

- The ROM shows good agreement with the CFD results for cases where M_{step} differed from the Mach number of the simulation, demonstrating that the ROM is applicable to a wider range of parameters than one single set around which it is constructed.

- For the multi-modal oscillation test cases, the corrected ROM results again match well with the full-order CFD solutions, with mean drag coefficient L_1 errors of under 3% for the test cases considered. Lift results also show good agreement with the full-order solutions.

Chapter 4

Applicability to Transonic Regime

This chapter presents the application of the reduced-order modeling methodology to Mach numbers close to 1. The basic geometric model used for testing here is the AGARD 445.6 wing, which has been used previously in aeroelastic studies. The effects of sinusoidal input amplitude, oscillation frequency, and step response Mach number are all investigated by first considering simulations with only a single mode of oscillation. Then, multi-modal test cases are conducted, and the results over the parameter space are assessed. In order to both apply the error assessment methodology as well as help choose the sampling points to use for the correction factor kriging surfaces, the simplified Method of Segments model has been developed. Results comparing full-order CFD simulations with Method of Segments and CFD-based ROMs are displayed.

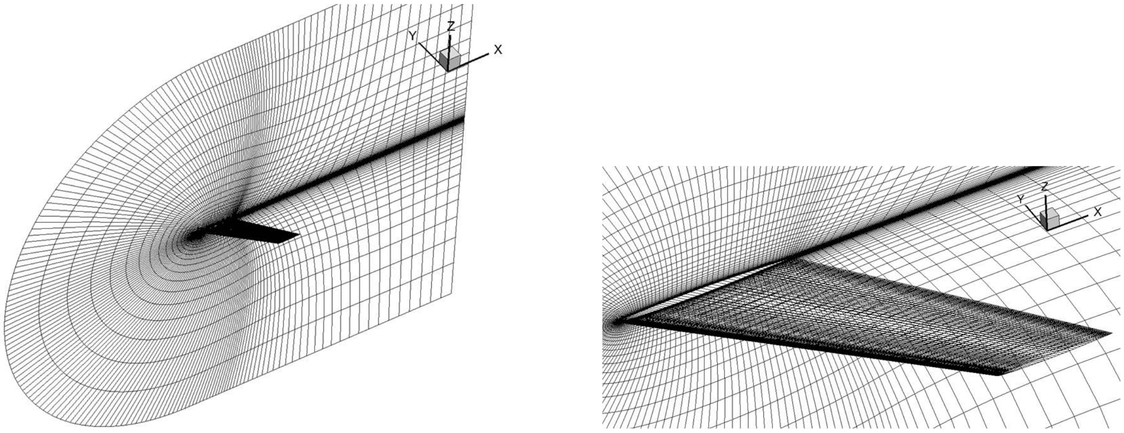
4.1 Transonic Problem Setup

The transonic flow regime is characterized by the appearance and motion of shock waves as portions of the flow reach Mach 1. These highly nonlinear flow fields create challenges when modeling the aerodynamic loads in this regime. Accurate modeling here is important due to the various aeroelastic and aerodynamic phenomena encountered, including the flutter transonic dip as well as a significant increase in the drag as flight speed nears Mach 1.

4.1.1 AGARD 445.6 Wing

The geometry upon which these test cases are performed is the AGARD 445.6 wing,[97] which has been used in wind tunnel aeroelastic tests as well as for computational aeroelastic studies.[65,66,98–100] A structured CFD grid has been obtained from NASA Langley and has dimensions of $65 \times 193 \times 41$, with the i direction being along the span, j direction along the chord, and k direction normal to the wing surface. Figure 4.1 shows the grid as well as a zoomed-in figure of the wing itself. Oscillations of the first three elastic mode shapes of the wing are considered. These mode shapes, shown in Fig. 4.2, are the same that have been used in other studies as well.[91,101] Note that, for each mode shape, the unit step input corresponds to a maximum wing deflection of just around 0.1% of the span (3.5% of the root thickness).

The AGARD 445.6 wing has a very thin cross-sectional geometry, with a maximum root thickness of about 4%. Because of this, the onset of transonic effects, signaled by the presence of mixed sub- and supersonic flow, will be delayed until very close to Mach 1 in comparison with other, thicker airfoils. Evidence of mixed flow can be found by looking at the pressure contours over the wing, as the transonic shock waves will cause pressure rises along the chord of the wing. Figure 4.3 shows the nondimensional pressure and Mach number contours along the top half of the wing in steady flow at Mach 0.9889 and zero angle of attack; note that the pressure values of Fig. 4.3(a) are nondimensionalized in such a way that $p_\infty = \frac{1}{\gamma}$, where γ is the ratio of specific heats and equals 1.4. The presence of a shock on the airfoil can be seen from the sharp pressure increase in the chordwise direction along all spanwise stations of the wing, along with the corresponding decrease from supersonic to subsonic Mach numbers for this inviscid solution. In this chapter, results are obtained from simulations both from within this relatively narrow band of transonic effects, which are first seen typically



(a) AGARD grid

(b) AGARD wing close-up

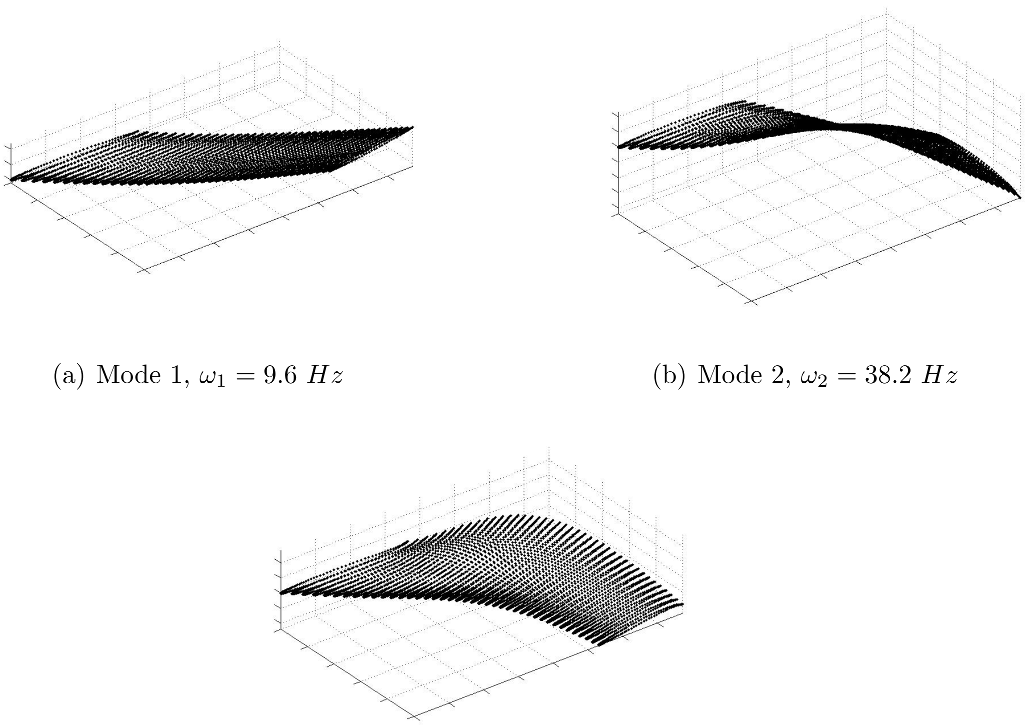
Figure 4.1: AGARD 445.6 wing

for as low of Mach numbers as around 0.98, and from a wider range of values from Mach 0.9-1.1. These simulations provide a broader picture of the aerodynamic loads expected to be encountered by the wing as it approaches Mach 1.

4.1.2 ROM Assessment

The CFD test cases used for ROM assessment near Mach 1 can be divided into sets similar to those used for the hypersonic test cases of Chapter 3. The first set consists of single mode of oscillation test cases and is subdivided into amplitude and frequency tests, in which all parameters except for amplitude and frequency, respectively, are held constant. A sense of how the ROM errors vary with increasing values of these two parameters is presented.

Next, the ROM is extended to multiple modes of oscillation within this same Mach regime. In order to construct the ROM, the error estimation methodology presented in Chapter 2 is applied here. The Method of Segments simplified model has been developed for this purpose and is detailed in a subsequent section. Correction factor kriging surface sampling point coefficient values are obtained both through direct CFD computation as well as the Method of Segments and compared to one another

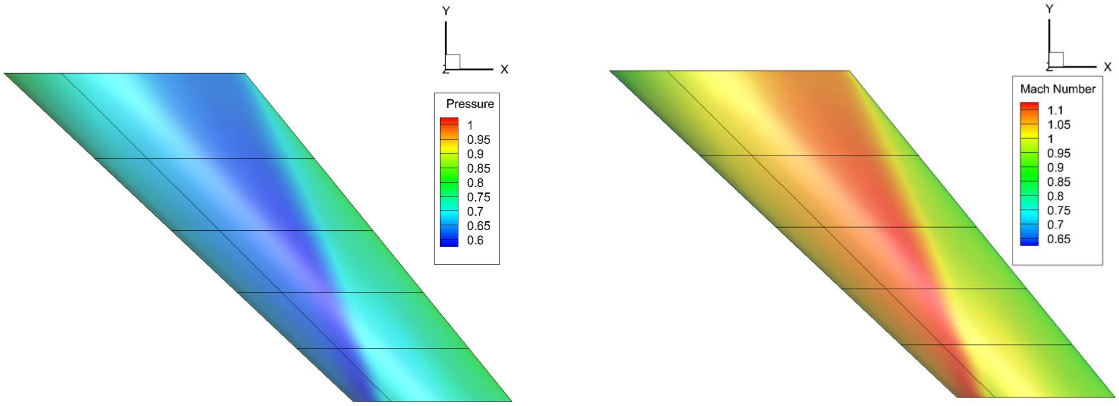


(a) Mode 1, $\omega_1 = 9.6\ Hz$

(b) Mode 2, $\omega_2 = 38.2\ Hz$

(c) Mode 3, $\omega_3 = 48.2\ Hz$

Figure 4.2: AGARD 445.6 wing mode shapes



(a) Pressure

(b) Mach number

Figure 4.3: Contours on AGARD 445.6 wing, Mach 0.9889, α=0

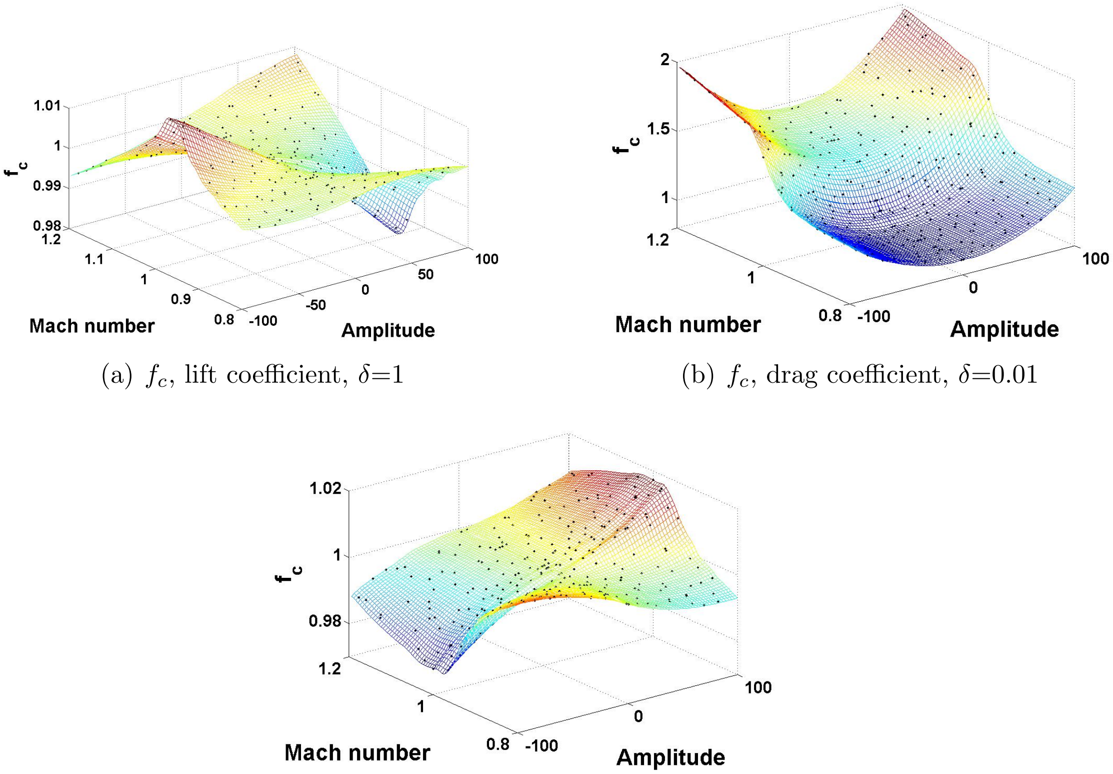
Table 4.1: **Single modal oscillation parameter values**

Parameter	**Min**	**Max**
M	0.8	1.2
Amplitude	-100	100

in order to determine the accuracy of the simplified model and thus its applicability to the problem. Finally, full-order CFD solutions are compared to ROMs constructed with both Method of Segments and CFD-calculated correction factor values. All CFD solutions presented here are Euler solutions.

4.2 Single Mode Tests

The first set of investigations consist of characterizing the applicability of the ROM methodology to oscillations of the first mode only. The first items to consider are the ranges of parameters, including Mach number, oscillation amplitude, and oscillation frequency, to use during testing; these ranges are selected to be those shown in Table 4.1. The corresponding kriging surfaces, shown in Fig. 4.4, are constructed for the correction factors corresponding to each of the lift, drag, and moment coefficients. Due to the relatively large gradients around Mach 1, separate surfaces are constructed for sub- and supersonic Mach numbers. Also, since the AGARD 445.6 airfoil is symmetric, negative amplitude lift and drag coefficient values are simply the opposite of and the the same as, respectively, those found for the corresponding positive amplitude, thus reducing the total number of CFD runs required. A total of 81 sampling points, found using Latin hypercube sampling of the parameter space, is used for the subsonic kriging surface, while 63 points are used for the supersonic surface. An offset value (δ) of 10^6 is used. However, note that, in the plots of Fig. 4.4, smaller δ values are used as indicated; this is done to emphasize the difference of the correction factor values compared to unity.

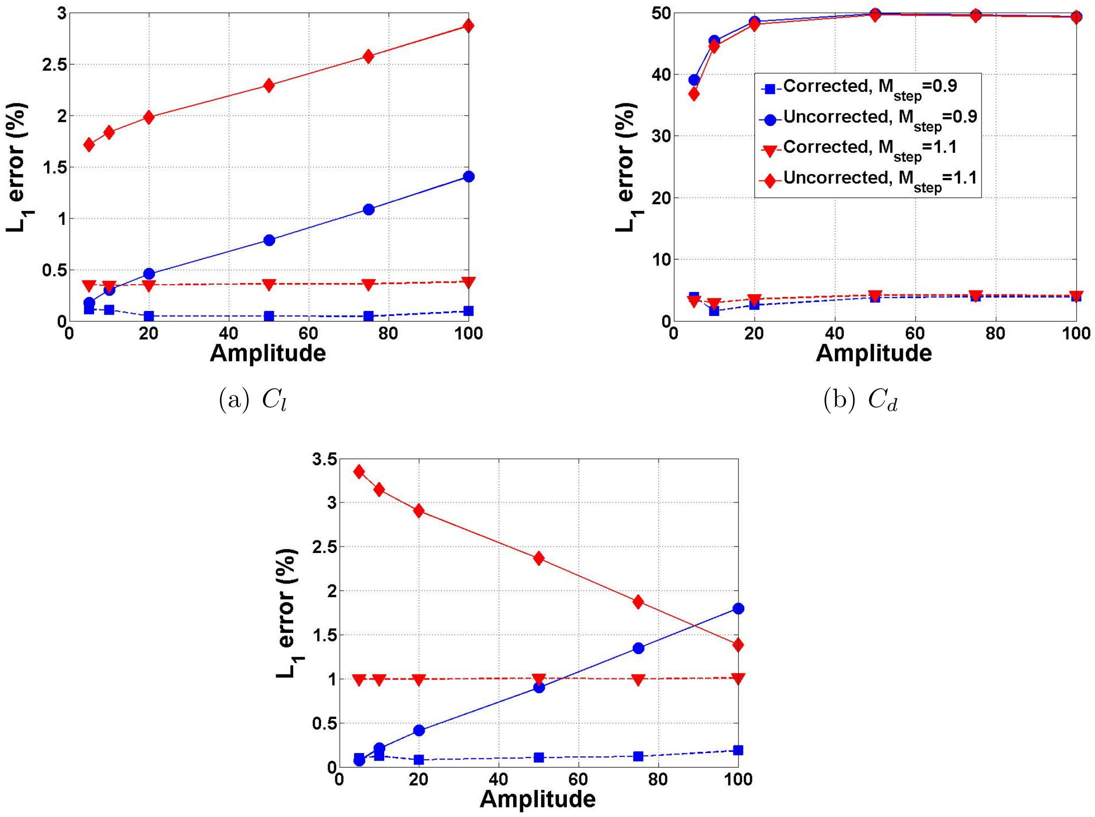


(a) f_c, lift coefficient, δ=1

(b) f_c, drag coefficient, δ=0.01

(c) f_c, moment coefficient, δ=1

Figure 4.4: Kriging surfaces for mode 1



(a) C_l

(b) C_d

(c) C_m

Figure 4.5: Amplitude tests, Mach 0.9

4.2.1 Amplitude Tests

To test the accuracy of the ROM as the amplitude of oscillation increases, sinusoidal oscillation test cases are generated with constant Mach number and oscillation frequency but varying oscillation amplitude. These tests are repeated for both Mach 0.9 and Mach 1.1. For each test, two ROMs are constructed, one with the step input computed at Mach 0.9 ($M_{step} = 0.9$) and the other with a step input at Mach 1.1 ($M_{step} = 1.1$); the oscillation frequency is $\omega_1 = 9.6\ Hz$. The results from each of the two ROMs for tests at Mach 0.9 (dashed lines) are compared to uncorrected ROM results (solid lines) in Fig. 4.5.

Fig. 4.5 shows that, while the correction factor ROM shows improved agreement

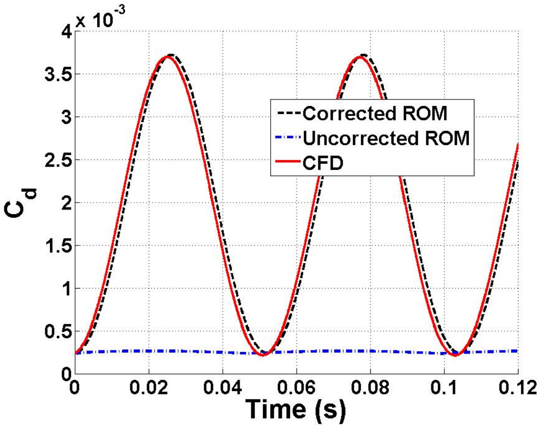
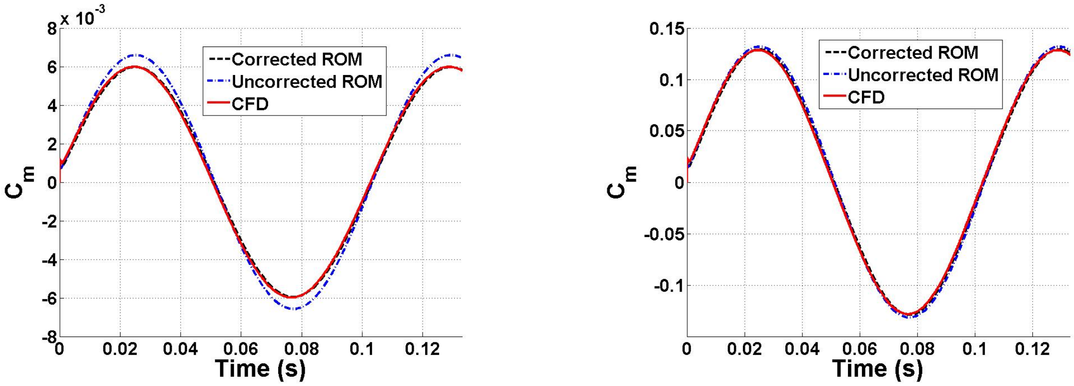


Figure 4.6: ROM-CFD comparison, Mach 0.9, amplitude=100, M_{step}=0.9

with the CFD results for the lift and moment coefficients, the overall errors seen for both the corrected and uncorrected ROMs are relatively small. Also, the errors for the ROM constructed with the same Mach number as the simulation ($M_{step} = M_{sim}$) are smaller than the errors from when $M_{step} \neq M_{sim}$, which is to be expected, though overall the errors for each of the corrected ROMs remain small, under 5% for all cases tested. As with the hypersonic test cases presented in Chapter 3, the uncorrected drag coefficient results do not match well at all with the CFD solution data. A qualitative comparison of the drag ROMs at an amplitude of 100 is shown in Fig. 4.6. One interesting feature of Fig. 4.5(c) is the fact that the uncorrected ROM errors for $M_{step} = 1.1$ decrease with oscillation amplitude. For a qualitative picture, of this, consider Fig. 4.7, which shows the comparisons for the moment coefficient at amplitudes of 5 (Fig. 4.7(a)) and 100 (Fig. 4.7(b)). Initially, the step Mach number of 1.1 results in an over-prediction of peak moment coefficient values at lower amplitudes. However, as the oscillation amplitudes increase, this over-prediction lessens, resulting in lesser error values.

Figure 4.8 shows the results for the tests conducted at Mach 1.1. The overall trends and errors are similar to those seen at Mach 0.9, though rather than the moment coefficient, the lift coefficient shows a decrease in error with increasing amplitude for



(a) Amplitude 5

(b) Amplitude 100

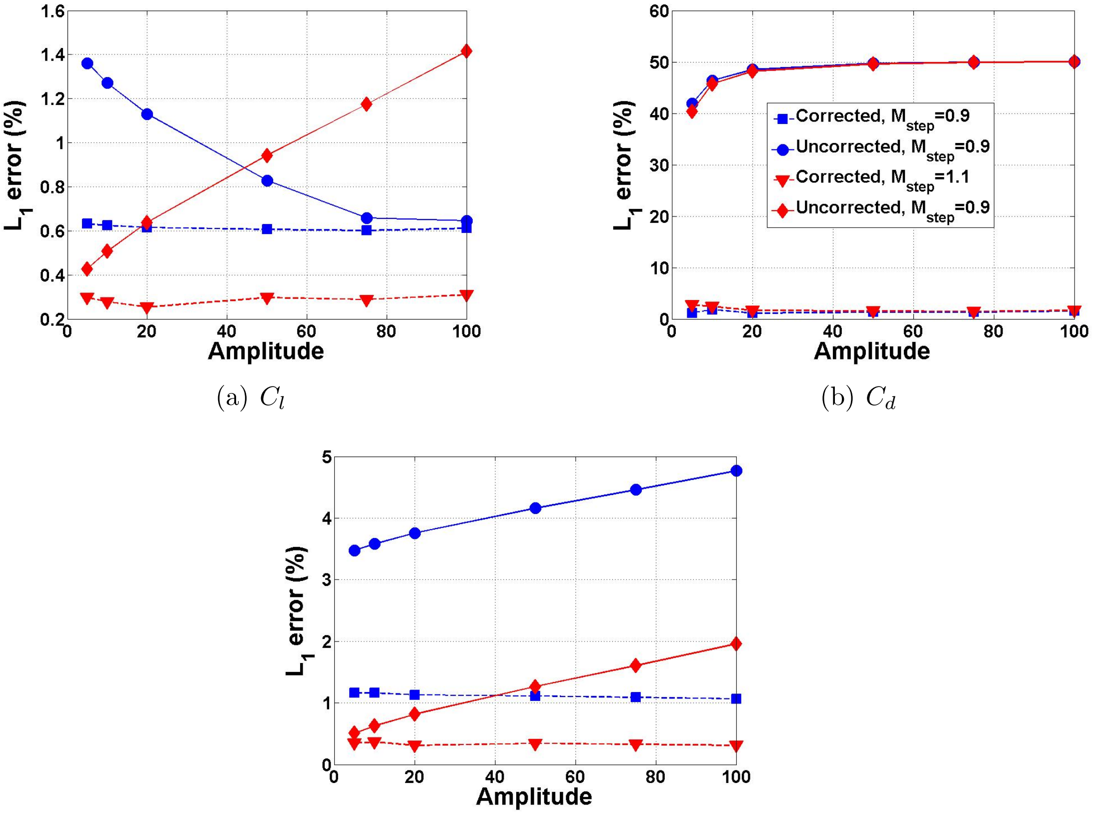
Figure 4.7: C_m results, Mach 0.9, M_{step}=1.1

the uncorrected ROM results for $M_{step} \neq M_{sim}$.

4.2.2 Frequency Tests

To investigate the accuracy of the ROM as oscillation frequency increases, tests are conducted at constant Mach number and oscillation amplitude but with oscillation frequencies ranging from ω_1 (9.6 Hz) to $5\omega_1$. As with the amplitude tests, these tests are repeated for both Mach 0.9 and Mach 1.1, and oscillation amplitude is held constant at 20. These parameters result in reduced frequencies ranging from 0.06-0.29 for M=0.9 and 0.05-0.24 for M=1.1. Also, for each value of M_{sim}, results from ROMs constructed with M_{step} = 0.9 and 1.1 are given. The results, displayed in Figs. 4.9 and 4.10, show that the errors do increase with oscillation frequency. This is likely due to the increased unsteadiness inherent with increased reduced frequencies combined with a steady correction factor formulation. Also, for the most part, the ROM in cases where $M_{step} = M_{sim}$ performs better than the ROM in cases where $M_{step} \neq M_{sim}$. The only exception is the drag coefficient for the tests with M_{sim} = 1.1, for which the ROM calculated with M_{step} = 0.9 has slightly lower errors, though the errors for each ROM increase at a very similar rate.

For a more qualitative comparison, Fig. 4.11 shows the lift and drag comparisons

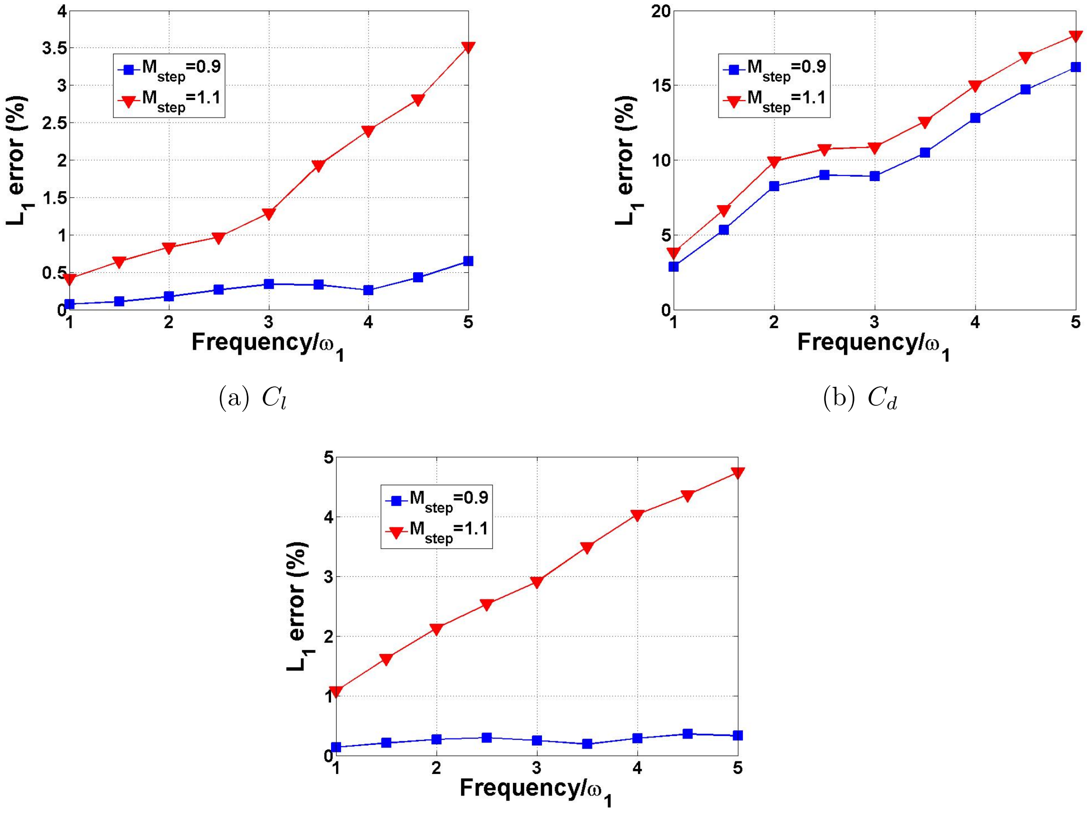


(a) C_l

(b) C_d

(c) C_m

Figure 4.8: Amplitude tests, Mach 1.1

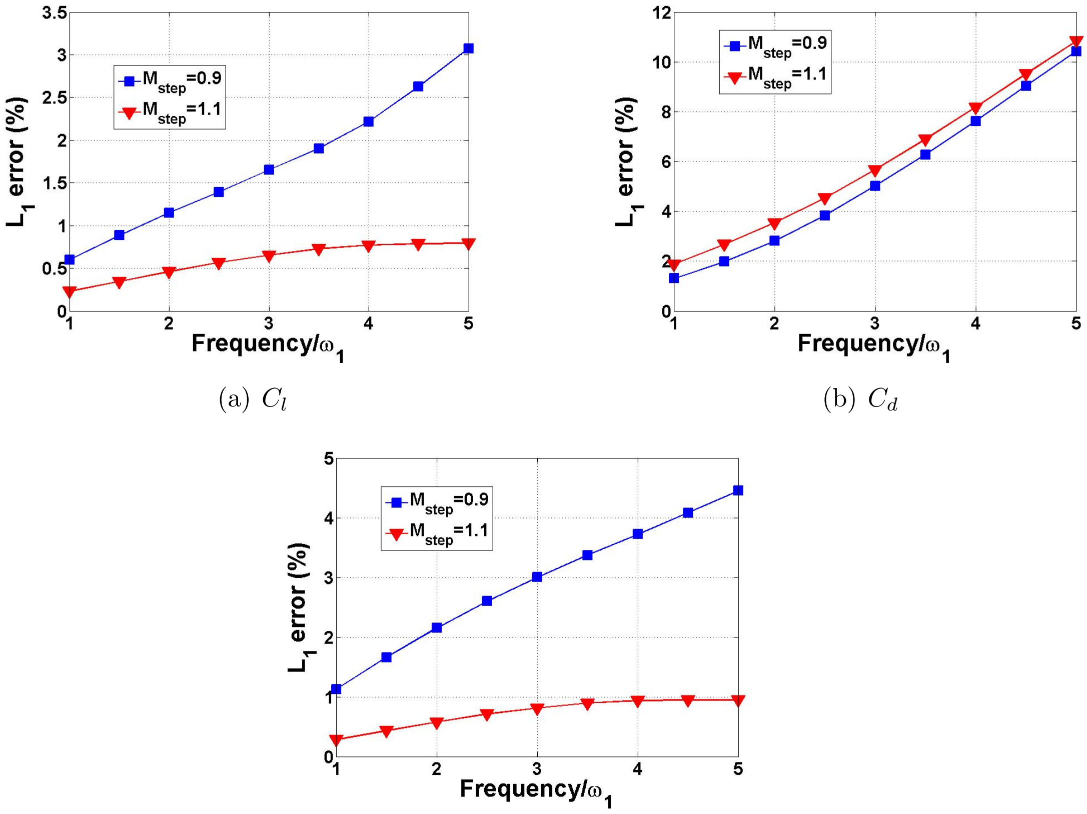


(a) C_l

(b) C_d

(c) C_m

Figure 4.9: Frequency tests, Mach 0.9



(a) C_l

(b) C_d

(c) C_m

Figure 4.10: Frequency tests, Mach 1.1

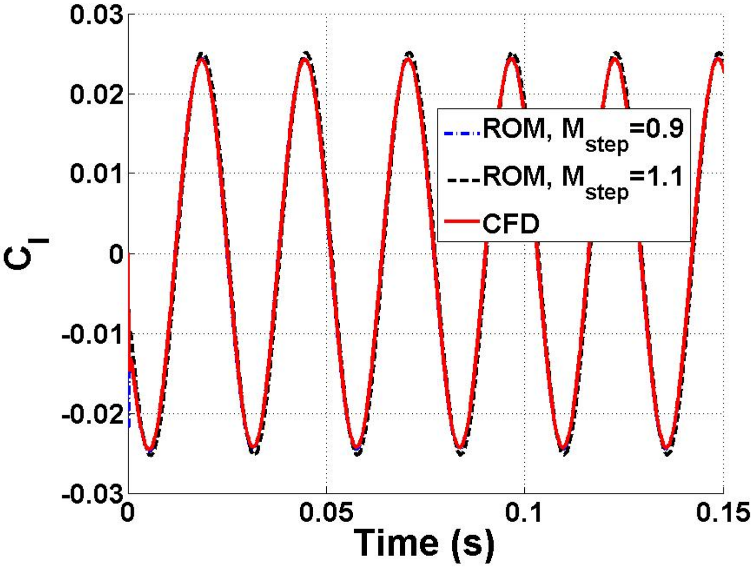
Table 4.2: **ROM phase shift test parameters**

Test	M	M_{step}	d_1	ω_{min} (rad/s)	ω_{max} (rad/s)	k range
1	0.9	0.9	20	60.3	$5\omega_{min}$=301	0.06-0.29
2	1.1	1.1	20	60.3	$5\omega_{min}$=301	0.05-0.24

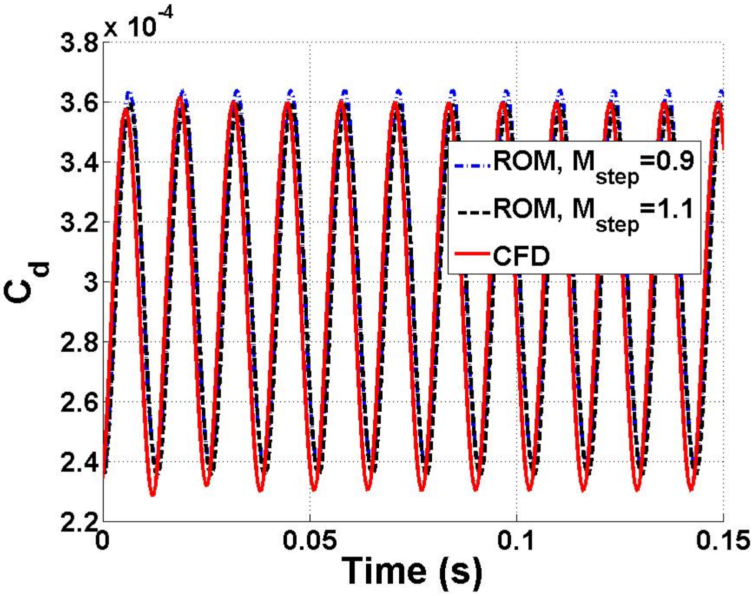
for the test case corresponding to a frequency of $4\omega_1$ and $M_{sim} = 0.9$. For the lift coefficient, each of the two ROMs match well with the CFD results. For the drag coefficient, two sources of error can be seen. First, a slight amplitude discrepancy has developed. Second, a phase shift is observed between the ROM and CFD results.

To investigate this phase shift, the Fast Fourier Transform[102] is computed for both the ROM and CFD results for each set of runs shown in Table 4.2, and the phase difference between the two responses is calculated. Figure 4.12 shows the lift, drag, and moment coefficient results for two separate series of results conducted at Mach 0.9 and Mach 1.1. Note that, for these tests, a single-mode ROM is used in which the correction factor is calculated through the use of a kriging surface with sampling points computed using direct CFD simulations.

For each of the two sets of results, the drag coefficient shows a greater phase difference over the range of frequencies than the lift and moment coefficients, though this increase is more pronounced for the Mach 0.9 tests. This shift may be the result of a slight ROM-CFD offset in time that is being manifested as an increasing phase difference with increasing oscillation frequency. Because of this overall phase difference increase with frequency, when looking at a specific test case, it is recommended to run a sample simulation at the highest frequency expected to be encountered and evaluate how the error fits in with the error tolerances for the problem.

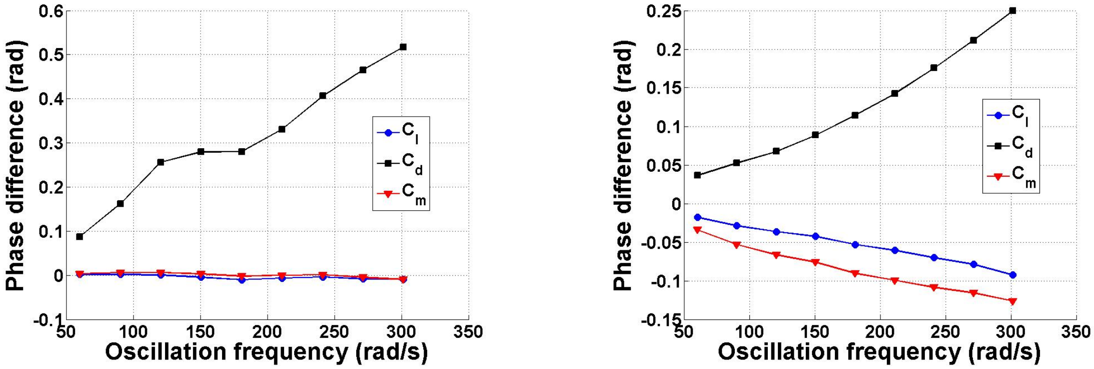


(a) C_l



(b) C_d

Figure 4.11: ROM-CFD comparisons for $\omega = 4\omega_1$, Mach 0.9



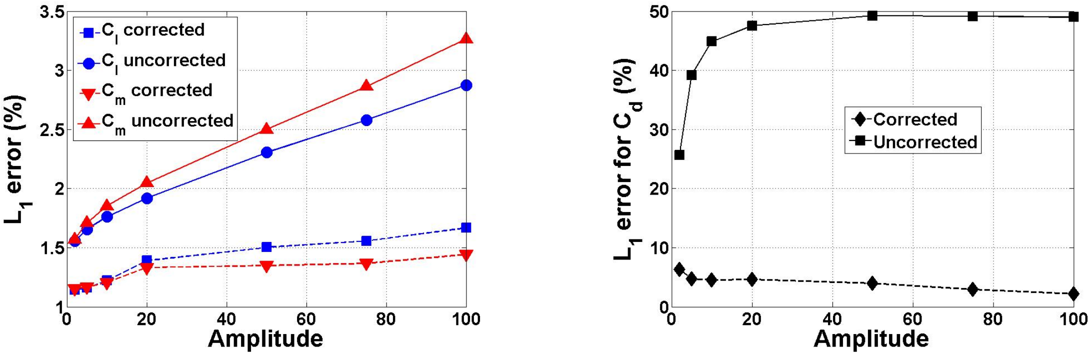
(a) Mach 0.9

(b) Mach 1.1

Figure 4.12: ROM-CFD phase shift

4.2.3 Single Mode Tests in Mixed Flow

The final item to discuss regarding the single mode results is the specific performance of the ROM in cases with mixed flow. To investigate this, additional amplitude and frequency tests are conducted for Mach 0.99; the other parameters for the tests are the same as those mentioned previously. Figures 4.13 and 4.14 show the amplitude and frequency test results, respectively, for this study; note that M_{step}=0.99 for these cases as well. Similar trends are seen here as compared to the amplitude and frequency tests conducted at Mach 0.9 and 1.1. The corrected ROM errors remain small over the entire range of amplitudes, while the uncorrected drag ROM results have the largest errors by far. For the frequency tests, lift and moment coefficient errors remain small, while the drag errors increase. To obtain a more qualitative sense of the errors, consider Fig. 4.15, which shows the ROM-CFD comparisons for the test cases corresponding to the nondimensionalized frequencies of 1 and 4 in Fig. 4.14. In addition to the slight phase shift as observed in previous frequency tests, an amplitude discrepancy develops as well for increased oscillation frequencies. However, the addition of extra kriging surface sampling points near the maximum modal amplitude in this run of 20 would likely help remedy this discrepancy. Overall, the single



(a) C_l and C_m

(b) C_d

Figure 4.13: Amplitude test, Mach 0.99

mode tests in the mixed-flow transonic regime show similar results to the other Mach numbers considered.

4.3 Multi-modal Tests

To investigate the ROM's applicability to multiple modes of oscillation, the first three modes of oscillation of the AGARD wing are considered. The first challenge in this process is to develop the simplified model to use for error estimation and sampling point determination. To address this issue, the Method of Segments has been developed and utilized for this purpose.

4.3.1 Method of Segments

The Method of Segments has been created to efficiently calculate correction factor values at locations throughout the parameter space without requiring a separate CFD run at each of the locations. The basic idea is that, when the wing is in an elastically-deformed position, it can be approximated as a series of chordwise-rigid segments along the span which are at different angles of attack, as shown in Fig. 4.16.

Since the correction factor methodology relies on the steady-state coefficients after a certain modal deformation has been input, the lift and drag at each of the chordwise

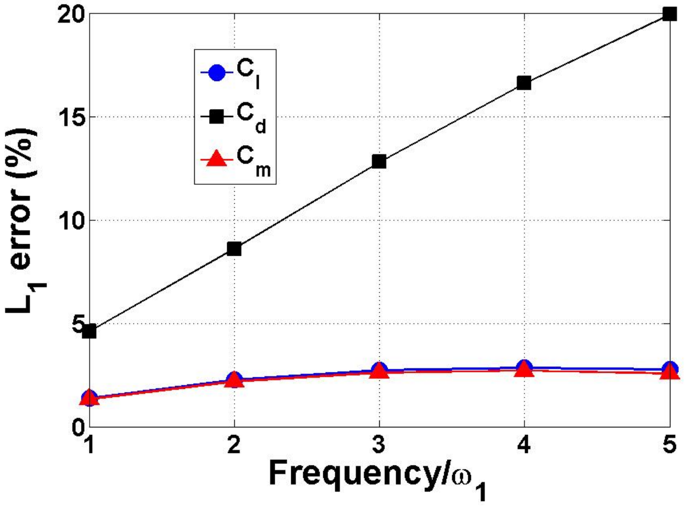
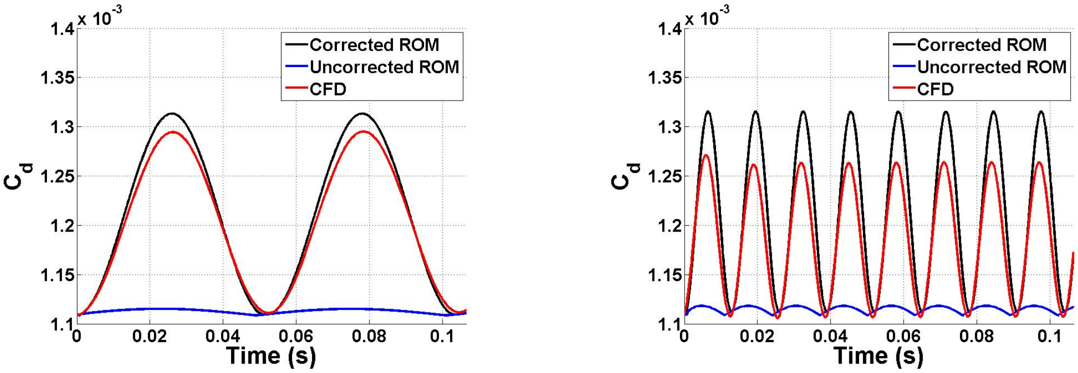


Figure 4.14: Frequency test, Mach 0.99



(a) Frequency/ω_1=1

(b) Frequency/ω_1=4

Figure 4.15: ROM-CFD comparisons, frequency test cases, Mach 0.99

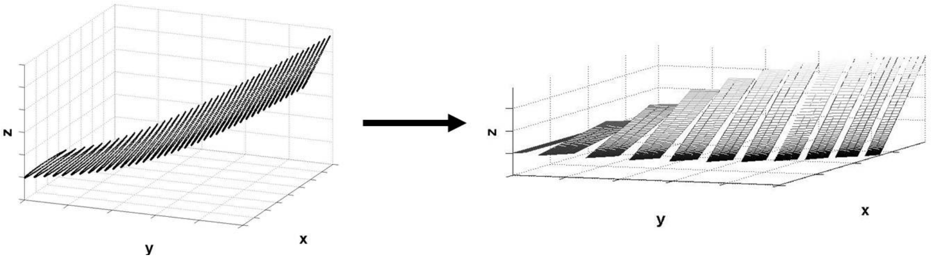


Figure 4.16: AGARD wing divided into segments

segments along the span are found through the use of steady rigid CFD simulations conducted at varying angles of attack and Mach numbers. While the individual segments will also undergo a plunge motion in addition to pitching motion during elastic deformations, these plunge motions are neglected here due to the steadiness of the CFD solutions being found. The specific steps to the method, shown graphically in Fig. 4.17, are as follows:

1. Divide the wing into chordwise segments along the span, which are assumed to be rigid in the chordwise direction. The AGARD 445.6 wing has been divided into 11 of these chordwise-rigid segments.

2. Conduct steady, rigid CFD runs throughout the parameter space, which consists of Mach number and angle of attack. The parameter space dimensionality will remain at 2 regardless of how many modes are being considered. Thus, the total number of runs remains relatively low, and steady runs are computationally cheaper than unsteady ones.

3. Track the lift and drag forces as well as the pitching moment on each of the chordwise segments, taking into consideration the spanwise width of each segment. Construct separate kriging surfaces for the force and moment quantities at each of the segments. Steps 1-3 are all performed up-front, prior to the ROM simulations.

4. For a certain wing deformation at a particular time step in a simulation, calculate the local angle of attack at each wing segment.

5. Pick the lift, drag, and moment off the kriging surfaces for each segment corresponding to the specific Mach number and local angle of attack; sum them together to find the lift, drag, and moment for the entire wing.

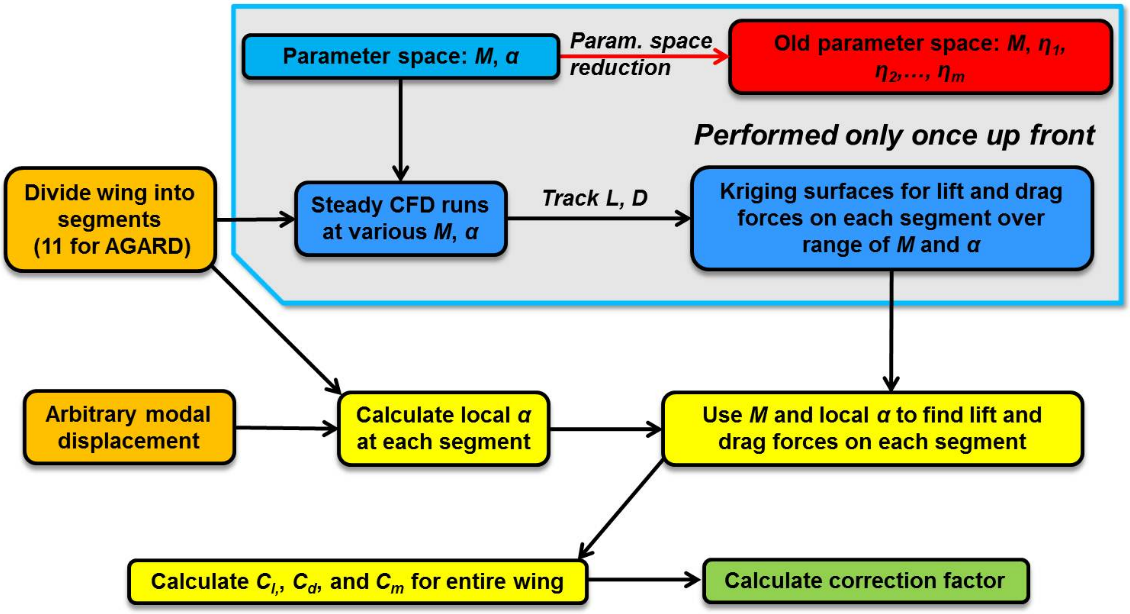

Parameter space: *M*, *α*
Param. space reduction
Old parameter space: *M*, *η*1, *η*2,…, *η*m
Performed only once up front
Divide wing into segments (11 for AGARD)
Steady CFD runs at various *M*, *α*
Track L, D
Kriging surfaces for lift and drag forces on each segment over range of *M* and *α*
Arbitrary modal displacement
Calculate local *α* at each segment
Use *M* and local *α* to find lift and drag forces on each segment
Calculate *C*l, *C*d, and *C*m for entire wing
Calculate correction factor

Figure 4.17: Method of segments process

6. Calculate the coefficients for the wing. These values can then be used to formulate the correction factor for that particular set of parameters.

The first test of the accuracy of the Method of Segments is to compare lift and drag coefficients found with MoS with those found with direct CFD simulations for the same sample cases. For this purpose, a total of 40 test cases in each of the Mach number ranges from 0.9-1.0 (denoted the subsonic runs) and 1.0-1.1 (supersonic runs) is selected. The various parameter ranges for these sample cases are shown in Table 4.3. The MoS kriging surfaces themselves are computed using a total of 152 sampling points arranged in a lattice-type pattern in each of the sub- and supersonic portions of the overall Mach number range considered here. The tip and root segment drag force kriging surfaces are shown in Fig. 4.18; the black dots represent the values found at each of the sampling points using CFD, corresponding to step 2 in the MoS procedure outlined above. Table 4.4 shows the results of the comparison; the mean L_1 errors are under 5% for the lift coefficient in each of sub- and supersonic ranges, while the drag coefficient mean errors are under 6%. These results demonstrate

Table 4.3: **Method of Segments testing parameters**

Parameter	**Min**	**Max**
M	0.9	1.1
d_1	-60	60
d_2	-40	40
d_3	-40	40

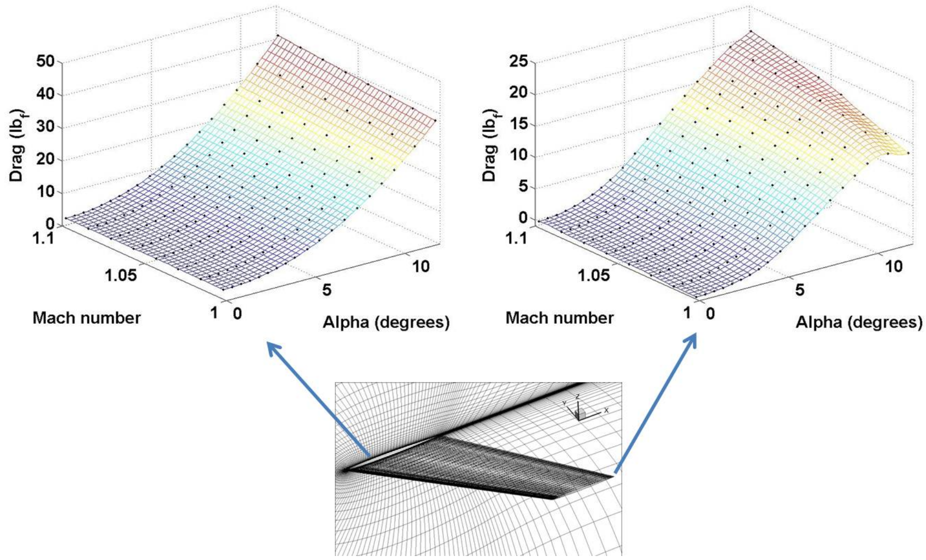


Figure 4.18: Method of Segments kriging surfaces, tip and root segments, Mach 1-1.1

the potential of the method for use in error estimation and ROM correction factor sampling point determination.

4.3.2 Multi-modal Oscillation Results

The goals of applying the ROM methodology to multiple modes of oscillation of the AGARD wing are twofold. The first goal is to characterize the overall errors seen from the ROM, as has been the goal of the other sets of results discussed in this dissertation. The second goal is to investigate the accuracy of a ROM constructed with the correction factor values calculated using the Method of Segments versus a ROM constructed with those same values computed with full CFD simulations. The

Table 4.4: **Method of Segments L_1 errors over 40 test cases (mean/standard deviation)**

	Subsonic	**Supersonic**
C_l	4.18/2.39	4.23/3.27
C_d	5.60/5.20	5.90/5.81

preceding section demonstrated the potential for the method to calculate individual coefficient values, but now the challenge is to demonstrate whether or not it can be applied to the full ROM. With these goals in mind, three different correction factor-based ROMs, along with a purely linear ROM, are constructed and compared with full-order CFD simulations; these ROMs are displayed in Table 4.5. Note that each ROM is constructed twice, once for Mach numbers less than 1 (Mach 0.9-1) and once for Mach numbers greater than 1 (Mach 1-1.1).

The first ROM, denoted from here on as ROM A, takes advantage of the relative computational efficiency of the Method of Segments by directly calculating the MoS correction factor value at each time step throughout the simulation without the computation of any correction factor kriging surfaces. Thus, this ROM eliminates any uncertainties based on the kriging surface fit of the data. The second MoS-based ROM, denoted ROM B, is created by first generating an initial number of sampling points in the Mach number-modal amplitude parameter space via Latin hypercube sampling. Using MoS, the correction factor at each of these points is found, and a kriging surface is computed. Next, the sampling point procedure from Chapter 2 is implemented by finding location of surface's maximum error through the use of the built-in MATLAB `predictor` function.[87] Then, an additional point is placed at this location, the surface is re-computed, and the maximum error is re-calculated. The whole process is repeated until stopping criterion is met, which, for this ROM, consists of finding total of 1,000 sampling points. Additionally, 11 more sampling points are specifically placed at zero amplitudes and varying Mach numbers to reduce errors

Table 4.5: **ROM variations**

ROM	**Sampling points** $(M \leq 1/M \geq 1)$	f_c **calc.**	**Comment**
A	N/A	MoS	Direct calculation of f_c at each time step using MoS
B	1,011/1,011	MoS/kriging	Points placed at location of maximum surface error
C	990/992	CFD/kriging	Mostly the same sampling points as ROM B
D	N/A	N/A	Linear ROM y_{conv}

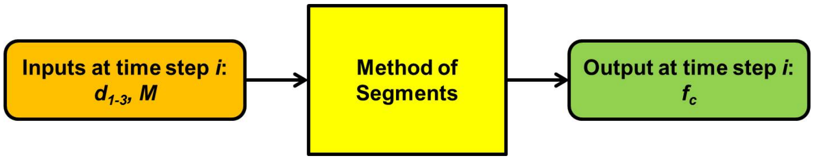


Figure 4.19: Schematic for ROM A

seen at those locations. ROM A can be thought of as the limit of ROM B if infinite sampling points are used. Figures 4.19 and 4.20 show schematics highlighting the differences between ROMs A and B.

The next ROM considered here, ROM C, is calculated using the same sampling points as in ROM B, but the correction factor values at each point are computed using individual direct CFD simulations rather than the Method of Segments. Its schematic is shown in Fig. 4.21 and differs from that of ROM B by replacing the Method of Segments block with CFD simulations. Note that in actuality, ROM C consists of fewer sampling points than ROM B. This is due to CFD code limitations, as some of the runs at higher modal deformations ran into numerical issues and thus did not produce results. Finally, ROM D consists of the linear ROM y_{conv} found using linear convolution and superposition.

Results obtained from these four ROMs are compared to a total of 100 full-order CFD simulations with sinusoidal oscillations of the first three modes of the AGARD 445.6 wing; half of these cases are in the subsonic portion of the Mach regime (Mach 0.9-1), and half are in the supersonic portion (Mach 1-1.1). For the subsonic test

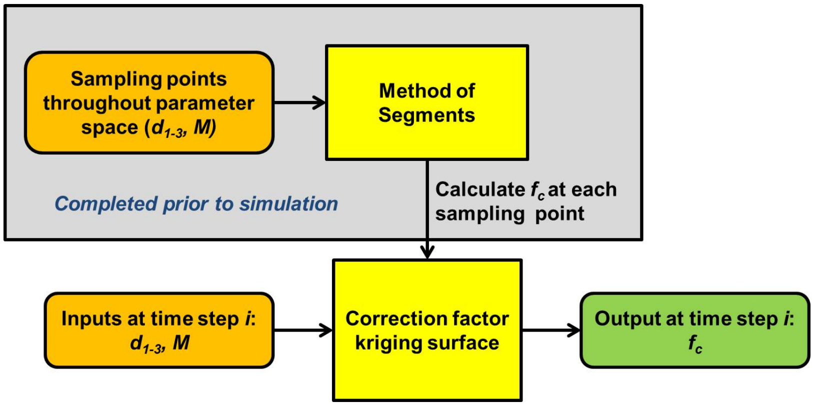

Sampling points throughout parameter space ($d_{1\text{-}3}$, M)
Method of Segments
Completed prior to simulation
Calculate f_c at each sampling point
Inputs at time step i: $d_{1\text{-}3}$, M
Correction factor kriging surface
Output at time step i: f_c

Figure 4.20: Schematic for ROM B

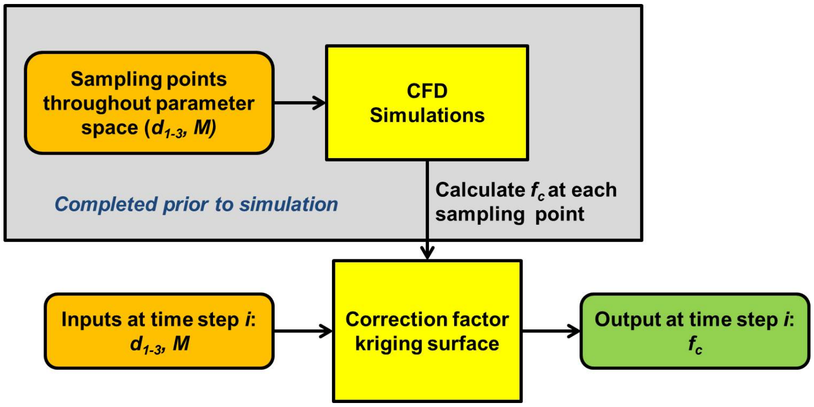
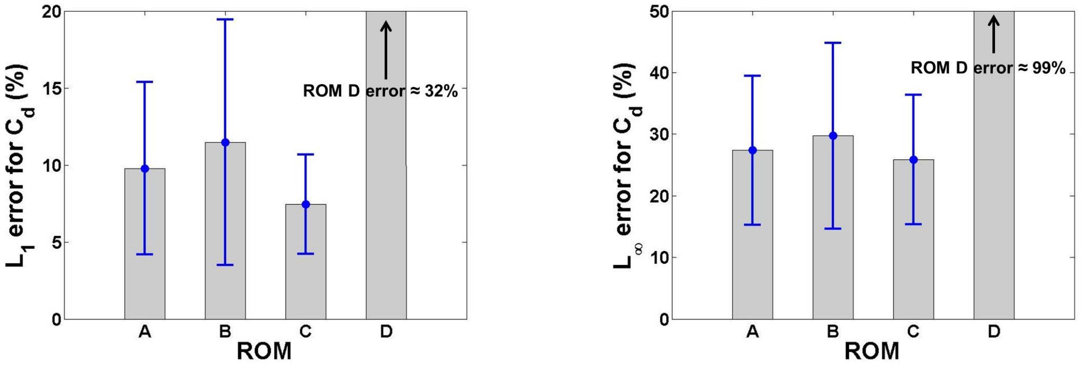

Sampling points throughout parameter space ($d_{1\text{-}3}$, M)
CFD Simulations
Completed prior to simulation
Calculate f_c at each sampling point
Inputs at time step i: $d_{1\text{-}3}$, M
Correction factor kriging surface
Output at time step i: f_c

Figure 4.21: Schematic for ROM C

Table 4.6: **Parameter ranges for sinusoidal test cases, AGARD 445.6 wing**

Parameter	Min	Max
M	0.9	1.1
d_1	-50	50
d_2	-35	35
d_3	-30	30
ω	$60.3\ rad/s$	$350\ rad/s$
k	0.05	0.29



(a) L_1

(b) L_∞

Figure 4.22: C_d error results over 100 test cases

cases, $M_{step} = 0.9$ is used for ROM construction, while $M_{step} = 1.05$ is used for the supersonic cases. Table 4.6 highlights the ranges of the various parameters used for these tests. Note that the frequency range is chosen such that the minimum value corresponds to $\omega_1 = 60.3\ rad/s$, and the maximum value corresponds to slightly over $5\omega_1$.

Figure 4.22 shows both the L_1 and L_∞ drag coefficient error results for each of the four ROMs listed in Table 4.5. The bars show the mean value of each of the error metrics over all 100 runs, while the error bars show the standard deviations.

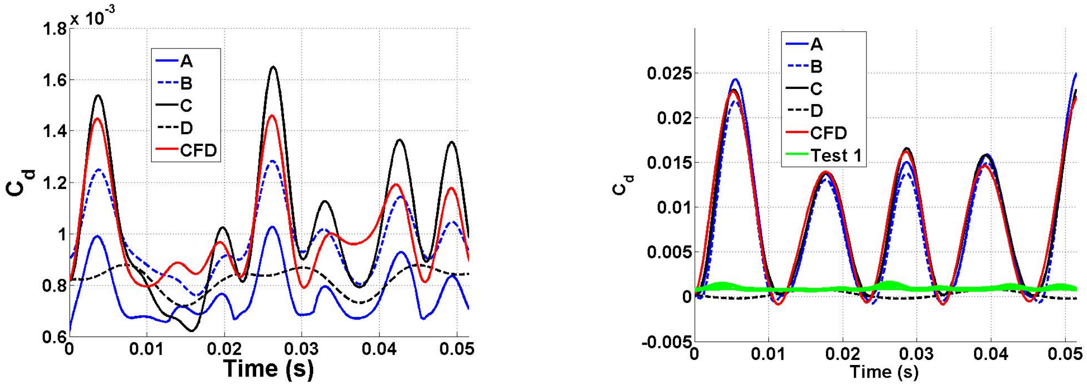
For the two Method of Segments ROMs, ROM A shows a slight improvement over ROM B for each of the error metrics, decreasing the mean L_1 error from around 11.5% to 9.8% and the L_∞ error from around 30% to just over 27%. This is to be expected

Table 4.7: **Test case parameters and errors**

Test	M	d_1	d_2	d_3	ω_1	ω_2	ω_3	ROM A	ROM B	ROM C
						(rad/s)			C_d errors: L_1/L_∞	
1	0.98	-11.1	3.30	-4.20	109	289	541	36.2/68.4	12.7/30.3	15.9/35.0
2	0.91	25.2	-29.2	5.80	158	274	492	5.11/17.3	6.26/22.9	3.63/12.1
3	0.94	28.4	-9.00	-22.4	137	269	557	9.90/29.1	12.0/30.9	6.14/25.8
4	0.96	13.4	-23.7	21.3	180	245	379	3.77/14.0	4.58/16.1	6.92/29.3

due to the fact that ROM A calculates the MoS correction factor value at each time step, while ROM B obtains the correction factor value from a previously-constructed kriging surface. Next, ROM D clearly performs the worst, with L_1 and L_∞ errors of around 32% and 99%, respectively. This demonstrates that the linear ROM is not suitable to model the drag coefficient. Finally, ROM C performs the best in terms of each of the error metrics when compared with the linear and MoS-based ROMs. For a better illustration of these comparisons, the ROM and CFD results for a number of specific test cases are shown in Figs. 4.23 and 4.24; the parameters and errors for these runs are listed in Table 4.7.

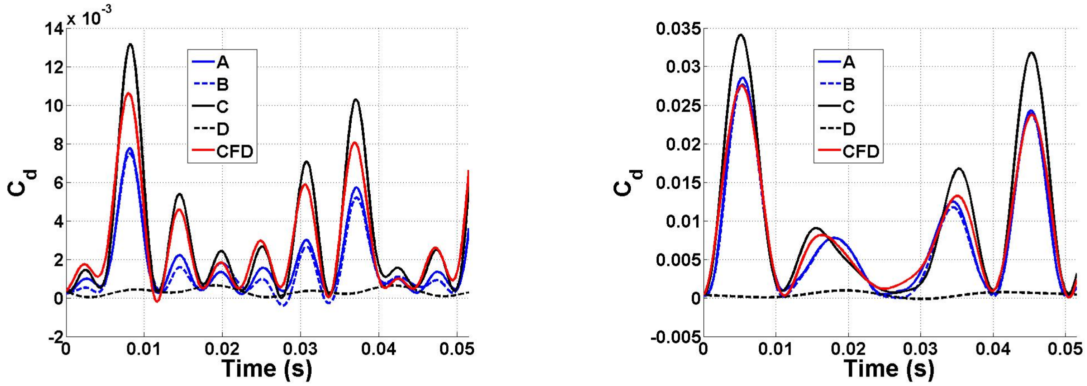
In general, graphically speaking, two main sources of error can be seen. Figure 4.23(a) displays the results for Test 1, which has the highest L_1 error for ROM C over all runs. Qualitatively, relatively large discrepancies can be seen between each of the ROMs and the CFD results. However, when looking at the total range spanned by the drag coefficient response value, it is relatively small. This is further illustrated in Fig. 4.23(b), which shows the ROM-CFD comparisons for Test 2, which has some of the smallest error values out of all test cases. In addition to the ROMs for that test case, the values from Test 1 are superimposed on the plot with the green lines. As can be seen, though the errors for Test 1 are larger than those for Test 2, the range spanned by the response of Test 2 is much larger. This shows that some of the large error values are due to small ranges spanned by the response quantity, resulting in small denominators for the error metric equations (Eqs. 2.19 and 2.20)



(a) Test 1

(b) Test 2

Figure 4.23: Example test cases, large and small C_d errors



(a) Test 3

(b) Test 4

Figure 4.24: Example test cases, mean C_d errors

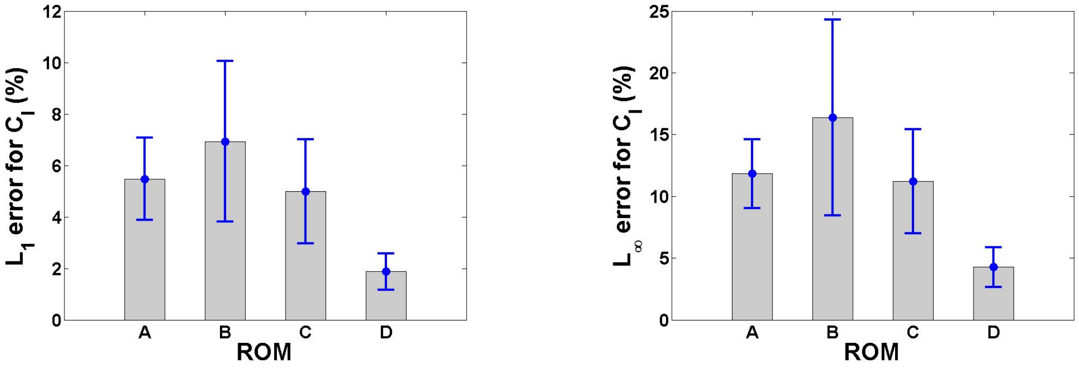
and hence larger error values. Note that, for Test 1, the unexpected result of ROM A having a larger error than ROM B is observed. This appears to be the result of a slight DC-type of offset introduced by ROM A for this particular case which has been magnified due to the small ranges of coefficient values spanned in the simulation.

For the second source of error, consider Fig. 4.24, which shows test cases having error values right around the mean. In these cases, the largest source of error appears to be amplitude discrepancies between the ROM and CFD results. In many situations, the predictions from ROM C tend to over-predict peak drag coefficient values, resulting in some error, while the peak comparisons for the MoS-based ROMs vary.

In addition to the drag coefficient, lift coefficient results for these same test cases are found as well, and the mean errors and standard deviations can be found in Fig. 4.25. These results are strikingly different than those found for the drag coefficient and show two main points. The first is that the errors for each of the ROMs are relatively small, under 6% for the mean L_1 error for ROM C. The second is that the best agreement is found between ROM D (linear) and the CFD results, and each of the correction factor ROMs give slightly higher errors. This suggests that, for these test cases, the linear ROM is sufficient to model the lift response. The higher errors for the nonlinear correction factor ROMs (A-C) likely result from the fact that the addition of the nonlinear correction factor inherently results in some approximation errors. When looking at quantities like the drag coefficient, which are very nonlinear, the application of the correction factor, despite these approximation errors, still greatly improves the results and shows a significant decrease in error over the linear ROM. However, when the system itself can be well-modeled as linear, these approximation errors in the application of the nonlinear correction factor result in a slight error overhead, thus increasing the errors in this case over the linear model. Figure 4.26 shows the ROM-CFD comparisons for test cases 3 and 4 from Table 4.7, in which the improvement of results with the purely linear ROM over the nonlinear ROMs can be seen. For ROM C, the introduction of the correction factor generally results in a slight over-prediction of peak lift coefficient values, resulting in larger error values. The results for the pitching moment are found to be similar in nature to the lift coefficient results in that ROM D is a good predictor of the response.

4.3.3 Mixed Flow Results

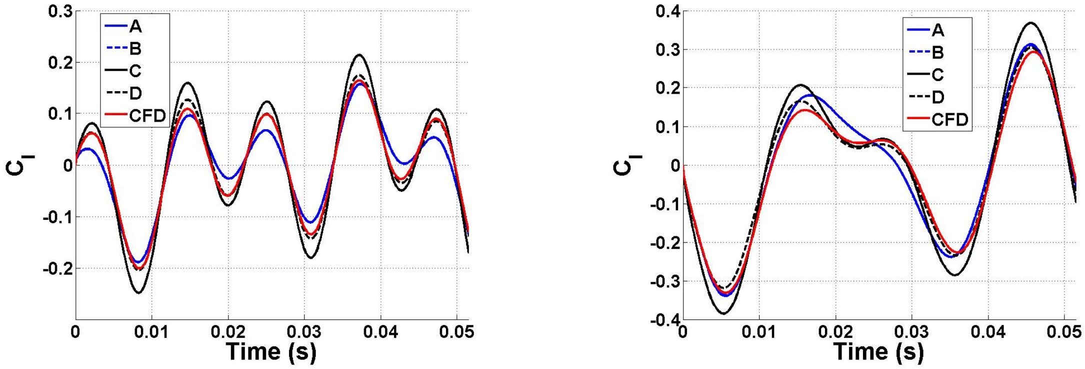
As before with the single mode tests, an important item to evaluate is the ROM's performance for multi-modal oscillations in the presence of mixed transonic flow. To do so, consider the specific test cases out of the 100 multi-modal oscillation test cases



(a) L_1

(b) L_∞

Figure 4.25: C_l error results over 100 test cases



(a) Test 3

(b) Test 4

Figure 4.26: Example test cases, C_l errors

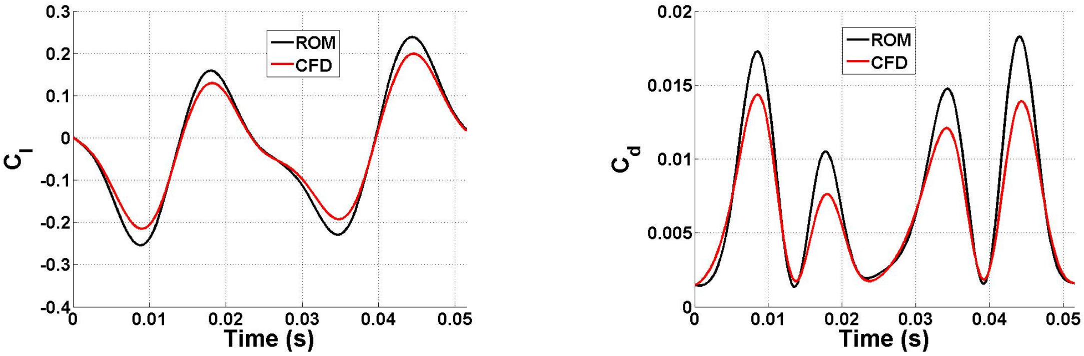
Table 4.8: **Results for mixed-flow transonic cases**

Coefficient	L_1 **error**	**Standard deviation**
C_l	4.86	0.78
C_d	9.78	2.41
C_m	5.73	0.60

Table 4.9: **Parameters for Case T1**

M	0.9963
d_{1-3}	35.3, -18.8, -10.7
ω_{1-3} (rad/s)	75.1, 244.0, 474.7
L_1 error (C_l, C_d, C_m)	4.45, 9.43, 5.25

that are in the Mach 0.98-1 mixed-flow transonic regime for this wing. In all, 10 cases fall into that range. The mean L_1 errors and error standard deviations of these cases for each coefficient are shown in Table 4.8. The results show that the mean drag error for these specific cases is slightly higher than for all of the cases, while the lift mean error is comparable. To get a qualitative sense of where the errors are coming from, consider Fig. 4.27, which shows the ROM-CFD comparisons for the lift and drag coefficient for a sample case, dubbed here as Case T1, having a drag error approximately that of the mean value for these mixed-flow cases; the parameters for this run are shown in Table 4.9. The main differences that can be seen between the ROM and CFD results for each coefficient are peak discrepancies. The ROM results over-predict the peak amplitude values, resulting in some error for these test cases. While the ROM results still show overall good agreement with the CFD solutions for the mixed-flow Mach range, *a priori* knowledge of the location of this range for a specific geometry would aid in identifying the regions within the parameter space where nonlinear transonic effects are prevalent. Additional correction factor sampling points may be required in order to reduce errors in those locations.



(a) Lift comparison

(b) Drag comparison

Figure 4.27: ROM-CFD comparisons for Case T1

4.4 Conclusion

The following are the relevant conclusions which can be deduced from this chapter:

- For tests with a single mode of oscillation, the ROM methodology in general works well. The ROM errors remain small as oscillation amplitude increases with other variables being held constant. However, the drag coefficient errors do increase with oscillation frequency at a larger rate than in the hypersonic regime, an effect that is artifact of a phase shift between the ROM and CFD results that develops at higher oscillation frequencies.

- The ROM methodology works well over the range of multi-modal oscillation test cases near Mach 1 considered here, having a mean drag coefficient L_1 error of just over 7%. The other coefficients show generally less errors than the drag.

- The Method of Segments shows promise as a simplified model for use within this regime, especially for the determination of the correction factor kriging surface sampling point values. Additionally, the ROMs constructed with the sampling point values computed with MoS rather than direct CFD simulations for this case had only slightly higher drag coefficient errors than the ROM with the values calculated through direct CFD simulations, further demonstrating potential applications.

Chapter 5

Applicability to Single Configuration Over Multiple Mach Regimes

This chapter presents studies which investigate the ROM methodology's accuracy over several different Mach regimes for a single test case geometry. Simulations are conducted for a range of Mach numbers, spanning from relatively low subsonic (M=0.3) up to supersonic (M=3.0). ROMs are constructed using varying values of step response Mach number M_{step} and number of correction factor kriging surface sampling points N_{samp}. These sampling point values themselves are calculated two separate ways, using direct CFD simulations and the Method of Segments. Errors are tracked and analyzed as functions of all of these parameters, and the results give a sense of the overall accuracy and range to which the ROM can be applied.

5.1 Problem Setup

The geometry used for testing in this chapter is the same AGARD 445.6 wing as used in Chapter 4; the mode shapes are the same as well. This wing would be expected to fly up through the low supersonic Mach regime, which encompasses much of the Mach number parameter space considered here. For this chapter, two specific Mach number ranges are considered: Mach 0.3-Mach 0.9 (denoted the subsonic range) and Mach 1.1-Mach 3.0 (denoted the supersonic range). The transonic range of Mach

0.9-1.1 is the specific subject of Chapter 4. As in Chapter 4, an offset δ value of 100 is used for ROM construction.

The results presented in this chapter can be subdivided into two separate sets, each of which consists of ROM-CFD comparisons of simulations with oscillations of the first three mode shapes. The first set, denoted Set A, consists of 50 test cases in each of the subsonic and supersonic Mach ranges with sinusoidal oscillations given to each mode shape. The Mach number, modal amplitudes, and modal frequencies are all determined through Latin hypercube sampling. A number of different values are used for M_{step}, and the results using each value are compared. The analysis of this set of results achieves two separate goals. The first is to evaluate the accuracy of the ROM as M_{step} moves away from the simulation Mach number M_{sim}. In a realistic setting, M_{step} would not be expected to be equal to M_{sim}, so these results allow for a systematic study of the effect of the choice of M_{step} over a much wider Mach range than previously tested for this AGARD 445.6 configuration. The second goal is to investigate the ROM accuracy with varying number of correction factor kriging surface sampling points N_{samp}. The error methodology described in previous chapters is employed, and the mean errors over all tests are plotted as functions of N_{samp}.

The second set of results, denoted Set B, consists of defining 25 sets of modal parameters (both amplitudes and oscillation frequencies) and conducting simulations for all of those modal parameters at each one in a specific range of Mach numbers. The modal parameters are obtained through Latin hypercube sampling. The goals for this set of results are again twofold. First, these results allow for the error of the methodology in general to be analyzed as a function of Mach number, as the same modal parameters will be tested at different Mach numbers. This way, potential conclusions of the range of the ROM's applicability can be deduced. The second goal is the same as that for Set A, to look at the effect of N_{samp} on the results. However, this set of results will allow the effect of Mach number to be examined more closely.

5.2 Set A Results

The first results to be examined are those from Set A, which consist of 50 simulations over the parameter spaces of interest. The specific parameters for each run are determined through Latin hypercube sampling, and the parameter spaces for both the sub- and supersonic test cases are displayed in Table 5.1. Three different ROMs are computed for each test, the same ROMs A-C as described in Table 4.5 in Chapter 4. For ROMs B and C, which rely on a pre-determined number of sampling points for the correction factor kriging surfaces, varied numbers of sampling points are used, the effects of which will be discussed subsequently. These kriging sampling points are determined by first using Latin hypercube sampling over the parameter space to select 100 initial sampling points. Then, a certain number of points (15 for the subsonic Mach range and 41 for the supersonic Mach range) are specifically placed at locations of zero modal amplitudes but varying Mach number in an attempt to reduce potential errors in those locations. Then, the sampling point procedure detailed in Chapter 2 is implemented, with additional sampling points placed at locations of maximum kriging surface error; the Method of Segments is chosen as the simplified model to use when generating the correction factor sampling point values used in the error analysis. Note that, for ROM C, individual CFD runs are then conducted to determine the correction factor values at the pre-determined sampling points, while ROM B relies on the Method of Segments values themselves for the correction factor values. A total of 400 additional points is found using this method, and ROM-CFD comparisons are made using kriging surfaces re-calculated each time an additional 100 points have been added.

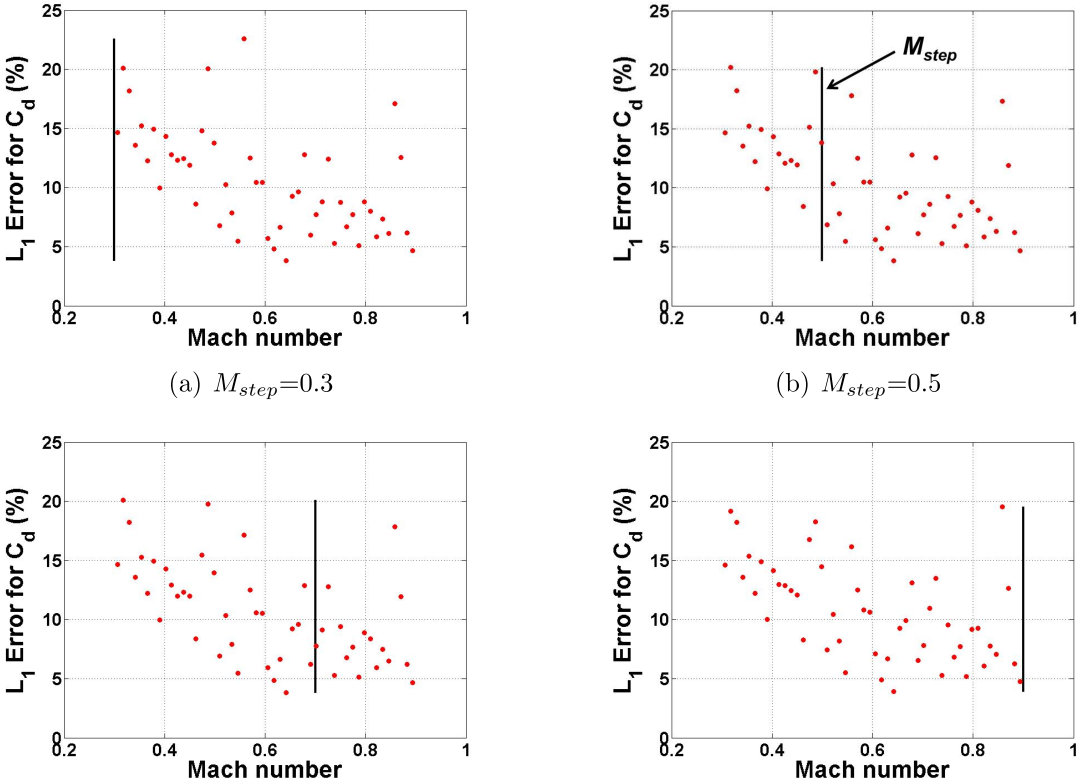
Table 5.1: **Parameter ranges for Set A results**

Parameter	**Min**	**Max**
M (sub/super)	0.3/1.1	0.9/3.0
d_1	-50	50
d_2	-35	35
d_3	-30	30
ω	63 rad/s	350 rad/s
k (sub/super)	0.06/0.02	1.02/0.28

5.2.1 Drag Coefficient Results

The first item to investigate within these results is the range of applicability of the ROM in terms of Mach number. Figure 5.1 shows the ROM C (correction factor calculated using CFD and kriging surfaces) drag coefficient errors for each of the 50 tests arranged in order of increasing Mach number for the subsonic tests, while Fig. 5.2 shows the same plots for the supersonic Mach range. The subplots show results calculated using various values of M_{step}, which are indicated on the plots with the solid black lines. For all of these tests, the ROMs are calculated using the full amount of 500+ kriging surface sampling points. For visualization purposes, some of the higher error values have been omitted from several of the plots; when this is the case, the number of omitted points as well as the maximum error value are noted on the specific plot.

In Fig. 5.1, the errors, in a purely qualitative sense, do appear to generally decrease as Mach number increases for the subsonic range. Some of the highest error values are seen at the lowest Mach numbers, in this case around 0.3, likely due to the increased reduced frequencies which result from lower flow velocities. For the supersonic range shown in Fig. 5.2, no overall trend of errors with respect to Mach number is readily evident. In each of the two Mach ranges, the case with the lowest value for M_{step} appears to give the largest overall errors. For the supersonic tests, the number of



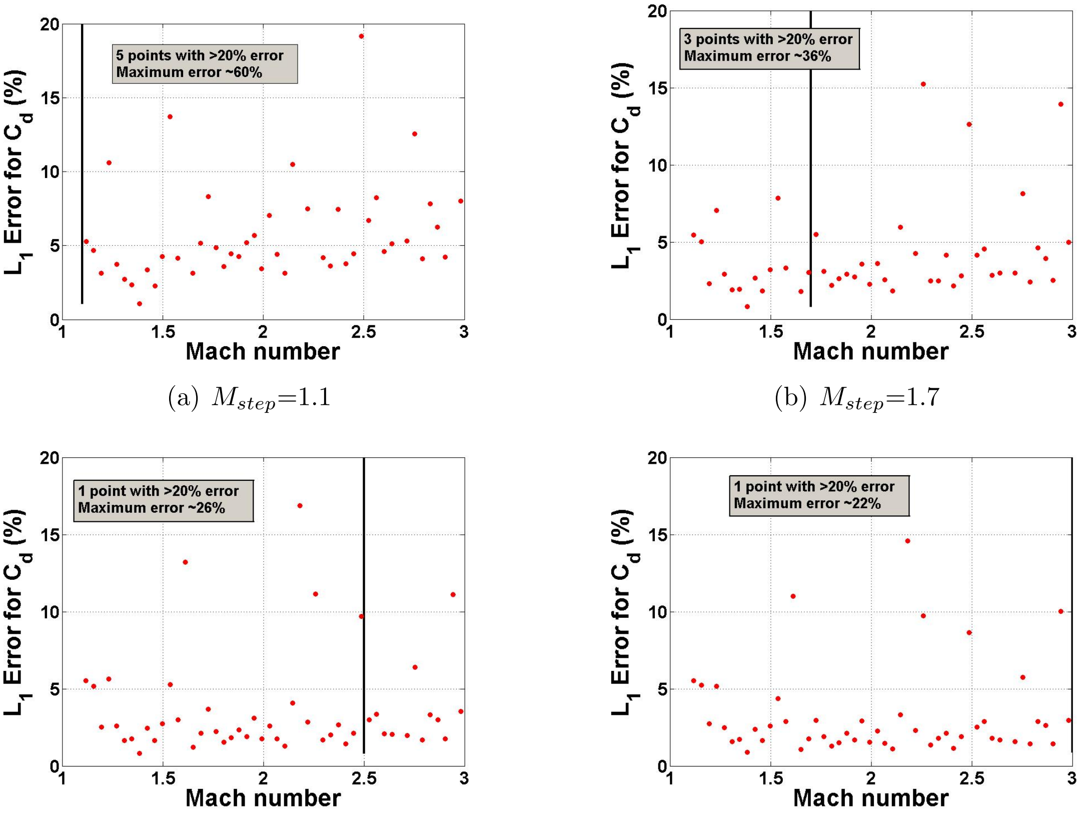
(a) M_{step}=0.3

(b) M_{step}=0.5

(c) M_{step}=0.7

(d) M_{step}=0.9

Figure 5.1: Drag coefficient errors for subsonic test cases as function of Mach number, results Set A, ROM C



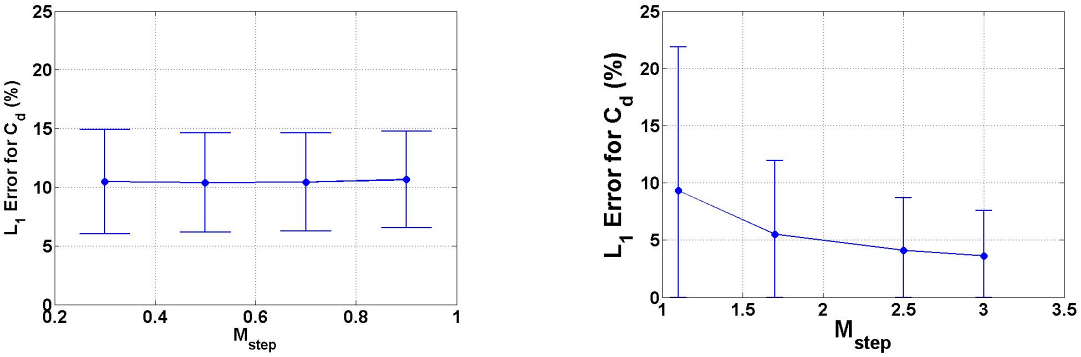
(a) M_{step}=1.1

(b) M_{step}=1.7

(c) M_{step}=2.5

(d) M_{step}=3.0

Figure 5.2: Drag coefficient errors for supersonic test cases as function of Mach number, results Set A, ROM C



(a) Subsonic Mach range

(b) Supersonic Mach range

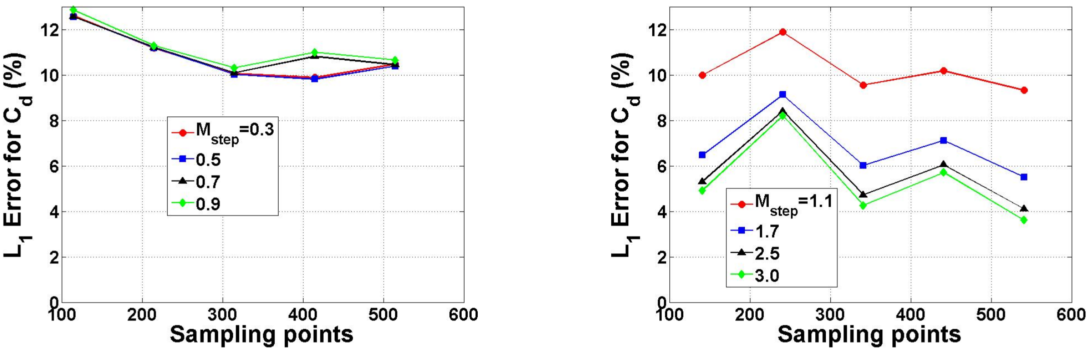
Figure 5.3: C_d errors for subsonic test cases as function M_{step}, results Set A, ROM C

errors over 20% as well as the maximum error seen both decrease as M_{step} increases, a fact likely supporting the previously-seen result that the ROM accuracy improves with increasing Mach number/decreasing reduced frequency. To evaluate these data in a more quantitative sense, consider Fig. 5.3, which shows the mean error and standard deviation (in the form of error bars) over all tests for ROM C for each of the values of M_{step} tested, i.e., the mean and standard deviations from the data shown in Figs. 5.1 and 5.2. For the subsonic test cases, the error values are fairly constant for each value of M_{step}. For the supersonic test cases, a decrease in mean error value is seen as M_{step} increases. To compare the errors seen in the subsonic and supersonic tests, the supersonic errors are less than the subsonic errors in general. The subsonic mean errors are just over 10%, while the error for M_{step}=1.1 for the supersonic tests is just under 10%, and the errors decrease from there. This suggests that the ROM's accuracy may increase with flow velocity, and hence lower reduced frequency, as a result of the general decrease in unsteadiness in flows.

The next item that needs to be examined is how the ROM's accuracy increases or decreases with a varying number of kriging surface sampling points. Figure 5.4 shows the ROM C errors for a range of correction factor kriging surface sampling points for both the subsonic and supersonic Mach ranges. For the subsonic cases, the data show the errors initially decreasing with increasing number of sampling points until around 300 points have been reached. At that point, a slight increase is seen at just over 400 points for the cases with M_{step}=0.7 and M_{step}=0.9, while a decrease is seen for M_{step}=0.3 and M_{step}=0.5. Other than this one value of N_{samp} of just over 400, each of the M_{step} values follows similar trends. Overall, solely in terms of mean error value, the effect of increasing the number of sampling points in the range shown here appears to be minimal, decreasing by only around 2 percentage points from the fewest sampling points to the most sampling points. To look at what happens to the spread of the error values, consider Fig. 5.5(a), which shows the mean and standard deviation of the errors for the M_{step}=0.7 case. From the plot, it can be seen that the standard deviation of error values is significantly decreased from the initial number of sampling points up to the final value of just over 500, suggesting that the addition of these sampling points does in fact help the overall accuracy of the ROM.

For the supersonic cases, the trend appears less clear, as the increase in sampling points alternately causes the overall errors to increase and then decrease. However, two items to note from Fig. 5.4(b) are that the errors for the supersonic cases again are generally smaller than those for the subsonic cases and that the omission of the largest error value does not change the agreement or trends among the various M_{step} ROMs as it does for the subsonic cases. To investigate the spread of the errors in the same way as for the subsonic cases, consider Fig. 5.5(b), which shows the mean and standard deviations for the M_{step}=2.5 ROM. Unlike the subsonic case, the standard deviations do not show much of a trend, as they alternately increase and decrease in size along with the mean.

The observation that, in some circumstances, the errors actually increase with increasing number of sampling points is interesting. This may be explained by the fact that some of the additional sampling points could be added along the parameter space boundaries or other locations which do not influence many of the simulations. Also,



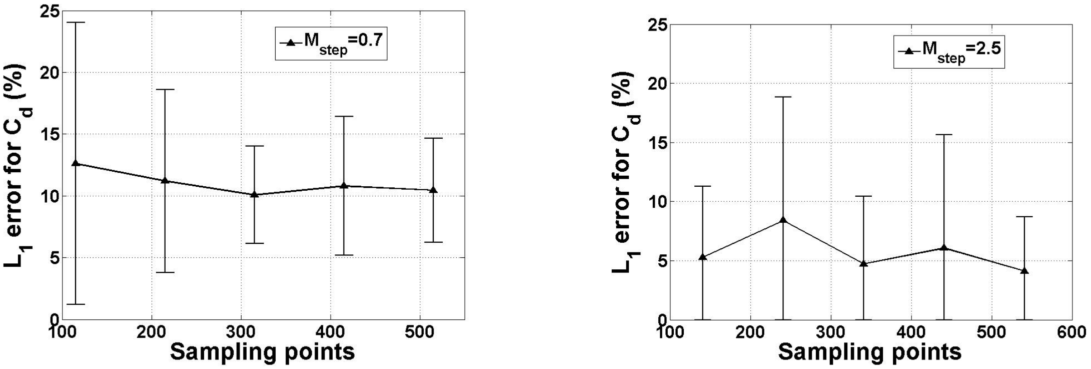
(a) Subsonic cases

(b) Supersonic cases

Figure 5.4: Drag coefficient errors as function number of sampling points, results Set A, ROM C

if points end up being bunched relatively closely in the parameter space, this may cause artificial undulations in the kriging surface, which may result in a degradation of accuracy. Finally, the Method of Segments is used to calculate the locations of the additional sampling points, whereas CFD is used for the final correction factor value calculations. Due to the inherent approximation errors of the MoS simplified model, it is possible that the additional sampling points are not placed at the location of maximum error on the kriging surface as would have been calculated by a higher-fidelity model and thus not in the optimal location for error reduction. However, overall, the spread of mean error values seen is relatively small. For a single value of M_{step}, the maximum error range seen over all numbers of sampling points tested is just over 2.5 percentage points for the subsonic cases and just over 4.5 percentage points for the supersonic cases.

The preceding results have explored the ROM methodology in general, with all correction factor values found using direct CFD simulations. Now, it is necessary to investigate the accuracy of the Method of Segments for the Mach ranges considered here in order to characterize the effectiveness of using that model as an error estimation methodology to calculate the locations of additional kriging surface sampling



(a) Subsonic, M_{step}=0.7

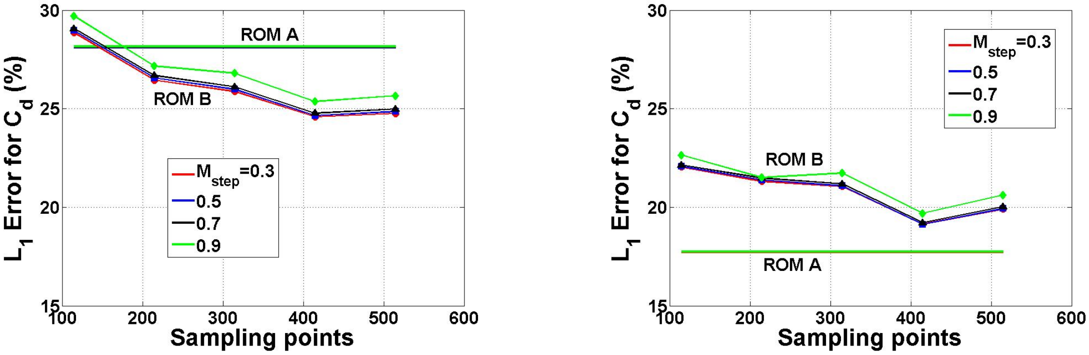
(b) Supersonic, M_{step}=2.5

Figure 5.5: Mean errors and standard deviations for specific M_{step} cases

points as well as for ROM construction itself. In Chapter 4, two separate MoS-based ROMs are constructed; ROM A is based on the direct calculation of correction factor values at each time step throughout a simulation, while ROM B uses a pre-determined number of correction factor sampling points, each calculated using MoS, to construct a correction factor kriging surface. ROM B is the same as ROM C, the results for which are shown above, except that MoS has replaced CFD as the correction factor calculation method. These ROMs are summarized in Table 4.5 in Chapter 4.

Figures 5.6 and 5.7 show the errors for the two Method of Segments ROMs over a range of correction factor kriging surface sampling points. In the plots, note that ROM A is constant over the range; this is due to the fact that the correction factor is directly calculated at each point, and therefore the number of sampling points quantity does not apply to it. First consider the subsonic Mach range. Figure 5.6(a) shows the errors over all 50 test cases, from which two main items become evident. The first item is that the errors are high in this case, over 24% for all cases tested. This suggests that the accuracy of the Method of Segments degrades with decreasing flow velocity in the subsonic regime. Second, the curious result of ROM B having lower errors than ROM A for the majority of data points is seen. Since ROM A is effectively the limit of ROM B as the number of data points goes to infinity, this is unexpected. However, when looking at Fig. 5.6(b), a different picture emerges. All ROM A results have smaller errors than ROM B results, meaning that the largest two errors over all tests for ROM A are very large and have a significant effect on both the mean error over all the runs as well as the comparative errors between ROMs A and B. However, due to the relatively large ROM-CFD errors seen here, it does not appear that the Method of Segments is a viable tool for the construction of the ROM itself for the subsonic regime, though it is still used as a rough error estimation tool for the generation of kriging surface sampling point parameters.

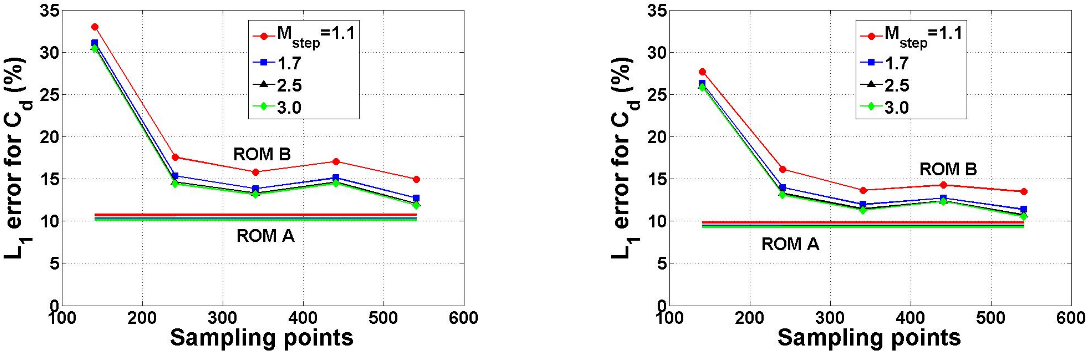
The results for the supersonic test cases shown in Fig. 5.7 are strikingly different. First, the increase from the initial number of sampling points (100+) to the second data point at 200+ points results in a drop in mean error for ROM B by roughly 50%, down to just around 15% depending on the specific value of M_{step}. Next, the values for ROM A are less than those for ROM B over all numbers of sampling points, which is to be expected. Also, the omission of the largest error value, while still obviously resulting in the expected decrease in mean error, does not affect the agreement trends between ROMs A and B as in the subsonic case. Overall, the errors for the supersonic Mach range are much smaller than the those for the subsonic range, with ROM A errors calculated to be just around 10% for each of the values of M_{step}; this is likely due to the decreased reduced frequencies of the supersonic regime as compared to the subsonic regime. While still larger than the errors computed using the direct CFD correction factor computations of ROM C, the results show the potential applicability of the Method of Segments for the generation of ROMs in the supersonic Mach regime as well as the usefulness in both error estimation and correction factor kriging surface sampling point parameter determination in this regime.



(a) All tests

(b) Largest two errors omitted

Figure 5.6: Method of Segments subsonic ROM C_d errors, results Set A



(a) All tests

(b) Largest error omitted

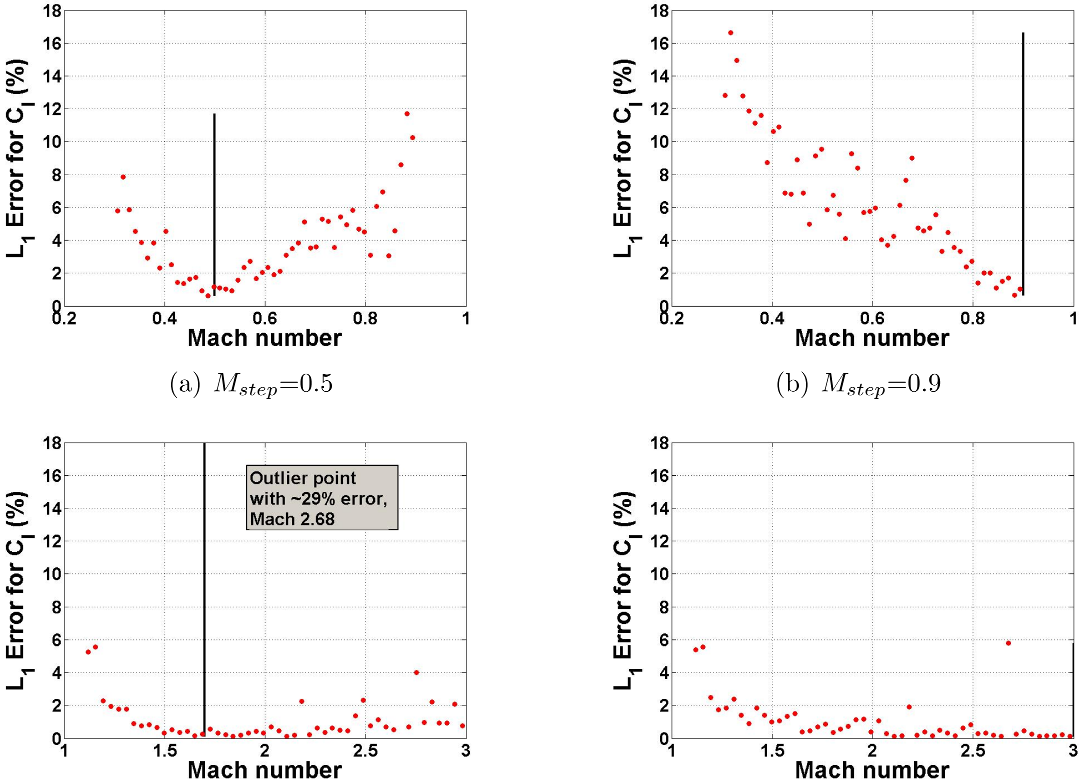
Figure 5.7: Method of Segments supersonic ROM C_d errors, results Set A

5.2.2 Lift and Moment Coefficient Results

The lift and moment coefficient results are presented together in the same section due to the fact that the overall trends are generally similar. The first series of results, displayed in Figs. 5.8 (C_l) and 5.9 (C_m), corresponds to the drag results of Figs. 5.1 and 5.2, showing the ROM C errors as a function of Mach number for specific values of M_{step}. Unlike the drag coefficient results, in which the errors generally decreased with Mach increasing number (M_{sim}), the errors for the lift and moment coefficients tend to be lower at Mach numbers closest to M_{step}, suggesting that these two coefficients show a larger degree of sensitivity to changes in M_{step} than the drag coefficient. The errors are also smaller than those for the drag coefficient across all Mach numbers, under 15% for the majority of the subsonic test cases and 5% for the majority of the supersonic test cases for both coefficients.

To quantitatively evaluate how the errors change with M_{step}, consider Figs. 5.10 and 5.11, which show the mean and standard deviation of errors over all test cases for each value of M_{step} by averaging the individual data points of Figs. 5.8 and 5.9, respectively.

The results show that the error values for the subsonic tests are lowest for M_{step} values toward the middle of the parameter space (i.e. M_{step}=0.5,0.7). This makes sense because the middle values of M_{step} are closer to more of the values of M_{sim} than the edge values of M_{step} simply by function of being in the middle of the parameter space. The previous plots show a sensitivity to M_{step}-M_{sim} separation for the lift and moment coefficient, so the increased distance away from more specific values of M_{sim} results in larger mean errors for the edge M_{step} values of 0.3 and 0.9. However, the supersonic results remain relatively constant across M_{step} values after a sharp drop from M_{step}=1.1 to M_{step}=1.7. This suggests an improvement in ROM accuracy as flow velocity increases, a phenomenon also that matches with the corresponding drag coefficient results.



(a) M_{step}=0.5

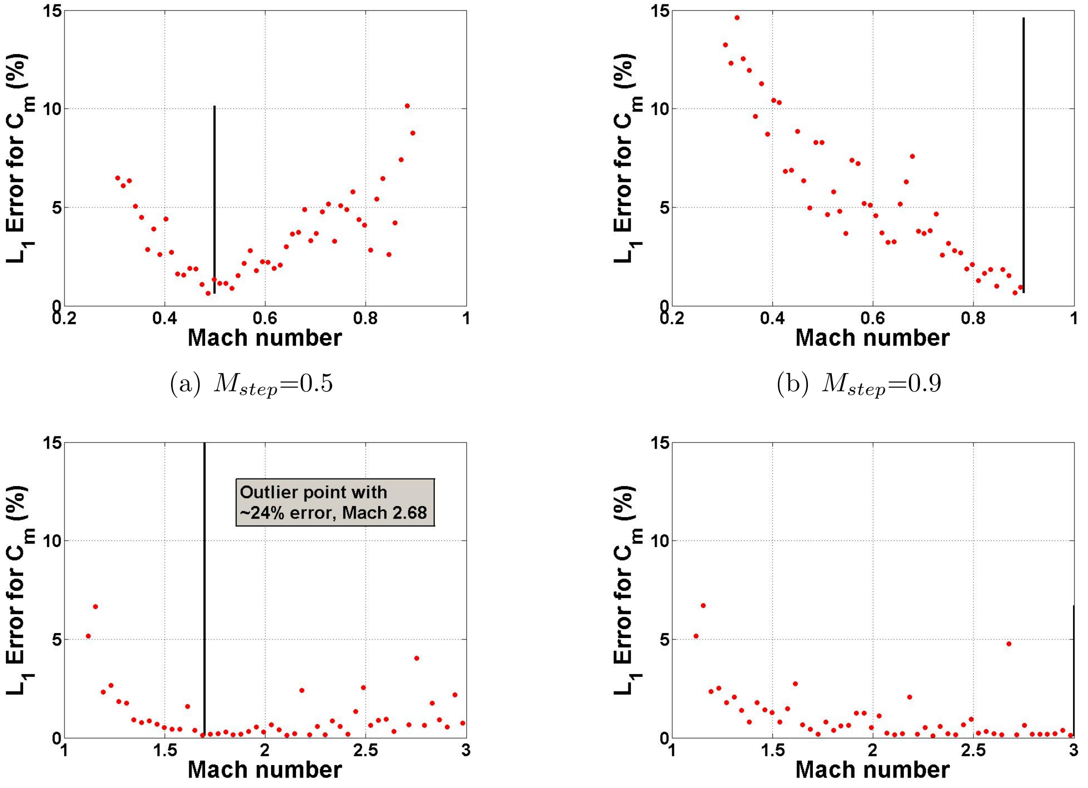
(b) M_{step}=0.9

Outlier point with ~29% error, Mach 2.68

(c) M_{step}=1.7

(d) M_{step}=3.0

Figure 5.8: Lift coefficient errors for as function of Mach number



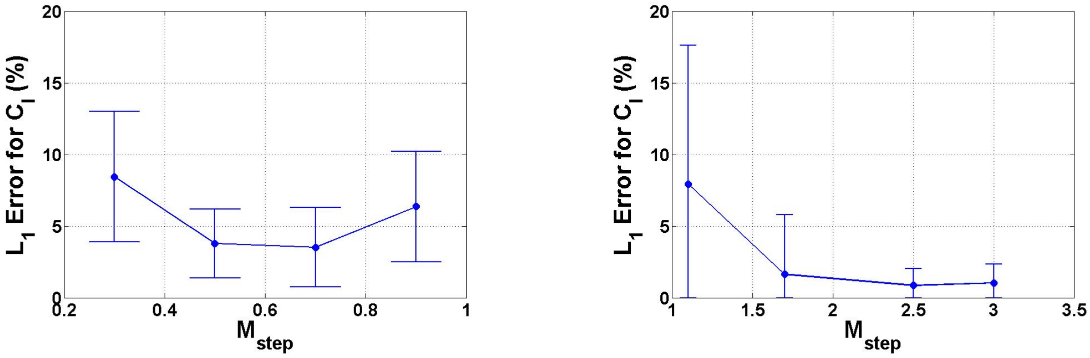
(a) M_{step}=0.5

(b) M_{step}=0.9

(c) M_{step}=1.7

(d) M_{step}=3.0

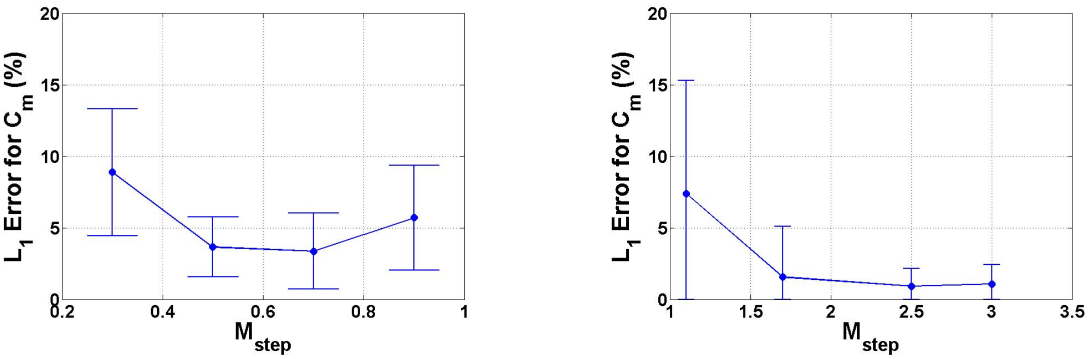
Figure 5.9: Moment coefficient errors for as function of Mach number



(a) Subsonic Mach range

(b) Supersonic Mach range

Figure 5.10: C_l errors for subsonic test cases as function M_{step}, results Set A



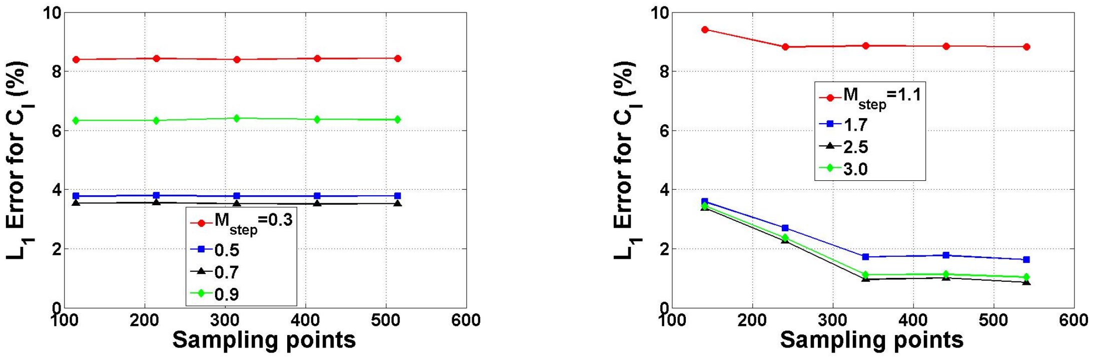
(a) Subsonic Mach range

(b) Supersonic Mach range

Figure 5.11: C_m errors for subsonic test cases as function M_{step}, results Set A

Next, the effect of the number of kriging surface sampling points needs to be evaluated. Figs. 5.12 and 5.13 show the ROM C errors for the lift and moment coefficients, respectively, over all Set A test cases and a range of sampling point numbers. The subsonic results, displayed in Figs. 5.12(a) and 5.13(a), show the errors to be constant over the range of sampling points, suggesting that the initial amount is sufficient to model the system in this case. The supersonic results, displayed in Figs. 5.12(b) and 5.13(b), show a slight decrease in errors for most of the M_{step} values up through just over 300 sampling points before leveling off. However, the total error percentage point decrease of just over 2 is relatively small, demonstrating that the necessary number of sampling points has been reached in these cases as well.

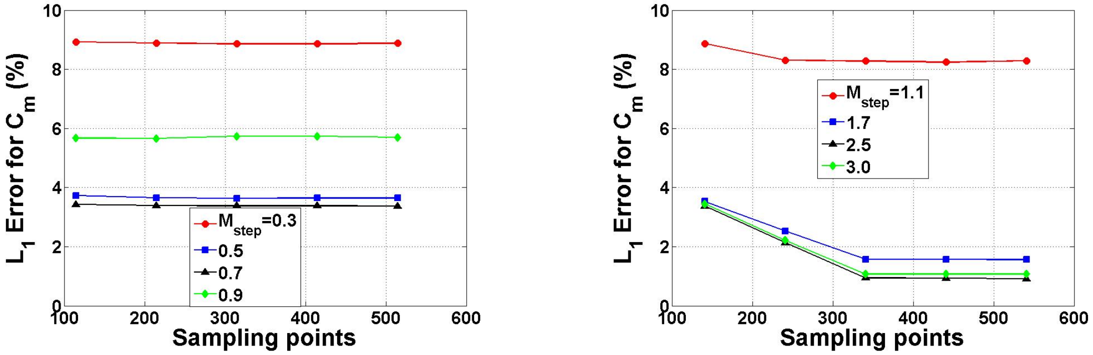
The last items in Set A to explore are the ROM-CFD comparisons from the two Method of Segments-based ROMs described in Section 5.2.1. Figs. 5.14 and 5.15 show the lift and moment coefficient error results, respectively, for each of ROMs A and B over a range of kriging surface sampling points. Note again that ROM A errors are constant due to the fact that it does not use kriging surface sampling points, instead directly calculating the correction factor value at each time step throughout the simulation. The first item to note is that the subsonic errors for both coefficients (Figs. 5.14(a) and 5.15(a)) are significantly lower than the corresponding drag coef-



(a) Subsonic Mach range

(b) Supersonic Mach range

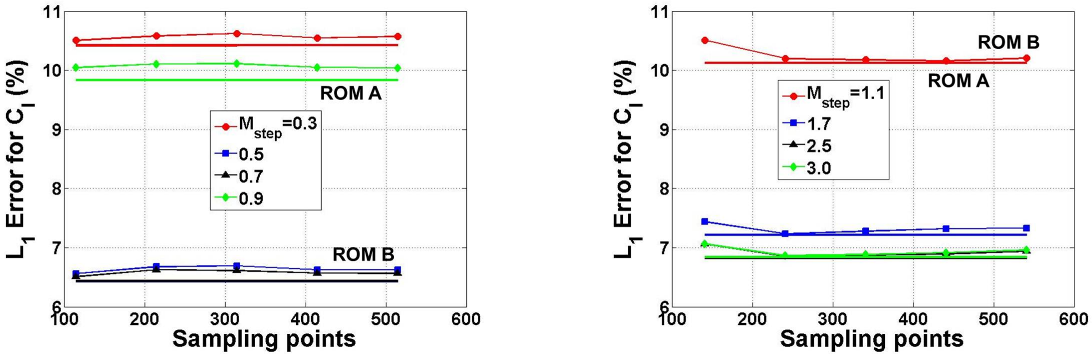
Figure 5.12: C_l errors as function number of sampling points, results Set A



(a) Subsonic Mach range

(b) Supersonic Mach range

Figure 5.13: C_m errors as function number of sampling points, results Set A



(a) Subsonic Mach range

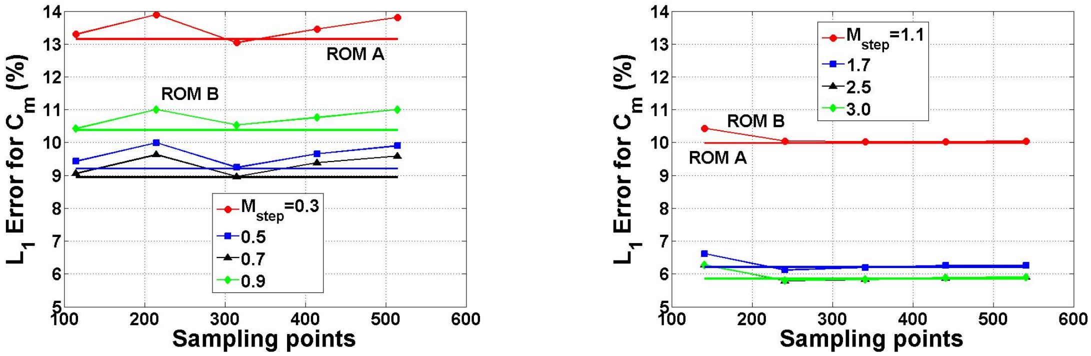
(b) Supersonic Mach range

Figure 5.14: Method of Segments ROM lift coefficient errors, results Set A

ficient errors, and the supersonic errors (Figs.5.14(b) and 5.15(b)) are lower as well. Second, the errors are generally close to being constant across the range of sampling points, again showing that the necessary amount of points has been reached. Finally, as with the drag coefficient results, there are several unexpected cases in which the ROM *B* errors are less than the ROM *A* errors, right around the 200+ sampling point value for the lift coefficient. However, in these cases, the difference between the error values is so small (less than 0.1 percentage point) that this could be due simply to noise in the kriging surface happening to slightly improve the results or a host of other rounding issues. Unlike the drag coefficient results, the omission of the largest two errors does not change the error comparison trends between the ROMs.

5.3 Set B Results

The second set of results to be examined are those from Set B, in which simulations of 25 sets of modal parameters are conducted at each Mach number in a range of Mach numbers. The parameters are selected using Latin hypercube sampling from the same parameter spaces as used in Set A, which are shown in Table 5.1. The main goal of these tests is to get a better picture of how the errors of the methodology are affected by Mach number; results showing how the errors are affected by number



(a) Subsonic Mach range

(b) Supersonic Mach range

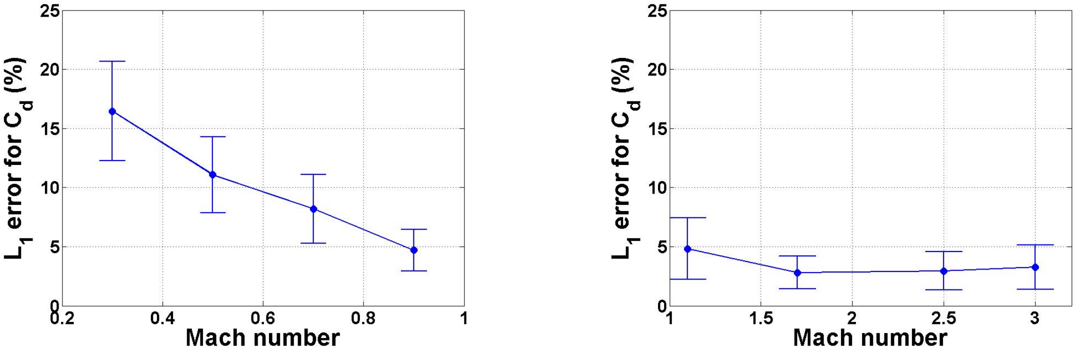
Figure 5.15: Method of Segments ROM moment coefficient errors, results Set A

of kriging surface sampling points are presented as well. All results shown in this section are for ROM C, in which the correction factor is computed through direct CFD simulations along with kriging surfaces. Also, except where otherwise noted, the maximum number of 500+ sampling points are used for the correction factor kriging surfaces.

5.3.1 Drag Coefficient Results

The first plot to consider is Fig. 5.16, which displays the mean drag coefficient errors over all cases in which the step response Mach number M_{step} is equal to the simulation Mach number M_{sim}. Note that each data point on the plot is the mean of the 25 sinusoidal test cases conducted at that particular Mach number, and the error bars represent one standard deviation.

Several items become apparent by looking at the plots. First, in the subsonic Mach range, the errors decrease significantly as Mach number increases, falling from just over 16% at Mach 0.3 to just under 5% for Mach 0.9. Along with the mean values, the spread in error values decreases as well, a fact which can be seen by the reduction in standard deviation values. Next, after a slight decrease in error from the Mach 1.1 value, the supersonic Mach range errors are relatively constant, hovering at



(a) Subsonic Mach range

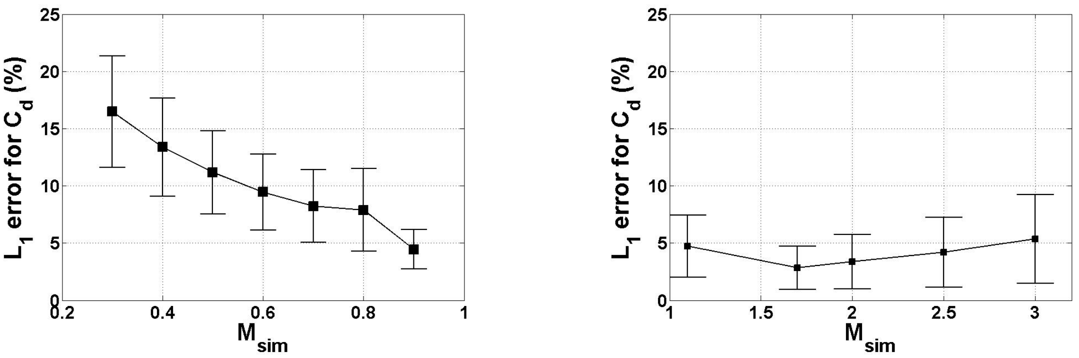
(b) Supersonic Mach range

Figure 5.16: Drag coefficient errors for M_{sim}=M_{step}, results Set B

just around 3%. These values are in general smaller than the subsonic values, again pointing to an increase in ROM accuracy with increasing Mach number.

As has been discussed previously, the values for M_{sim} in general would not be expected to be equal to M_{step}. Because of this, it is necessary to investigate the ROM errors for each of the simulations at a particular value of M_{sim} that have been calculated using a range of M_{step} values. Figure 5.17 displays the mean errors and standard deviations for each value of M_{sim} for ROMs which have been calculated at each of the values of M_{step}. For example, consider the value M_{sim}=0.5. To calculate the corresponding data point in Fig. 5.17(a), ROMs need to be computed for each of the 25 sinusoidal simulations conducted at M_{sim}=0.5 for M_{step}=[0.3 0.5 0.7 0.9], which gives a total of 25×4=100 ROM-CFD comparisons. The errors from these 100 comparisons are then used to compute the mean value and standard deviation data in the figure.

In general, the trends seen in Fig. 5.17 are similar to those observed in Fig. 5.16. The errors decrease with Mach number in the subsonic range, and the supersonic errors are smaller than the subsonic errors. A slight increase in error with Mach number is seen in the higher Mach numbers of the supersonic range, even climbing higher than the M_{sim}=1.1 error, but the overall error values remain small, with mean



(a) Subsonic Mach range

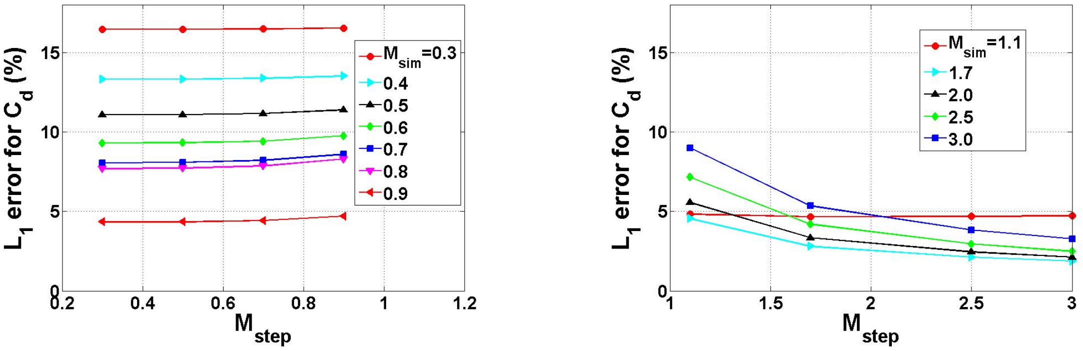
(b) Supersonic Mach range

Figure 5.17: Drag coefficient errors for M_{sim} using all values of M_{step}, results Set B

values under 6% in all cases.

The last item to visualize for the drag coefficient errors for ROM C is how the errors change with M_{step}. Consider Fig. 5.18, which shows the errors for each value of M_{sim} as a function of M_{step}. Each data point in this plot is computed by calculating the mean error over all 25 runs conducted at a specified value of M_{sim} using a constant value of M_{step}. For the subsonic Mach range shown in Fig. 5.18(a), the errors remain fairly constant for each value of M_{sim} over the range of M_{step}. Also, as seen previously, the errors decrease with Mach number, with all data points for M_{sim}=0.9 having the smallest error at each M_{step} value and all data points at M_{sim}=0.3 having the largest.

However, the supersonic results in Fig. 5.18(b) show different trends. First, all values of M_{sim} follow the same progression of decreasing error as M_{step} increases except for M_{sim}=1.1, which remains fairly constant over the range. Next, the greatest value of M_{sim}, 3.0, has the largest errors in general over the Mach range, except for the two highest values of M_{step} tested, where the constant value of the M_{sim}=1.1 error is the largest. In general, for a specific value of M_{step}, the error decreases as M_{sim} is reduced, which is the opposite of what is seen in Fig. 5.18(a). This result may be a function of how far apart M_{sim} and M_{step} become in some circumstances. The spread of Mach values in the supersonic range tested is larger than the spread in the



(a) Subsonic Mach range

(b) Supersonic Mach range

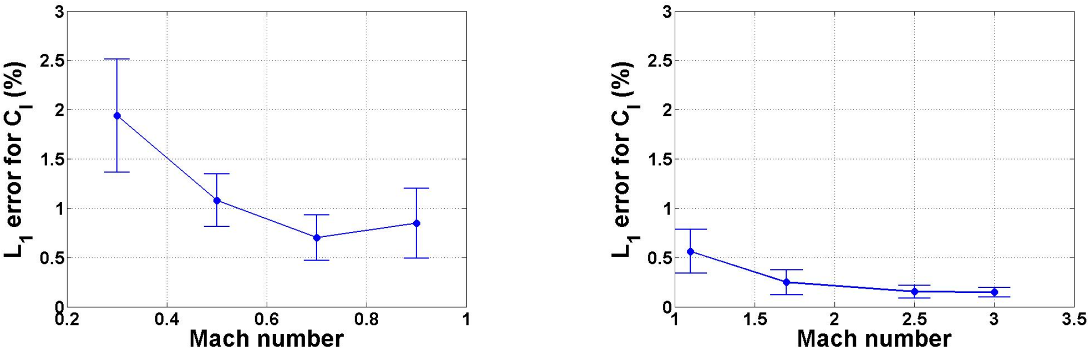
Figure 5.18: Drag coefficient errors as functions of M_{step}, results Set B

subsonic range. For example, when an M_{step} value of 1.1 is used to calculate the ROM for an M_{sim} value of 3, the difference between the two is 1.9 Mach number units, which is over twice the entire span of the subsonic Mach range. This would also explain the different trend seen for M_{sim}=1.1. As M_{step} increases, the M_{sim}-M_{step} gap is constantly increasing. For all other cases, as M_{step} increases, the gap is either decreasing or decreasing at first before increasing. This suggests the expected result that the errors will eventually increase as M_{step} moves away from M_{sim}.

5.3.2 Lift Coefficient Results

The next series of results show the lift coefficient results for these Set B test cases. Consider first Figs. 5.19 and 5.20, which shows the lift coefficient errors for test cases in which M_{sim}=M_{step}; it corresponds to Fig. 5.16, which shows the drag coefficient results over the same cases. The main item to note is that the errors for the lift coefficients are much smaller than those for the drag, with a maximum mean error in the subsonic range of under 3% and in the supersonic range of under 1.5%. Also, the supersonic errors are smaller than the subsonic errors, continuing the trend seen from the drag coefficient results.

Next, consider Fig. 5.20, which shows the lift coefficient ROM errors in which the



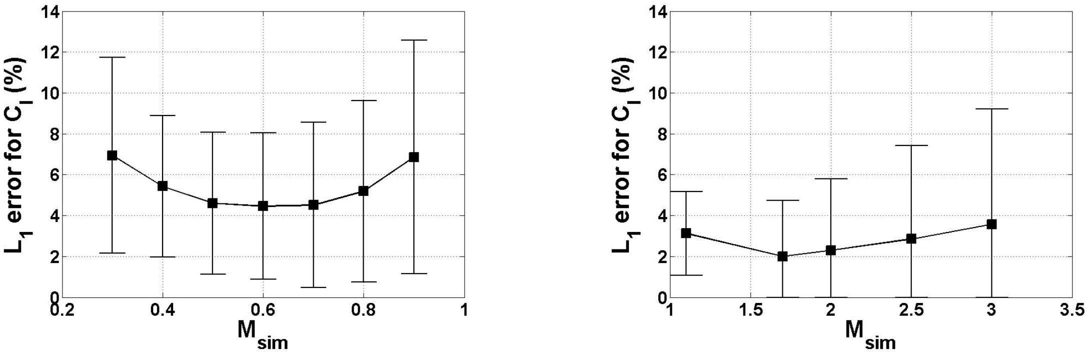
(a) Subsonic Mach range

(b) Supersonic Mach range

Figure 5.19: Lift coefficient errors for M_{sim}=M_{step}, results Set B

ROMs at each value of M_{sim} have been calculated using each of the different values of M_{step}. This corresponds to the drag coefficient results shown in Fig. 5.17. Unlike the drag coefficient results, the lift coefficient errors do show a significant increase over the errors calculated using only M_{sim}=M_{step}, increasing to maximum mean error of around 7% for the subsonic cases and 3% for the supersonic cases. Taking a closer look at Fig. 5.20(a), the errors are largest for the extreme values of M_{sim} and smallest for those in the middle of the range. This result makes sense when looking at it in terms of the gap between M_{sim} and M_{step}. As has been shown previously, the ROM errors will eventually increase as M_{step} continues to move away from M_{sim}; thus, the errors in general will increase with increasing M_{sim}-M_{step} gap. The values of M_{sim} on the edges of the Mach range will have the largest mean values of this M_{sim}-M_{step} gap, as some of the ROMs will be calculated using an M_{step} value at or near the other edge of the range. However, values in the middle will have lower M_{sim}-M_{step} gap values due to the central location. As a result of this, the M_{sim} values in the middle of the Mach range in the plot have the lowest errors.

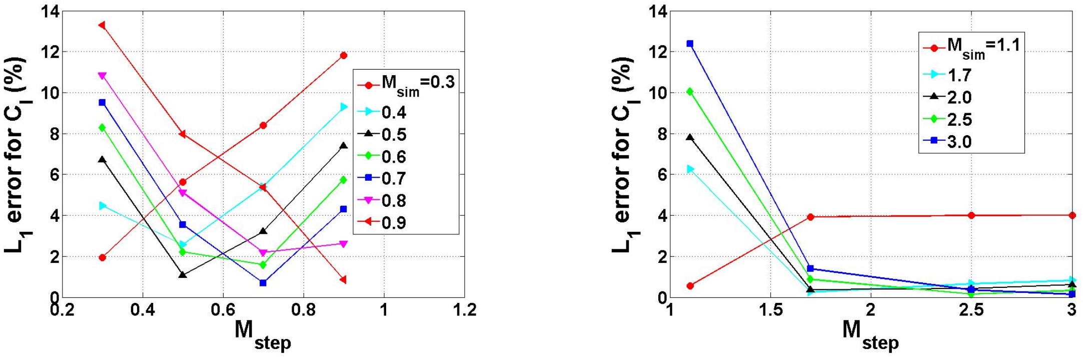
This trend of lift coefficient errors being a function of M_{sim}-M_{step} gap can be further seen by examining Fig. 5.21, which is the lift coefficient plot corresponding to the drag results displayed in Fig. 5.18. As before, each data point corresponds to the



(a) Subsonic Mach range

(b) Supersonic Mach range

Figure 5.20: Lift coefficient errors for M_{sim} using all values of M_{step}, results Set B



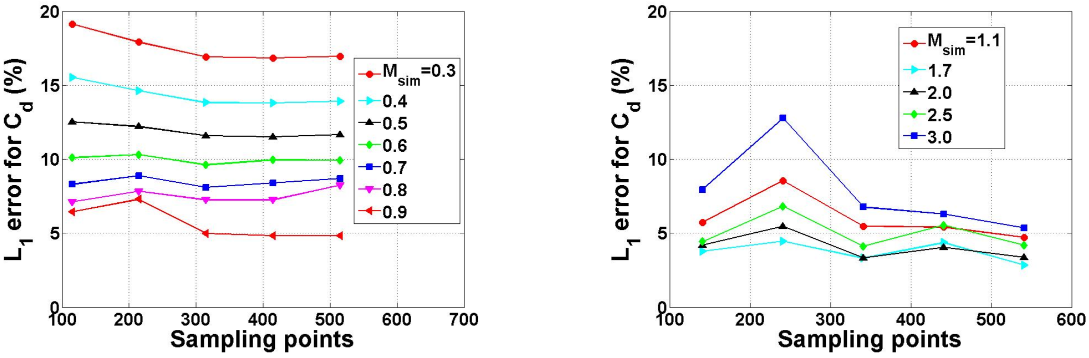
(a) Subsonic Mach range

(b) Supersonic Mach range

Figure 5.21: Lift coefficient errors as functions of M_{step}, results Set B

mean error of all test cases at a specific value of M_{sim} with a ROM constructed at a certain value of M_{step}. In Fig. 5.21(a), the data point with the least error for each of the values of M_{sim} is either found at M_{step}=M_{sim} or, for cases in which an exact corresponding value of M_{step} is not calculated, very close to that value. The trend continues in the supersonic Mach range for all points in Fig. 5.21(b) as well.

The trends and error values of the moment coefficient reflect those of the lift coefficient for these test cases and thus are not presented here.



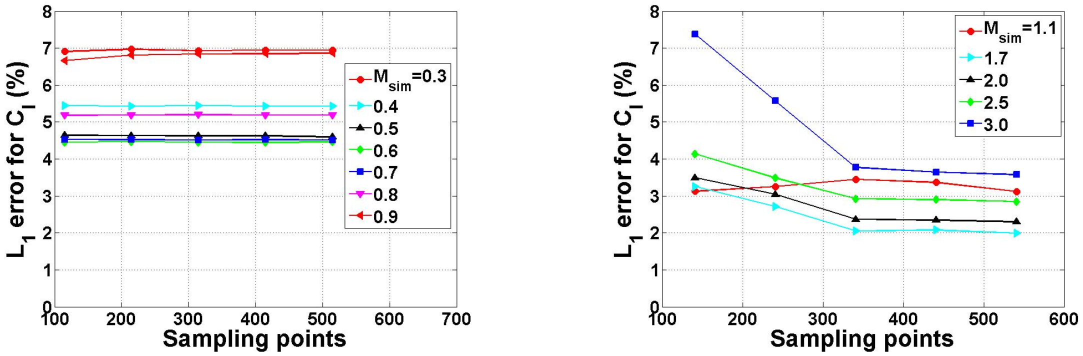
(a) Subsonic (b) Supersonic

Figure 5.22: C_d errors as function of number of sampling points, results Set B

5.3.3 Effect of Kriging Surface Sampling Points

The final item to investigate for the Set B results is how the errors change with the number of sampling points. Consider Fig. 5.22, which shows the drag coefficient errors for each value of M_{sim} in both the subsonic (Fig. 5.22(a)) and supersonic (Fig. 5.22(b)) ranges as functions of the number of sampling points used in correction factor kriging surface construction. For this plot, the ROM-CFD comparisons at each value of M_{sim} are made for each value of M_{step}. For a point of reference, the right-most values on the plots, corresponding to over 500 sampling points being used, are the same values as plotted in Fig. 5.17.

The data do not show any significant decrease in errors from the fewest to most sampling points for the subsonic range. Slight initial decreases are seen for the values of $M_{sim} \leq 0.5$ as the number of sampling points increases up to around 300 before leveling off. For the supersonic range, slight error decreases again are seen over the sampling point range. An uptick in the errors is seen at just over 200 sampling points. One possible explanation for this is that the kriging fit for that particular set of parameters caused some type of undulations in the kriging surface that is not seen in reality. However, despite that uptick, the errors for the supersonic range remain smaller than the subsonic range in general.



(a) Subsonic

(b) Supersonic

Figure 5.23: C_l errors as function of number of sampling points, results Set B

Next, consider Fig. 5.23, which shows the lift coefficient results for the same cases as shown in Fig. 5.22. For the subsonic range, the errors are essentially constant across the sampling point values considered. Slight initial error decreases are seen in the supersonic cases until around 300 sampling points before the errors level off; the general trends seen for the lift coefficient reflect those observed for the drag coefficient for this case in each of the supersonic and subsonic Mach ranges.

5.4 Example Case Over Entire Mach Number Range

It is now important to show how this methodology can be practically applied to a specific example test case spanning a wide range of Mach numbers. This section evaluates the ROM's performance over the entire span of Mach numbers that have been tested for the AGARD 445.6 wing, from Mach 0.3 to Mach 3. In doing so, the transonic ROM of Chapter 4 has been integrated with these test cases. In practice, one will not have access to an unlimited number of step responses to use for each different value of M_{sim}. Thus, some method must be chosen for how to best deal with values of M_{sim} that fall somewhere between the different values of M_{step}.

For this section, suppose that the values of M_{step} found in the first row of Table 5.2 are the only values for which step responses are available. The goal is to compute

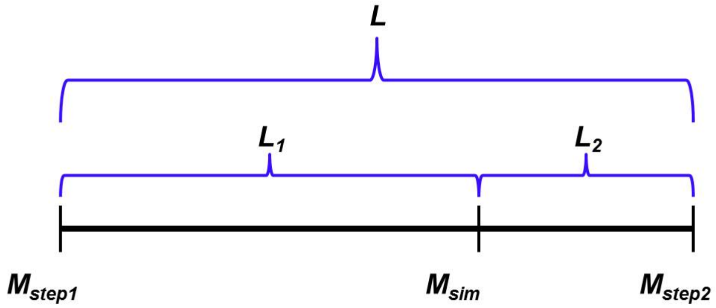
Table 5.2: **Mach values for example case**

Parameter	**Values**
M_{step}	0.3, 0.9, 1.05, 1.1, 3.0
M_{sim}	0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.7, 2.0, 2.5, 3.0

ROM results having the least error possible for the M_{sim} values found in the second row of Table 5.2. The test cases used here are the same as those used in results Set B, where, for each value of M_{sim}, 25 separate simulations are conducted. These 25 test cases have the same modal amplitudes and oscillations for each value of M_{sim} (the parameter space is shown in Table 5.1), so the only variation among the tests at different Mach numbers is M_{sim}. Each data point in the following results represents the mean L_1 error value over all 25 cases.

For each test case, the ROM is calculated in two separate methods, shown graphically in Fig. 5.24. Method 1 uses a weighted average of the two ROM responses computed using the next higher and next lower M_{step} values. Method 2 simply uses the ROM response computed from the closest value of M_{step} to that particular value of M_{sim}. Note that, for cases where M_{step}=M_{sim}, the methods are identical. Also, all ROMs in this section use kriging surface sampling points calculated directly from CFD simulations (ROM C). Figs. 5.25 and 5.26 show block diagrams for Method 1 and Method 2, respectively, outlining how they would be implemented in a full simulation framework. For Method 1, the weighted average of two ROM responses is passed back to the simulation framework, while for Method 2, the response from the closest M_{step} is passed back. Note that among the main differences in the methods is the fact that two ROM computations are necessary for Method 1, as individual ROM responses must be calculated at each of two M_{step} values, while only one such calculation is necessary for Method 2.

The lift, drag, and moment coefficient results of the tests are shown in Figs. 5.27, 5.28,



$$y_{Method1} = \left(\frac{L_2}{L}\right) y_{ROM,M_{step1}} + \left(\frac{L_1}{L}\right) y_{ROM,M_{step2}}$$

$$y_{Method2} = \begin{cases} y_{ROM,M_{step1}} & L_1 < L_2 \\ y_{ROM,M_{step2}} & L_1 > L_2 \end{cases}$$

Figure 5.24: Diagram for ROM calculation methods

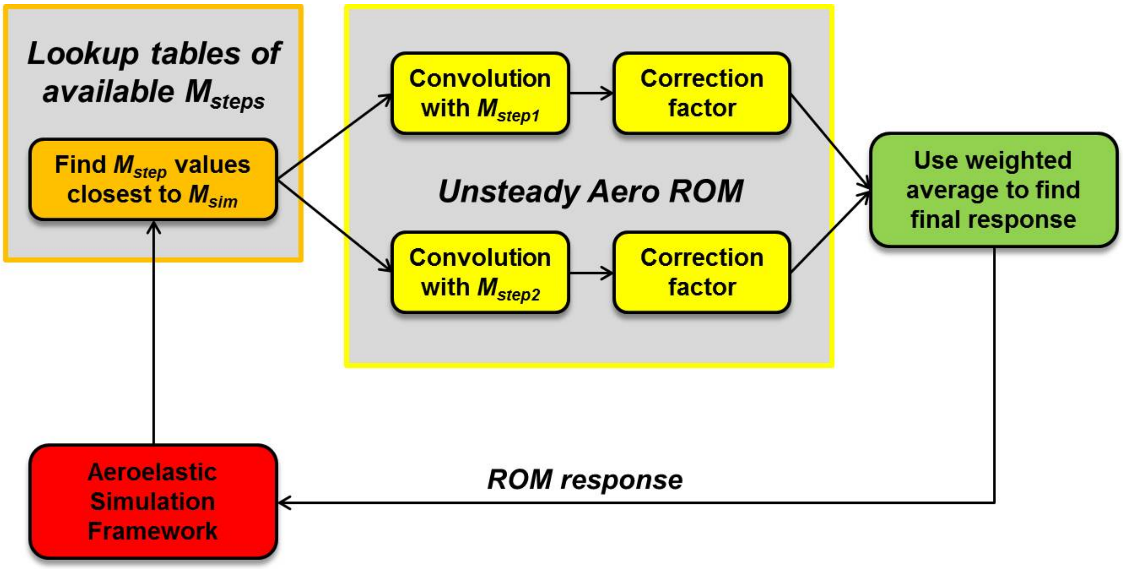

Lookup tables of available Msteps
Find Mstep values closest to Msim
Convolution with Mstep1
Correction factor
Unsteady Aero ROM
Convolution with Mstep2
Correction factor
Use weighted average to find final response
ROM response
Aeroelastic Simulation Framework

Figure 5.25: Block diagram for Method 1 implementation

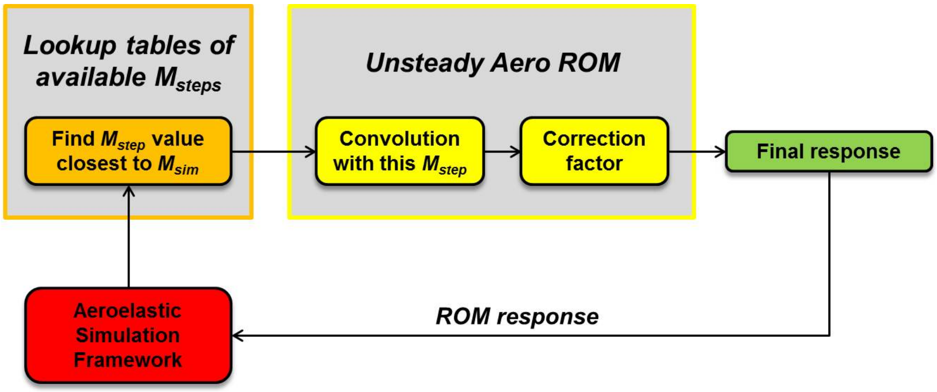
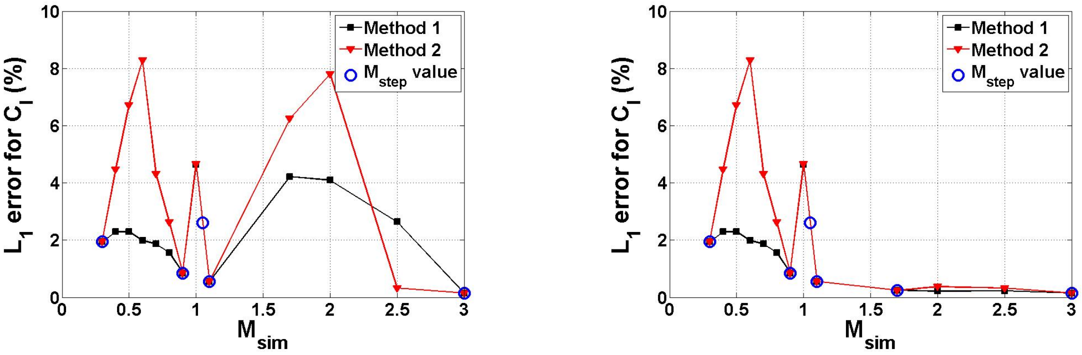

Lookup tables of available M_{steps}
Find M_{step} value closest to M_{sim}
Unsteady Aero ROM
Convolution with this M_{step}
Correction factor
Final response
ROM response
Aeroelastic Simulation Framework

Figure 5.26: Block diagram for Method 2 implementation

and 5.29, respectively, from which several important points can be gathered. First, the lift and moment results show very similar trends in that the errors generally increase as M_{sim} moves away from M_{step}. The exception to this is the Method 1 errors in the subsonic Mach range, which remain relatively steady between M_{step}=0.3 and M_{step}=0.9. To see what happens when another value of M_{step} is added, consider Fig. 5.27(b), in which M_{step}=1.7 has been added to the list of available step responses (the results in Fig. 5.27(a) contain only the M_{step} values in Table 5.2). As can be seen, this greatly reduces the error in the supersonic Mach range for each method. In terms of the methods, Method 1 (weighted averages) seems to be equal to or better than Method 2 in most situations. This is most strikingly seen in the subsonic range around Mach 0.6, where the Method 2 error is just above 8% for the lift coefficient, while the Method 1 error is down to just over 2%; the corresponding moment coefficient errors at the same location for each of the methods are around 9% and 3%, respectively. Next, an error spike can be seen for both the lift and moment coefficients around Mach 1, in the transonic regime. Even though the errors do spike, they are still relatively small at just over 5%. Also, given the nonlinearities present within the transonic regime, a good recommendation would be to have as many values of M_{step} as feasible within this region.

The drag coefficient results differ in trends than the other two coefficients. Each of the two methods performs similarly throughout the Mach range, and less sensitivity is seen to $M_{step} - M_{sim}$ difference. The addition of another M_{step} value of 1.7 (Fig. 5.28(b)) does slightly improve the errors in the supersonic Mach range, though errors were already low to begin with ($\sim 5\%$). The results also show that the ROM generally improves in accuracy with Mach number, as the Mach 0.3 results show the highest errors. Finally, as with the lift and moment coefficient results, an error spike is seen around Mach 1 in the transonic region, though errors here remain well below the errors at the low end of the Mach number parameter space around $M = 0.3$.

Overall, both methods performed reasonably well, though Method 1 is shown to have generally smaller errors overall for the lift and moment coefficients. For the drag, both methods perform very similarly. Thus, the recommendation for the method of constructing ROMs for situations where $M_{sim} \neq M_{step}$ is to use Method 1. However, one potential advantage of Method 2 over Method 1 is computational expense. As mentioned previously, whereas Method 2 only requires the computation of one ROM response, Method 1 requires the computation of two responses in order to find the weighted average. In general, the ROM is computationally cheap to compute, and thus this may not be a significant issue. However, if a situation arises in which a very large number of ROM responses will need to be computed, the increased efficiency of Method 2 may need to be considered. Finally, a brief word must be given as to how these methods would extend to higher-dimension flight condition parameter spaces. For example, what happens if altitude is considered as well? In this case, Method 1 could extend to be the result of a weighted function of the responses from the nearest pre-determined number of step response parameter values within the multi-dimensional parameter space; further work would be required to identify the optimal method for doing so. For Method 2, one could still find the nearest set of step response parameters value by calculating the Euclidean distances from the simulation parameters to the various sets of nearby step response parameters.



(a) M_{step} values from Table 5.2

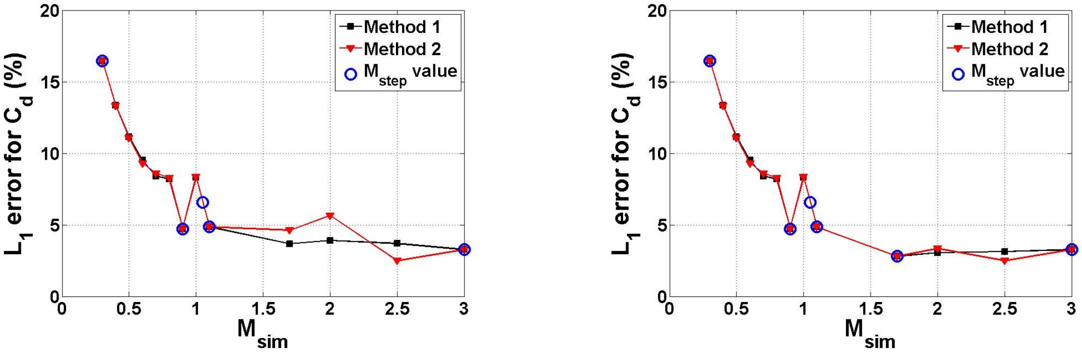
(b) Addition of M_{step}=1.7

Figure 5.27: Lift coefficient results over entire Mach range

A note needs to be mentioned about the potential applicability of the ROM to the extreme low end of the subsonic regime, below Mach 0.3. One factor that would determine how low the ROM should go would be the takeoff speed of the vehicle. While on the ground, vibrations from the wheels, ground effects, special takeoff configurations, and other factors would all influence the dynamics and aerodynamics of the vehicle, thus complicating the analysis. For a point of reference, the takeoff speed of the SR-71 Blackbird is right around 200 knots,[103] corresponding to a Mach number of 0.3 assuming 70°F ambient temperature. If the ROM is desired to be applied for these low Mach numbers, the ROM errors would be expected to increase with decreasing Mach number.

5.5 Computational Savings

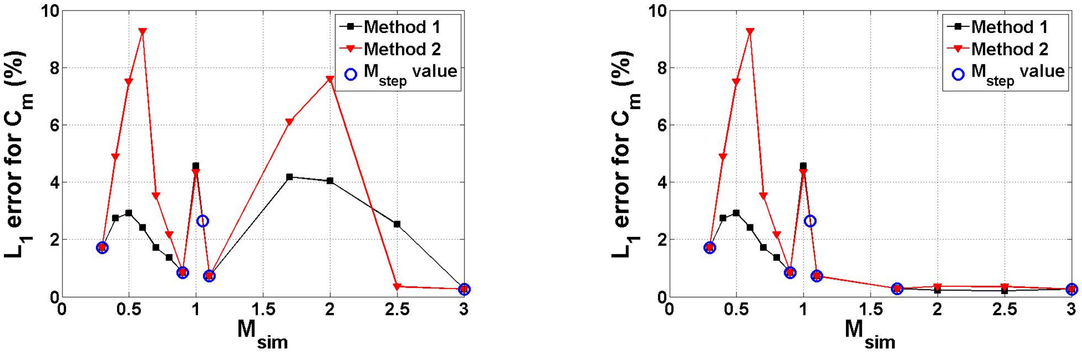
Finally, a quantification of the computational time savings needs to be given between the ROM and full-order CFD simulations. The unsteady sinusoidal CFD simulations were computed using 16 total processors on the NASA Pleiades Supercomputer consisting of Intel Xeon E5-2670, X5670, and X5675 processors with at least



(a) M_{step} values from Table 5.2

(b) Addition of M_{step}=1.7

Figure 5.28: Drag coefficient results over entire Mach range



(a) M_{step} values from Table 5.2

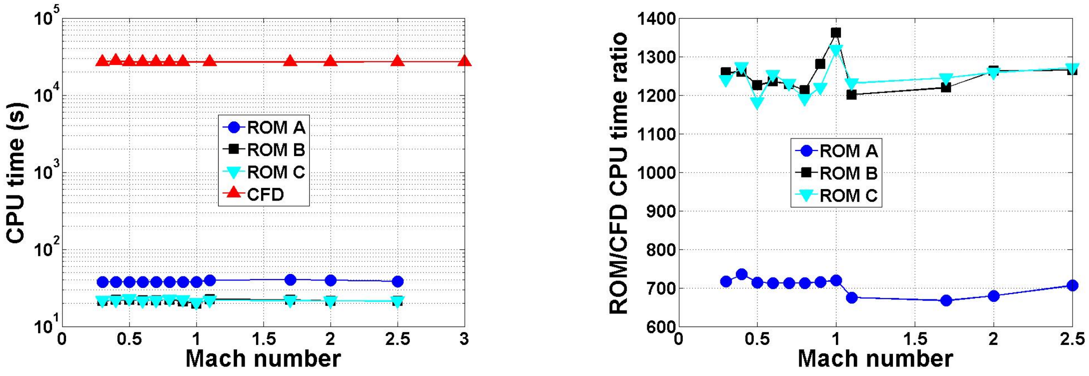
(b) Addition of M_{step}=1.7

Figure 5.29: Moment coefficient results over entire Mach range

2.6 GHz processor speed[a]. Conversely, the ROM simulations for the same cases were conducted on a desktop computer with a 2.66-GHz Intel Core 2 CPU with 3 GB of RAM. Figure 5.30(a) shows a logarithmic plot of the comparative times between the ROM and CFD runs over a range of Mach numbers; note that the ROM simulation time is per coefficient, and the CFD CPU time is the total time of all processors. Each data point represents the mean value over the same 25 test cases conducted in the previous section at that particular Mach number. Figure 5.30(b) shows the ratio between the CFD time and the times for each of the ROMs in order to quantify the improvements. The results show a roughly three order-of-magnitude decrease in computational cost for the ROMs over the full CFD simulations. Note that the ROM values are the time it takes to run after all up-front computations have been completed. For ROM C, these up-front costs consist of individual CFD runs for each correction factor kriging surface sampling point, while for ROMs A and B, these up-front costs only included the steady CFD runs for Method of Segments calculations. Also, between the two Method of Segments ROMs, ROM A has a higher computational cost than ROM B due to the fact that direct calculation of the correction factor takes place at each time step rather than just picking the value off of a kriging surface. Lastly, note that multiple ROM computations will be necessary to calculate more than one coefficient, so the computational reduction will be slightly decreased if additional coefficients are calculated. However, for more complex geometries, the CPU time for the full CFD simulations will inevitably increase with the number of grid points, while this will have no effect on the ROM solutions. Additionally, it could be possible to find more efficient convolution algorithms which would improve the computational time of the ROM, making the computational savings even greater.

The preceding discussion compares the ROM and CFD computational time results assuming that the ROM has already been constructed. However, in deciding whether

[a] Information about the Pleiades Supercomputer available online via http://www.nas.nasa.gov/hecc/resources/pleiades.html



(a) CPU times

(b) CFD/ROM CPU time ratio

Figure 5.30: ROM-CFD CPU time comparisons

or not to use the ROM methodology (versus the full-order CFD solutions themselves) for a particular problem, the up-front computational cost of the required CFD solutions for ROM construction must be taken into consideration. For example, for the AGARD 445.6 geometry, if only a limited number of full-order simulations would need to be conducted to analyze a certain problem, then it would be much more efficient to simply conduct the full-order CFD simulations rather than go through the ROM construction process, as the full-order solutions themselves are not overly computationally expensive in a relative sense for this geometry. However, as the number of simulations to be conducted increases, as in an aeroelastic simulation framework, then the benefits of using the ROM begin to overshadow the brute-force method of direct computation of the full-order solutions. An additional item to take into consideration is the coupling with other codes and analyses. For example, the structural analysis may be performed using a finite element-type of code. Information will need to be passed to the structural analysis from the aerodynamic analysis, and vice versa. If the CFD code is incompatible with the other components of the aeroelastic analysis, then the full-order solutions will not be able to be used for the overall simulation. For each individual problem, given the CPU time and component compatibility requirements, the number of simulations that will need to be conducted, and other factors, the cost/benefit analysis of using the ROM versus full-order solutions will be different.

As an example of the ROM construction computational cost, consider the ROM constructed for the subsonic Mach regime from M=0.3-0.9. For this ROM, a total of 515 CFD correction factor kriging surface sampling points were computed, along with 77 steady-state runs for Method of Segments calculations and 12 different individual modal step input runs for convolution. The total CPU time for these runs is on the order of 10^7 seconds. However, due to parallelization of the CFL3D computations, the actual run time is much less, as each simulation is parallelized into 15 separate processors. Table 5.3 summarizes the relative time requirements, as percentage of total ROM construction CPU time, for each of the necessary types CFD simulations as applied to the subsonic ROM computed in this chapter. As seen in the table, the CFD correction factor sampling point runs account for just over 95% of this total computational time. However, the Method of Segments CFD computations accounted for only around 3.6% of the total CPU time, so using these runs to limit the number of correction factor sampling points can play a significant role in reducing ROM construction expense. For this Mach regime, rather than using a quantifiable error-based stopping criterion for sampling point determination, separate ROMs were constructed using anywhere from 100 to 500 sampling points, as described previously. The tests showed that, for the most part, using only around 100 sampling points would have been sufficient for this ROM, and doing so would have eliminated the need for around 80% of the correction factor CFD runs, significantly reducing the time needed for ROM construction. A quantifiable stopping criterion which captures the sufficiency of the ∼100 sampling points using the computationally-cheap Method of Segments thus is extremely useful to have.

5.6 Conclusion

The following are the relevant conclusions which can be deduced from this chapter:

Table 5.3: **Relative CPU time for CFD simulations used for ROM construction**

CFD run type	**CPU time (% of total)**
Steady	0.3
Steps	0.9
MoS	3.6
Correction factor	95.2

- Errors generally decrease with increasing Mach number in both the subsonic and supersonic regimes for the drag coefficient results. The highest errors are generally seen at Mach 0.3. Due to these increasing errors, the reduced frequencies/Mach numbers at this end of the parameter space form a general boundary of where the ROM can be expected to be applicable.

- The Method of Segments generally performs better in the supersonic regime than the subsonic regime due to the smaller reduced frequency values here.

- Increasing the number of kriging surface sampling points in most cases does not significantly affect the accuracy of the results, showing that a sufficient number of points has been reached.

- For cases in which $M_{sim} \neq M_{step}$, using a method of weighted averages of ROMs computed from neighboring values of M_{step} successfully reduces the ROM errors.

- Overall, the ROM methodology works well in the Mach ranges tested, with the supersonic errors generally being smaller than the subsonic errors.

- For the AGARD 445.6 wing, computational savings of well over two orders of magnitude are achieved by using the ROM over full-order CFD solutions.

Chapter 6

Concluding Remarks

This chapter ties the dissertation together by highlighting the important conclusions and other takeaway points. These items include a summary of the thesis, the important overall conclusions which have been reached, the key contributions of thesis, a note about how the method can be practically used, and finally recommendations for future work.

6.1 Summary

This thesis has presented a reduced-order modeling methodology for the calculation of unsteady aerodynamic loads for general vehicle configurations over a wide range of Mach regimes. In a controls simulation framework, it is imperative that these loads be computed both accurately and efficiently. Thus, computationally-efficient simplified models generally will not work due to their lack of accuracy, while accurate high-fidelity models will not work due to the high level of computational cost. A reduced-order model (ROM) is used for this thesis work due to the fact that it is able to extract and retain data from high-fidelity simulations while running orders of magnitude faster computationally, approaching the level of the simplified models.

The overall goal of the project encompassing this and several other research tasks is to extend a two-dimensional hypersonic vehicle controls simulation framework to three dimensions. Since a hypersonic vehicle must pass through the sub-, trans-, and supersonic regimes on the way up to hypersonic flight, the aerodynamic model used must be able to account for all of these different flight conditions. The most efficient way to do so is to use the same mathematical form for the aerodynamic loads over all regimes, which eliminates the need to switch out individual models depending on the current flight condition of the vehicle.

The specific ROM methodology combines linear convolution, which accounts for unsteadiness, with a nonlinear correction factor, which accounts for modal amplitudes and flight conditions away from those around which the model is constructed. The first step in ROM construction is to convolve the time derivative of arbitrary modal inputs with the modal step responses to find the linear, uncorrected response. The correction factor is calculated over the parameter range of interest through the use of a limited number of full-order CFD simulations. The results of these simulations are in turn used to construct a surrogate-type kriging surface of correction factor values throughout the parameter space, which for this work has consisted of Mach number and modal input amplitudes. In order to help out in determining the number and location of kriging surface sampling points, simplified models are employed. The final, corrected ROM response is computed by combining the correction factor value at the specific point in the parameter space with the linear, uncorrected ROM response. In this work, the response quantities of interest have been the lift, drag, and moment coefficients, though the ROM will work for other quantities, such as the generalized aerodynamic forces, as well.

The applications of the ROM to the subsonic, transonic, and super/hypersonic regimes are investigated in this thesis. ROM results are generated and compared with full-order CFD simulations of test cases involving the oscillation of one or more elastic mode shapes. In the hypersonic regime, the testing geometry is a two-dimensional half-diamond airfoil. The simplified model used to assist in correction factor kriging surface sampling point determination is piston theory. In the other Mach regimes, the testing platform is a three-dimensional AGARD 445.6 wing geometry. The simplified model used here for sampling point determination is the newly-developed Method of Segments (MoS). For each of these regimes, several different tests are conducted. Single-modal oscillation tests focus on the ROM-CFD comparisons as either oscillation amplitude or oscillation frequency is increased while all other variables remain constant. Multi-modal oscillation tests focus on finding the mean ROM errors over many different CFD simulations in which the modal and Mach number parameters are spread evenly throughout the parameter space. The applicability boundaries of the ROM methodology are investigated by tracing the ROM errors as the Mach number of the simulation moves away from the Mach number of the step response on which the ROM is constructed. Finally, the number of correction factor kriging surface sampling points necessary is investigated by constructing the model with varying numbers of points and comparing the results.

6.2 Conclusions

The following are the principal conclusions from this thesis, obtained from the analysis of the ROM over the different Mach regimes examined:

- The correction factor has been successfully applied to the calculation of the lift, drag, and moment coefficients. The largest improvement of the corrected ROM over the uncorrected ROM occurred consistently for the drag coefficient. In many instances the uncorrected, linear ROM produced satisfactory results for the lift coefficient.

- Though the ROM can be applied in each of the regimes tested, the results show that the errors generally increase with reduced frequency and hence unsteadiness in the flow. This is shown in two ways. First, the ROM errors are generally lowest in the super/hypersonic regimes, where the increased flow velocity serves to decrease the reduced frequency. Second, the single modal oscillation test cases demonstrate a decrease in accuracy corresponding to an increase in oscillation frequency, which increases the reduced frequency. These increased errors for the single mode cases are manifested qualitatively as a phase shift between the ROM and CFD results.

- A single ROM has been shown to work over a range of Mach numbers. The framework is there to extend this to other flight parameters, such as altitude, by the addition of further variables into the correction factor parameter space. However, it would be more of a challenge to extend the ROM to changes in vehicle geometry. While, for example, the span of the wing could be used as a variable in the correction factor parameter space, the practical application of the methodology would dictate that a new CFD grid would have to be established for each geometric modification, likely rendering the ROM for this case as infeasible.

- The use of simplified models greatly assisted in the construction of the ROM by providing some basis for the determination of where in the parameter space the correction factor kriging surface sampling points should be located.

- Upon completion of all necessary CFD runs, the ROM runs significantly faster by well over two orders of magnitude. For more complex geometries that require larger grid sizes, these computational savings will be even greater.

6.3 Key Contributions

This dissertation makes several key contributions to the literature, listed as follows:

- Introduces a robust new method to correct linear convolution using a steady nonlinear correction factor. Whereas CFD code limitations inherently cap the size of the step input that can be given, the addition of the correction factor provides a methodology to calculate the aerodynamic loads at deformations much larger than what can be used for indentification/interrogation of the flow in a CFD code.

- Increases parameter space applicability for a convolution-type ROM. In addition to input amplitude, the correction factor methodology permits the ROM to be used at flight conditions different from any one specific set of conditions used for model construction. In this thesis, the Mach number has been the flight condition variable considered. Thus, in a controls simulation framework, the aerodynamic model will not have to be re-computed every time the flight conditions are modified, and the same form of the equations can be used at all times. In general, these flight conditions, in addition to Mach number, may include other parameters such as altitude and dynamic pressure.

- Proposes way to construct continuously-applicable ROM from subsonic up to super/hypersonic Mach regimes. Based on a limited number of step inputs, the ROM is shown to provide quality results for a single test case with simulation Mach number values ranging from M=0.3 to M=3.0 by using a method of weighted averages of responses calculated from nearby values of step response Mach numbers.

- Implements framework to use simplified models to assist in ROM construction. In a preliminary analysis of the parameter space, it would be time-consuming to run full-order CFD simulations just to characterize, for example the number of sampling points that would be necessary. By employing simplified models for this purpose, a significant amount of time is saved.

- Develops new method for efficient aerodynamic calculations to aid in ROM construction. The Method of Segments utilizes a limited number of steady-state CFD computations to calculate an unlimited number of correction factor values within the Mach number-modal amplitude parameter space, thus doing away with the 1-to-1 correspondence between CFD runs and number of sampling points. This greatly helps in the determination of where in the parameter space further sampling points should be located. The MoS values increase in accuracy with decreasing reduced frequency, so the accuracy is greatest in the supersonic regime over all regimes tested. However, for the subsonic regime, the Method of Segments can be used as a simplified model to assist in correction factor kriging surface sampling point determination.

6.4 Suggested ROM Usage

This dissertation would not be complete without a note detailing when this ROM methodology would be of practical use for the designer, simulation/control engineer, or other person who wants to evaluate the unsteady aerodynamic loads of some particular vehicle. In general, the advantages of the ROM are that it works for a wide range of parameters, including flight conditions and large modal deformations. This, then, lends itself to a controls simulation framework, where the model would be expected to be valid over a wide range of these parameters. For flutter boundary prediction and other analyses where only small modal perturbations are required, the method would work, though the engineer would need to determine whether or not a different method, such as convolution or Volterra analysis, would be more efficient due to the fact that the aerodynamics of large amplitudes of oscillation would not need to be calculated. However, for post-flutter behavior, such as limit cycle oscillations that have large modal amplitudes, the ROM presented here would be of great use.

In terms of the design process, the ROM would be used when higher-fidelity analyses are needed, not as a first-step to determine potential candidate geometries/configurations. For the preliminary design of configurations for a vehicle, simplified model analysis (i.e. piston theory in the hypersonic regime) could be used to generate one or a small number of candidate configurations. Then, these candidate configurations can be analyzed by the reduced-order model in a controls sense. In terms of specific computational software, the methodology is not limited to any specific CFD code or mesh generator. Though CFL3D and ICEM CFD are used here as the flow solver and mesh generator, respectively, the ROM is certainly not limited to these specific programs. As for the post processing, ROM implementation, etc., MATLAB has been used here for those computations, though nothing prevents other computer programming languages from being used as well.

Currently, the ROM works best for lower reduced frequencies corresponding to higher flow velocities. This is shown by the fact that the highest errors were generally seen around Mach 0.3; for some perspective, of the 25 test cases conducted for results Set B in Chapter 5, the runs at Mach 0.3 had a maximum reduced frequency of 1.00, corresponding to a maximum oscillation frequency of five times the first natural frequency. For the hypersonic tests of Chapter 3, the maximum reduced frequencies are around 0.7, though both in practice as well as in the chapter, most hypersonic reduced frequencies are lower than this maximum value. In terms of the literature, Ref. 104 provides a sample control surface geometry for a hypersonic vehicle. With the given geometric and modal parameters, oscillations at the first modal natural frequency at Mach 8 would give a reduced frequency of 0.17; at Mach 0.3, this would give a reduced frequency of 3.99 at sea level. In general, for flight vehicles designed to fly in the low subsonic regime, the steadiness of the correction factor term would likely degrade the accuracy. However, the accuracy would improve with the general decrease in reduced frequency for the same geometric and oscillation frequency parameters.

Next, it is important to know approximately how many values of M_{step} will be needed for the actual simulation framework. The results showed that the lift and moment coefficients are much more sensitive to the separation of M_{sim} and M_{step}, while the drag coefficient's errors do not increase as dramatically with the separation. In general, the coefficient errors were low in the subsonic (Mach 0.3-0.9) and supersonic (Mach 1.1-3.0) Mach ranges of Chapter 5 with only two and three values, respectively, of M_{step} in each. However, the increased error in the transonic regime right around Mach 1 suggested that the M_{step} interval should be as refined as possible in that region. Though the calculation of additional step responses adds computational expense, each response needs only to be computed once. Thus, the addition of the up-front cost corresponding to more step responses can lead to reductions of error of the ROM while not increasing the ROM computational time at all. The altitude was held constant throughout this research for each set of runs, though that may be included as another variable in the correction factor kriging surface. Another possibility is to have the altitude and Mach number be combined and have dynamic pressure be used as the flight condition variable. Finally, no limit was seen in terms of the size of modal amplitudes that can be applied. Rigid body modes in the ROM will be treated in the same way as elastic mode shapes.

6.5 Recommendations for Future Work

A number of areas remain open to be explored further in future work:

- Further analysis into the ROM's performance with full-order Navier-Stokes solutions would be beneficial. Significant challenges have been encountered in this research in obtaining a structured grid capable of providing quality unsteady viscous solutions with CFL3D. A further discussion of this point is found in Appendix B. Additionally, studies of CFD preconditioning effects may be useful at lower Mach numbers.

- The addition of an unsteady term to the correction factor itself would open up potential for error reduction at lower flow velocities and higher oscillation frequencies. This would possibly take the form of a velocity term in the correction factor calculations rather than just a modal deformation.

- The ROM itself should be implemented in a full controls simulation framework for a hypersonic or other type of flight vehicle.

- The ROM's application to more complex geometric configurations should be explored. While this would not be expected to affect the ROM methodology itself, the potential application of simplified models for correction factor kriging surface sampling point determination would need to be investigated. For example, the Method of Segments would not be expected to work as well for long, slender vehicle with significant longitudinal modal deformations, so other options would need to be considered.

Appendices

Appendix A

Steps for ROM Construction

This purpose of this appendix is to give a “cookbook-style” description of steps necessary to construct and conduct simulations with the ROM. Figure A.1 displays a detailed block diagram showing the steps necessary for ROM construction and simulation; note that Q_0 refers to the steady-state value of some particular quantity of interest.

ROM Construction:

1. Select the geometric configuration of interest and create a CFD grid. At this point, low-fidelity analyses should have been performed on an array of candidate configurations, and based on those results, the group should be down-selected to one or a very small group of final candidate configuration geometries.
2. Determine the parameter space of interest of the problem. This includes the range of modal deformations as well as the Mach number and any other flight condition parameter under consideration.
3. Conduct CFD simulations to find the step responses to the modes of interest in the problem. If the geometry is asymmetric, then the positive and negative step responses will need to be found. From these responses, the linear ROM result y_{conv} can be computed.

4. Select the simplified model(s) to use for sampling point determination as well as the stopping criterion for the addition of further kriging surface sampling points. If the Method of Segments simplified model is selected, conduct steady-state CFD runs over a range of Mach numbers and rigid-body angles of attack.
5. Use the sampling point determination methodology to find the correction factor sampling points for the problem, and conduct the corresponding correction factor CFD runs. Additionally, steady-state CFD runs at zero angle of attack are necessary to calculate and subtract off the steady-state offset values during ROM computations.
6. From these CFD sampling point values, construct the lift, drag, and moment correction factor kriging surfaces.

ROM Simulation:

1. At each time step during the simulation, determine the modal and flight condition inputs. For comparisons with direct CFD simulations, these could be prescribed modal inputs, such as those used for ROM-CFD comparisons in this thesis. However, for a full aeroelastic simulation, these would come from a thermoelastic structural analysis.
2. Compute the linear response y_{conv} using linear convolution with the modal step responses.
3. Add the nonlinear correction factor by picking the value off the kriging surface corresponding to the specific combination of geometric and flight conditions.
4. Add on the steady-state value of the coefficient of interest to find the final response value. Repeat these steps for each time step in the simulation.

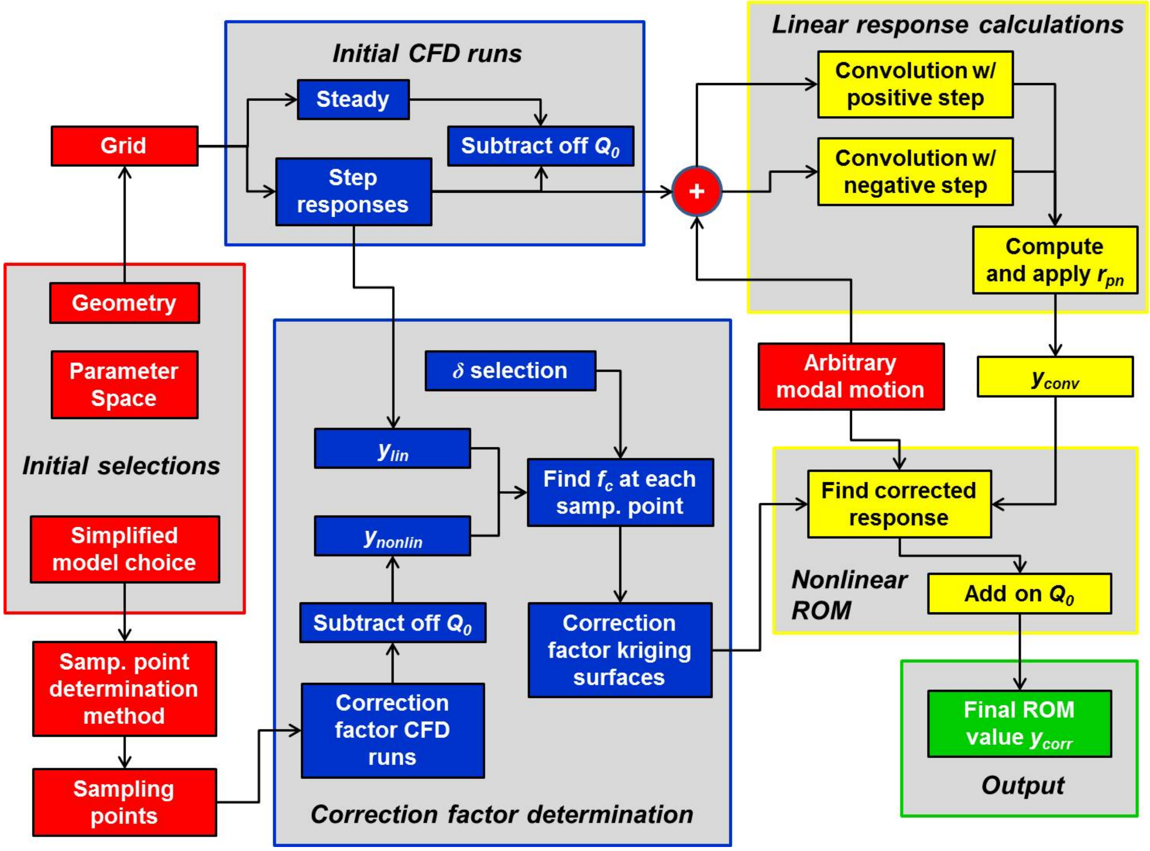

Initial CFD runs
Steady
Subtract off Q_0
Step responses
Grid
+
Linear response calculations
Convolution w/ positive step
Convolution w/ negative step
Compute and apply r_{pn}
Geometry
Parameter Space
Initial selections
Simplified model choice
Samp. point determination method
Sampling points
δ selection
y_{lin}
y_{nonlin}
Find f_c at each samp. point
Subtract off Q_0
Correction factor CFD runs
Correction factor kriging surfaces
Correction factor determination
Arbitrary modal motion
y_{conv}
Find corrected response
Nonlinear ROM
Add on Q_0
Final ROM value y_{corr}
Output

Figure A.1: Block diagram of overall ROM steps

Appendix B

CFD Highlights

B.1 Governing Equations

Computational fluid dynamics solvers find the solutions to the aerodynamic governing equations at discrete locations throughout the flow field of interest. These equations describe the conservation of mass, momentum, and energy of the flow and are known collectively as the Navier-Stokes equations. In general, the flow field variables of interest are the density ρ, pressure p, velocity $\mathbf{V}$, and internal energy per unit mass e. In this section, each equation will be shown in conservation, partial differential equation form. Other forms of the equations are the nonconservation forms, which contain the substantial derivative, and the integral forms. The first equation is the continuity equation, showing the conservation of mass and written as follows:[105]

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{V}) = 0 \tag{B.1}$$

where the quantity $\nabla \cdot (\mathbf{A})$ is the divergence of some vector $\mathbf{A}$. Practically speaking, the continuity equation is a representation of the fact that the net mass flow out of some arbitrary control volume is equal to the rate of decrease of mass inside that same control volume.[105]

The next equation is the momentum equation, which describes the conservation of momentum and is the manifestation of Newton's Second Law, $F = ma$, in the flow field, noting that force is the time rate of change of momentum. It is expressed as follows,[105] where u, v, and w are the x, y, and z components, respectively, of the velocity field:

$$\begin{aligned} \tfrac{\partial(\rho u)}{\partial t} + \nabla \cdot (\rho u \mathbf{V}) &= -\tfrac{\partial p}{\partial x} + \rho f_x + F_{x,viscous} \\ \tfrac{\partial(\rho v)}{\partial t} + \nabla \cdot (\rho v \mathbf{V}) &= -\tfrac{\partial p}{\partial y} + \rho f_y + F_{y,viscous} \\ \tfrac{\partial(\rho w)}{\partial t} + \nabla \cdot (\rho w \mathbf{V}) &= -\tfrac{\partial p}{\partial z} + \rho f_z + F_{z,viscous} \end{aligned} \tag{B.2}$$

The f terms in the momentum equation represent the body force per unit mass on the fluid, while the $F_{viscous}$ terms represent the viscous forces due to fluid viscosity.

The final governing equation is the energy equation. It represents the conservation of energy, the fact that energy can change form but can neither be created nor destroyed:[105]

$$\frac{\partial}{\partial t}\left[\rho\left(e+\frac{V^2}{2}\right)\right]+\nabla\cdot\left[\left(e+\frac{V^2}{2}\right)\mathbf{V}\right]=\rho\dot{q}-\nabla\cdot(p\mathbf{V})+\rho\left(\mathbf{f}\cdot\mathbf{V}\right)+\dot{Q}'_{viscous}+\dot{W}'_{viscous} \tag{B.3}$$

In this equation, $\dot{Q}'_{viscous}$ is the net heat flux due to viscous forces, and $\dot{W}'_{viscous}$ is the rate of work due to viscous forces. In order to close these equations, two more equations are necessary;[105] assuming a perfect gas, these equations are the definition of e (Eq. B.4, in which c_v is the specific heat at constant volume) and the perfect gas equation (Eq. B.5, in which R is the gas constant and T is the temperature).

$$e = c_v T \tag{B.4}$$

$$p = \rho R T \tag{B.5}$$

When the effects of viscosity and heating in the above equations are neglected, they are referred to as the Euler equations.[106] This serves to simplify the analysis, as items such as turbulence models and other aspects of solving the full-order Navier-Stokes equations no longer need to be considered.

B.2 CFL3D

CFL3D[27,107] is a structured flow solver first developed in the 1980's at NASA Langley Research Center in Hampton, Virginia. It is capable of solving the time-dependent conservation form of the Navier-Stokes equations for steady and unsteady flows on two and three-dimensional geometries. Other features include a semi-discrete finite volume method for spatial discretization and upwind-biasing for the pressure and convective terms.[27] It has many parameters and options which can be set by the user. These include a number of choices for the time-stepping scheme, time step ramping, matrix inversion method, flux limiter, spatial differencing scheme, and various other parameters. To aid in solution convergence, multigridding and mesh sequencing may be employed. An array of turbulence models can be used for viscous flows. Ref. 27 contains a detailed description of these and other parameters. For unsteady flows with deforming meshes, two mesh deformation methods are built in, an exponential decay method combined with Trans-Finite Interpolation and a finite macro-element method combined with Trans-Finite Interpolation. Ref. 107 contains a description of these methods as well as an overview of CFL3D capabilities and a primer on unsteady computations. The reader is referred to this and Ref. 27 as excellent user's guides for code information.

The code does not have a graphical user interface, so all code inputs must be placed in an input file (an example input file can be found at the end of this appendix). For steady non-restarted cases, the required files for simulation are the input file, grid file, and CFL3D executable. For unsteady computations with mesh deformation, the additional files required are a restart file containing the solution from a previous run and a modal input file (*aesurf.dat*). This modal input file lists the x, y, and z values of each grid point for each mode shape under consideration. Then, in the overall input file, the deformation of a particular mode is given in terms of the magnitude of the modal deformation in the *aesurf.dat* file.

B.2.1 CFD Simulations

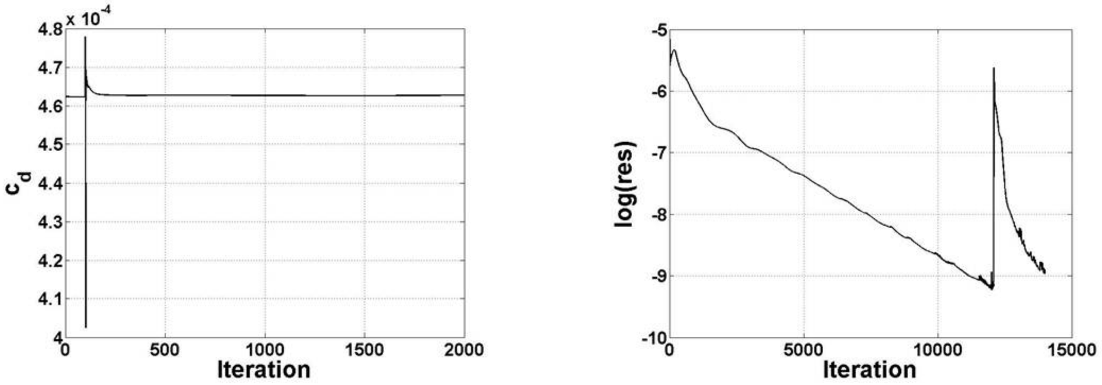
In general, all CFD runs in this thesis are conducted by first converging a steady-state solution. Then, the code is switched to unsteady mode (iunst=2), and the desired modal inputs are given. An example CFL3D input file with sinusoidal modal inputs for the half-diamond geometry is given at the end of this appendix.

For the hypersonic half-diamond airfoil simulations, the runs are conducted at right around 27 kilometers in altitude with a dimensional time step of 2×10^{-5} s. For the unsteady runs, 15 sub-iterations are used to aid in convergence. For each of the full-order solutions with sinusoidal modal inputs, 2500 time steps are computed per solution. The grid itself is split into eight separate blocks to allow for parallel computing, thus aiding computational efficiency of the simulations. Multigridding with 5 total grid levels is employed. Some of the various CFL3D parameters utilized are a second-order accurate time scheme (ita=-2), scalar tridiagonal matrix inversions (idiag=1), min-mod flux limiter (iflim=2), flux-vector splitting (ifds=0), an upwind-biased third-order spatial differencing parameter (rkap0=1/3), and an entropy fix (epsa_r=0.3).

For the AGARD 445.6 wing simulations, runs were conducted at just over 5 kilometers in altitude with a dimensional time step of around 2.6×10^{-5} s. For the unsteady runs, 10-20 sub-iterations are used, depending on the specific run. For each full-order solution with sinusoidal inputs, 2000 time steps are computed per solution. As for the grid, it is split into 96 separate blocks, each with dimensions $9 \times 33 \times 21$, and a total of 3 multigrid levels are used. Many of the same parameters are used as in the half-diamond simulations, one exception being Roe's flux-difference scheme

Table B.1: **Parameters for residual example cases**

Test Case	**Mach number**	**Frequencies** (rad/s)	**Amplitudes**
Step input	0.96	N/A	1, 0, 0
Sinusoidal input	1.04	204, 259, 409	-18.4, 31.4, -6.3
Correction factor	0.95	N/A	42.1, 38.7, -13.8



(a) Drag coefficient

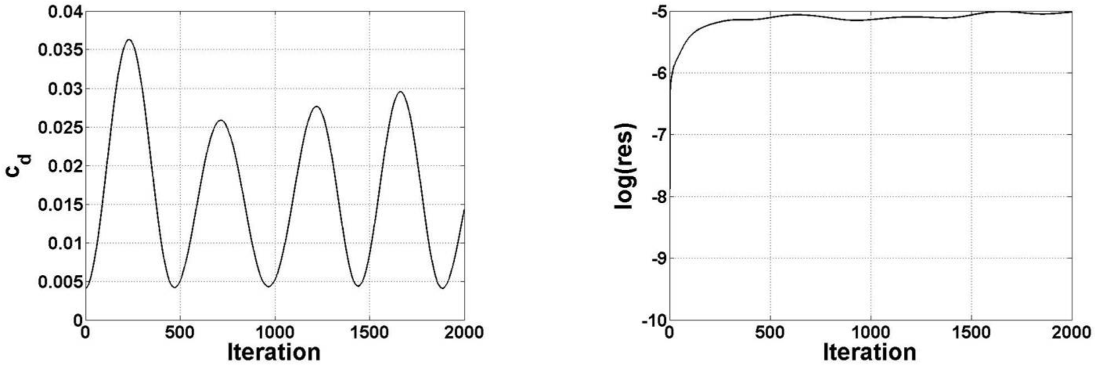
(b) Residuals

Figure B.1: Example residual plot for unsteady AGARD 445.6 step solution

(ifds=1).

In terms of the convergence of the CFD simulations, the residual values for all runs are generally right around or under 10^{-5}. Figures B.1, B.2, and B.3 display example logarithmic plots of the residuals, as well as the corresponding drag coefficient values, for three cases with a step input, sinusoidal input, and correction factor calculation input for the AGARD 445.6 wing. Note that, for the step and correction factor examples, the residual plots contain more iterations than the coefficient plots. This is due to the fact that the steady-state iteration are included on those particular residual plots. Also, the parameters for each of the runs can be found in Table B.1, where the order of the listed frequencies and amplitudes correspond to modes 1-3.

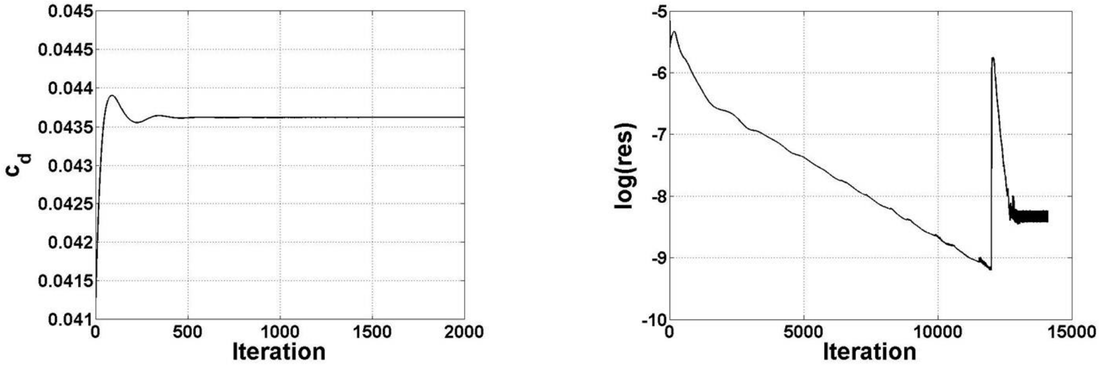
The code produces a number of different output files. The lift, drag, and moment coefficients are found in the *cfl3d.res* file, listing the values for each steady-state



(a) Drag coefficient

(b) Residuals

Figure B.2: Example residual plot for unsteady AGARD 445.6 sinusoidal solution



(a) Drag coefficient

(b) Residuals

Figure B.3: Example residual plot for unsteady AGARD 445.6 correction factor computation solution

iteration as well as unsteady time step. Other output files of interest include the plot3D grid/solution files. These files contain information on the flow field variables at each grid point and are used to calculate the lift and drag forces as well as moments on the wing segments for Method of Segments calculations. Data from these files is extracted for Method of Segments computations by using the post-processing program Tecplot.[108] All other post-processing is accomplished using MATLAB.[109]

B.2.2 Euler Solutions

An additional point that needs to be addressed is the selection of using Euler solutions in this research over full-order Navier-Stokes solutions, which include viscosity. Overall, efforts were made to acquire a quality viscous-compatible structured grid for use in the CFL3D code, including a viscous 3-D hypersonic vehicle control surface-type of geometry, AGARD 445.6 wing viscous grid, and a 2-D viscous grid of the AGARD 445.6 airfoil. However, each ran into significant challenges in obtaining quality unsteady solutions, most of which resulted in CFL3D crashing due to a myriad of errors.

The addition of viscosity would certainly have an effect on the response coefficients, especially for the drag. This effect would most largely be felt as the addition of skin friction drag around the steady-state, undeformed configuration. However, the changes in the drag due to elastic modal oscillations would still be largely due to the unsteady pressures, which are captured by the Euler formulations. Thus, while the steady-state value for the drag is different than what would be expected if the full-order Navier-Stokes equations are used, the perturbations of the drag around this steady-state are captured using an Euler formulation; capturing these perturbations is the goal of this research.

B.2.3 CFL3D: Lessons Learned

This section is slightly different from the rest of the thesis in that it is basically a memory dump of some of the issues encountered while running CFL3D and the solutions to them. Although, a better section title may have been "Lessons Learned (And Some That Have Not)," as there are a few problems that have been faced to which no satisfactory solutions have been found that do not involve work-arounds. The goal here is to provide a resource to anyone who may be using CFL3D and coming across some of the same issues. Some lessons were first learned by reading the user's guides and then reinforced through experience, some were learned by systematic searches through the various output files, and a few were learned by arbitrarily trying random things after becoming frustrated that nothing else was working.

- The *cfl3d.error* file is not always very helpful. Instead of relying on this, be sure to look at all the output files, including the outputs written by the specific computer cluster, *cfl3d.out*, *precfl3d.out*, and others. On many occasions, the best clue to figuring out what went wrong is the location in the *cfl3d.out* file where the output stops.

- Make sure you have all the necessary files in your directory. There is no guarantee that the error messages will immediately point to this being the issue when the code bombs, so that is a good first check when something bad happens.

- When creating grids, make sure all of the indices of the blocks are all oriented correctly with each other. Not changing them to do so prior to exporting the mesh to CFL3D will result in errors which can seem quite puzzling. When all else fails in trying to get a grid to run, open up the formatted grid file to see if number of grid points in each direction for each block (in the header of the file) are correct.

- When creating a two-dimensional grid, two parallel planes are required, i.e., all the grid points are created at y=1, and the same x and z values are also created at y=2. Make sure these two planes are exactly parallel with each other. If they are not, even by the slightest bit, the code will crash. If a two-dimensional grid will not run at all, check to make sure that all points in each plane are exactly planar. There have been cases where just a few of the points were off by something on the order of 10^{-10} or less despite inputs for the points to be exactly planar, and that caused the code to crash.

- In terms of where the runs were conducted, the half-diamond computations were performed on the nyx cluster at the University of Michigan, while the AGARD computations were conducted on the Pleiades cluster at NASA. In an issue that has yet to be completely solved, the half-diamond runs would not run properly on the NASA cluster, while the AGARD grid would never run properly on the Michigan cluster. The grids were constructed on different computers, so the issue may have something to do with the endianness of the various computer architectures. In all likelihood, a person with a stronger computer science background would be able to come up with a solution, but being limited to a specific cluster for each grid did not turn out to be much of a hindrance.

- Block splitting is very nice. The splitter feature automatically creates a new input file when the grid is split into blocks, so you do not have to worry about manually inputting potentially hundreds of block-to-block interface index values. Thus, you can create a grid in the simplest manner possible in terms of number of blocks and then use the splitter tool to split the grid and make it more efficient for parallel processing.

- Multigridding is good for convergence, but sometimes it can result in a grid crashing. If the grid is crashing for no apparent reason with multigridding, try not using it.

B.2.4 CFL3D Input File

The following pages show a sample input file for the half-diamond airfoil geometry. This particular case is an unsteady sinusoidal simulation, signaled by the moddfl value of 1. As the number of blocks in the grid increases, the length of the input file also increases accordingly, as many input values need to be input one line per block. To give a sense as to how large these files can become, the corresponding input file for the AGARD 445.6 grid, with 96 blocks, takes up 29 pages if shown in the same manner as the half-diamond grid input file here.

```
I/O FILES
diamond.bin
plot3dg.bin
plot3dq.bin
cfl3d.out
cfl3d.res
cfl3d.turres
cfl3d.blomax
cfl3d.out15
cfl3d.prout
cfl3d.out20
ovrlp.bin
patch.bin
restart.bin
>----User Based Input
epsa_r 0.3
<-------------------
Two-dimensional diamond airfoil
     XMACH     ALPHA      BETA  REUE,MIL   TINF,DR     IALPH    IHSTRY
    8.3589    0.0000    0.0000    5.9775  400.4100   90.0000    0.0000
      SREF      CREF      BREF       XMC       YMC       ZMC
   1.60000   1.60000    1.0000   0.00000      0.00      0.00
        DT     IREST   IFLAGTS      FMAX     IUNST    CFLTAU
   0.00598         1         0       0.0         2         5
     NGRID   NPLOT3D    NPRINT    NWREST      ICHK       I2D    NTSTEP       ITA
        -8         0         4       100         0         1      2500        -2
       NCG       IEM  IADVANCE    IFORCE  IVISC(I)  IVISC(J)  IVISC(K)
         4         0         0       000         0        0          0
         4         0         0       000         0        0          0
         4         0         0       002         0        0          0
         4         0         0       001         0        0          0
         4         0         0       002         0        0          0
         4         0         0       001         0        0          0
         4         0         0       000         0        0          0
         4         0         0       000         0        0          0
    IDIM    JDIM    KDIM
      02      33     337
      02      33     337
      02     241     337
      02     241     337
      02     241     337
      02     241     337
      02      33     337
      02      33     337
    ILAMLO    ILAMHI    JLAMLO    JLAMHI    KLAMLO    KLAMHI
         0         0         0         0         0         0
         0         0         0         0         0         0
         0         0         0         0         0         0
         0         0         0         0         0         0
         0         0         0         0         0         0
         0         0         0         0         0         0
         0         0         0         0         0         0
         0         0         0         0         0         0
     INEWG    IGRIDC        IS        JS        KS        IE        JE        KE
         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0
  IDIAG(I)  IDIAG(J)  IDIAG(K)  IFLIM(I)  IFLIM(J)  IFLIM(K)
```

```
         1         1         1         2         2         2
         1         1         1         2         2         2
         1         1         1         2         2         2
         1         1         1         2         2         2
         1         1         1         2         2         2
         1         1         1         2         2         2
         1         1         1         2         2         2
         1         1         1         2         2         2
   IFDS(I)   IFDS(J)   IFDS(K)  RKAP0(I)  RKAP0(J)  RKAP0(K)
         0         0         0    0.3333    0.3333    0.3333
         0         0         0    0.3333    0.3333    0.3333
         0         0         0    0.3333    0.3333    0.3333
         0         0         0    0.3333    0.3333    0.3333
         0         0         0    0.3333    0.3333    0.3333
         0         0         0    0.3333    0.3333    0.3333
         0         0         0    0.3333    0.3333    0.3333
         0         0         0    0.3333    0.3333    0.3333
      GRID     NBCI0   NBCIDIM     NBCJ0   NBCJDIM     NBCK0   NBCKDIM    IOVRLP
         1         1         1         1         1         1         1         0
         2         1         1         1         1         1         1         0
         3         1         1         1         1         1         1         0
         4         1         1         1         1         1         1         0
         5         1         1         1         1         1         1         0
         6         1         1         1         1         1         1         0
         7         1         1         1         1         1         1         0
         8         1         1         1         1         1         1         0
I0:   GRID   SEGMENT    BCTYPE      JSTA      JEND      KSTA      KEND     NDATA
         1         1      1001         0         0         0         0         0
         2         1      1001         0         0         0         0         0
         3         1      1001         0         0         0         0         0
         4         1      1001         0         0         0         0         0
         5         1      1001         0         0         0         0         0
         6         1      1001         0         0         0         0         0
         7         1      1001         0         0         0         0         0
         8         1      1001         0         0         0         0         0
IDIM: GRID   SEGMENT    BCTYPE      JSTA      JEND      KSTA      KEND     NDATA
         1         1      1002         0         0         0         0         0
         2         1      1002         0         0         0         0         0
         3         1      1002         0         0         0         0         0
         4         1      1002         0         0         0         0         0
         5         1      1002         0         0         0         0         0
         6         1      1002         0         0         0         0         0
         7         1      1002         0         0         0         0         0
         8         1      1002         0         0         0         0         0
J0:   GRID   SEGMENT    BCTYPE      ISTA      IEND      KSTA      KEND     NDATA
         1         1      1000         0         0         0         0         0
         2         1      1000         0         0         0         0         0
         3         1         0         0         0         0         0         0
         4         1         0         0         0         0         0         0
         5         1         0         0         0         0         0         0
         6         1         0         0         0         0         0         0
         7         1         0         0         0         0         0         0
         8         1         0         0         0         0         0         0
JDIM  GRID   SEGMENT    BCTYPE      ISTA      IEND      KSTA      KEND     NDATA
         1         1         0         0         0         0         0         0
         2         1         0         0         0         0         0         0
         3         1         0         0         0         0         0         0
         4         1         0         0         0         0         0         0
         5         1         0         0         0         0         0         0
         6         1         0         0         0         0         0         0
         7         1      1003         0         0         0         0         0
         8         1      1003         0         0         0         0         0
K0:   GRID   SEGMENT    BCTYPE      ISTA      IEND      JSTA      JEND     NDATA
```

```
         1         1      1003         0         0         0         0         0
         2         1         0         0         0         0         0         0
         3         1      1003         0         0         0         0         0
         4         1      1005         0         0         0         0         0
         5         1      1003         0         0         0         0         0
         6         1      1005         0         0         0         0         0
         7         1      1003         0         0         0         0         0
         8         1         0         0         0         0         0         0
KDIM: GRID   SEGMENT    BCTYPE      ISTA      IEND      JSTA      JEND     NDATA
         1         1         0         0         0         0         0         0
         2         1      1003         0         0         0         0         0
         3         1      1005         0         0         0         0         0
         4         1      1003         0         0         0         0         0
         5         1      1005         0         0         0         0         0
         6         1      1003         0         0         0         0         0
         7         1         0         0         0         0         0         0
         8         1      1003         0         0         0         0         0
      MSEQ    MGFLAG    ICONSF       MTT      NGAM
         1         1         0         0         2
      ISSC EPSSSC(1) EPSSSC(2) EPSSSC(3)      ISSR EPSSSR(1) EPSSSR(2) EPSSSR(3)
         0       0.3       0.3       0.3         0       0.3       0.3      0.3
      NCYC    MGLEVG     NEMGL     NITFO
        15         5         0         0
      MIT1      MIT2      MIT3      MIT4      MIT5      MIT6      MIT7      MIT8
         1         1         1         1         1         1         1         1
  1-1 BLOCKING DATA:
      NBLI
         8
 NUMBER   GRID     :    ISTA   JSTA   KSTA   IEND   JEND   KEND  ISVA1  ISVA2
      1      1             1     33      1      2     33    337      1      3
      2      3             1    241      1      2    241    337      1      3
      3      5             1    241      1      2    241    337      1      3
      4      2             1     33      1      2     33    337      1      3
      5      4             1    241      1      2    241    337      1      3
      6      6             1    241      1      2    241    337      1      3
      7      1             1      1    337      2     33    337      1      2
      8      7             1      1    337      2     33    337      1      2
 NUMBER   GRID     :    ISTA   JSTA   KSTA   IEND   JEND   KEND  ISVA1  ISVA2
      1      3             1      1      1      2      1    337      1      3
      2      5             1      1      1      2      1    337      1      3
      3      7             1      1      1      2      1    337      1      3
      4      4             1      1      1      2      1    337      1      3
      5      6             1      1      1      2      1    337      1      3
      6      8             1      1      1      2      1    337      1      3
      7      2             1      1      1      2     33      1      1      2
      8      8             1      1      1      2     33      1      1      2
  PATCH SURFACE DATA:
    NINTER
         0
  PLOT3D OUTPUT:
   GRID IPTYPE ISTART   IEND   IINC JSTART   JEND   JINC KSTART   KEND   KINC
 IMOVIE
      1
  PRINT OUT:
   GRID IPTYPE ISTART   IEND   IINC JSTART   JEND   JINC KSTART   KEND   KINC
      3      0      0      0      0      1    241      1    337    337      1
      4      0      0      0      0      1    241      1      1      1      1
      5      0      0      0      0      1    241      1    337    337      1
      6      0      0      0      0      1    241      1      1      1      1
   CONTROL SURFACE:
  NCS
    0
   GRID ISTART   IEND   JSTART   JEND   KSTART   KEND  IWALL  INORM
```

```
Moving Grid Data - Deforming surface (forced motion)
 NDEFRM
      0
   LREF
   GRID IDEFRM RFREQI OMEGAX OMEGAY OMEGAZ XORIG YORIG ZORIG
   GRID   ICSI   ICSF   JCSI   JCSF   KCSI  KCSF
Moving Grid Data - Aeroelastic surface (aeroelastic motion)
    NAESURF
          1
    IAESRF     NGRID     GREFL      UINF      QINF    NMODES   ISKYHK
       1.0      -4.0       1.0    2499.3  108433.6       3.0       0.0
    FREQ  GMASS    DAMP  xO(2n-1)  xO(2n)  gfO(2n)
   6.005    1.0  0.9999       0.0     0.0      0.0
   6.005    1.0  0.9999       0.0     0.0      0.0
   6.005    1.0  0.9999       0.0     0.0      0.0
  MODDFL         AMP      FREQ        tO
    1.0000   -0.0738  265.6026    0.0020
    1.0000   -0.1486  973.1940    0.0020
    1.0000   -0.0877  816.5346    0.0020
    GRID   IAEI  IAEF  JAEI  JAEF   KAEI    KAEF
       3      1     2     1   241    337     337
       4      1     2     1   241      1       1
       5      1     2     1   241    337     337
       6      1     2     1   241      1       1
Moving Grid Data - Data for field/multiblock mesh movement
 NSKIP   ISKTYP   BETA1   ALPHA1   BETA2   ALPHA2   NSPRGIT
     0       -2     1.5      1.0     5.0    0.025         1
 GRID    ISKIP    JSKIP   KSKIP
Moving Grid Data - Multi-motion coupling
 NCOUPL
      0
 SLAVE   MASTER   XORIG   YORIG   ZORIG
```

Appendix C

Correction Factor Offset δ Selection

This appendix provides some details involving the selection of the correction factor offset value δ and an explanation of the corrected ROM solution's δ-independence.

C.1 Effect of δ on Solution

Recall that the equation for the correction factor is written as follows:

$$f_c = \frac{y_{nonlin} + \delta}{y_{lin} + \delta} \tag{C.1}$$

The purpose of the δ term is to give the correction factor a finite value as the denominator approaches zero for situations when the various linear responses sum to zero or near zero. Thus, the most important item to consider when selecting δ is to ensure that it is larger than the largest expected absolute value of y_{lin}. However, the upper boundaries of δ do not appear to be as clearly-defined. To get a more quantitative understanding of how the ROM is affected by the specific choice of δ, a study is conducted in order to investigate errors as δ changes. Then, the corrected response y_{corr} is deconstructed to obtain a better practical understanding of the correction factor's application.

For this study, the example test case using the 2-D half-diamond airfoil geometry listed as Test 1 in Table 3.5 in Chapter 3 is considered here. The parameters for this case are shown below in Table C.1. The lift, drag, and moment errors are tracked for increasing values of δ. The error metric used in this study is the L_∞ error. This is chosen due to the fact that using a very small δ value (or even δ=0) would not cause the correction factor value to tend towards infinity for all time steps in the simulation. Rather, only certain areas would be affected by this singularity, which will be manifested as large ROM errors for those particular time steps. These areas are easily isolated by the L_∞ error, which picks the single largest error out of all time steps in the simulation. Additionally, instead of using δ itself as the independent variable, the results are plotted in terms of the variable r_δ, defined as follows:

$$r_\delta = \frac{\delta}{\max\left(y_{CFD}\right) - \min\left(y_{CFD}\right)} \tag{C.2}$$

where the denominator is the range spanned by coefficient values over the particular run and is the same value used to normalize the L_1 and L_∞ errors. This quantity is chosen in order give a better sense to the relative magnitude differences between δ and the coefficient values under consideration.

The results are shown in Fig. C.1, in which the value of r_δ used in Chapter 3 is highlighted by the circles around the appropriate data points. For each of the quantities here, the errors are largest for small values of r_δ, showing that having correction factor denominators equal to or close to zero is an issue that will be seen in practice and will result in error increases. The most significant increase in error is seen for the moment coefficient in which the maximum error for r_δ=0.60 is just over 263%. For values of $r_\delta > 1$, in which δ is now larger than the range of CFD values, the errors are seen to level off to constant values. To see how the largest errors are manifested graphically, consider Fig. C.2, which shows the comparisons for the drag and moment coefficients among the CFD results, ROM results from Chapter 3 using δ=100, and ROM results using the δ value resulting in the largest error for that particular coefficient. For the drag results of Fig. C.2(a), the ROM with small δ value

Table C.1: δ **test case parameters**

M	d_1	d_2	d_3	ω_1	ω_2 (rad/s)	ω_3
8.75	36.3	11.5	-24.7	648.0	854.9	850.2

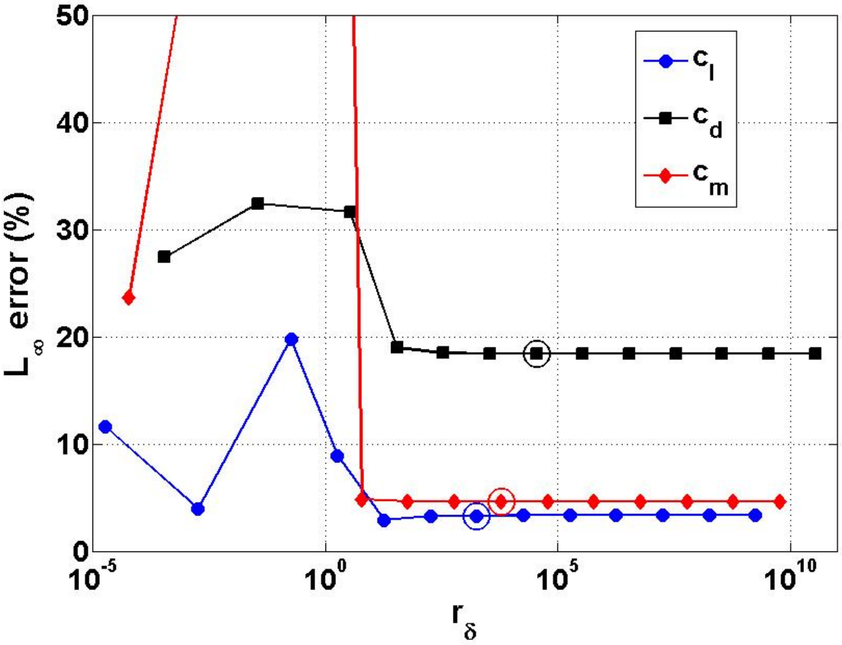
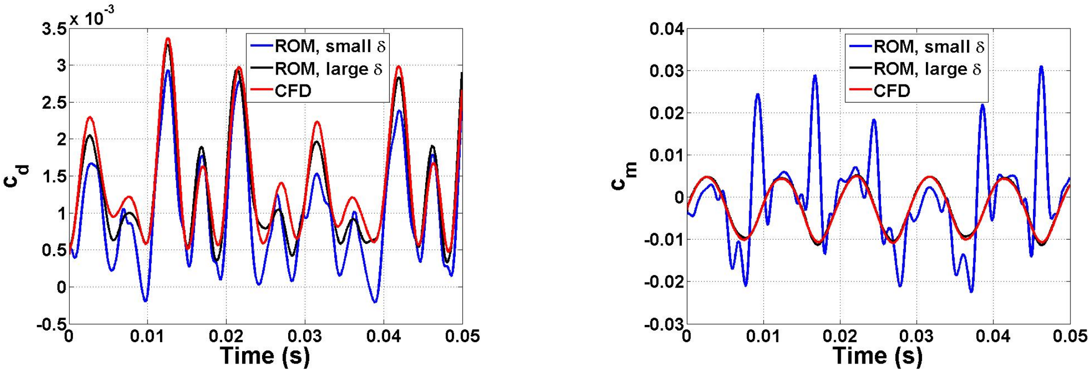


Figure C.1: ROM errors as function of δ

is qualitative not as good a representation as the other ROM, with peaks/valleys that do not match well with the CFD data. The discrepancy between the ROMs is even more striking for the moment coefficient, shown in Fig. C.2(b). The near-zero correction factor denominator values result in significant spikes and undulations in the small-δ ROM response, while the large-δ ROM is qualitatively a very good representation of the CFD data.

Overall, these results suggest that the most critical factor in choosing the value of δ to use is that it is larger than the maximum value of y_{lin} expected to be encountered. If that specific quantity cannot be estimated *a priori*, then the maximum range expected



(a) Subsonic cases

(b) Supersonic cases

Figure C.2: ROM-CFD comparisons using different values of δ

to be spanned by the response quantity of interest can be a good comparison quantity to use in place of y_{lin}, as shown by the errors leveling off for $r_\delta > 1$. Errors remain constant with increasing δ after this point has been reached.

C.2 Corrected ROM Solution δ-Independence

The final item to discuss here is exactly why the corrected ROM solution is δ-independent. Recall again that the correction factor can be written as,

$$f_c = \frac{y_{nonlin} + \delta}{y_{lin} + \delta} \tag{C.3}$$

It follows, then, that the corrected response, y_{corr}, can be found by the expression,

$$y_{corr} = f_c \left(y_{conv} + \delta \right) - \delta \tag{C.4}$$

which can be re-written as,

$$y_{corr} = f_c y_{conv} + \delta \left(f_c - 1 \right) = \alpha + \beta \tag{C.5}$$

where the two terms have been renamed α and β for convenience.

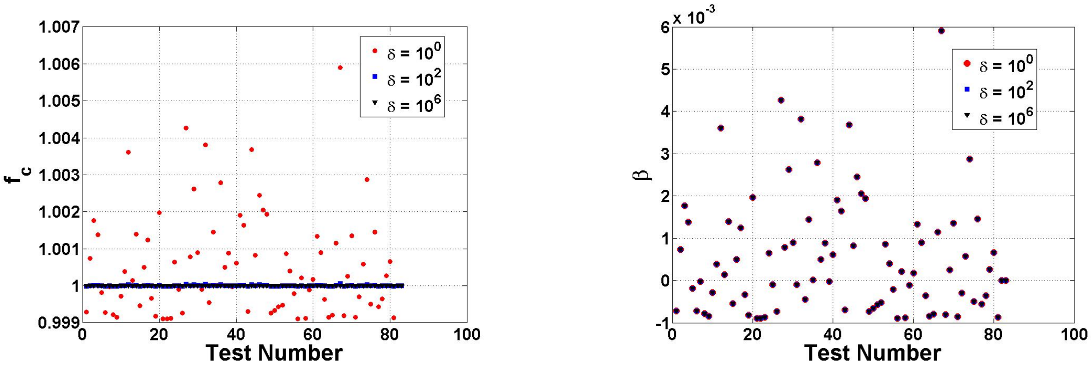
The first item to investigate is whether or not each of the individual α and β terms are δ-independent. Consider Fig. C.3, which shows correction factor (Fig. C.3(a)) and β (Fig. C.3(b)) values obtained from the CFD runs used to construct the single-mode ROM for the AGARD 445.6 wing in Chapter 4 for varying values of δ. As can be seen in the plots, f_c shows a degree of δ-dependence, as the values move closer to 1 with increasing δ. However, the β term shows very little δ-dependence, as the values remain effectively constant over the δ range. Extremely small differences do exist for β at different δ values, but these differences are virtually indistinguishable on the plot, demonstrating that β is such a weak function of δ that it is essentially δ-independent.

Now, consider the α term. As described above, the f_c term is δ-dependent, approaching 1 as δ increases. Consider again Fig. C.3(a). Though a clear δ-dependence for f_c can be seen, all values are close to 1, as the largest single value over all cases is around 1.006. Because of this, the α term will be very close to y_{conv}, and the δ-dependence of f_c will not translate much to the overall value of α.

Next, consider the second term, β, which is shown to be virtually δ-independent in Fig. C.3(b). This is where the correction factor's difference from 1, and hence the nonlinearity of the problem, is emphasized. If f_c=1, then β will obviously be zero. However, the relatively small difference from 1 is magnified by the δ term. Rather than a multiplicative factor for the uncorrected solution, the β term serves as an additive factor to move from the uncorrected to the corrected value. By replacing f_c with its definition, β can be rewritten as:

$$\beta = \frac{\delta}{y_{lin} + \delta} \left(y_{nonlin} - y_{lin} \right) \tag{C.6}$$

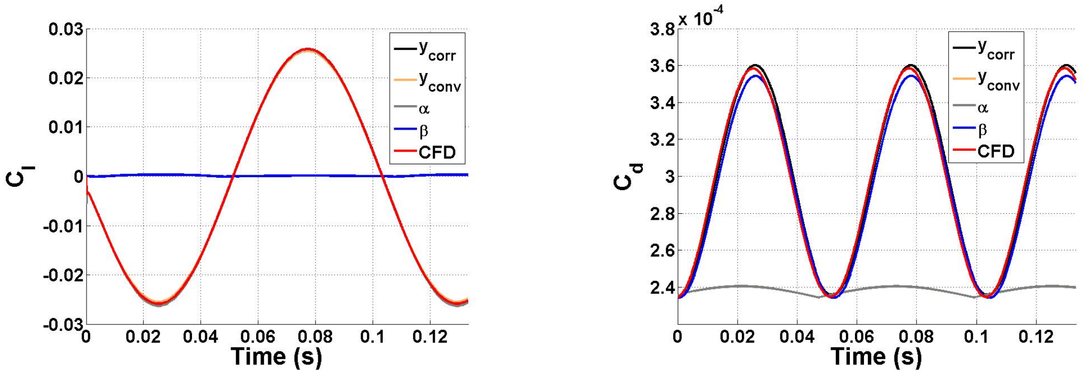
From this equation, it can be seen that β is driven largely by the difference between the quantities y_{nonlin} and y_{lin}. Consider the sample ROM-CFD comparison case shown in Fig. C.4, in which y_{corr}, y_{conv}, α, β, and the CFD lift and drag coefficient results are all plotted for a single-mode of oscillation (amplitude 20) of the AGARD 445.6 wing



(a) f_c values for sample tests

(b) β values for sample tests

Figure C.3: f_c and β values over range of δ, c_d



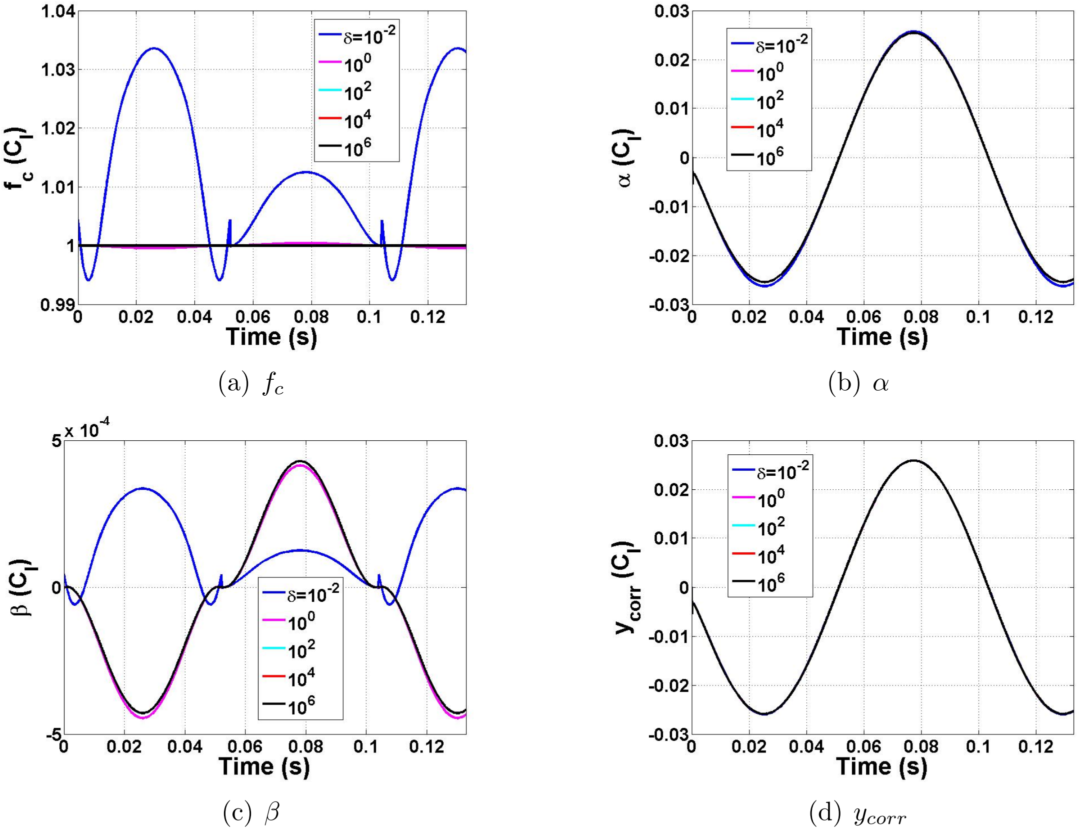
(a) C_l

(b) C_d

Figure C.4: ROM component plots, single mode of oscillation, amplitude 20

at Mach 0.9. For the drag coefficient plot (Fig. C.4(b)), β is the dominant term. In an absolute sense, β is small for all time steps. However, when added to the uncorrected ROM values, the β values do make a significant difference percentage-wise in the final corrected solution as compared to the uncorrected one, as shown in the plot. For the lift coefficient plot (Fig. C.4(a)), α is the dominant term, which agrees from the previously-seen results of the linear response showing good agreement with the CFD results.

For a further picture of how δ does or does not affect the various correction factor-related components, consider Figs. C.5 and C.6, which plot the lift and drag coefficient values, respectively, for f_c, α, β, and y_{corr} for the same test case as in Fig. C.4 over a range of δ values. For the lift coefficient, the α and β curves for δ values other than 10^{-2} are virtually indistinguishable from each other. Figure C.5(a) shows that the f_c values move closer to 1 as δ increases, which agrees with the previously-seen results. For $\delta = 10^{-2}$, the offset value is approaching the values of the coefficients themselves, potentially introducing some singularity issues into the solution. That is one possible reason for the discrepancy in the β plot for this δ value, though the overall corrected ROM solution y_{corr} still agrees with those computed using the other δ values. For the drag coefficient plots in Fig. C.6, the trend of f_c moving closer to 1 with increasing δ is again observed. The other quantities are very close to each other over the δ range; as described above, the correction factor values, while changing with δ, are all relatively close to 1 and thus have a minimal effect on α, and β is largely δ-independent.



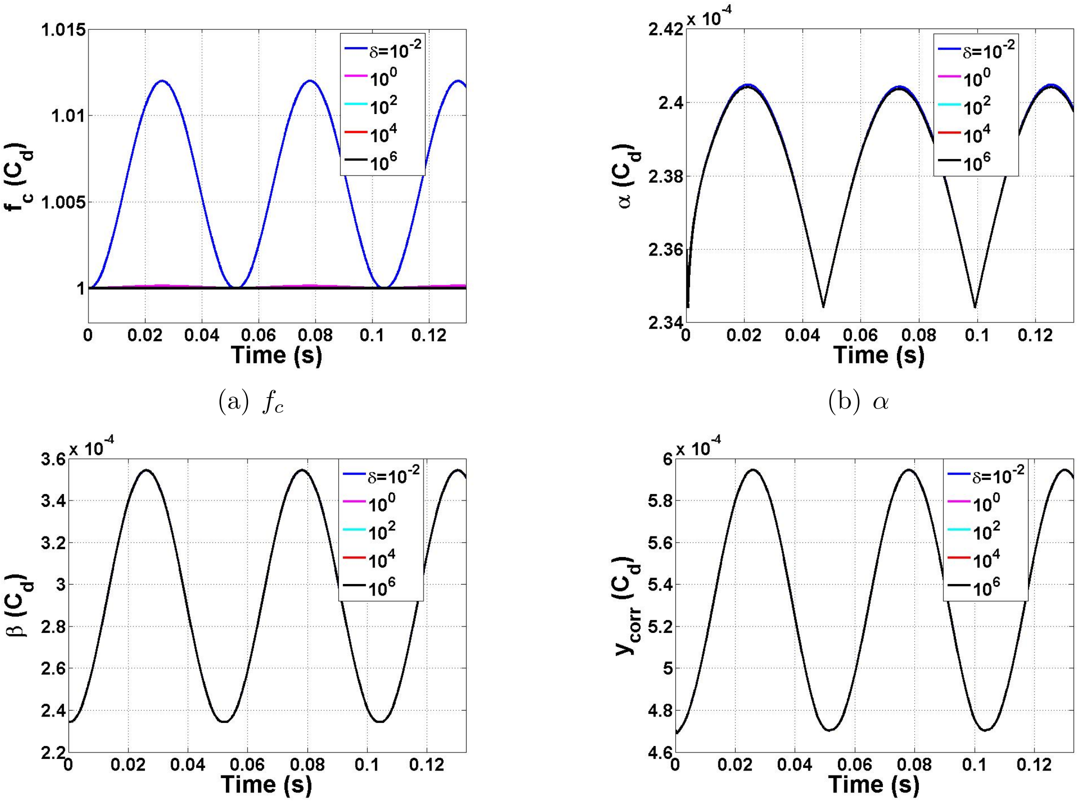
(a) f_c

(b) α

(c) β

(d) y_{corr}

Figure C.5: Lift coefficient quantities with varying δ



(a) f_c

(b) α

(c) β

(d) y_{corr}

Figure C.6: Drag coefficient quantities with varying δ

Appendix D

Potential for Non-CFD Applications

One aspect of the ROM methodology that has not as of yet received very much attention is its potential application to non-CFD problems. In theory, the method could be applied to any system of differential equations for which a step response can be computed. This appendix provides the brief highlights of a study looking into this potential, focusing on one possible significant limitation, the length of the memory of the system being modeled. The system's memory is characterized by looking at how many time steps after some sort of perturbation has been given that the effects of that perturbation will be felt. To investigate this, the Riccati equation for a nonlinear circuit is chosen as the model equation to be utilized, and it is given by:

$$\dot{y} + \alpha y + \epsilon y^2 = x\left(t\right) \tag{D.1}$$

where the input is the circuit voltage $x\left(t\right)$, and the output is the current $y\left(t\right)$; α and ϵ are resistance parameters. This equation describes a nonlinear circuit and has been used previously as a model equation for Volterra series-type of ROM analyses.[43]

In this case, the system's memory is measured by how long it takes the step response to reach 95% of its final, steady value given a set of parameters. This quantity, denoted T_{95}, is shown graphically in Fig. D.1 and is measured in terms of the number of time steps to reach the 95% value. In order to alter the system's

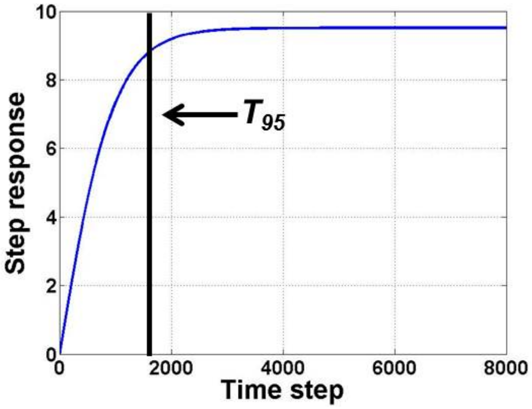
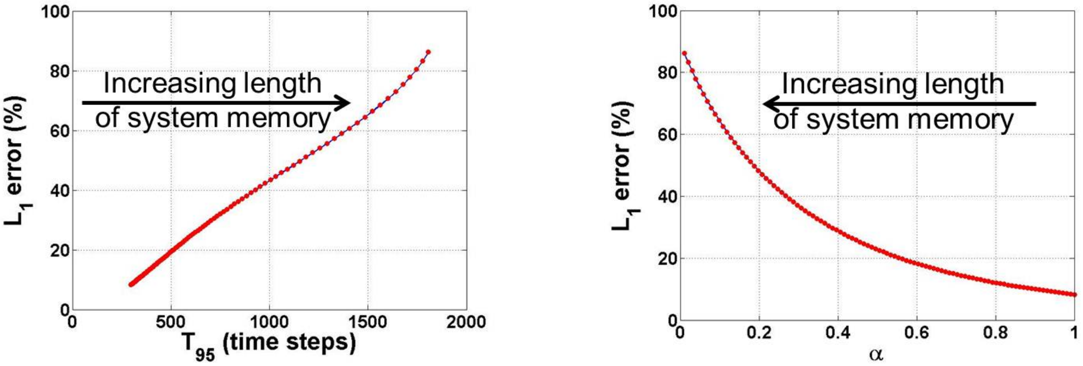


Figure D.1: T_{95} calculation

memory, the quantity α is changed for each run. For the ROM results, the correction factor is directly calculated at each time step throughout the course of the run, and the truth model used is the direct solution of the differential equation. For each test run, the input voltage $x\left(t\right)$ is given as follows,

$$\begin{aligned} x\left(t\right) &= A\left(1 - cos2\pi ft\right) \\ A &= 25 \\ f &= 0.1 \text{ Hz} \end{aligned} \tag{D.2}$$

where A is the input amplitude given in multiples of the step input. $\epsilon = 0.01$ is used for each run. Figure D.2 shows the ROM errors change both with increasing system memory (Fig. D.2(a)) and α (Fig. D.2(b)). As can be seen, an increase of system memory length adversely affects the accuracy of the ROM, with L_1 errors increasing to over 80% for the largest values of T_{95}. These results show that, when applying the ROM methodology to a new system, it may be important to investigate the memory length of that system in comparison with other systems for which the ROM applicability has previously been determined.



(a) Errors as function of T_{95}

(b) Errors as function of α

Figure D.2: Effect of system memory on ROM errors

Bibliography

[1] Bolender, M. and Doman, D., "Nonlinear Longitudinal Dynamical Model of an Air-Breathing Hypersonic Vehicle," *Journal of Spacecraft and Rockets*, Vol. 44, No. 2, March-April 2007, pp. 374–387.

[2] Bolender, M., Oppenheimer, M., and Doman, D., "Effects of Unsteady and Viscous Aerodynamics on the Dynamics of a Flexible Air-breathing Hypersonic Vehicle," *Proceedings of the AIAA Atmospheric Flight Mechanics Conference and Exhibit*, AIAA Paper No. 2007-6397, August 2007.

[3] Oppenheimer, M., Skujins, T., Bolender, M., and Doman, D., "A Flexible Hypersonic Vehicle Model Developed With Piston Theory," *Proceedings of the AIAA Atmospheric Flight Mechanics Conference and Exhibit*, AIAA Paper No. 2007-6396, August 2007.

[4] McNamara, J. and Friedmann, P., "Aeroelastic and Aerothermoelastic Analysis in Hypersonic Flow: Past, Present, and Future," *AIAA Journal*, Vol. 49, No. 6, June 2011, pp. 1089–1122.

[5] Falkiewicz, N., "Reduced-Order Aerothermoelastic Analysis of Hypersonic Vehicle Structures," Ph.D. dissertation, University of Michigan, 2012.

[6] Ashley, H. and Zartarian, G., "Piston Theory - A New Aerodynamic Tool for the Aeroelastician," *Journal of the Aeronautical Sciences*, Vol. 23, No. 12, 1956, pp. 1109–1118.

[7] Batina, J., Seidel, D., Bland, S., and Bennett, R., "Unsteady Transonic Flow Calculations for Realistic Aircraft Configurations," *Journal of Aircraft*, Vol. 26, No. 1, 1989, pp. 21–28.

[8] Ashby, D., "Potential Flow Theory and Operation Guide for the Panel Code PMARC_14," Tech. Rep. NASA TM-1999-209582, NASA Ames Research Center, Moffett Field, California, 1999.

[9] Bisplinghoff, R., H.Ashley, and Halfman, R., *Aeroelasticity*, Dover Publications, Mineola, New York, 1996.

[10] Hodges, D. and Pierce, G., *Introduction to Structural Dynamics and Aeroelasticity*, Cambridge University Press, New York, 2002.

[11] Dowell, E., *A Modern Course in Aeroelasticity: Fourth Revised and Enlarged Edition*, Kluwer Academic Publications, Boston, 2004.

[12] Friedmann, P., "Renaissance of Aeroelasticity and Its Future," *Journal of Aircraft*, Vol. 36, No. 1, January-February 1999, pp. 105–121.

[13] Dowell, E. and Tang, D., "Nonlinear Aeroelasticity and Unsteady Aerodynamics," *AIAA Journal*, Vol. 40, No. 9, September 2002, pp. 1697–1707.

[14] Livne, E., "Future of Airplane Aeroelasticity," *Journal of Aircraft*, Vol. 40, No. 6, November-December 2003, pp. 1066–1092.

[15] Chavez, F. and Schmidt, D., "Analytical Aeropropulsive/Aeroelastic Hypersonic-Vehicle Model with Dynamic Analysis," *Journal of Guidance, Control, and Dynamics*, Vol. 17, No. 6, November-December 1994, pp. 1308–1319.

[16] Parker, J., Serrani, A., Yurkovich, S., Bolender, M., and Doman, D., "Control-Oriented Modeling of an Air-Breathing Hypersonic Vehicle," *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 3, May-June 2007, pp. 865–869.

[17] Bendiksen, O., "Modern Developments in Computational Aeroelasticity," *Proceedings of the Institution of Mechanical Engineers*, Vol. 218, No. 3, June 2004, pp. 157–177.

[18] Bendiksen, O., "Review of Unsteady Transonic Aerodynamics: Theory and Applications," *Progress in Aerospace Sciences*, Vol. 47, No. 2, 2011, pp. 135–167.

[19] Batina, J., Bennett, R., Seidel, D., Cunningham, H., and Bland, S., "Recent Advances in Transonic Computational Aeroelasticity," Tech. Rep. NASA-TM-100663, NASA Langley Research Center, Hampton, Virginia, September 1988.

[20] Theodorsen, T., "General Theory of Aerodynamic Instability and the Mechanism of Flutter," Tech. Rep. 496, NACA, 1935.

[21] Peters, D., Karunamoorthy, S., and Cao, W., "Finite State Induced Flow Models Part I: Two-Dimensional Thin Airfoil," *Journal of Aircraft*, Vol. 32, No. 2, 1995, pp. 313–322.

[22] Guendel, R., "Unsteady Aerodynamics for Aeroelastic Applications Using the Impulse Response Method," Master's dissertation, Massachusetts Institute of Technology, June 2000.

[23] Guendel, R. and Cesnik, C., "Aerodynamic Impulse Response of a Panel Method," *Proceedings of the 42nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference and Exhibit*, AIAA Paper No. 2001-1210, April 2001.

[24] Batina, J., "An Efficient Algorithm for Solution of the Unsteady Transonic Small-Disturbance Equation," *Proceedings of the AIAA 25th Aerospace Sciences Meeting*, AIAA Paper No. 87-0109, January 1987.

[25] Bennett, R., Batina, J., and Cunningham, H., "Wing-Flutter Calculations with the CAP-TSD Unsteady Transonic Small-Disturbance Program," *Journal of Aircraft*, Vol. 26, No. 9, 1989, pp. 876–882.

[26] Raveh, D., "Computational-fluid-dynamics-based Aeroelastic Analysis and Structural Design Optimization - A Researcher's Perspective," *Computer Methods in Applied Mechanics and Engineering*, Vol. 194, No. 30, 2005, pp. 3453–3471.

[27] Rumsey, C. and Biedron, R., "CFL3D User's Manual, Version 5.0, Second Edition," Tech. rep., NASA Langley Research Center, Hampton, Virginia, September 1997 (last updated August 2009).

[28] Hong, M., Kuruvila, G., Bhatia, K., SenGupta, G., and Kim, T., "Evaluation of CFL3D for Unsteady Pressure and Flutter Predictions," *Proceedings of the 44th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, AIAA Paper No. 2003-1923, April 2003.

[29] McNamara, J. and Friedmann, P., "Aeroelastic and Aerothermoelastic Analysis of Hypersonic Vehicles: Current Status and Future Trends," *Proceedings of the 48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, AIAA Paper No. 2007-2013, April 2007.

[30] McNamara, J., Crowell, A., Friedmann, P., Glaz, B., and Gogulapati, A., "Approximate Modeling of Unsteady Aerodynamics for Hypersonic Aeroelasticity," *Journal of Aircraft*, Vol. 47, No. 6, November-December 2010, pp. 1932–1945.

[31] Scott, R. and Pototzky, A., "Quasisteady Aerodynamics for Flutter Analysis Using Steady Computational Fluid Dynamics Calculations," *Journal of Aircraft*, Vol. 33, No. 1, January-February 1996, pp. 191–197.

[32] Lucia, D., Beran, P., and Silva, W., "Reduced-Order Modeling: New Approaches for Computational Physics," *Progress in Aerospace Sciences*, Vol. 40, No. 1, 2004, pp. 51–117.

[33] Silva, W., Beran, P., Cesnik, C., Guendel, R., Kurdila, A., Prazenica, R., Librescu, L., Marzocca, P., and Raveh, D., "Reduced-Order Modeling: Cooperative Research and Development at the NASA Langley Research Center," *Proceedings of the 2001 International Forum of Aeroelasticity and Structural Dynamics*, IFASD-2001-008, 2001.

[34] Kim, T., "Frequency-Domain Karhunen-Loeve Method and Its Application to Linear Dynamic Systems," *AIAA Journal*, Vol. 36, No. 11, November 1998, pp. 2117–2123.

[35] Hall, K., Thomas, J., and Dowell, E., "Proper Orthogonal Decomposition Technique for Transonic Unsteady Aerodynamic Flows," *AIAA Journal*, Vol. 38, No. 10, October 2000, pp. 1853–1862.

[36] Zaide, A. and Raveh, D., "Numerical Simulation and Reduced-Order Modeling of Airfoil Gust Response," *AIAA Journal*, Vol. 44, No. 8, August 2006, pp. 1826–1834.

[37] Raveh, D., "CFD-Based Models of Aerodynamic Gust Response," *Journal of Aircraft*, Vol. 44, No. 3, May-June 2007, pp. 888–897.

[38] Gupta, K. and Bach, C., "Systems Identification Approach for a Computational-Fluid-Dynamics-Based Aeroelastic Analysis," *AIAA Journal*, Vol. 45, No. 12, December 2007, pp. 2820–2827.

[39] Glaz, B., Liu, L., and Friedmann, P., "Reduced-Order Nonlinear Unsteady Aerodynamic Modeling Using a Surrogate-Based Recurrance Framework," *AIAA Journal*, Vol. 48, No. 10, 2010, pp. 2418–2429.

[40] Glaz, B., Liu, L., Friedmann, P., Bain, J., and Sankar, L., "A Surrogate Based Approach to Reduced-Order Dynamic Stall Modeling," *Proceedings of the 51st AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, AIAA Paper No. 2010-3042, April 2010.

[41] Glaz, B., Liu, L., Friedmann, P., Bain, J., and Sankar, L., "A Surrogate-Based Approach to Reduced-Order Dynamic Stall Modeling," *Journal of the American Helicopter Society*, Vol. 57, No. 2, April 2012.

[42] DaRonch, A., Ghoreyshi, M., and Badcock, K., "On the Generation of Flight Dynamics Aerodynamic Tables by Computational Fluid Dynamics," *Progress in Aerospace Sciences*, Vol. 47, No. 8, 2011, pp. 597–620.

[43] Silva, W., "Discrete-Time Linear and Nonlinear Aerodynamic Impulse Responses for Efficient CFD Analyses," Ph.D. dissertation, College of William & Mary, December 1997.

[44] Silva, W., "Identification of Nonlinear Aeroelastic Systems Based on the Volterra Theory: Progress and Opportunities," *Nonlinear Dynamics*, Vol. 39, No. 1, 2005, pp. 25–62.

[45] Volterra, V. and Fantappie, L., *Theory of Functionals and of Integral and Integro-differential Equations*, Blackie & Son Limited, London, 1930.

[46] Rugh, W., *Nonlinear System Theory: The Volterra/Wiener Approach*, The Johns Hopkins University Press, Baltimore, 1981.

[47] Baumann, W., Herdman, T., and Stalford, H., "Recent Work Using Volterra Series as a Methodology to Analyze Nonlinear Aircraft Dynamic Properties," *Mathematical and Computer Modeling*, Vol. 11, 1988, pp. 883–888.

[48] Wagner, H., "Über die Entstehung des dynamischen Auftriebes von Tragflügeln," *ZAMM*, Vol. 5, No. 1, February 1925, pp. 17–35.

[49] Jones, R., "The Unsteady Lift of a Wing of Finite Aspect Ratio," Tech. Rep. No. 681, NACA Langley Memorial Aeronautical Laboratory, 1940.

[50] Heaslet, M. and Lomax, H., "Two-Dimensional Unsteady Lift Problems in Supersonic Flight," Tech. Rep. No. 945, NACA Ames Aeronautical Laboratory, 1949.

[51] Tobak, M., "On the Use of the Indicial Function Concept in the Analysis of Unsteady Motions of Wings and Wing-Tail Combinations," Tech. Rep. No. 1188, NACA Ames Aeronautical Laboratory, 1954.

[52] Tobak, M., Chapman, G., and Schiff, L., "Mathematical Modeling of the Aerodynamic Characteristics in Flight Dynamics," Tech. Rep. NASA TM-85880, NASA Langley Research Center, 1984.

[53] Beam, R. and Warming, R., "Numerical Calculations of Two-Dimensional, Unsteady Transonic Flows With Circulation," Tech. Rep. NASA TN D-7605, NASA Ames Research Center, Moffett Field, California, 1974.

[54] Ballhaus, W. and Goorjian, P., "Computation of Unsteady Transonic Flows by the Indicial Method," *AIAA Journal*, Vol. 16, No. 2, February 1978, pp. 117–124.

[55] Clancy, S. and Rugh, W., "A Note on the Identification of Discrete-Time Polynomial Systems," *IEEE Transactions on Automated Control*, Vol. 24, No. 6, December 1979, pp. 975–978.

[56] Silva, W. and Bartels, R., "Development of reduced-order models for aeroelastic analysis and flutter prediction using the CFL3Dv6.0 code," *Journal of Fluids and Structures*, Vol. 19, No. 6, 2004, pp. 729–745.

[57] Silva, W., "Simultaneous Excitation of Multiple-Input/Multiple-Output CFD-Based Unsteady Aerodynamic Systems," *Journal of Aircraft*, Vol. 45, No. 4, July-August 2008, pp. 1267–1274.

[58] Kim, T., "Efficient Reduced-Order System Identification for Linear Systems with Multiple Inputs," *AIAA Journal*, Vol. 43, No. 7, July 2005, pp. 1455–1464.

[59] Kim, T., Hong, M., Bhatia, K., and SenGupta, G., "Aeroelastic Model Reduction for Affordable Computational Fluid Dynamics-Based Flutter Analysis," *AIAA Journal*, Vol. 43, No. 12, December 2005, pp. 2487–2495.

[60] Gaitonde, A. and Jones, D., "Reduced Order State-Space Models from the Pulse Responses of a Linearized CFD Scheme," *International Journal for Numerical Methods in Fluids*, Vol. 42, No. 6, 2003, pp. 581–606.

[61] Allen, C., Taylor, N., Fenwick, C., Gaitonde, A., and Jones, D., "A Comparison of Full Non-linear and Reduced Order Aerodynamic Models in Control Law Design Using a Two-dimensional Aerofoil Model," *International Journal for Numerical Methods in Engineering*, Vol. 64, No. 12, 2005, pp. 1628–1648.

[62] Singh, R. and Baeder, J., "Direct Calculation of Three-Dimensional Indicial Lift Response Using Computational Fluid Dynamics," *Journal of Aircraft*, Vol. 34, No. 4, July-August 1997, pp. 465–471.

[63] Lind, R., Prazenica, R., and Brenner, M., "Estimating Nonlinearity Using Volterra Kernels in Feedback with Linear Models," *Nonlinear Dynamics*, Vol. 39, No. 1, 2005, pp. 3–23.

[64] Munteanu, S., Rajadas, J., Nam, C., and Chattopadhyay, A., "Reduced-Order-Model Approach for Aeroelastic Analysis Involving Aerodynamic and Structural Nonlinearities," *AIAA Journal*, Vol. 43, No. 3, March 2005, pp. 560–571.

[65] Raveh, D., "Reduced-Order Models for Nonlinear Unsteady Aerodynamics," *AIAA Journal*, Vol. 39, No. 8, August 2001, pp. 1417–1429.

[66] Raveh, D., Levy, Y., and Karpel, M., "Efficient Aeroelastic Analysis Using Computational Unsteady Aerodynamics," *Journal of Aircraft*, Vol. 38, No. 3, May-June 2001, pp. 547–556.

[67] Balajewicz, M., Nitzsche, F., and Festzy, D., "Application of Multi-Input Volterra Theory to Nonlinear Multi-Degree-of-Freedom Aerodynamic Systems," *AIAA Journal*, Vol. 48, No. 1, January 2009, pp. 56–62.

[68] Balajewicz, M., Nitzsche, F., and Feszty, D., "Reduced Order Modeling of Nonlinear Transonic Aerodynamics Using a Pruned Volterra Series," *Proceedings of the 50th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, AIAA Paper No. 2009-2319, May 2009.

[69] Kurdila, A., Prazenica, R., Rediniotis, O., and Strganac, T., "Multiresolution Methods for Reduced-Order Models for Dynamical Systems," *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 2, March-April 2004, pp. 193–200.

[70] Prazenica, R., , and Kurdila, A., "Multiwavelet Constructions and Volterra Kernel Identification," *Nonlinear Dynamics*, Vol. 43, No. 3, 2006, pp. 277–310.

[71] Khawar, J., Zhigang, W., and Chao, Y., "Volterra Kernel Identification of MIMO Aeroelastic System Through Multiresolution and Multiwavelets," *Computational Mechanics*, Vol. 49, No. 4, 2012, pp. 431–458.

[72] Marzocca, P., Silva, W., and Librescu, L., "Nonlinear Open-/Closed-Loop Aeroelastic Analysis of Airfoils via Volterra Series," *AIAA Journal*, Vol. 42, No. 4, April 2004, pp. 673–686.

[73] Ghoreyshi, M., Jirasek, A., and Cummings, R., "Computational Investigation into the Use of Response Functions for Aerodynamic-Load Modeling," *AIAA Journal*, Vol. 50, No. 6, June 2012, pp. 1314–1327.

[74] Lisandrin, P., Carpentieri, G., and vanTooren, M., "Investigation over CFD-Based Models for the Identification of Nonlinear Unsteady Aerodynamics Responses," *AIAA Journal*, Vol. 44, No. 9, September 2006, pp. 2043–2050.

[75] Lucia, D. and Beran, P., "Reduced-Order Model Development Using Proper Orthogonal Decomposition and Volterra Theory," *AIAA Journal*, Vol. 42, No. 6, June 2004, pp. 1181–1190.

[76] Lucia, D. and Beran, P., "Aeroelastic System Development Using Proper Orthogonal Decomposition and Volterra Theory," *AIAA Journal*, Vol. 42, No. 2, March-April 2005, pp. 509–518.

[77] Crowell, A. and McNamara, J., "Model Reduction of Computational Aerothermodynamics for Hypersonic Aerothermoelasticity," *AIAA Journal*, Vol. 50, No. 1, January 2012, pp. 74–84.

[78] Silva, R., Mello, O., Azevedo, J., Chen, P., and Liu, D., "Investigation on Transonic Correction Methods for Unsteady Aerodynamics and Aeroelastic Analyses," *Journal of Aircraft*, Vol. 45, No. 6, November-December 2008, pp. 1890–1903.

[79] Thomas, J., Dowell, E., and Hall, K., "Static/Dynamic Correction Approach for Reduced-Order Modeling of Unsteady Aerodynamics," *Journal of Aircraft*, Vol. 43, No. 4, July-August 2006, pp. 865–878.

[80] Lind, R., Prazenica, R., Brenner, M., and Baldelli, D., "Identifying Parameter-Dependent Volterra Kernels to Predict Aeroelastic Instabilities," *AIAA Journal*, Vol. 43, No. 12, December 2005, pp. 2496–2502.

[81] Prazenica, R., Reisenthel, P., Kurdila, A., and Brenner, M., "Volterra Kernel Extrapolation for Modeling Nonlinear Aeroelastic Systems at Novel Flight Conditions," *Journal of Aircraft*, Vol. 44, No. 1, January-February 2007, pp. 149–162.

[82] Omran, A. and Newman, B., "Piecewise Global Volterra Nonlinear Modeling and Characterization for Aircraft Dynamics," *Journal of Guidance, Control, and Dynamics*, Vol. 32, No. 3, May-June 2009, pp. 749–759.

[83] Omran, A. and Newman, B., "Full Envelope Nonlinear Parameter-Varying Model Approach for Atmospheric Flight Dynamics," *Journal of Guidance, Control, and Dynamics*, Vol. 35, No. 1, January-February 2012, pp. 270–283.

[84] Silva, W., "Recent Enhancements to the Development of CFD-Based Aeroelastic Reduced-Order Models," *Proceedings of the 48th*

AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, AIAA Paper No. 2007-2051, April 2007.

[85] Fung, Y., *An Introduction to the Theory of Aeroelasticity*, Dover Publications, Mineola, New York, 1993.

[86] Matheron, G., "Principles of Geostatistics," *Economic Geology*, Vol. 58, 1963, pp. 1246–1266.

[87] Lophaven, S., Nielsen, H., and Sondergaard, J., "DACE: A MATLAB Kriging Toolbox, Version 2.0," Tech. Rep. IMM-TR-2002-12, Informatics and Mathematical Modelling, Denmark, August 1, 2002.

[88] Martin, J. and Simpson, T., "Use of Kriging Models to Approximate Deterministic Computer Models," *AIAA Journal*, Vol. 43, No. 4, April 2005, pp. 853–863.

[89] McKay, M., Beckman, R., and Conover, W., "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code," *Technometrics*, Vol. 21, No. 2, May 1979, pp. 239–245.

[90] Cioppa, T. and Lucas, T., "Efficient Nearly Orthogonal and Space-Filling Latin Hypercubes," *Technometrics*, Vol. 49, No. 1, February 2007, pp. 45–55.

[91] Raveh, D., Levy, Y., and Karpel, M., "Aircraft Aeroelastic Analysis and Design Using CFD-Based Unsteady Loads," *Proceedings of the 41st AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference and Exhibit*, AIAA Paper No. 2000-1325, April 2000.

[92] Jones, R., Schonlau, M., and Welch, W., "Efficient Global Optimization of Expensive Black-Box Functions," *Journal of Global Optimization*, Vol. 13, No. 4, 1998, pp. 455–492.

[93] Glaz, B., Friedmann, P., and Liu, L., "Surrogate Based Optimization of Helicopter Rotor Blades for Vibration Reduction in Forward Flight," *Journal of Structural and Multidisciplinary Optimization*, Vol. 35, No. 4, 2008, pp. 341–363.

[94] Sacks, J., Welch, W., Mitchell, T., and Wynn, H., "Design and Analysis of Computer Experiments," *Statistical Science*, Vol. 4, No. 4, 1989, pp. 409–423.

[95] Anon., "Documentation for ANSYS ICEM CFD 12.0," Tech. rep., ANSYS, Canonsburg, Pennsylvania, 2009.

[96] Lighthill, M., "Oscillating Airfoils at High Mach Number," *Journal of the Aeronautical Sciences*, Vol. 20, No. 6, 1953, pp. 402–406.

[97] Jr., E. Y., "AGARD Standard Aeroelastic Configurations for Dynamic Response. Candidate Configuration 1.-wing 445.6," Tech. Rep. NASA TM-100492, NASA Langley Research Center, 1987.

[98] Raveh, D., "Identification of Computational-Fluid-Dynamics Based Unsteady Aerodynamic Models for Aeroelastic Analysis," *Journal of Aircraft*, Vol. 41, No. 3, May-June 2004, pp. 620–632.

[99] Cowan, T., A.S. Arena, J., and Gupta, K., "Accelerating Computational Fluid Dynamics Based Aeroelastic Predictions Using System Identification," *Journal of Aircraft*, Vol. 38, No. 1, January-February 2001, pp. 81–87.

[100] Munteanu, S., Rajadas, J., and Chattopadhyay, A., "Nonlinear Aeroelastic Analysis Using a CFD-Based State Space ROM," *Proceedings of the 47th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, AIAA Paper No. 2006-2024, May 2006.

[101] Silva, W. and Raveh, D., "Development of Unsteady Aerodynamic State-Space Models from CFD-Based Pulse Responses," *Proceedings of the 42nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference and Exhibit*, AIAA Paper No. 2001-1213, April 2001.

[102] Brigham, E., *The Fast Fourier Transform and Its Applications*, Prentice Hall, Englewood Cliffs, New Jersey, 1988.

[103] Shul, B., *Sled Driver: Flying the World's Fastest Jet, Third Edition*, Mach 1, Inc., Chico, California, 1991.

[104] Falkiewicz, N., Cesnik, C., Crowell, A., and McNamara, J., "Reduced-Order Aerothermoelastic Framework for Hypersonic Vehicle Control Simulation," *AIAA Journal*, Vol. 49, No. 8, August 2011, pp. 1625–1646.

[105] Anderson, J., *Fundamentals of Aerodynamics, Fourth Edition*, McGraw-Hill, New York, 2007.

[106] Hirsch, C., *Numerical Computation of Internal and External Flows: Fundamentals of Computational Fluid Dynamics, Second Edition*, Butterworth-Heinemann, New York, 2007.

[107] Bartels, R., Rumsey, C., and Biedron, R., "CFL3D Version 6.4 - General Usage and Aeroelastic Analysis," Tech. Rep. NASA/TM-2006-214301, NASA Langley Research Center, Hampton, Virginia, April 2006.

[108] Anon., "Tecplot 360 2011 User's Manual," Tech. rep., Tecplot, Inc., Bellevue, Washington, 2011.

[109] Anon., "MATLAB R2011a User Guides," Tech. rep., MathWorks, Inc., Natick, Massachusetts, 2011.