

Methods for the Calculation of Aerodynamic Models for
Flight Simulation

Thesis submitted in accordance with the requirements of
the University of Liverpool for the degree of Doctor in Philosophy
by
Andrew McCracken (BEng)

September 2014

Copyright © 2014 by Andrew McCracken

All rights reserved.

A model is a caricature that overemphasizes some features at the expense of others.

Emanuel Derman

Abstract

Flight dynamics analysis using computational models is a key stage in the design of aircraft. The models used in industry consist of two main parts. The first is a tabular aerodynamic model which is essentially a large database of aerodynamic data. The tabular aerodynamic model is a highly dimensional database containing aerodynamic loads and moments for different parameter combinations. In order to reduce the size of the tables, a number of assumptions are made. These include having sufficient resolution of the parameter space to capture the variation in the flow dynamics; decoupling certain parameters to reduce the dimensionality; using a single dynamic derivative, assuming independence from the flow conditions; and finally neglecting flow history effects which are dominant during manoeuvres with highly unsteady flow phenomena.

Secondly is the use of dynamic derivatives to simulate unsteady motion effects. These are calculated using small-amplitude forced oscillatory motions. In order to accelerate their computation, frequency domain methods are used. The Linear Frequency Domain and Harmonic Balance are two such methods used in this work. As part of the frequency domain calculations, linear solvers are used to provide solution to the frequency domain problem. These solvers use preconditioners to accelerate the time to solution. An alternative method of preconditioning is proposed in this work based on the first and second order spatial discretisation Jacobian matrices. It is shown that there is significant speed up achieved by varying the proportions of the first and second order terms in the preconditioner matrix.

In order to assess the performance of the tabular models, an initial assessment is carried out using a hierarchy of manoeuvres of increasing complexity. For each test case, the replay from the tabular model is compared with the fully unsteady time-accurate CFD solution. This is in line with a framework proposed in the literature. It is shown that the tabular model performs well through the linear aerodynamic regime, although breaks down where history effects become significant. The assessment continues with a study of each of the assumptions used to formulate the tables. Again a hierarchy of test cases of increasing complexity is used. Also used are both forced and free-response manoeuvres. It is shown that the resolution and coupling assumptions have little impact on the performance of the tabular model. The use of a single dynamic derivative is not shown to have an impact either, although it is suggested that for

more complicated manoeuvres, this could be important. Finally, the most significant error is introduced through neglecting history effects. It is shown that for manoeuvres where history effects dominate, such as those at the extremes of the flight envelope, the tabular model is not sufficient to effectively model the aerodynamics during these manoeuvres.

Acknowledgements

I would firstly like to thank my supervisors Professor Ken Badcock and Professor George Barakos for making it possible to carry out this work. Also to my examiners Dr Dorian Jones and Dr Mark White for their time in examining this thesis and for their input on improvements. As always, thanks must go to all those in the CFD lab past and present for making each day survivable and always a good laugh. The extra guidance from Dr Andrea Da Ronch, Dr Sebastian Timme and Dr David Kennett were invaluable in overcoming numerous problems, both intellectually and politically.

Thanks must also go to the Engineering and Physical Sciences Research Council (EPSRC) and Airbus Ltd for their funding of this project. Also, to the Flight Physics department at Airbus Filton for their input throughout. Further to this, the help of staff in DLR's Institute of Aerodynamics and Flow Technology, particularly Markus Widhalm, was invaluable in the work on the Linear Frequency Domain.

Final thanks must go to my family for their support from start to finish, particularly to my wife Laurette, without whom I would neither have started nor finished this thesis.

Declaration

I confirm that the thesis is my own work, that I have not presented anyone else's work as my own and that full and appropriate acknowledgement has been given where reference has been made to the work of others.

Andrew McCracken

September 2014

List of Publications

Da Ronch, A., McCracken, A., Badcock, K.J., “Assessing the Impact of Aerodynamic Modelling on Manoeuvring Aircraft,” *AIAA Science and Technology Forum and Exposition 2014*, National Harbor, MD, 13–17 Jan 2014.

McCracken, A., Badcock, K.J., “Uncertainty Quantification in Flight Dynamics Modelling,” *ECCOMAS Young Investigators Conference 2013*, Bordeaux, 2–6 Sept 2013.

McCracken, A., Kennett, D.J., Badcock, K.J., “Assessment of Tabular Models Using CFD,” *AIAA Atmospheric Flight Mechanics Conference*, Boston, 19–22 Aug 2013.

Da Ronch, A., McCracken, A., Badcock, K.J., Widhalm, M., and Campobasso, M.S., “Linear Frequency Domain and Harmonic Balance Predictions of Dynamic Derivatives,” *Journal of Aircraft*, Vol. 50, No. 3, pp 694–707, 2013.

McCracken, A., Da Ronch, A., Timme, S., Badcock, K.J., “Solution of Linear Systems in Fourier-Based Methods for Aircraft Applications,” *International Journal of Computational Fluid Dynamics*, Vol. 27, No. 2, pp 79–87, 2013.

McCracken, A., Akram, U., Da Ronch, A., Badcock, K.J., “Requirements for Computer Generated Aerodynamic Models for Aircraft Stability and Control Analysis,” *5th Symposium on Integrating CFD and Experiments in Aerodynamics*, Tokyo, October 2012.

McCracken, A., Timme, S., Badcock, K.J., “Accelerating Convergence of the CFD Linear Frequency Domain Method by a Preconditioned Linear Solver,” *ECCOMAS 2012*, Vienna, September 2012.

Da Ronch, A., McCracken, A., Badcock, K.J., Ghoreyshi, M., and Cummings, R.M., “Modeling of Unsteady Aerodynamic Loads,” AIAA–2011–2376, *AIAA Atmospheric Flight Mechanics Conference*, Portland, Oregon, 8–11 Aug 2011.

Contents

Abstract	v
Acknowledgements	vii
Declaration	ix
List of Publications	xi
List of Figures	xv
List of Tables	xvii
Nomenclature	xix
1 Introduction	1
1.1 Computational Aerodynamic Models	1
1.2 Frequency Domain Methods	5
1.3 Preconditioning of Linear Systems	8
1.4 Summary	10
2 Formulation	13
2.1 CFD Solvers	13
2.1.1 Governing Flow Equations	13
2.1.2 Unsteady Solution	14
2.1.3 DLR TAU-code	15
2.1.4 Parallel Meshless (PML)	16
2.2 Control Surface Modelling	17
2.3 Tabular Models	19
2.3.1 Dynamic Derivatives	20
2.4 Frequency Domain Methods	21
2.4.1 Linear Frequency Domain	21
2.4.2 Harmonic Balance	23
2.4.2.1 Implicit Solution	25

2.4.3	ILU Preconditioner	26
3	Performance of Frequency Domain Methods	29
3.1	Linear Frequency Domain	29
3.1.1	Solver Options	29
3.1.2	Implicit LFD	31
3.1.3	ILU _α Preconditioner	42
3.2	Harmonic Balance	47
3.2.1	TAU Implementation	47
3.2.2	PML Implementation	48
4	Tabular Aerodynamic Model	51
4.1	Background	51
4.2	CFD Validation	53
4.3	Dynamic Derivatives	54
4.4	2D Aerofoil	55
4.4.1	Manoeuvres	55
4.5	2D Aerofoil with Control Surface	58
4.5.1	Manoeuvres	59
4.6	LANN Wing	66
5	Assessment of Assumptions	69
5.1	Tabular Model Assumptions	69
5.2	Forced	72
5.3	Free Response	84
5.3.1	Simulation	85
5.3.2	Control Inputs	87
5.4	Dynamic Derivative Error Quantification	90
5.4.1	Error Estimation	91
5.4.2	Application	91
6	Conclusions	95
6.1	Future Work	97
	Bibliography	99
S	Supplementary Information – Implicit Implementation of LFD	105
S.1	Matrix Storage	105
S.2	Setting up the System Matrix	106
S.3	Parallelisation	107
S.4	Memory Reduction	107

List of Figures

1.1	Flow conditions of interest	3
2.1	Unit cube control volume [1]	13
2.2	Regular grid and associated dual-cells	16
2.3	Point cloud with ellipse	17
2.4	NACA 0012 aerofoil point distribution with underlying structural model	18
3.1	NACA 0012 grids	32
3.2	NACA 64A010 grid	33
3.3	SDM grid	33
3.4	Goland wing grids	34
3.5	LANN wing grid	34
3.6	NACA0012 fine AGARD CT2 solution comparison	36
3.7	NACA0012 fine AGARD CT5 solution comparison	36
3.8	Convergence comparison	38
3.9	Parallel performance (Goland Euler)	41
3.10	Real vs Complex solver convergence (Goland Euler)	42
3.11	Second order preconditioner comparison	43
3.12	First order preconditioner	44
3.13	Weighted preconditioner $\alpha = 0.90$	44
3.14	Influence of preconditioner weighting on convergence	45
3.15	Convergence and Real Positive eigenvalue (NACA 0012 fine AGARD CT2)	46
3.16	Error from exact solution (NACA 0012 fine AGARD CT2)	47
3.17	Pitching moment coefficient (NACA 0012 fine AGARD CT2)	48
3.18	NACA 0012 point distributions	49
3.19	Pitching moment coefficient loops from PML HB	49
3.20	Speed up of PML HB	50
4.1	CFD validation (McCroskey case 8)	54
4.2	NACA 0012 RANS point distribution	55
4.3	Ramp manoeuvre replay $2^\circ/\text{s}$	56
4.4	Ramp manoeuvre replay $10^\circ/\text{s}$	56

4.5	Oscillatory manoeuvre replays	57
4.6	NACA 0012 point distributions	58
4.7	NACA 0012 point distributions with underlying structure	59
4.8	Ramp motion at $10^\circ/s$	60
4.9	Pressure coefficient distribution at two locations in the manoeuvre at Mach 0.8	62
4.10	Pressure coefficient distribution at an incidence of -2.0° in the manoeuvre at Mach 0.8	62
4.11	Obstacle avoidance manoeuvre(Euler)	64
4.12	Obstacle avoidance manoeuvre (RANS)	65
4.13	LANN wing grid	67
4.14	LANN wing replay comparison	67
5.1	Effect of resolution on lift coefficient prediction for aerofoil with control surface at Mach 0.8	71
5.2	Effect of input parameters in pitch damping derivative	72
5.3	Aerofoil with control surface ramp replays with and without coupling	73
5.4	Aerofoil with control surface obstacle replays with and without coupling in the derivative	75
5.5	LANN Wing: coupled and decoupled replays	76
5.6	Aerofoil table resolution comparison	78
5.7	LANN table resolution comparison	79
5.8	Aerofoil: Effect of dynamic derivative value on replay	80
5.9	LANN Wing: Effect of dynamic derivative value on replay	81
5.10	Oscillatory manoeuvre replays with quasi-steady	82
5.11	Turbulent eddy viscosity at 19.24° on the upstroke	83
5.12	Turbulent eddy viscosity at 10.0° on the downstroke	84
5.13	Effect of history on the replay	85
5.14	Two degree of freedom aerofoil [2]	86
5.15	Free-response control inputs	87
5.16	Tabular replays with and without coupling	88
5.17	Table resolution comparison	88
5.18	Effect of pitch damping derivative value on replay	89
5.19	Lift damping derivative variation with flight parameters	90
5.20	$M = 0.3, \alpha_0 = 0.0^\circ, \alpha_A = 5.0^\circ, k = 0.001$	92
5.21	$M = 0.3, \alpha_0 = 0.0^\circ, \alpha_A = 5.0^\circ, k = 0.1$	93
5.22	$M = 0.3, \alpha_0 = 8.0^\circ, \alpha_A = 5.0^\circ, k = 0.001$	93
5.23	$M = 0.3, \alpha_0 = 8.0^\circ, \alpha_A = 5.0^\circ, k = 0.1$	94

List of Tables

2.1	Example Aerodynamic Table (x indicates non-zero entry)	20
3.1	AGARD test case conditions (Euler)	32
3.2	Solver Parameters	35
3.3	Solver speed test results	37
3.4	PETSc option test cases	38
3.5	PETSc option test results NACA 0012 AGARD CT2	39
3.6	Solver memory requirement	40
3.7	Solver memory requirement relative to steady state	40
3.8	Augmentation comparison (Goland Euler)	42
3.9	TAU-HB memory requirement (NACA 0012 fine AGARD CT2)	48
3.10	PML-HB memory requirement (MB)	50
4.1	Example Aerodynamic Table (x indicates non-zero entry)	51
4.2	Reduced table for Mach and incidence	52
4.3	Reduced table for Mach and control surface deflection	52
4.4	Control surface ramp manoeuvre parameters	60
4.5	Control surface obstacle manoeuvre parameters	63
4.6	LANN wing manoeuvre parameters	66
5.1	Coupled Aerodynamic Table (x indicates non-zero entry)	69
5.2	Decoupled Aerodynamic Table for sideslip (x indicates non-zero entry)	69
5.3	Decoupled Aerodynamic Table for control surface deflection (x indicates non-zero entry)	69
5.4	Coupled and decoupled table parameter ranges	72
5.5	LANN coupling manoeuvre parameters	76
5.6	Aerofoil with control surface table resolutions per Mach number	77
5.7	LANN wing table resolutions per Mach number	78
5.8	NACA 0012 dynamic derivative table parameters	92

Nomenclature

Symbols

\tilde{A}	Approximation of A
A	System matrix
A_α	Mixed order Jacobian matrix
A_f	First order spatial accurate fluid Jacobian
A_s	Second order spatial accurate fluid Jacobian
\mathbf{b}	Right hand side vector
C_L, C_D, C_Y	Force coefficients (lift, drag, side-force)
$C_{L\alpha}$	Lift-curve slope (1/rad.)
C_{Lq}	Lift damping derivative (1/rad./s)
C_l, C_m, C_n	Moment coefficients (roll, pitch, yaw)
$C_{m\alpha}$	Pitching moment slope (1/rad.)
C_{mq}	Pitch damping derivative (1/rad./s)
c	Chord length (m)
h	Plunge height (m)
I_y	Moment of inertia about pitch axis (kg.m ²)
k	Reduced frequency
L	Lift force (N)
L_h	Change in lift due to heave (N/m)
$L_{\dot{h}}$	Change in lift due to heave velocity (N/m/s)
L_α	Change in lift due to pitch (N/rad.)
$L_{\dot{\alpha}}$	Change in lift due to pitch rate (N/rad./s)
M	Mach number
M_h	Change in pitching moment due to heave (N.m/m)
$M_{\dot{h}}$	Change in pitching moment due to heave velocity (N.m/m/s)
M_α	Change in pitching moment due to pitch (N.m/rad.)
$M_{\dot{\alpha}}$	Change in pitching moment due to pitch rate (N.m/rad./s)
m	Mass (kg)

m	Number of Krylov subspace vectors
N_H	Number of harmonics retained
P	Preconditioner matrix
P_α	Preconditioner based on mixed order Jacobian
q	Pitch rate (rad./s)
R	Residual
r	Krylov residual vector
Re	Reynolds' number
t	Time (s)
U_∞	Free stream velocity (m/s)
$\bar{\mathbf{w}}$	Steady state component
$\hat{\mathbf{w}}$	Vector of Fourier coefficients
w	Vector of conserved flow variables
$\tilde{\mathbf{w}}$	Perturbation component
x	Grid point coordinates
$\dot{\mathbf{x}}$	Grid point velocities

Greek Symbols

α, β	Angle of attack, Sideslip ($^\circ$)
α_0	Mean incidence
$\Delta\alpha$	Amplitude of oscillation
$\delta_{ele}, \delta_{ail}, \delta_{rud}$	Control surface deflections (elevator, aileron, rudder) ($^\circ$)
ϵ	Amplitude factor for finite difference step
ω	Reduced circular frequency

Subscripts

α	Weight applied to second order Jacobain terms
f	First order spatial accuracy
s	Second order spatial accuracy

Acronyms

BCSR	Block Compressed Sparse Row
CGS	Conjugate Gradients
CSR	Compressed Sparse Row
GCR	Generalised Conjugate Residual
GMRes	Generalised Minimal Residual
HB	Harmonic Balance
ILU	Incomplete Lower Upper

LFD	Linear Frequency Domain
LU	Lower Upper
LU-SGS	Lower Upper Symmetric Gauss Seidel
NLFD	Non-Linear Frequency Domain
PETSc	Portable Extensible Toolkit for Scientific Computation
RANS	Reynolds Averaged Navier-Stokes
RCM	Reverse Cuthill-McKee
SDM	Standard Dynamics Model
TFQMR	Transpose-Free Quasi-Minimal Residual

Chapter 1

Introduction

1.1 Computational Aerodynamic Models

Industrial practice is seeing ever increasing use of computer simulations for flight dynamics analysis. The aerodynamic models used in the simulation must be fit for purpose to ensure reliable results. The tabular aerodynamic model is one such computational model that is used frequently as part of aircraft loads assessment during the design phase, as well as in onboard control systems. Despite the frequent and long term use of this model, it has not been fundamentally assessed for civil domain problems.

Computational models used for flight simulation consist of a number of components. These typically include an aerodynamic database, a method to account for unsteady effects and a method to include the flight mechanics, in order to model the aircraft response for given loads and moments. The aerodynamic database contains the force and moment coefficients for a given parameter set covering the flight envelope, which are obtained by empirical or computational methods. For manoeuvres where the rates become significant, or where the aerodynamics begin to deviate from the linear regime, unsteady modelling is required. A number of approaches have been proposed for this which will be discussed shortly. The final part is that of the flight mechanics modelling. The equations of motion are set up for the given configuration and describe the response of the loads and moments present at each point within the manoeuvre. The manoeuvre is then simulated by stepping through time and moving the aircraft to its new position until a complete trajectory can be traced.

The aerodynamic database is stored as large tables and forms the tabular aerodynamic model. These tables can be in the order of millions in terms of required data points. It therefore becomes necessary to have calculation methods which minimise the time to form the database. One such study is that from Ghoreyshi et al. [3]. In this paper, a method is presented to accelerate the generation of the aerodynamic model through the use of a Kriging-based sampling technique. Two scenarios were considered to test the method. The first was to make use of Kriging interpolation, more specif-

ically calculating a confidence interval for the predictor, in order to determine where samples should be taken. This approach enables nonlinearities to be better captured in the parameter space by locating high-fidelity simulations in these regions. The second scenario was that of a changing geometry. A data fusion approach was used, whereby the original geometry aerodynamic model was augmented by a few high-fidelity samples for the new configuration. The new aerodynamic model was then used for the new configuration. This approach has the benefit of only requiring a few expensive simulations to obtain an updated model. It was shown that, for the case presented, the required number of samples in the first scenario was reduced from 2000 to 35, and for the second scenario, down to just 10 samples.

Work by Da Ronch et al. [4] used the method described above for further test cases. The focus of this work was on the applicability of the models for flight dynamics simulation. Again, a number of aerodynamic models were used with a hierarchy of fidelities as the source of data including high-fidelity CFD. The method was assessed using five different test cases across a range of regimes. All the models were full aircraft of conventional and unconventional configurations. The performance of the aerodynamic model was compared with wind tunnel and flight test data and it was shown that the model was suitable for the cases presented.

A further study by Mackman et al. [5] looked to the use of surrogate models to reduce the number of samples required to generate an aerodynamic model. In this work, CFD simulations were used as the source of aerodynamic data. Two interpolation approaches were used to create the surrogate models. The first was the same Kriging method used in the previously described studies. The second made use of radial basis functions (RBF). The sample locations were then chosen based on the mean squared error of the interpolant. The two approaches were tested on the DLR-F12 aircraft and an RAE 2822 aerofoil. It was shown that both methods required fewer points to form the aerodynamic model than space-filling techniques.

Although efforts have been made to reduce the computational effort to generate the tables, there has been limited work into what the limits of the tabular model are. A framework to establish the limits was proposed in [6]. In a number of previous studies, tabular models have been validated against flight test data, however, this work proposed the use of CFD simulations as the baseline comparator. The idea is to run a manoeuvre using both the tabular model and the CFD time-accurate solver, any discrepancies between the two solutions are considered as limitations of the tables. This methodology was tested using the AGARD standard dynamics model fighter configuration for a number of manoeuvres. The tabular model performed well for slow motions, but began to break down for faster motions at high angles of attack. The discrepancies were ascribed to vortical flow hysteresis. It was also shown in this work that the addition of the dynamic stability derivatives to modify the static tabulated data, improved the accuracy of the tabular replay solution.

This framework was then used by Vallespin et al. in [7]. The assessment was extended to additional test cases, and for a wider range of manoeuvres, with the main focus on the use of an unmanned combat air vehicle (UCAV). This case was designed to cover a large flight envelope, which at high angles of attack, included complex vortical flow adding to the difficulty in modelling the unsteadiness. Again it was seen that there was good agreement across most of the manoeuvres, although when very high rates were present, the tabular model began to breakdown. The fundamental assumptions in the tabular model are mentioned as possible sources of error, although they are not individually assessed.

There are a number of examples where the tabular model is no longer adequate but little work has been carried out to assess the fundamental assumptions in the formulation of this model, namely resolution of tables, decoupling parameters, quasi-steady modelling and dynamics modelling. Initial work to assess the assumptions was presented in [8], where an aerofoil case was taken with no control surface to better understand the performance of the tables through a number of regimes as shown in Fig. 1.1.

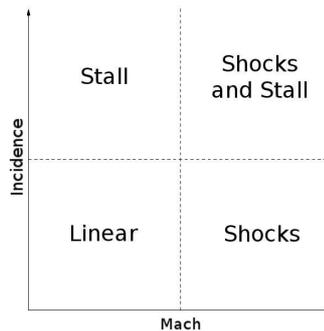


Figure 1.1: Flow conditions of interest

Manoeuvres in the linear portion of the figure, which was up to around Mach 0.5 and incidence = 10.0° for the case used, were well modelled. However, in the shock and stall region, which was the dynamic stall case in the paper, large discrepancies began to show displaying the inadequacy of modelling the time history. The tables used were two-dimensional across a small parameter space, and as a result meant it was not possible to properly assess the other assumptions. Also assessed was the dependence of the value of the dynamic stability derivatives on the Mach number, incidence, amplitude of oscillation and reduced frequency. It was shown that through the linear regime, the dynamic derivatives remained fairly constant in value, although again when there was large unsteadiness, the values began to vary.

A number of unsteady aerodynamic models for flight dynamics applications have been proposed. A review of several methods was carried out in [9] with a companion paper [10] providing practical examples of the methods using a delta wing at high angles of attack. The models reviewed consisted of a hierarchy of complexities. The most

simple was that of the dynamic stability derivative model. This was shown in previous work by the author to be inadequate for combat aircraft manoeuvre simulation. It is however used as the base method to which others are compared. The next approach is that of a Volterra series [11]. This approach is similar to the dynamic derivative method in that it expands the coefficients as a series. However, in the Volterra series, there is a method to account for time history effects in the Volterra kernel. This kernel describes how the output varies with changes in the inputs through time and must be computed using training data. The source of this data can be empirical or computational such as from CFD. A further set of methods come under Indicial approaches [12, 13]. Indicial approaches can be either linear or nonlinear in nature. The linear methods consider the response of an aerodynamic load to be linearly varying with the forcing function. For example, the lift coefficient has a constant variation with the change in incidence. This is limited in its approach due to the variation changing as the aerodynamics become more complex, such as at high angles of attack. This can be worked around by using the nonlinear indicial methods. The difference from the linear method is that the variation of the output with the input is computed for a number of input values. This provides a history of responses. It was shown that the traditional stability derivative approach was not sufficient to capture the unsteady flow phenomena and that the indicial methods were much more suitable.

A further analysis of unsteady modelling approaches with a focus on flight dynamics was carried out in [14], which covered some of those previously mentioned, although extended the work to include the Surrogate-Based Recurrence Framework (SBRF) [15]. The SBRF method requires forming a surrogate model to describe how the outputs, aerodynamic loads and moment, relate to the inputs. The surrogate is formed using a number of CFD solves as training data, with a specific number of historical solutions used to describe the variations. Increasing the number of historical data points, improves the approximation power of the model. Once the surrogate is formed, the inputs can be prescribed, and the outputs determined easily. The test case was a pitching NACA 0012 aerofoil in the transonic regime. The performance of each method was compared to a time-accurate CFD solution. It was shown that the SBRF model was best able to model the flow phenomena, and at a similar cost to the conventional stability derivative approach.

Although each of the above have their benefits, the most widely used in practice remains the stability derivative model originally proposed by Bryan [16]. This model is based upon the assumption that the load and moment coefficients can be broken down into a steady state component and an unsteady component (i.e. changes in the values are relative to the rate of the motion). The application of this method is discussed later.

A number of other papers have explored the variation of the dynamic derivative with the flight parameters at which it is calculated. Work by Bratt and Wight [17] in

the 1940s used oscillatory wind tunnel tests with various aerofoil cross sections to view the effect of the flight conditions on the dynamic derivatives. It was shown that there can be significant variation with the oscillatory frequency, oscillatory amplitude, mean incidence, location of the axis of oscillation, Reynolds number and the aerofoil profile. It was particularly noted that the pitch damping derivative can vary between negative and positive values with changes in the mean incidence. This is of importance around the stall, where the difference between a damped (negative derivative) and under-damped (positive derivative) system can be significant. Further work was carried out by Greenwell in [18]. In this work, there was a focus on the effect of the oscillatory frequency on the values of the static and dynamic stability derivatives. Again this was done using wind tunnel tests. A delta wing test case was used, where it was shown that the frequency of oscillation can have a significant effect on the dynamic derivative values, particularly at high angles of attack. When a manoeuvre simulation is carried out, a single value for the dynamic derivative is usually taken, which is assumed to be independent of the flight parameters. This is however not the case.

1.2 Frequency Domain Methods

Running full-order time-accurate unsteady CFD calculations to obtain all the dynamic derivative values required for a complete aircraft flight envelope assessment, can be very computationally expensive. The periodicity in the simulations used to compute the derivative values allow the use of frequency domain methods to accelerate the computation time. These were initially developed for turbomachinery flows [19, 20]. The methods have been adapted for use in many areas of aerospace including the prediction of unsteady air loads [21], flutter analyses [22] and on the application of predicting dynamic derivatives for flight dynamics [23]. These methods make use of the periodic nature of certain simulations, to allow a frequency domain representation of the solution. This has the benefit of being able to compute the periodic state directly, without the need to resolve an initial transient. Two methods are used in this work. The first is the Linear Frequency Domain (LFD) method. LFD assumes periodic, small amplitude variations about a steady state, which allows linearisation of the time-accurate flow equations and subsequent solution in the frequency domain. This is described further in Section 2.4.1. The second is the Harmonic Balance (HB) approach which models the flow equations as a Fourier series, and truncates this to a specified number of harmonics. This is described in Section 2.4.2.

The implicit solvers used to solve the frequency domain problems in this work, make use of linear solution methods. A Krylov subspace method is used with a preconditioner to accelerate the convergence of the system. The preconditioner is widely accepted as the most significant part in the acceleration of linear solvers. Available methods can be unstable for certain problems, particularly when indefinite matrices occur. Stabilisation

of the incomplete factorisation preconditioner has received attention with many papers being published, although the outcomes are usually only focused on stabilising for a particular type of matrix. In this thesis, a method of stabilisation is explored for general cases arising from CFD problems.

The Linear Frequency Domain method was originally developed for use in turbomachinery flows by Hall and Crawley [24] as a small disturbance Euler method. This paper set out to address the limitations of linearisation of the unsteady problem, namely the assumptions of isentropy and irrotationality of the flow, to allow shock waves to be accurately predicted. The method was demonstrated using a hyperbolic channel, and a cascade. The results from the cascade are of most interest here, featuring a pitching aerofoil section. The method was compared against incompressible analytical solutions. The method showed very good agreement in the pressure plots with just a little over-estimation in the real and imaginary components of pressure.

The linearized Euler equations have been used for CFD solution. A small disturbance Euler equations solver which was developed by Kreiselmaier and Laschka in [25], and which was then developed for use with the Navier-Stokes equations by Pechloff and Laschka in [21]. The underlying flow solver was Technische Universität München's FLM Navier-Stokes solver. The method was demonstrated on sinusoidally pitching NACA 64A010 and NLR 7301 aerofoils at AGARD CT8 and CT5 [26] viscous conditions respectively. The results from the FLM.SD.NS method were compared against the FLM.NS and FLOWer codes showing good agreement even for the stronger shock dynamics in the CT5 case.

The LFD formulation is derived from the same principles as the small disturbance Navier-Stokes method however, the perturbations are transformed into the frequency domain to allow subsequent solution of a linear system in the form $\mathbf{Ax} = \mathbf{b}$. This has been implemented within the DLR TAU-code. It was first presented by Widhalm et al. [22] being demonstrated on both 2D and 3D cases including the full DLR-F12 aircraft model. The method is run on an inviscid subsonic heaving NACA 0012 case, a viscous transonic pitching NACA 64A010 case, the viscous transonic pitching LANN wing and a viscous transonic DLR-F12 full aircraft case. The paper shows very good performance of the LFD method in terms of predicting the pressure distributions against the RANS calculations for all cases. There are only small discrepancies at large gradients, particularly at the peaks for the LANN wing. A limitation of this method is in the formulation of the right-hand side of the linear system using finite differences which introduces problems when choosing a general finite difference step size. This method is the basis for the work presented in this thesis.

In [23], Da Ronch et al. carried out a study of the LFD (and Harmonic Balance) methods for their ability to predict the values of dynamic derivatives in place of unsteady time-accurate calculations. Both 2D and 3D test cases are used, from aerofoils up to the DLR-F12 model. The paper found that an order of magnitude improve-

ment in CPU time is achieved using the LFD method, whilst retaining the main flow features to provide a good estimate of the dynamic derivatives. This paper offers a comprehensive study of the methods for several test cases.

The Harmonic Balance was originally proposed for modelling unsteady nonlinear flows by Hall et al [27]. The purpose of this paper was to solve the Navier-Stokes and Euler equations in Harmonic Balance form to then be able to solve these as a steady state problem when applied to turbomachinery cascade flows. The method involves expanding the flow unknowns in a Fourier series and then retaining a limited number of harmonics. The method is validated on a front stage compressor rotor in viscous transonic conditions with an inflow Mach number of 1.27 and Reynold's number of 1.35×10^6 . The rotor blades undergo a harmonic pitching motion with a reduced frequency of 1.0 and pitch amplitudes of 0.01° and 1.0° . For the smaller amplitude case, it is shown that even using one harmonic is sufficient to represent the flow due to nonlinear effects being very small for this case. The larger amplitude pitch however requires at least three harmonics to represent the flow to within engineering accuracy. The method was found to be between one and two orders of magnitude quicker in terms of CPU time than the equivalent unsteady time-accurate computation.

In [28], an implicit version of the Harmonic Balance technique was described for use in flight dynamics analysis. The purpose of implementing the implicit solver was to speed up the solution time by removing the reliance on explicit convergence acceleration methods such as Multigrid. The new solver is tested using a pitching NACA 0012 aerofoil under AGARD CT1 conditions and a pitching F-5 wing with a wing tip launcher and missile at Mach 0.896, $\alpha=0.004^\circ$, $\Delta\alpha=0.117^\circ$ and $k = 0.275$. It was shown that for both cases, one harmonic was sufficient to obtain accurate solutions at certain points in the pressure plots and moment loops, but that accurate reconstruction through the whole cycle required higher numbers of harmonics. The Harmonic Balance technique was shown to be an order of magnitude quicker in terms of CPU time than the equivalent unsteady time-accurate calculation. The paper sets out to use the method to generate the dynamic terms for flight simulation and the use for calculating dynamic derivatives is listed as future work.

In [29], He develops a version of the Harmonic Balance technique which takes the unsteady perturbations with first order accuracy and retains only the first harmonic. The purpose of this paper was to develop a nonlinear harmonic methodology to calculate unsteady viscous flows applied to turbomachinery. The method was validated using three different test cases. The first was a flat plate with both laminar and turbulent boundary layers. The second was a transonic diffuser with oscillating back pressure and the final case was an oscillating transonic compressor cascade. For all cases, the new method showed very good agreement with the experimental results and a greatly improved accuracy compared to the fully linear methods, which this was designed to replace. A speed up of around 5 was demonstrated compared to the equivalent unsteady

time-accurate calculation.

The Non-Linear Frequency Domain (NLFD) method was first proposed by McMullen in [30] and [31]. The NLFD method is very similar to that of LFD but does not linearise the problem. Instead, a number of harmonics of the flow are retained thus being able to resolve a greater number of the non-linear flow features at higher frequencies. The motivation for this research was to improve the solver technologies for calculating the direct periodic state of a flow. The NLFD formulation is very similar to that of the Harmonic Balance technique. The method was validated for two cases; first a cylinder undergoing vortex shedding then a pitching NACA 64A010 aerofoil at AGARD CT6 conditions [26]. It was shown that for the vortex shedding case, the NLFD method compares well with experimental results when three or more harmonics are retained. The same is shown for the aerofoil case. It was stated that the NLFD method is an order of magnitude more efficient than dual-time stepping methods.

1.3 Preconditioning of Linear Systems

As part of the implicit methods used in solving the LFD and HB problems, linear solvers are employed with preconditioners. For iterative linear solvers, the rate of convergence is strongly influenced by the preconditioning strategy employed [32]. There are many methods of preconditioning of which Incomplete Lower–Upper (ILU) is widely considered one of the most effective for Krylov type solvers. A review of preconditioning techniques until 2002 was carried out by Benzi [33]. This review will cover parts of the pre-2002 literature directly related to this work, and the published research since. A comprehensive review of ILU preconditioning methods is described in Saad [34].

Nejat [35] assesses the effect of fill-in for an ILU preconditioner when applied to 2nd, 3rd and 4th order accurate spatial discretisations. The preconditioning efficiency and memory requirement are compared to that of using a direct LU preconditioner. A NACA 0012 aerofoil and a 15% thick diamond aerofoil were chosen as test cases at high Mach numbers, in order to generate systems that were difficult to solve. To obtain a good initial guess, some implicit iterations were run before the linear solve took place to enhance the stability of the solver, particularly for the higher-order schemes. A preconditioner based on the first-order discretisation was used for all cases due to the memory requirement of storing the higher-order matrices. A GMRes Krylov solver was used. The baseline solution used LU preconditioning, against which the efficiency of the ILU method was compared. It was found that ILU with four levels of fill-in provided a rate of convergence very similar to that of the exact LU decomposition, but with significantly reduced memory requirements. Two levels of fill-in were recommended for most cases as the most efficient.

Dwight [36] carried out a similar assessment as above with a GMRes-ILU solver, but using an RAE 2822 aerofoil in the viscous transonic regime. The DLR TAU code

was used for this. The same conclusion was reached, with ILU(4) being most effective, but memory intensive. A comparison with the LU-SGS implicit solver was also made, showing the linear solver to be superior, converging twice as many orders of magnitude in the L2 norm of the density residual in the same CPU time of 20,000 secs.

Duff and Meurant [37] carried out a study of a number of reordering techniques for bandwidth optimisation and reduced fill-in. More importantly, they introduced the use of the Frobenius norm of the residual matrix $R = A - LU$, to help determine the convergence of the system through the accuracy of the incomplete factorisation. This has been used in a number of subsequent papers. Various orderings were used on four different model problems arising from Laplace's equation with different boundary conditions. Results showed that the number of iterations for convergence is directly related to the norm of the residual matrix.

A key paper was written by Chow [38] in which the author tried to establish conditions for the failure of ILU preconditioners. Test matrices were chosen to be of varying difficulty to solve due to a mix of zero pivots and unstable triangular solves, along with matrices of various sizes and sparsity patterns. Three parameters were established to analyse the effectiveness of the ILU factorisation. The first was the condition number of $(LU)^{-1}$, which gives insight into the stability of the triangular solve. The second was the value of $1/pivot$, which finds small or zero pivots. Finally, the size of the largest element in the L and U factors gives information on inaccuracies due to the dropping strategy used. It was decided that the condition estimate was the most useful of the three statistics, although this can be very expensive to compute for large matrices. A framework was developed to give reasons for the preconditioner break-down based on the condition estimate, and the value of $1/pivot$. An analysis was then carried out using this framework for a number of different ILU preconditioners, giving the reason for failure. This paper concluded that there is no generalisable ILU factorisation, but that there are several methods that can be tried to converge a system. Even though reasons were given for each failed case, there is no explanation as to why the factorisations become unstable.

ILU preconditioners have proven to be useful across a wide range of problems. However, when certain types of matrix, such as indefinite matrices are encountered, the stability of the preconditioner can become poor and even make the condition number of the input matrix worse. This leads to the search for methods to stabilise the preconditioner. Stabilisation methods typically involve some form of permutation of the diagonal terms.

In [39], a number of stabilisation methods were looked at. The method chosen was dependent on the properties of the matrix, as per the framework in [40]. The purpose was to improve the diagonal dominance of the matrices arising from convection dominated problems. The main idea for the stabilisation involved looking at the values within the finite difference stencils. Ratios were calculated between stencil points,

and depending on the value, a quantity would be subtracted from the diagonal term. This method proved very beneficial for the case used, and performed better than the standard ILU preconditioner, although it was not as robust.

Chapman [41] looked at scaling of matrix terms to improve the diagonal dominance. CFD problems were the focus. A number of approaches were studied. One method took the terms close to the diagonal as a way to reduce bandwidth and improve diagonal dominance. Another method involved scaling of the diagonal blocks by adding a multiple of the identity matrix. Matrices were looked at from the Harwell-Boeing and FIDAP libraries as well as others from CFD simulations. The stabilisation proved effective for the more difficult CFD matrices, although across the majority of cases, there was little difference in the performance compared with standard ILU, when equivalent levels of fill-in were used.

An approach to stabilisation taken by Duff et al. [42], which has been further analysed in [43] and [44], looked to improve the diagonal dominance of the matrices, but this time with both reordering of the values and scaling. Both direct and iterative methods, along with preconditioning are all described as potential beneficiaries. The method involves making sweeps across all the matrix terms within each row, and searching for the largest terms. These are then permuted to the diagonal, with an entry added to the permutation matrix for later use in the solver. It was shown in [42, 43, 44] that the stabilisation is effective, being able to converge systems which would not converge with standard ILU, and in some cases improved the convergence by more than one order of magnitude.

Finally, a paper by Pueyo and Zingg [45], and used by Wong and Zingg [46], looked to the use of Newton-Krylov solvers for the calculation of aerodynamic flows. The preconditioning is done with the use of a level of fill based ILU method, which is reordered using Reverse Cuthill-McKee. An ILU(0) preconditioner based on a second order Jacobian matrix, and an approximate Jacobian Matrix were tested. The latter was shown to have better convergence properties. The approximation used was to take the first order discretisation, and then add the numerical dissipation terms, where the dissipation coefficient is a linear combination of the second order coefficient, and the inverse fourth order coefficient. A weighting is applied to the fourth order term.

1.4 Summary

Based on the current literature, there is a gap in the knowledge about the adequacy of the tabular aerodynamic model with dynamic derivatives. The work carried out in this thesis looks to fill this gap. A systematic study of the fundamental behaviour is carried out with a focus on civil aircraft problems, with recommendations given as to when the models are useful, as well as when they are no longer fit for purpose.

Further to this, two frequency domain methods are studied for use in calculating the dynamic derivatives in the above model. The solution methods however are not optimal. As part of this work, the Linear Frequency Domain method is implemented with an implicit solver in order to accelerate time to solution. This implicit approach is assessed against the previously available for a number of test cases of various complexities. The preconditioner in the implicit solver is also studied. A preconditioner is then developed which improves the performance of the implicit solver further by up to a factor of 5.

This thesis begins with a review of the formulation behind the CFD solvers and modelling approaches used in this work. It will then continue with an assessment of Frequency Domain methods and the preconditioning of linear systems. A systematic assessment of the Tabular Aerodynamic Model is carried out, before finishing with a review of the Dynamic Derivative model and a method for quantifying the uncertainty introduced.

Chapter 2

Formulation

2.1 CFD Solvers

The benefits of using CFD over physical experiments are numerous, including cost savings in time and money, being able to explore the finer details of flows, and enabling greater control over simulations. With the applications of CFD growing, the models are becoming of larger dimension, enabled by the rapid growth in computational power.

For the majority of CFD solvers, a finite volume approach is taken. A unit cube control volume such as that shown in Fig. 2.1, will have a quantity of fluid flowing in and out, with varying velocity vectors and energy. The conservation laws of mass, momentum and energy can be applied to the volume in order to obtain the fluxes at each face. This leads to partial differential equations that describe the fluid flow.

2.1.1 Governing Flow Equations

In this thesis, both the inviscid Euler equations (where viscosity is equal to zero) and the Reynolds Averaged Navier-Stokes (RANS) equations are used to model the flow.

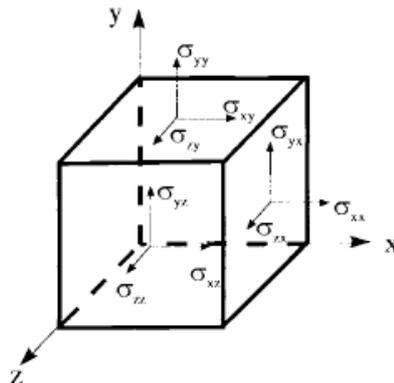


Figure 2.1: Unit cube control volume [1]

The more general Navier-Stokes equations in three dimensions, can be written in vector form as:

$$\frac{\partial \mathbf{w}}{\partial t} + \nabla \cdot (f_c(\mathbf{w}) - f_v(\mathbf{w})) = 0 \quad (2.1)$$

where the functions $f_j(\mathbf{w})$ for $j \in [c, v]$ are:

$$\begin{aligned} f_c(\mathbf{w}) &= \mathbf{E}_c + \mathbf{F}_c + \mathbf{G}_c \\ f_v(\mathbf{w}) &= \mathbf{E}_v + \mathbf{F}_v + \mathbf{G}_v \end{aligned} \quad (2.2)$$

with the subscripts c and v representing the convective and viscous fluxes respectively, and

$$\begin{aligned} \mathbf{w} &= \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{pmatrix}, \mathbf{E}_c = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ \rho Hu \end{pmatrix}, \mathbf{F}_c = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ \rho Hv \end{pmatrix}, \mathbf{G}_c = \begin{pmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ \rho Hw \end{pmatrix}, \\ \mathbf{E}_v &= \begin{pmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xw} \\ \tau_x \mathbf{U} + q_x \end{pmatrix}, \mathbf{F}_v = \begin{pmatrix} 0 \\ \tau_{yx} \\ \tau_{yy} \\ \tau_{yw} \\ \tau_y \mathbf{U} + q_y \end{pmatrix}, \mathbf{G}_v = \begin{pmatrix} 0 \\ \tau_{wx} \\ \tau_{wy} \\ \tau_{ww} \\ \tau_w \mathbf{U} + q_w \end{pmatrix} \end{aligned} \quad (2.3)$$

where the velocity vector $\mathbf{U} = [u, v, w]^T$, q is the heat flux vector, τ is the viscous shear stress, ρ is the density and p is the pressure obtained from the ideal gas equation of state.

In order to solve turbulent flow problems in a computationally efficient manner, time averaging of the turbulent terms is carried out through Reynolds Averaging. Reynolds averaging decomposes the instantaneous flow variables into a mean and a varying component. Time averaging is then applied to each of these components. The time averaged mean and varying components are then substituted back into the Navier–Stokes equations, but are now the Reynolds Averaged Navier–Stokes equations. The turbulence model used in this thesis is that of Spalart and Allmaras [47].

2.1.2 Unsteady Solution

CFD solvers are particularly useful for the solution of unsteady problems. In order to model the unsteadiness, a method to capture the flow history is required. In the solvers used in this work, this is done using the dual-time stepping approach described in [48]. The dual-time method defines an outer time step, the real time steps within a manoeuvre, and an inner time step in pseudo time. The dual-time method also requires

additional terms in the residual to capture the flow history. The semi-implicit form of the governing flow equations is written for the outer iterations, solved in real time, as:

$$\frac{\partial \mathbf{w}}{\partial t} = -\mathbf{R}^{n+1}, \quad (2.4)$$

where n is the current time step. The residual is then modified to include the flow variables at the previous two time steps for a second order accurate solution. The unsteady residual R^* is written as:

$$\mathbf{R}^* = \mathbf{R}^{n+1} + \frac{3\mathbf{w}^{n+1} - 4\mathbf{w}^n + \mathbf{w}^{n-1}}{2\Delta t} \quad (2.5)$$

This residual term is then used at the inner iterations to solve the following equation, as a steady-state problem, $\mathbf{R}^* = \mathbf{0}$, in pseudo time.

$$\frac{\partial \mathbf{w}}{\partial \tau} = -\mathbf{R}^{*,m+1}, \quad (2.6)$$

where m is the pseudo time step and τ the pseudo time. During a time-accurate unsteady simulation, equation (2.6) is converged to a desired level, with the solution being equal to that at the real time step. In addition to capturing the history effects in an unsteady simulation, it is also necessary to capture the motion effects. This is done by applying velocities to the points in the computational domain. For example, a moving boundary will cause fluid points close to the boundary to move. This is simulated by applying a velocity to the points in the normal direction to the velocity vector at the surface.

A modification that can be made to the fully unsteady approach above, is to remove the dual-time terms so that only the point velocities are retained. This leads to solution of equation (2.6), but with the following residual.

$$\mathbf{R}^* = \mathbf{R}^{n+1} \quad (2.7)$$

This quasi-steady model will be of particular use for comparisons later in this thesis.

2.1.3 DLR TAU-code

The TAU-code [49, 36, 50], developed by DLR (German Aerospace Centre), is an unstructured finite volume compressible RANS code, which is widely used in industry across Europe, in particular by Airbus.

The TAU-code is a software package with stand-alone modules for grid partitioning, a preprocessor, solver, grid adaptation and grid deformation. The code is capable of calculating from low subsonic through to hypersonic flows.

The module of interest in this work is the solver. TAU uses an edge-based dual-cell approach with the inviscid fluxes being calculated to second-order accuracy through a

central scheme, or a variety of first-order upwind schemes with linear reconstruction to regain second-order accuracy. The dual-grid approach used in TAU takes the primary grid, defined by the mesh, and forms a secondary grid on top of this to provide the faces between the primary grid vertices at which to calculate the fluxes. The dual-cell faces are formed by taking the centroids of the primary grid cells and then connecting these, whilst passing through the mid-point of the perpendicular primary grid face as shown in Fig. 2.2, where the primary grid has the solid lines and the dual-cells have the dashed lines.

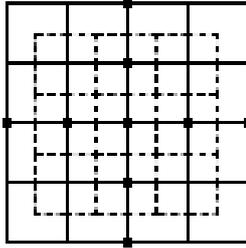


Figure 2.2: Regular grid and associated dual-cells

The solver uses either an explicit Runge-Kutta scheme or a semi-implicit LU-SGS (Lower-Upper Symmetric Gauss-Seidel) method. The results presented in this thesis use the implicit solver with a central finite difference for the discretisation of the flow equations. Each of the methods uses a dual-time stepping [48] approach and a multi-grid [51] convergence acceleration algorithm. Multigrid accelerates the convergence to solution using varying levels of grid coarseness. The accelerated convergence occurs from the use of the fine grids to resolve the high frequency modes, and the coarser grids to resolve the lower frequency modes. The grid levels are formed by merging the dual-grid cells. During the solution process, the residuals from the fine grid solution are interpolated onto the coarser mesh, where only the lower frequency modes can be resolved on the coarser grid spacing. A refinement of the solution is then done when the coarse grid residuals are passed back up to the finer grids. This cycle is carried out at each iteration, with the number of levels, and number of passes made between the levels specified by the user.

2.1.4 Parallel Meshless (PML)

The Parallel Meshless (PML) solver [52] has been developed at the University of Liverpool with particular focus on moving body simulations. The meshless approach offers an alternative to Chimera methods. The computational domains are point distributions rather than grids used in finite volume solvers. PML uses an implicit scheme developed for solving the Euler equations, as well as the laminar Navier Stokes and RANS equations. Spatial derivatives are approximated using a least squares method on clouds of

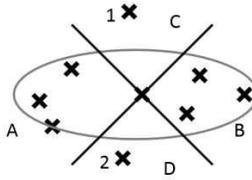


Figure 2.3: Point cloud with ellipse

points. Unlike the finite volume methods which calculate the fluxes at the faces of the control volume, PML calculates the fluxes halfway between the star point of a stencil and all other points in the stencil. The system of equations is linearised, and solved implicitly using approximate, analytical Jacobian matrices where the inviscid flux is obtained from an approximate Riemann solver, and turbulent terms modelled using a one-equation Spalart-Allmaras model. A preconditioned Krylov subspace iterative method is used as the linear solver.

The meshless approach removes the need to generate meshes for complex geometries, and instead only requires simple point distributions of component parts of a model, which PML assembles using the point data to create a large cloud to be used by the solver. The preprocessor to allow this requires a novel stencil selection algorithm which is described in [53]. The stencil selection process makes use of the connectivity in the underlying component meshes to guide the orientation of the ellipses used to select the stencils. For each star point, an ellipse is formed with the semi-major axis as close to perpendicular to the flow direction as possible. The ellipse is divided into quadrants, and a minimum number of points is required in each quadrant to form a stencil. An example is shown in Fig. 2.3, taken from [53].

Increasing the required number of points should increase the stability of the solver, through greater resolution in each stencil. This is however not always possible.

Several cases have been studied including a cylinder undergoing vortex shedding, turbulent aerofoil problems, a three-dimensional fighter aircraft configuration and store release problems with bodies in relative motion. The capability of this solver is particularly useful for modelling control surfaces. This will be demonstrated later in this thesis.

2.2 Control Surface Modelling

The modelling of control surfaces can be a problem for CFD solvers. A number of different approaches exist in the literature. Two different methods have been implemented within PML and have been used in this work. The first is a standard deformation technique. The first step in using this technique is to describe an underlying geometry

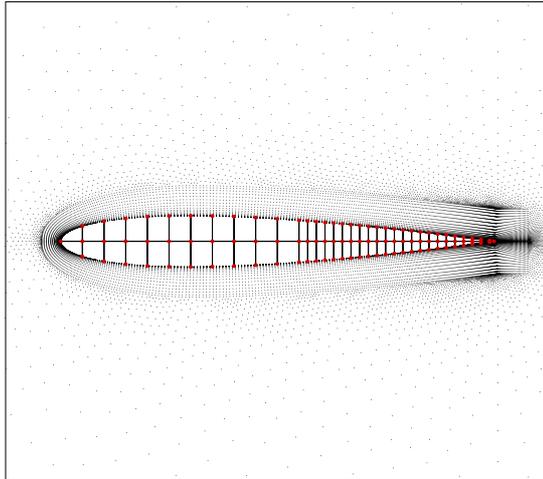


Figure 2.4: NACA 0012 aerofoil point distribution with underlying structural model

that represents the structural elements of a model. In this work, this is a simple beam stick model as shown in Fig. 2.4.

Deformations are prescribed for the structure, which in turn is used to deform the fluid domain. In order to communicate the deformation from the structure to the fluid domain, a mapping is carried out. This mapping describes how much the fluid domain points should be moved for a deformation in the underlying structural grid. At each step within a manoeuvre, the relevant deformation can rapidly be established and applied to the fluid domain points. At the point where the deformed section meets that of the undeformed, the surface is smoothed in a blending procedure. It is also possible, for three-dimensional cases, to insert a cut in the geometry, as would be the case in a physical test. A number of these methods were assessed in [54].

In this work, the deformation technique has been used for the RANS control surface simulations. For the mapping from structure to surface, an interpolation matrix, H is formed, which is used to transfer the displacements between the structural and fluid grids using Eq. (2.8).

$$(\delta y_f)_i = \sum_{j=1}^{j_s} h_{ij} (\delta y_s)_j, \quad (2.8)$$

where $(\delta y_f)_i$ is the displacement of the fluid mesh at node i , $(\delta y_s)_j$ is the displacement of the structure at node j , and h_{ij} are the coefficients of the displacement interpolation matrix H . The coefficients are obtained through the use of the Constant Volume Tetrahedron (CVT) mapping technique. This is described in detail in [55]. The method first overlaps the fluid and structural point distributions. From this overlap, tetrahedra are formed where the peak of each tetrahedron is a point in the fluid domain, and the base vertices are points in the structural domain. The fluid point is then projected onto the plane formed from the structural vertices. Shape functions are

calculated, so that when the structural points are deformed, the fluid point deforms in a manner that maintains the volume of the tetrahedral element. This method is used due to its simplicity and speed of calculation.

The boundary points are moved in line with the interpolation matrix and is described further in [56]. Once the boundary deformations have been determined, the points in the rest of the computational domain are deformed using an inverse distance approach, as described in [57].

The second approach is to make use of the capabilities of the meshless PML solver. Point distributions are defined for the body and control surface, which are then overlapped according to the desired deflection. The preprocessor then redefines the boundaries, removes any points inside the boundary, then reselects the stencils for computation. This has been used for all Euler simulations with a control surface in this work. This approach has the benefit of modelling the control surface in a more realistic manner than that of the deformation. There is no smoothing of the edges, as there wouldn't be on a real aircraft, and the flap is treated as a separate entity to the body.

2.3 Tabular Models

Tabular models are used to determine the flight mechanics loads (the loads subsequently referred to in this thesis) and moments on manoeuvring aircraft. They are frequently used in the design phase for flight mechanics loads assessment and control systems design. For commercial aircraft the flight envelope can be highly dimensional, with many data points required within the range for each parameter. This can require data points in the order of millions. The data stored in the aerodynamic tables must cover the parameter space in order to effectively simulate manoeuvres. In forming the tables, a number of assumptions are made which give rise to certain limitations. One initial assumption that is made in forming the tables, is that the resolution (i.e. how many data points are in the parameter space) is sufficient to represent the aerodynamic variations of interest. With the tables requiring data to be obtained at millions of points, if CFD is used as the source of the aerodynamic data, it is clearly not feasible to have a solution for each parameter combination. To reduce the number of points required, the parameters can be decoupled. For example, the six dimensional table in Table 2.1, $[M, \alpha, \beta, \delta_{ele}, \delta_{ail}, \delta_{rud}]$ can be reduced to four three dimensional tables of $[M, \alpha, \beta]$, $[M, \alpha, \delta_{ele}]$, $[M, \alpha, \delta_{ail}]$ and $[M, \alpha, \delta_{rud}]$. The assumption here is that the influence of each decoupled parameter on another is negligible, through the use of small perturbations, which may not be the case for certain flow conditions.

An extension to minimising the required number of high fidelity calculations is to use a hierarchy of methods of different fidelities. For instance, low fidelity semi-empirical data can be used. A data fusion approach is then applied to maintain sufficient fidelity of the tables, whilst reducing the number of CFD simulations required as originally

M	α	β	δ_{ele}	δ_{ail}	δ_{rud}	C_L	C_D	C_Y	C_l	C_m	C_n
x	x	x	-	-	-	x	x	x	x	x	x
x	x	-	x	-	-	x	x	x	x	x	x
x	x	-	-	x	-	x	x	x	x	x	x
x	x	-	-	-	x	x	x	x	x	x	x

Table 2.1: Example Aerodynamic Table (x indicates non-zero entry)

proposed in [58], where a 30% reduction in computational time was achieved without loss of accuracy. In [3] this approach was extended to use the DATCOM [59] database as the source of the low-fidelity data, which was then assessed using a commercial jet aircraft case with changing geometry. Kriging interpolation was also applied in this work to further minimise the number of calculations required to fill the tables. It was also used to locate points in the parameter space where a high-fidelity solution is required (i.e. location of high parametric sensitivity).

In this work, only high-fidelity CFD data are used due to the low number of data points required for the cases presented. Kriging is then used to obtain unknown data points within the manoeuvre parameter space.

2.3.1 Dynamic Derivatives

Dynamic derivatives describe how the forces and moments vary with rates of motion. For example, the pitch-damping derivative C_{M_q} describes how the pitching moment coefficient, C_M , varies with the pitch rate, q . The derivative values are used to account for motion effects, by taking the static load or moment coefficient and modifying this as shown in Eq. (2.9) for a pitching motion.

$$C_j(t) = C_{j_0} + C_{j_\alpha} \Delta\alpha(t) + C_{j_{\dot{\alpha}}} \frac{c}{2U_\infty} \dot{\alpha}(t) + C_{j_q} \frac{c}{2U_\infty} q(t) + C_{j_{\dot{q}}} \left(\frac{c}{2U_\infty} \right)^2 \dot{q}(t). \quad (2.9)$$

The j subscript represents the force or moment of interest (i.e. L,D,M), the zero subscript term is the steady value at time t . The non-dimensionalisation factor is taken from the reduced frequency $k = \frac{\omega c}{2U_\infty}$. The term ω is in radians per second, as are $\dot{\alpha}$ and \dot{q} . This factor maintains consistency with the prescribed inputs, usually k , for describing the oscillatory motion in the calculation of the dynamic derivatives.

For a harmonic pitching motion, the following can be defined

$$\begin{aligned} \Delta\alpha &= \alpha_A \sin(\omega t) & \dot{\alpha} &= q = \omega \alpha_A \cos(\omega t) \\ \ddot{\alpha} &= \dot{q} = -\omega^2 \alpha_A \sin(\omega t). \end{aligned} \quad (2.10)$$

This allows Eq. (2.9) to be rewritten as

$$C_j(t) = C_{j_0}(t) + \bar{C}_{j_\alpha} \alpha(t) + \bar{C}_{j_q} q(t) \frac{c}{2U_\infty}, \quad (2.11)$$

where the bar terms are formed as below again with $k = \frac{\omega c}{2U_\infty}$.

$$\begin{aligned} \bar{C}_{j_\alpha} &= (C_{j_\alpha} - k^2 C_{j_q}) \\ \bar{C}_{j_q} &= (C_{j_\alpha} + C_{j_q}) \end{aligned} \quad (2.12)$$

The derivative values can be calculated from forced periodic oscillations as described in [60]. The periodic time-dependent solution can then be written as a Fourier series, with the first Fourier coefficients corresponding to the values of the stability derivatives. These can be calculated directly using Eq. (2.13),

$$\begin{aligned} \bar{C}_{j_\alpha} &= \frac{2}{\alpha_A n_c T} \int_0^{n_c T} \Delta C_j(t) \sin(\omega t) dt \\ \bar{C}_{j_q} &= \frac{2}{k \alpha_A n_c T} \int_0^{n_c T} \Delta C_j(t) \cos(\omega t) dt \end{aligned} \quad (2.13)$$

where the terms α_A, k, n_c, T and ω are the oscillatory amplitude, reduced frequency, number of cycles, time period and circular frequency respectively.

Given this model consists of a steady and unsteady component dependent on the instantaneous motion rates, there is no accounting for history effects. As such, this approach is considered as quasi-steady, as described for the CFD solvers previously.

2.4 Frequency Domain Methods

2.4.1 Linear Frequency Domain

The Linear Frequency Domain method [22] uses the assumptions of periodicity and small amplitudes to reduce an unsteady nonlinear problem into a steady linear one.

The governing equations of a fluid are first written in the semi-discrete form as

$$\frac{\partial \mathbf{w}}{\partial t} + \mathbf{R}(\mathbf{w}, \mathbf{x}, \dot{\mathbf{x}}) = 0, \quad (2.14)$$

where \mathbf{R} is the residual, \mathbf{w} is the vector of conservative flow variables, with \mathbf{x} and $\dot{\mathbf{x}}$, the grid position and grid velocities respectively.

The assumption of small amplitudes allows the variables to be calculated as a steady state plus a small perturbation about that steady mean state. This gives rise to

$$\begin{aligned} \mathbf{w}(t) &= \bar{\mathbf{w}} + \tilde{\mathbf{w}}(t), \quad \text{where } \|\tilde{\mathbf{w}}\| \ll \|\bar{\mathbf{w}}\| \\ \mathbf{x}(t) &= \bar{\mathbf{x}} + \tilde{\mathbf{x}}(t), \quad \text{where } \|\tilde{\mathbf{x}}\| \ll \|\bar{\mathbf{x}}\|. \end{aligned} \quad (2.15)$$

Combining equations (2.14) and (2.15), leads to the following.

$$\frac{d\tilde{\mathbf{w}}}{dt} + \frac{\partial \mathbf{R}}{\partial \mathbf{w}} \Big|_{\bar{\mathbf{w}}, \bar{\mathbf{x}}} \tilde{\mathbf{w}} + \frac{\partial \mathbf{R}}{\partial \mathbf{x}} \Big|_{\bar{\mathbf{w}}, \bar{\mathbf{x}}} \tilde{\mathbf{x}} + \frac{\partial \mathbf{R}}{\partial \dot{\mathbf{x}}} \Big|_{\bar{\mathbf{w}}, \bar{\mathbf{x}}} \dot{\tilde{\mathbf{x}}} = \mathbf{0} \quad (2.16)$$

The small periodic time-dependent perturbation is then written as a Fourier series in terms of the base frequency ω and the mode k

$$\tilde{\mathbf{w}}(t) = \sum_{k=1}^{\infty} (\hat{\mathbf{w}}_k e^{ik\omega t}), \quad (2.17)$$

where $\hat{\mathbf{w}}$ is a vector of complex Fourier coefficients corresponding to the flow solution. This is also applied to $\tilde{\mathbf{x}}$. The LFD system can be rewritten by combining equations (2.16) and (2.17) as,

$$\left\{ ik\omega I + \frac{\partial \mathbf{R}}{\partial \mathbf{w}} \right\} \hat{\mathbf{w}}_k = -\frac{\partial \mathbf{R}}{\partial \mathbf{x}} \hat{\mathbf{x}}_k - ik\omega \frac{\partial \mathbf{R}}{\partial \dot{\mathbf{x}}} \hat{\mathbf{x}}_k. \quad (2.18)$$

Limiting interest to the perturbations which are harmonic in the forced frequency, k is taken to be 1, and hence the nonlinear Eq. (2.14) has been reduced to a single linear equation. The linear system is then solved for $\hat{\mathbf{w}}_k$.

The real and imaginary parts in Eq. (2.18) are taken to form two coupled real systems as

$$\begin{aligned} -\omega \Im(\hat{\mathbf{w}}) + \frac{\partial \mathbf{R}}{\partial \mathbf{w}} \Re(\hat{\mathbf{w}}) &= -\frac{\partial \mathbf{R}}{\partial \mathbf{x}} \Re(\hat{\mathbf{x}}) + \omega \frac{\partial \mathbf{R}}{\partial \dot{\mathbf{x}}} \Im(\hat{\mathbf{x}}) \\ \omega \Re(\hat{\mathbf{w}}) + \frac{\partial \mathbf{R}}{\partial \mathbf{w}} \Im(\hat{\mathbf{w}}) &= -\frac{\partial \mathbf{R}}{\partial \mathbf{x}} \Im(\hat{\mathbf{x}}) - \omega \frac{\partial \mathbf{R}}{\partial \dot{\mathbf{x}}} \Re(\hat{\mathbf{x}}). \end{aligned} \quad (2.19)$$

The above linear system is written in the form

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \quad (2.20)$$

where A is the system matrix written as

$$A = \begin{bmatrix} \frac{\partial \mathbf{R}}{\partial \mathbf{w}} & -\omega I \\ \omega I & \frac{\partial \mathbf{R}}{\partial \mathbf{w}} \end{bmatrix}, \quad (2.21)$$

where \mathbf{x} is the vector of Fourier coefficients to be calculated and \mathbf{b} is the right-hand side obtained through central finite differences from the steady state solution as

$$\begin{aligned} \frac{\partial \mathbf{R}}{\partial \mathbf{x}} \hat{\mathbf{x}} &\approx \frac{\mathbf{R}(\bar{\mathbf{w}}, \bar{\mathbf{x}} + \epsilon \hat{\mathbf{x}}, 0) - \mathbf{R}(\bar{\mathbf{w}}, \bar{\mathbf{x}} - \epsilon \hat{\mathbf{x}}, 0)}{2\epsilon} \\ \frac{\partial \mathbf{R}}{\partial \dot{\mathbf{x}}} \hat{\mathbf{x}} &\approx \frac{\mathbf{R}(\bar{\mathbf{w}}, \bar{\mathbf{x}}, \epsilon \hat{\mathbf{x}}) - \mathbf{R}(\bar{\mathbf{w}}, \bar{\mathbf{x}}, -\epsilon \hat{\mathbf{x}})}{2\epsilon}. \end{aligned} \quad (2.22)$$

The value of ϵ has to be chosen to be large enough to reduce errors due to the rounding

of higher-order terms, but also small enough to reduce the truncation error. Once this linear system has been obtained and set up, it can be solved using a linear solver as described in section 3.1.

2.4.2 Harmonic Balance

The Harmonic Balance method was initially proposed for use in turbomachinery flows for the rapid solution of periodic oscillatory simulations. The benefit is that there is no linearisation involved in the formulation, and as such, nonlinearities can be captured to varying degrees of accuracy depending on the number of harmonics retained in the solution. HB has been extended for use with aircraft aerodynamics and in particular for flight dynamics and the generation of dynamic derivatives.

The formulation again begins with the semi-discrete form of the governing flow equations as

$$\frac{\partial \mathbf{w}(t)}{\partial t} + \mathbf{R}(\mathbf{w}) = \mathbf{0}, \quad (2.23)$$

where \mathbf{w} is the vector of conserved variables, and \mathbf{R} is the residual of the flux terms. Assuming a periodic motion, these terms can be written as a Fourier series with frequency ω as

$$\begin{aligned} \mathbf{w}(t) &= \hat{\mathbf{w}}_0 + \sum_{k=1}^{\infty} (\hat{\mathbf{w}}_k e^{ik\omega t}) \\ \mathbf{R}(t) &= \hat{\mathbf{R}}_0 + \sum_{k=1}^{\infty} (\hat{\mathbf{R}}_k e^{ik\omega t}). \end{aligned} \quad (2.24)$$

The exponential is broken down into its sine and cosine components, along with the corresponding Fourier coefficients denoted by the subscripts a and b . The series is then truncated to a specified number of harmonics N_H leading to the following

$$\begin{aligned} \mathbf{w}(t) &\approx \hat{\mathbf{w}}_0 + \sum_{k=1}^{N_H} (\hat{\mathbf{w}}_{a_k} \cos(\omega kt) + \hat{\mathbf{w}}_{b_k} \sin(\omega kt)) \\ \mathbf{R}(t) &\approx \hat{\mathbf{R}}_0 + \sum_{k=1}^{N_H} (\hat{\mathbf{R}}_{a_k} \cos(\omega kt) + \hat{\mathbf{R}}_{b_k} \sin(\omega kt)). \end{aligned} \quad (2.25)$$

Combining Eqs. (2.23) and (2.25), then grouping similar harmonic terms yields

$$\begin{aligned} \hat{\mathbf{R}}_0 &= 0 \\ \omega n \hat{\mathbf{w}}_{b_n} + \hat{\mathbf{R}}_{a_n} &= 0 \\ -\omega n \hat{\mathbf{w}}_{a_n} + \hat{\mathbf{R}}_{b_n} &= 0. \end{aligned} \quad (2.26)$$

A system of $N_T = 2N_H + 1$ equations has now been obtained, which can be written in matrix form as

$$\omega \mathbf{A} \hat{\mathbf{w}} + \hat{\mathbf{R}} = 0, \quad (2.27)$$

where \mathbf{A} is a block matrix of size $N_T \times N_T$ containing blocks with diagonal terms $\mathbf{A}(n+1, N_H+n+1) = \mathbf{n}$, and $\mathbf{A}(N_H+n+1, n+1) = -\mathbf{n}$. The vectors $\hat{\mathbf{w}}$ and $\hat{\mathbf{R}}$ are composed of

$$\hat{\mathbf{w}} = \begin{pmatrix} \hat{\mathbf{w}}_0 \\ \hat{\mathbf{w}}_{a_1} \\ \vdots \\ \hat{\mathbf{w}}_{a_{N_H}} \\ \hat{\mathbf{w}}_{b_1} \\ \vdots \\ \hat{\mathbf{w}}_{b_{N_H}} \end{pmatrix} \quad \hat{\mathbf{R}} = \begin{pmatrix} \hat{\mathbf{R}}_0 \\ \hat{\mathbf{R}}_{a_1} \\ \vdots \\ \hat{\mathbf{R}}_{a_{N_H}} \\ \hat{\mathbf{R}}_{b_1} \\ \vdots \\ \hat{\mathbf{R}}_{b_{N_H}} \end{pmatrix}, \quad (2.28)$$

where the coefficients are those seen in Eq. (2.25). A solver could be written to solve Eq. (2.27), however this could be complicated due to the complex Fourier terms, particularly when dealing with viscous flows, as well as finding a relationship between $\hat{\mathbf{w}}$ and $\hat{\mathbf{R}}$. To overcome this problem, the system is transformed back to the time domain. The solution is discretised into N_T equally spaced intervals over the cycle to obtain

$$\mathbf{w}_{hb} = \begin{pmatrix} \mathbf{w}(t_0 + \Delta t) \\ \mathbf{w}(t_0 + 2\Delta t) \\ \vdots \\ \mathbf{w}(t_0 + T) \end{pmatrix} \quad \mathbf{R}_{hb} = \begin{pmatrix} \mathbf{R}(t_0 + \Delta t) \\ \mathbf{R}(t_0 + 2\Delta t) \\ \vdots \\ \mathbf{R}(t_0 + T) \end{pmatrix}, \quad (2.29)$$

where T is the time for one cycle and $\Delta t = 2\pi/(N_T\omega)$. The vectors in Eq. (2.29) are initialised from steady state solutions at each of the intervals to provide a good initial guess. The vectors of Fourier terms are then related to the corresponding HB vectors via an $N_T \times N_T$ transformation matrix \mathbf{E}

$$\hat{\mathbf{w}} = \mathbf{E} \mathbf{w}_{hb} \quad \text{and} \quad \hat{\mathbf{R}} = \mathbf{E} \mathbf{R}_{hb}, \quad (2.30)$$

where

$$\mathbf{E} = \frac{2}{N_T} \begin{bmatrix} 0.5 & 0.5 & \dots & 0.5 \\ \cos(2\pi \frac{1 \times 1}{N_T}) & \cos(2\pi \frac{1 \times 2}{N_T}) & \dots & \cos(2\pi \frac{1 \times N_T}{N_T}) \\ \vdots & & & \\ \cos(2\pi \frac{N_H \times 1}{N_T}) & \cos(2\pi \frac{N_H \times 2}{N_T}) & \dots & \cos(2\pi \frac{N_H \times N_T}{N_T}) \\ \sin(2\pi \frac{1 \times 1}{N_T}) & \sin(2\pi \frac{1 \times 2}{N_T}) & \dots & \sin(2\pi \frac{1 \times N_T}{N_T}) \\ \vdots & & & \\ \sin(2\pi \frac{N_H \times 1}{N_T}) & \sin(2\pi \frac{N_H \times 2}{N_T}) & \dots & \sin(2\pi \frac{N_H \times N_T}{N_T}) \end{bmatrix}. \quad (2.31)$$

It is possible to compute one large Fourier transform on the full system, however, in using this matrix with one column per time slice, this has the effect of carrying out lots of small transforms and reduces the computational cost. Substituting Eq. (2.30) into Eq. (2.27) and premultiplying by \mathbf{E}^{-1} gives

$$\omega \mathbf{E}^{-1} \mathbf{A} \mathbf{E} \mathbf{w}_{hb} + \mathbf{E}^{-1} \mathbf{E} \mathbf{R}_{hb} = 0. \quad (2.32)$$

This can then be reduced to

$$\omega \mathbf{D} \mathbf{w}_{hb} + \mathbf{R}_{hb} = 0, \quad (2.33)$$

where

$$\mathbf{D} = \mathbf{E}^{-1} \mathbf{A} \mathbf{E} = \frac{2}{N_T} \sum_{k=1}^{N_H} k \sin(2\pi k(j-i)/N_T). \quad (2.34)$$

Equation (2.33) is then solved by introducing a pseudo-time term to be able to time-march the system to achieve a converged solution.

$$\frac{d\mathbf{w}_{hb}}{dt} + \omega \mathbf{D} \mathbf{w}_{hb} + \mathbf{R}_{hb} = 0. \quad (2.35)$$

Equation (2.35) only differs from Eq. (2.23) by the HB source term $\omega \mathbf{D} \mathbf{w}_{hb}$. This allows for simple extension of existing CFD solvers for the solution of this system. The time domain response can be reconstructed from the \mathbf{w}_{hb} and \mathbf{R}_{hb} vectors through transformation to the frequency domain, where the Fourier coefficients in Eq. (2.25) are obtained. Solution approaches to equation (2.35) are described in section 3.2.

2.4.2.1 Implicit Solution

As part of this thesis, an implicit solution approach is taken for solving the Harmonic Balance problem. The method applied is that of Woodgate et al. [28]. In order to solve the system implicitly, a global Jacobian matrix needs to be defined containing all the

time slices. This is defined as follows

$$\begin{bmatrix} \frac{\partial \mathbf{R}}{\partial \mathbf{w}}|_{t_0+\Delta t} & \omega \mathbf{D}_{1,2} & \dots & \omega \mathbf{D}_{1,N_T} \\ \omega \mathbf{D}_{2,1} & \frac{\partial \mathbf{R}}{\partial \mathbf{w}}|_{t_0+2\Delta t} & & \\ \vdots & & \ddots & \\ \omega \mathbf{D}_{N_T,1} & \omega \mathbf{D}_{N_T,2} & & \frac{\partial \mathbf{R}}{\partial \mathbf{w}}|_{t_0+T} \end{bmatrix} \quad (2.36)$$

The diagonal terms are the Jacobian matrices of the individual time slices, and the off-diagonal terms are taken from the HB source term in Eq. (2.35). This approach allows rapid solution of the Harmonic Balance problem with linear solvers and are used in this thesis with the PML solver.

2.4.3 ILU Preconditioner

The linear system in Eq. (2.20) ideally would be solved by finding the inverse of A and then multiplying this by the right hand side to obtain \mathbf{x} . However, finding the inverse of the very large sparse matrices encountered in CFD using direct methods is computationally expensive. The alternative used is to take the linear system and represent it as an equivalent system, which is better conditioned, and thus quicker to iteratively solve. This is done with the use of a preconditioning matrix P which is an approximation of A , and can be inverted easily. This is used as shown in Eq. (2.37) for left preconditioning:

$$P^{-1}A\mathbf{x} = P^{-1}\mathbf{b} \quad (2.37)$$

and in Eq. (2.38) for right preconditioning

$$\begin{aligned} AP^{-1}P\mathbf{x} &= AP^{-1}\mathbf{y} = \mathbf{b} \\ \text{where } \mathbf{x} &= P^{-1}\mathbf{y}. \end{aligned} \quad (2.38)$$

If P^{-1} is equal to the inverse of A , an identity matrix is obtained on the left hand side and the system is solved. The most simple preconditioning technique is *Jacobi Preconditioning*, where the preconditioner matrix is a diagonal matrix with the inverse of the diagonal terms of A along it.

The method used here is that of Incomplete Lower Upper (ILU) preconditioning [34]. ILU preconditioning, as with LU decomposition, forms a lower triangular matrix L and an upper triangular matrix U , where $A = LU - R$ with R termed the Residual matrix, whereby the norm of the matrix indicates the accuracy of the incomplete factorisation. The difference between ILU and LU is that there is a limit on the number of new non-zero terms that are generated when using ILU. The sparsity pattern in the lower and upper matrices is set with the level of fill-in indicated in brackets. ILU(0) is zero fill-in which means the sparsity of LU is equal to that of A , i.e. no new non-

zeros have been added. In this work up to 1 level of fill-in is used, which usually adds around two further non-zeros for every one in the A matrix, although this varies slightly depending on the problem. The algorithm for the ILU factorisation is as follows

1. For all non-zero elements a_{ij} define $lev(a_{ij}) = 0$
 2. For $i = 2, \dots, n$ Do:
 3. For $k = 1, \dots, i - 1$ and for $lev(a_{ik}) \leq p$ Do:
 4. Compute $a_{ik} = a_{ik}/a_{kk}$
 5. Compute $a_{i*} = a_{i*} - a_{ik}a_{k*}$
(where * indicates operation on all non-zero terms in the row)
 6. Update the levels of fill of the non-zero a_{ij} 's using:
$$lev_{ij} = \min\{lev_{ij}, lev_{ik} + lev_{kj} + 1\}$$
 7. EndDo
 8. Replace any element in row i with $lev(a_{ij}) > p$ by zero
 9. EndDo
-

This algorithm determines the sparsity pattern of the ILU factorisation for a given level of fill-in.

Chapter 3

Performance of Frequency Domain Methods

Frequency domain methods have proven useful for flight dynamics purposes. They offer accelerated computation of the aerodynamic response to a periodic oscillation required to calculate the dynamic derivative terms in the unsteady aerodynamic model. This chapter looks to accelerate the Linear Frequency Domain and Harmonic Balance methods, through an approach to preconditioning for linear solvers.

3.1 Linear Frequency Domain

The origins of the Linear Frequency Domain method are in the small disturbance Euler method developed for use in turbomachinery flows by Hall and Crawley [24]. It has since been extended for use in external aerodynamic problems. It has been implemented in the DLR TAU code, as described in [22], and forms the basis of the method described in this chapter.

3.1.1 Solver Options

There are a number of different solution methods available for solving the linear systems in the LFD formulation. In order to improve upon them, it is first necessary to assess how well they perform in their current form. Three approaches are available within TAU, namely MG LU-SGS, PETSc and a Generalised Conjugate Residual (GCR) linear solver, the latter of which has been implemented as part of this work.

MG LU-SGS

The MG LU-SGS option is implemented within TAU and drives the solution to convergence by solving

$$\tilde{A}\mathbf{x} = A\mathbf{x} - \mathbf{b}, \quad (3.1)$$

where \tilde{A} is an approximation of A , and the matrix-vector product $\tilde{A}\mathbf{x}$ is driven to an L2 norm of zero. This is done using the semi-implicit LU-SGS iterative solver [36] with Multigrid [51] to accelerate the convergence. The MG LU-SGS option can also be used with a GMRes [61] Krylov solver, in order to further accelerate the convergence, but at the expense of memory. This method only operates on the matrix-vector product and never stores the full Jacobian matrix explicitly in memory. This minimises the memory requirement to enable very large grids to be run on relatively inexpensive machines, and gives this approach a competitive edge over the other two solver options, in this sense.

PETSc Linear Solvers

The second option is to use the linear solvers built into the PETSc linear libraries [62, 63]. TAU can be compiled with the PETSc libraries for solution of both the Adjoint and LFD problems. The implementation requires the storage of the Jacobian matrix explicitly in memory, and as such requires significantly more memory than MG LU-SGS. The PETSc libraries include many solvers, from the direct LU and Cholesky methods through to the approximate Krylov methods, including GCR and GMRes used in this work. PETSc also has a variety of preconditioners, ranging from the simple Jacobi preconditioning to the ILU factorisation used here. The solvers also have many options to monitor convergence properties, along with other options to improve convergence such as Reverse Cuthill-McKee reordering. This array of solver options makes PETSc a useful tool for optimising the linear solution.

Generalised Conjugate Residual Linear Solver

The final option is the GCR [64] linear solver. This method has been implemented in TAU for solution of the LFD problem as part of this work. It uses a block matrix structure rather than an element-wise structure to minimise the memory required to store the sparse matrix, and to increase the speed with which the data can be accessed. This requires a blocked version of the ILU(k) preconditioner mentioned in Section 2.4.3. As with PETSc, the full Jacobian matrix is stored in memory.

The GCR solver is a Krylov subspace method whereby the system is projected onto a subspace shown in Eq. (3.2).

$$\mathcal{K}_m(A, \mathbf{r}_0) = \text{span}\{\mathbf{r}_0, A\mathbf{r}_0, A^2\mathbf{r}_0, \dots, A^{m-1}\mathbf{r}_0\}, \quad (3.2)$$

where m is the number of subspace vectors allocated in advance, and the residual \mathbf{r}_0 is the term to be minimised. This minimisation is done using the following algorithm

-
1. First, set $\mathbf{p}_0 = \mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$
 2. Iterate, for $i=0,1,\dots$ until restart or convergence do:
 - Compute $\alpha_i = \frac{(\mathbf{r}_i, A\mathbf{p}_i)}{(A\mathbf{p}_i, A\mathbf{p}_i)}$,
 - $\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha_i \mathbf{p}_i$,
 - $\mathbf{r}_{i+1} = \mathbf{r}_i - \alpha_i A\mathbf{p}_i$,
 - $\mathbf{p}_{i+1} = \mathbf{r}_{i+1} + \sum_{j=0}^i \beta_j^{(i)} \mathbf{p}_j$,
 - where $\{\beta_j^{(i)}\}$ are chosen so that $(A\mathbf{p}_{i+1}, A\mathbf{p}_j) = 0$, for $0 \leq j \leq i$.
-

This is implemented in a restarted format, where the \mathbf{p}_i and $A\mathbf{p}_i$ terms are discarded after a number of iterations specified by the user, and the initial guess for the next loop is restarted from the final value of the previous set of vectors. This is repeated until convergence. The number of restart iterations is equal to the number of Krylov vectors, as such, a larger restart will represent the system more accurately, although this comes at the expense of memory. For convergence, the right preconditioned system uses the true residual which is obtained directly from the residual vector, however left preconditioning uses the preconditioned residual. In both cases, the convergence is determined relative to the initial residual to ensure consistency, regardless of the preconditioning side chosen.

Being a Krylov subspace method, during the solve, only matrix-vector products are stored with the size of the subspace defined by the user. In the following calculations 20 subspace vectors were used to allow for good rates of convergence, whilst minimising the run-time memory requirement.

3.1.2 Implicit LFD

This section shows the speed up and memory requirements of the three methods for both 2D and 3D test cases. The baseline comparator is the MG LU-SGS solution due to this being the standard method in TAU. PETSc is also included due to it being one of the TAU options, although this is not the benchmark method. The rate of convergence with respect to the number of iterations and the CPU time are compared for each case to demonstrate the improvement achieved with the various solvers. Finally, a weighted preconditioner is proposed and analysed, with reasons given for the improved performance.

Test Cases

In order to assess the different solution methods for the LFD problem, a number of test cases have been chosen. A hierarchy of complexities, ranging from Euler aerofoil problems to RANS wing cases, is used. All have been run using the LFD solver within

the DLR TAU code. All simulations are run solving the Euler equations, unless a Reynolds number is given.

NACA 0012 Aerofoil

Two grids of different point density were used for this case, a coarse grid Fig. 3.1(a) with 12,672 points (132 points around the aerofoil with 33 points in the normal direction), and a fine grid Fig. 3.1(b) with 31,416 points (212 around the aerofoil with 51 in the normal direction).

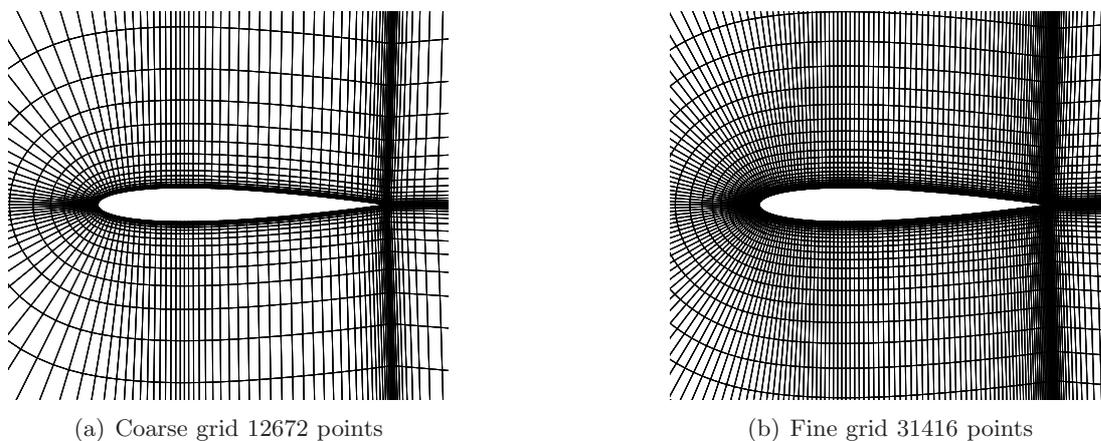


Figure 3.1: NACA 0012 grids

The two grids were run for AGARD CT2 and CT5 conditions, as shown in Table 3.1.

AGARD	CT2	CT5
Mach number, M	0.6	0.755
Mean incidence, α_0	3.16°	0.016°
Pitch amplitude, $\Delta\alpha$	4.59°	2.51°
Reduced frequency, k	0.0811	0.0814

Table 3.1: AGARD test case conditions (Euler)

NACA 64A010 Aerofoil

The NACA 64A010 aerofoil grid is shown in Fig. 3.2. A hybrid grid is used with regular cells in the boundary layer, with a wall spacing of 1×10^{-4} , and triangular elements in the farfield. There are 21,454 points. This case is used here as a simple viscous case with a Spalart-Allmaras one equation turbulence model. The case was run at AGARD CT8 conditions $M = 0.8$, $\alpha_0 = 0.0^\circ$, $\Delta\alpha = 0.5^\circ$, $\omega = 0.2$ and $Re = 12.5 \times 10^6$.

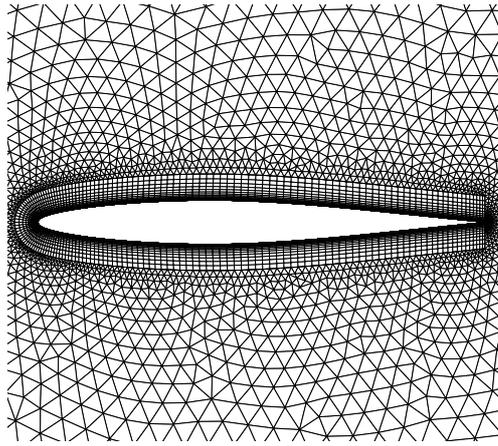


Figure 3.2: NACA 64A010 grid

SDM Generic Fighter Configuration

The SDM model is a generic fighter aircraft used for CFD studies, ranging from validation studies to flight dynamics simulation [65]. The grid, shown in Fig. 3.3, is fully unstructured with 59,542 points. This case was run at $M = 0.3$, $\alpha_0 = 0.0^\circ$, $\Delta\alpha = 2.0^\circ$ and $\omega = 0.0986$.

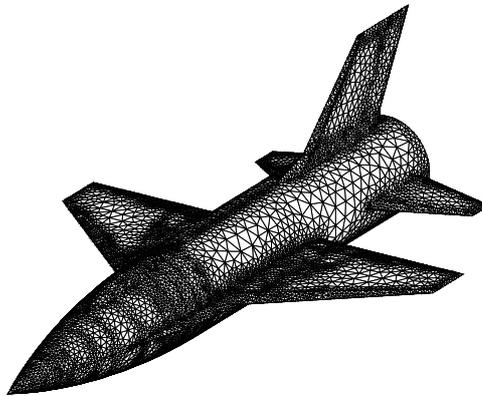


Figure 3.3: SDM grid

Goland Wing

The Goland wing is an academic test case, often used for aeroelasticity analyses, and is unswept with a circular arc section. Two grids were used here for an inviscid and a viscous calculation. The inviscid Euler grid, shown in Fig. 3.4(a) is made up of 201,909 points, and the viscous RANS grid in Fig. 3.4(b) is made up of 991,075 points. The Euler case was run at $M = 0.8$, $\alpha_0 = 1.0^\circ$, $\Delta\alpha = 1.0^\circ$, $\omega = 0.05$, with the RANS case being run at $M = 0.925$, $\alpha_0 = 0.0^\circ$, $\Delta\alpha = 1.0^\circ$, $\omega = 0.05$ and $Re = 15 \times 10^6$.

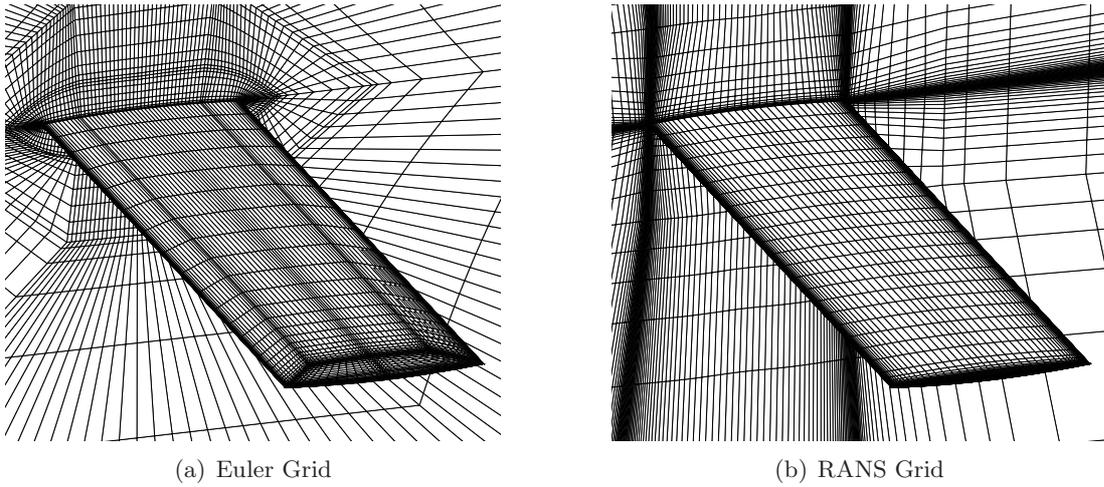


Figure 3.4: Golland wing grids

LANN Wing

The LANN wing grid shown in Fig. 3.5 is a fully unstructured grid, with 267,463 points. This case was run at $M = 0.82$, $\alpha_0 = 0.6^\circ$, $\Delta\alpha = 1.0^\circ$ and $\omega = 0.2$.

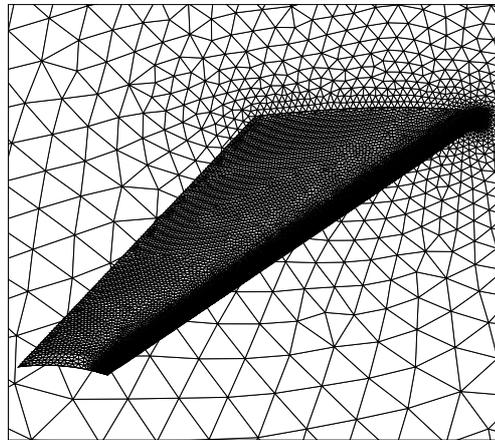


Figure 3.5: LANN wing grid

Solver Parameters

The Linear Frequency Domain method requires calculation of the steady nonlinear mean state, plus a linear solve for the frequency domain solution. The performance analysis is solely for the linear solve part of the calculation using a single processor, with the methods involving ILU shown for 0 or 1 level of fill-in. The steady-state in each case had previously been converged, with the solution used as a restart for the frequency domain solve. The parameters used for each test case are shown in Table 3.2.

		NACA0012	NACA64A010	SDM	Goland	LANN
MG LU-SGS	Min. Residual	1×10^{-8}	1×10^{-6}	1×10^{-5}	$1 \times 10^{-5}/1 \times 10^{-3}$	1×10^{-4}
	Max. Iteration	10000	10000	10000	10000	10000
	CFL	500/50	10	20	10/5	10
	MG cycle	3w++	3w++	3w++	3w++	3w++
PETSc	Min. Residual	1×10^{-8}	1×10^{-6}	1×10^{-5}	1×10^{-5}	1×10^{-4}
	Max. Iteration	10000	10000	10000	10000	10000
	Krylov Vectors	20	20	20	20	20
	Level of fill	1	1	0	1	1
GCR	Min. Residual	1×10^{-8}	1×10^{-6}	1×10^{-5}	1×10^{-5}	1×10^{-4}
	Max. Iteration	10000	10000	10000	10000	10000
	Krylov Vectors	20	20	20	20	20
	Level of fill	1	1	0	1	1

Table 3.2: Solver Parameters

In Table 3.2 the CFL numbers for NACA0012 and Goland cases are separated for the coarse and fine, and Euler and RANS grids respectively. The LANN wing and the Goland RANS case do not converge to the desired level with no Krylov solver, as such, a GMRes loop has been used for these cases with 20 Krylov vectors and 5 preconditioning iterations. The PETSc GMRes solver was run with right preconditioning and had Reverse Cuthill McKee reordering of the matrix, the GCR method used left preconditioning and used the in-built *Bandwidth optimisation* option in TAU.

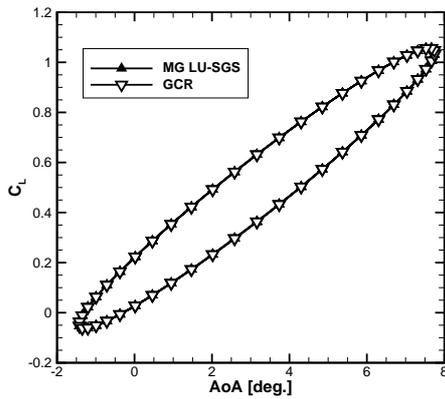
Solution Comparison

In order to determine that the solution method does not change the simulated loads and moments, it is necessary to compare the solutions from each. Presented in Figs. 3.6 and 3.7 are the LFD solutions for the NACA 0012 CT2 and CT5 cases using the MG LU-SGS and GCR solvers.

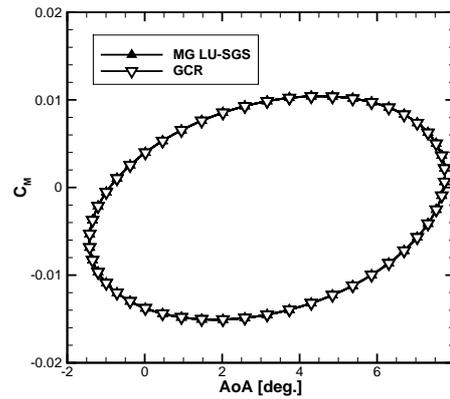
It is seen that the MG LU-SGS and GCR solutions are almost identical in both the lift and pitching moment loops. This is to be expected, and shows that the change of solution method does not change the outcome, as required.

Speed Tests

For the speed tests, the time shown is the CPU time for the linear solver to run, including the setup of the Jacobian matrix and preconditioner. Each test was run on the same processor for consistency across the solvers. All cases were run with the options as in Table 3.2, with all other parameters set as default. The MG LU-SGS (GMRes) option was run with 20 Krylov vectors and 10 preconditioning iterations. The results of the speed tests for each case are shown in Table 3.3.

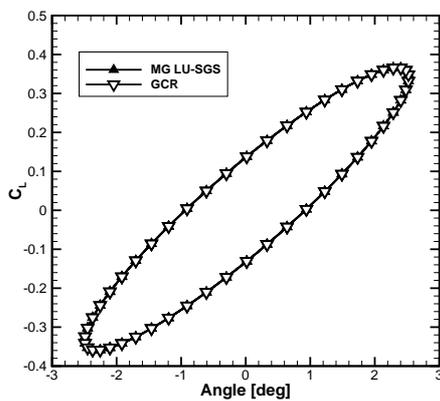


(a) Lift coefficient

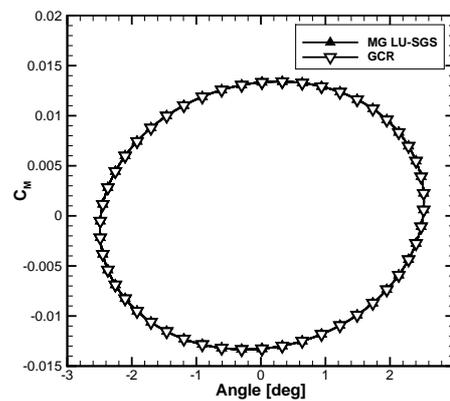


(b) Pitching moment coefficient

Figure 3.6: NACA0012 fine AGARD CT2 solution comparison



(a) Lift coefficient



(b) Pitching moment coefficient

Figure 3.7: NACA0012 fine AGARD CT5 solution comparison

It can be seen that the GCR solver offers about an order of magnitude improvement in CPU time over the MG LU-SGS method, although this drops to about a factor of five for the viscous cases. This could be due to the larger matrix blocks in the Jacobian matrix causing a lower diagonal dominance, thus making it more difficult to solve. The MG LU-SGS with GMRes Krylov solver is shown to be quicker than the MG LU-SGS with no Krylov solver in terms of iterations for convergence. This is expected given it is a semi-implicit method, however, in terms of CPU time, this option only offers an improvement for the large cases which are more difficult to solve, particularly the RANS cases. The improvement seen is between a factor of two to three.

A more important comparison is between the previously implemented PETSc solvers, and the newly implemented GCR ILU for the solution of the LFD problem. The PETSc GMRes solver with ILU preconditioning in some cases is a factor of two quicker

		MG LU-SGS	GMRes	PETSc	GCR
NACA 0012 coarse (CT2/CT5)	Iterations	3831 / 3158	334 / 340	116 / 170	245 / 207
	CPU Time (s)	220 / 179	198 / 201	6 / 8	13 / 12
NACA 0012 fine (CT2/CT5)	Iterations	3518 / 3162	322 / 345	244 / -	297 / 283
	CPU Time (s)	497 / 446	472 / 506	23 / -	39 / 37
NACA 64A010	Iterations	12782	544	-	292
	CPU Time (s)	1883	833	-	43
SDM	Iterations	1176	69	88	73
	CPU Time (s)	475	290	96	95
Goland Wing (Euler)	Iterations	13799	630	74	274
	CPU Time (s)	25773	11256	433	918
Goland Wing (RANS)	Iterations	2272	374	-	250
	CPU Time (s)	29861	30291	-	6626
LANN Wing	Iterations	-	127	-	62
	CPU Time (s)	-	3269	-	3006

Table 3.3: Solver speed test results

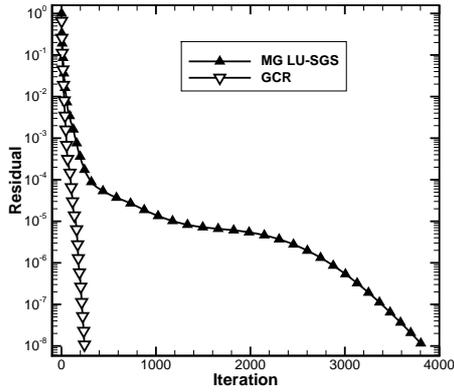
than the GCR ILU solver, however, in other cases it fails to converge or requires too much memory to run. The differences between the two methods is firstly that PETSc uses a GMRes solver compared to GCR, and secondly PETSc uses RCM reordering, which has been shown in the literature to greatly improve convergence versus the un-ordered matrix. The GMRes solver requires half the number of operations per step compared to GCR, so it would be expected to see about a factor of two difference in run time, which is observed. The use of different options in PETSc to accelerate the convergence is explored in the next section.

The convergence plots for four of the cases are shown in Fig. 3.8. It is clear that, in terms of iteration count, the implicit GCR formulation converges significantly quicker than the semi-implicit MG LU-SGS option. The convergence for the methods is comparable through the first two orders of magnitude, however, the rate of convergence slows significantly for the MG LU-SGS solver as the residual level reduces.

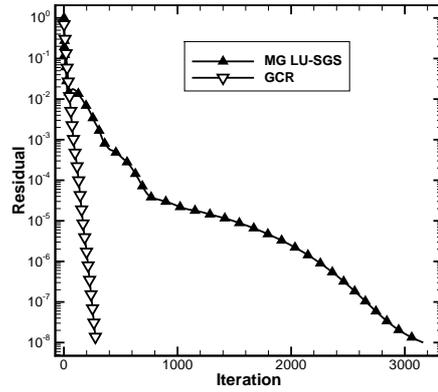
Options in PETSc

Given the large number of possible combinations of solver methods using PETSc, it is useful to view a few key combinations. The performance can be significantly affected by the options used.

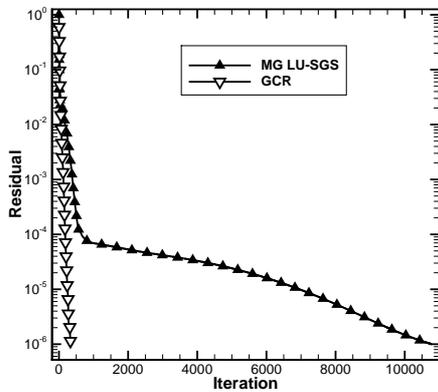
The PETSc library contains options for linear solvers, preconditioning types, re-ordering strategies and many other linear algebra functions. These options can be combined in many ways to affect the convergence of a system. For comparison, settings are tested similar to those of the GCR-ILU solver, along with slight variants. The combinations are shown in Table 3.4.



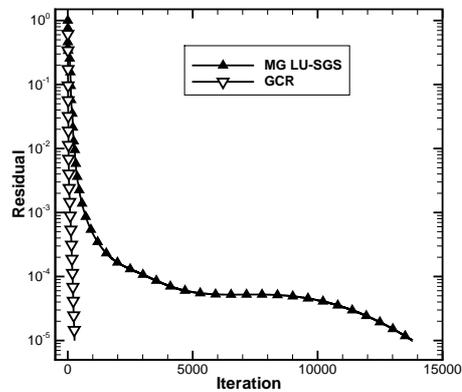
(a) NACA0012 coarse CT2



(b) NACA0012 fine CT5



(c) NACA64A010 viscous



(d) Goland Euler

Figure 3.8: Convergence comparison

Test Case	Krylov Type	Preconditioner	Reordering
1	GCR	ILU(0)	-
2	GCR	ILU(0)	RCM
3	GCR	ILU(1)	-
4	GCR	ILU(1)	RCM
5	GMRes	ILU(0)	-
6	GMRes	ILU(0)	RCM
7	GMRes	ILU(1)	-
8	GMRes	ILU(1)	RCM

Table 3.4: PETSc option test cases

The NACA 0012 fine grid at AGARD CT2 conditions has been used as a test case for simplicity. The converged residual has been set to 1×10^{-8} , with a maximum number of iterations at 2000. Results are shown in Table 3.5, with the residual shown

in brackets if convergence was not achieved.

Test Case	Converged Iteration	Run time (s)	Memory (MB)
1	(2.81×10^{-2})	140	808
2	(2.75×10^{-2})	143	816
3	(6.47×10^{-3})	189	1125
4	207	26	1162
5	1813	88	722
6	(3.05×10^{-3})	97	722
7	(6.76×10^{-5})	146	1028
8	244	23	1064

Table 3.5: PETSc option test results NACA 0012 AGARD CT2

As was seen previously, PETSc does not always converge, and in this instance, converges in only 3 out of the 8 cases. It is also seen that the use of RCM reordering does have an effect on the convergence and allows the system to converge the residual 8 orders of magnitude for some cases. The reordering does however carry a penalty in terms of memory to store the permutation matrix for the mapping of the value locations to their original positions.

Finally, a comparison between GCR and GMRes PETSc solvers shows GMRes to be more efficient in terms of memory and time. The time per iteration is about 30% quicker for the GMRes solver than GCR, with half the number of operations required per iteration. The solution time includes the setup of the preconditioner and carrying out the reordering.

The key consideration for the PETSc results is that the preconditioner is formed based on the second-order Jacobian matrix, which is often poorly conditioned. Using the different options shows that, independent of the Krylov solver type, a preconditioner formed from the incomplete factorisation of the second-order Jacobian matrix with either 0 or 1 level of fill-in is not robust, unless reordering is applied. This is also seen in the other cases during the speed tests.

Memory Tests

An important metric for solver performance is that of memory usage. When these methods are deployed in an industrial context, efficiency at every level is important. The memory requirement for each solver and test case is shown in Table 3.6. This is inclusive of the augmented system and preconditioner matrices for the PETSc and GCR options.

Table 3.6 shows that the PETSc and GCR methods require far more memory than MG LU-SGS due to the Jacobian and preconditioner matrices being stored explicitly in memory. The memory requirement is shown to be up to an order of magnitude higher for the GCR method than for MG LU-SGS, and up to about a factor of five against

	MG LU-SGS	GMRes	PETSc	GCR
NACA0012 (coarse)	130 MB	150 MB	468 MB	329 MB
NACA0012 (fine)	189 MB	241 MB	1064 MB	679 MB
NACA64A010	188 MB	231 MB	1168 MB	791 MB
SDM	382 MB	482 MB	3277 MB	2487 MB
Goland (Euler)	740 MB	1040 MB	9002 MB	6150 MB
Goland (RANS)	6067 MB	-	>64 GB	41.5 GB
LANN Wing	1853 MB	-	>64 GB	22.5 GB

Table 3.6: Solver memory requirement

	Steady State	MG LU-SGS	GMRes	PETSc	GCR
NACA0012 (coarse)	130 MB	1.00	1.15	3.60	2.53
NACA0012 (fine)	148 MB	1.28	1.63	7.19	4.59
NACA64A010	142 MB	1.32	1.63	8.23	5.57
SDM	188 MB	2.03	2.56	17.43	13.23
Goland (Euler)	284 MB	2.61	3.66	31.70	21.65
Goland (RANS)	1033 MB	5.87	-	>63.44	41.14
LANN Wing	414 MB	4.48	-	>158.30	55.65

Table 3.7: Solver memory requirement relative to steady state

MG LU-SGS with the GMRes option. The memory required for PETSc is greater still, being higher than that for the GCR solver, despite the GMRes solver requiring about half the storage of GCR. The main difference between the PETSc and GCR solvers is the use of floats instead of doubles in the preconditioner. This is described further in the supplementary information section S.4. A further comparison can be made through a normalisation by the steady-state memory requirement. This is shown in Table 3.7. The missing entry values or undefined values in the PETSc column are due to either the solver not converging for 0 or 1 fill-in, or the memory required is larger than the amount of memory available for one processor.

Parallel

For large grid sizes, being able to run simulations in parallel, whilst retaining good performance of the method, is a key consideration. The Goland Euler case has been used here to demonstrate the scalability of the GCR method, with the results shown in Fig. 3.9.

It can be seen from Fig. 3.9(a) that with an increase in the number of processors used, the number of iterations for convergence increases. ILU requires a global factorisation of the matrix to be fully effective, however, in parallel, this would require significant communication between processors. An assumption that is made, is to then carry out a factorisation using the part of the Jacobian matrix stored on the local processor.

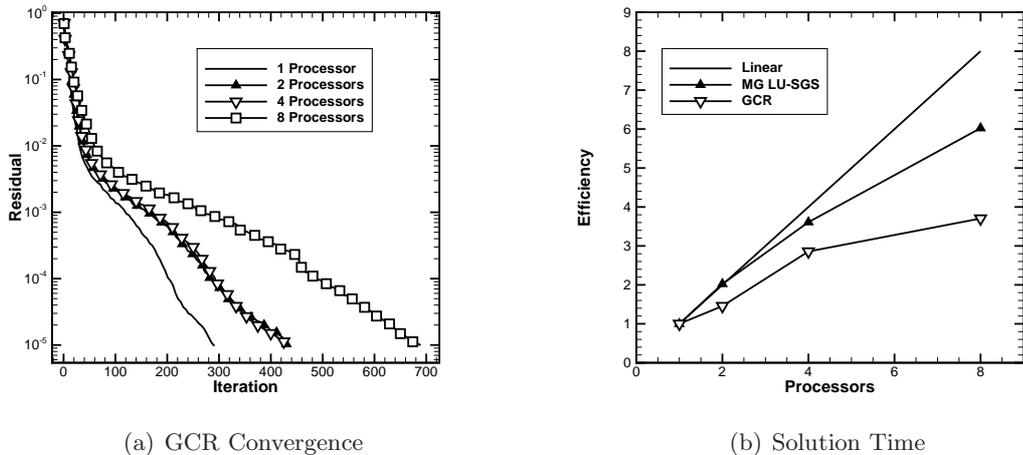


Figure 3.9: Parallel performance (Goland Euler)

The deteriorating performance of the preconditioner as a result of this assumption can be seen in the required number of iterations to convergence increasing with number of processors. In Fig. 3.9(b) the solution time for both the MG LU-SGS and GCR methods is shown for an increasing number of processors. The solution times have been normalised by the time on one processor, with the reciprocal value calculated to obtain the parallel efficiency. This allows for direct comparison of the scalability of the two methods. It is seen that the MG LU-SGS method scales more efficiently than the GCR method, although the difference is not as large as expected. However, when using 8 processors, the MG LU-SGS method is twice as efficient as the GCR approach. The ideal linear scaling is also shown for reference. Using any more than eight processors here causes the problem size to be too small on each processor.

Real vs Complex

There are two ways in which the LFD matrix shown in Eq. (2.21) can be augmented. The first is with four quadrants in the matrix. The leading diagonal containing the Jacobian matrices, and the off-diagonal quadrants containing matrices with the frequency along the diagonal. The second approach is to use a complex formulation. This requires half the memory to store due to taking a single Jacobian matrix and adding the frequency terms to the diagonal. This would also have the benefit of a reduction in the number of operations during matrix-vector multiplications.

The two methods of augmentation of the A matrix are compared for iterative convergence rate and run time. The comparison carried out here uses the Goland wing (Euler) case. The Real and Complex systems are set up, with the solvers being converged 5 orders of magnitude, using one level of fill-in for the preconditioner. The convergence rates of the two solvers are shown in Fig. 3.10.

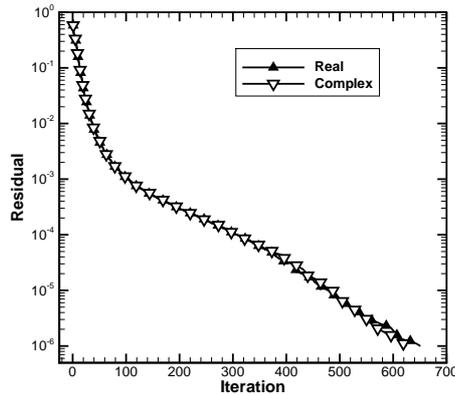


Figure 3.10: Real vs Complex solver convergence (Goland Euler)

	Real	Complex
Iterations	651	625
CPU Time (s)	2486	2128

Table 3.8: Augmentation comparison (Goland Euler)

The tests have shown that the Complex solver does not offer a significant improvement over the Real solver, with an improvement of just 4% in the number of iterations, and 14% in the solution time. The implementation of a complex solver could be considered for future work.

3.1.3 ILU_α Preconditioner

The ILU preconditioner was used for both the PETSc and GCR solution methods. It was apparent that for some cases, at least for PETSc, that the system was too poorly conditioned to achieve convergence. This problem needs to be understood. The difference between the two approaches, when GCR was used in PETSc, was that the preconditioner matrices were based on different Jacobian matrices. With PETSc, the preconditioner was based on the second order Jacobian matrix, whereas for GCR, it was based on the first order Jacobian matrix. There is a significant difference between the two in terms of sparsity and conditioning. A first order preconditioner is typically used due to the first order Jacobian matrix being better conditioned. This improves stability during factorisation in forming the preconditioner. This does however remove the benefit of the second order preconditioner being a better approximation to the inverse of the Jacobian matrix.

For CFD applications, a Jacobian matrix \mathbf{A}_2 , based on the second-order spatial discretisation, is often found to lead to a very poor preconditioner \mathbf{P} in the sense of bad convergence of the Krylov method. This was shown in [38] that for non-symmetric,

non-diagonally dominant matrices, the incomplete factors can be more ill conditioned than the original matrix. It is possible to view this effect by assessing how well the preconditioner approximates the matrix \mathbf{A}^{-1} from looking at the solution of $\mathbf{P}\mathbf{x} = \mathbf{b}$. If the preconditioner was obtained using a direct method (i.e. $\mathbf{P}^{-1} \equiv \mathbf{A}^{-1}$), the exact solution would result. However, as an incomplete factorisation is used, the preconditioner is only an approximation (i.e. $\mathbf{P}^{-1} \approx \mathbf{A}^{-1}$). The exact solution and second order preconditioner solution are shown in Fig. 3.11.

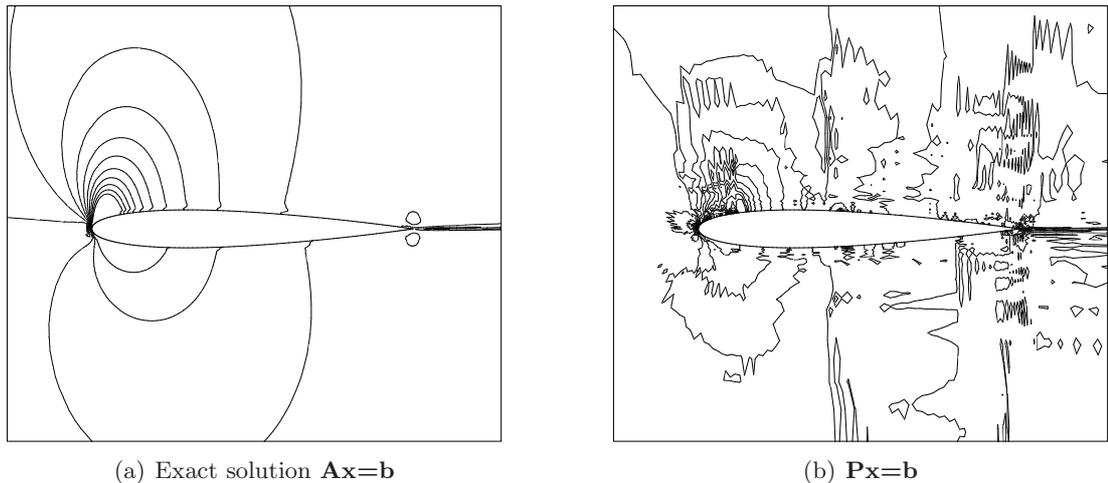


Figure 3.11: Second order preconditioner comparison

It can be seen that the preconditioner based on the pure second order spatial discretisation gives a solution which is highly oscillatory. This is consistent with poor convergence of the Krylov method. Similar unstable behaviour in the forward and backward solves were shown in [40] and [66]. A heuristic fix is to base \mathbf{P} on the Jacobian matrix \mathbf{A}_1 of the first-order spatial scheme, which seems to improve on this situation significantly. This has the benefit of being better conditioned with increased diagonal dominance, as was shown in [67]. The solution of the first order preconditioner is shown in Fig. 3.12.

The preconditioner based on the first order Jacobian has little oscillatory behaviour and is a reasonable approximation to the exact solution. This is due to the better conditioning of the first order Jacobian not causing any stability problems in the factorisation steps for forming the preconditioner. A variation on this approach is to calculate \mathbf{P} based on the matrix \mathbf{A}_α , where

$$\mathbf{A}_\alpha = \alpha\mathbf{A}_2 + (1 - \alpha)\mathbf{A}_1 \quad (3.3)$$

The weighted preconditioner matrix \mathbf{P}_α is then formed from the ILU factorisation of the matrix \mathbf{A}_α . The solution of the new weighted preconditioner is shown in Fig. 3.13.

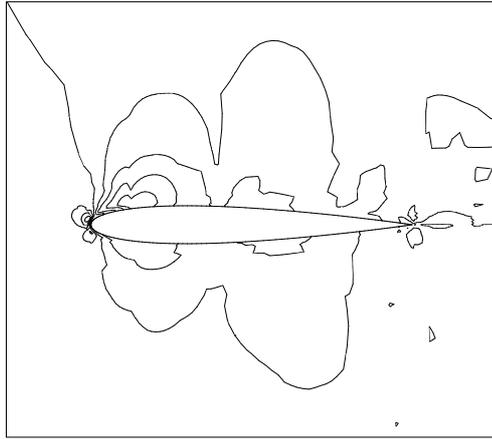


Figure 3.12: First order preconditioner

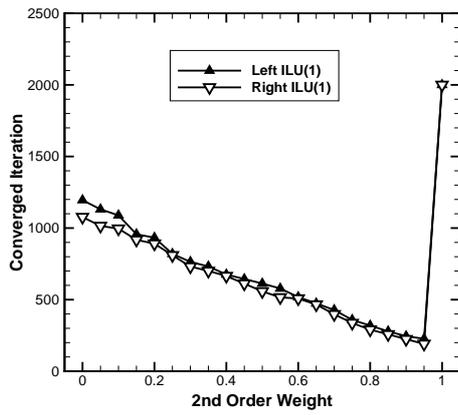


Figure 3.13: Weighted preconditioner $\alpha = 0.90$

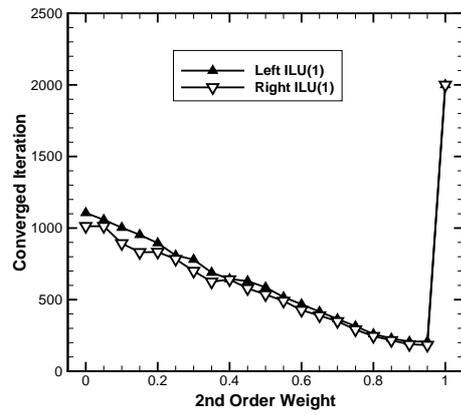
For $\alpha = 0.90$, the majority of the high frequency oscillations seen in Fig. 3.11(b) have been damped by the introduction of a small amount of the first order Jacobian terms. The improved stability in the factorisation and the better approximation from the second order terms would be expected to improve the convergence of the linear solver.

Performance

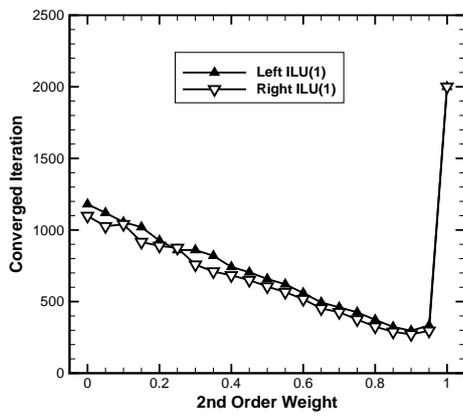
Seeing that a mixed order preconditioner can aid in the convergence of the LFD problem, a sweep of α has been carried out for the test cases previously used. The results are shown for the NACA0012 aerofoil cases in Figs. 3.14(a),3.14(b),3.14(c),3.14(d), the NACA64A010 viscous case in Fig. 3.14(e) and the Golland wing Euler case Fig. 3.14(f). Each case was converged to the same level as in Table 3.2, although the maximum number of iterations was limited to 2000, allowing sufficient iterations for convergence whilst limiting time to solution. Each case is shown for both left and right preconditioning.



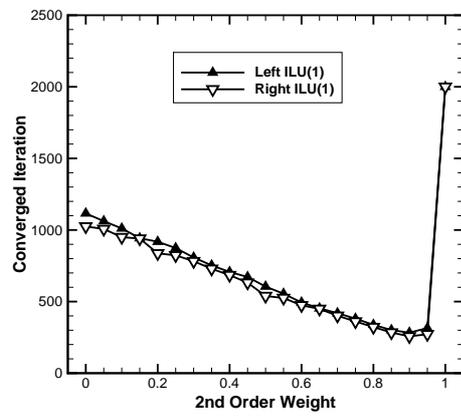
(a) NACA0012 coarse CT2



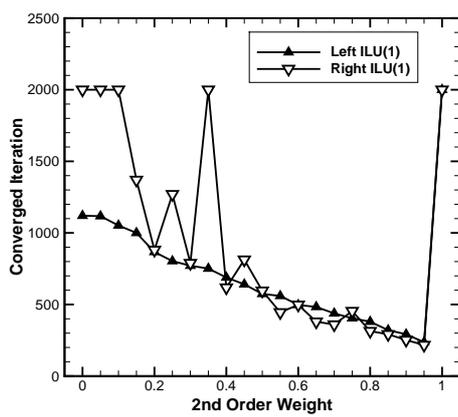
(b) NACA0012 coarse CT5



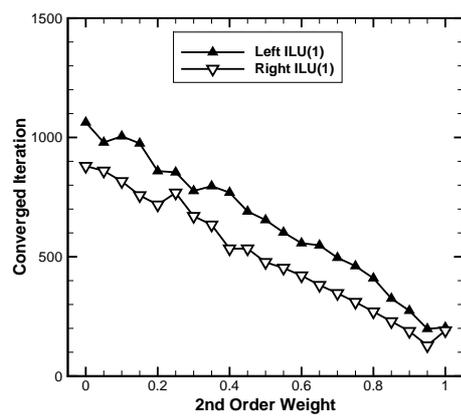
(c) NACA0012 fine CT2



(d) NACA0012 fine CT5



(e) NACA64A010 RANS case



(f) Goland Euler case

Figure 3.14: Influence of preconditioner weighting on convergence

It can be seen that having changed the grid, the flow conditions and the side of preconditioning used, the optimum weight lies between 85% to 90% 2nd order with 15% to 10% 1st order terms respectively. The lack of grid or condition dependence shows the possibility of generalising the weight to improve the solver convergence against the use of a first order preconditioner. The improvement seen is up to about a factor of five over the first order preconditioner, which given the time per iteration is not affected by the different weights, this also equates to a factor of five improvement in solution time.

An indicator of the conditioning of a matrix is to look at the distance of the real eigenvalues from the origin. The largest real positive eigenvalues are plotted against the value of α in Fig. 3.15.

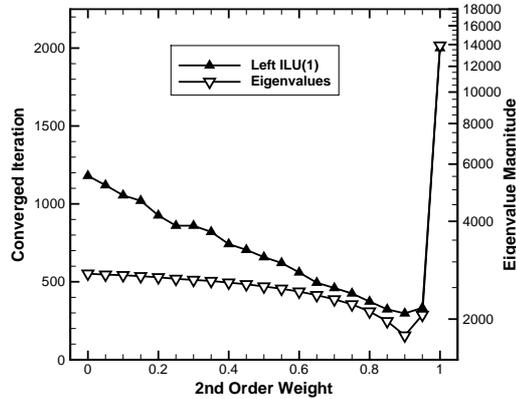


Figure 3.15: Convergence and Real Positive eigenvalue (NACA 0012 fine AGARD CT2)

It is seen that as the number of iterations to convergence reduces, the largest real eigenvalue reduces in magnitude. This indicates that using a mix of the first and second order Jacobian matrices to form the preconditioner, changes the conditioning, and thus the performance of the preconditioner matrix that is formed. A final method of analysis is to take the solution of $\mathbf{Ax} = \mathbf{b}$, and subtract the solution of $\mathbf{Px} = \mathbf{b}$. The L2 norm of the difference gives an indication of the approximation of the factorisation. This is a similar approach to that taken by Duff and Meurant [37] with the Frobenius norm of the remainder matrix, however this is using the solution vectors. The results of this are shown in Fig. 3.16.

As with the previous results, it is clear that the approximation to the inverse of \mathbf{A} varies with the proportion of first and second order Jacobian matrices in the preconditioner. Across all analyses, the preconditioner based on the second order Jacobian matrix is very poorly conditioned, and is a poor approximation. This explains the lack of convergence in the PETSc test cases.

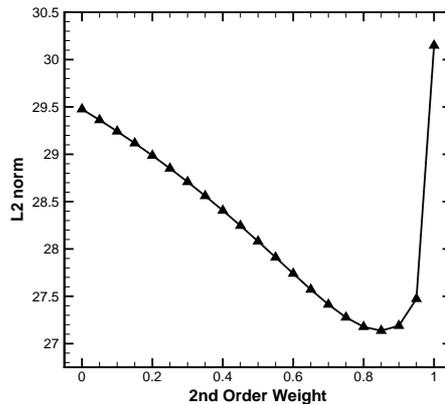


Figure 3.16: Error from exact solution (NACA 0012 fine AGARD CT2)

3.2 Harmonic Balance

The Harmonic Balance solver has the advantage of being able to capture nonlinearities. This proves very useful for calculating dynamic derivatives at the extremes of the flight envelope, where the values from LFD may no longer be adequate. A Harmonic Balance solver has been implemented in both the DLR TAU code and PML CFD solvers as part of this work. The HB implementations do not make use of the aforementioned preconditioner.

3.2.1 TAU Implementation

The HB solution method typically requires a coupled solution of the time slices, with the HB source term applied to the residuals. This proves to be a problem in TAU, where it requires significant memory reallocation, and corresponding code modification to solve the coupled problem. A work-around for this, is to solve the time slices in a decoupled manner, with the HB source term being updated at each step to ensure the solves use the correct \mathbf{w}_{hb} vector. The benefit of this implementation is that the user can select any TAU solver option.

The HB method in TAU is demonstrated here for the NACA 0012 aerofoil test case at CT2 conditions. The pitching moment solution is shown in Fig. 3.17, with increasing number of harmonics retained in the solution.

This case was chosen for demonstration due to the presence of nonlinearities. The LFD solution is not able to predict this loop accurately, as was shown in [23]. If the dynamic derivatives are required at these conditions, it is necessary to use the HB solver. It is seen here that at least three harmonics are required to reconstruct the time-domain solution, with increasing numbers of harmonics improving the match further. Three harmonics are often sufficient to represent the time-domain solution to within sufficient accuracy, although solutions for this are not shown here.

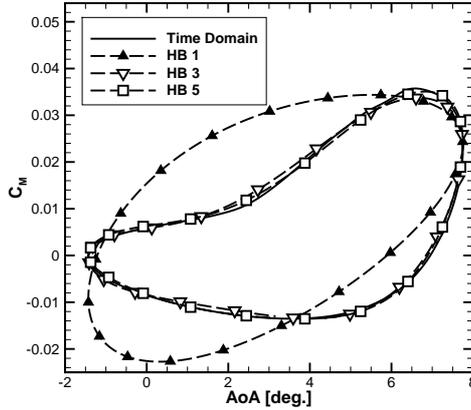


Figure 3.17: Pitching moment coefficient (NACA 0012 fine AGARD CT2)

Given the industrial application of the TAU code, it is necessary to consider the memory requirement for the HB method. This is shown in Table 3.9.

Solver	Memory (MB)
Time Domain	39.5
HB-1	75.4
HB-2	84.1
HB-3	90.6
HB-4	98.0
HB-5	102
HB-6	106

Table 3.9: TAU-HB memory requirement (NACA 0012 fine AGARD CT2)

It can be seen that the memory required to solve the HB system increases with the number of retained harmonics as expected. This is due to the time slice solutions and residuals being stored at a rate of $2N_H + 1$. However, the HB solver requires minimal extra memory than the time-domain solver.

3.2.2 PML Implementation

An alternative approach to the solution of the Harmonic Balance problem is to use an implicit formulation. This is described in Section 2.4.2.1. To demonstrate the capability of the implicit HB solver, the NACA 0012 aerofoil is again used. This time two flow conditions are used. Firstly, an inviscid simulation at AGARD CT2 conditions, and secondly, a viscous case at AGARD CT1 conditions. The point distributions used are shown in Figs. 3.18(a) and 3.18(b).

The pitching moment against angle of attack is shown in Fig. 3.19 for an increasing number of retained harmonics.

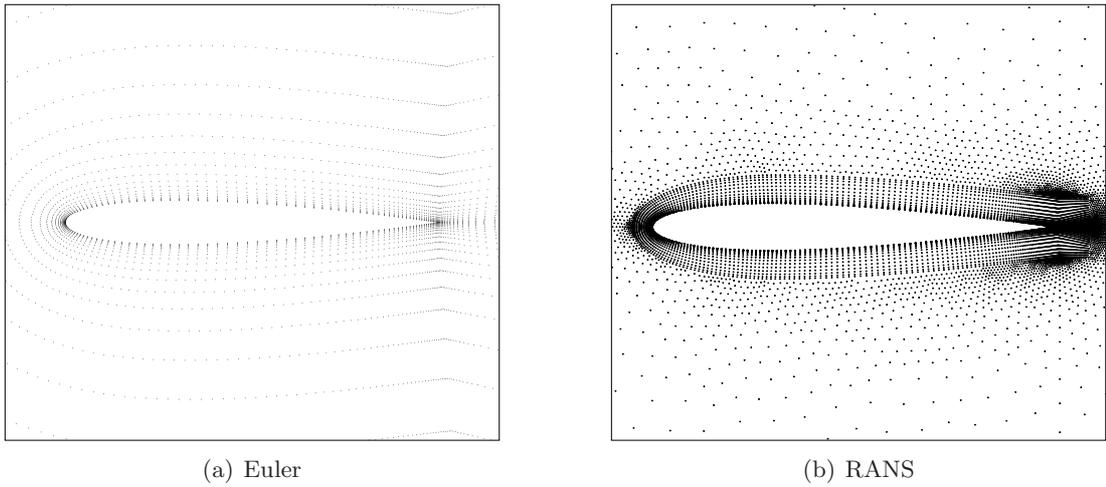


Figure 3.18: NACA 0012 point distributions

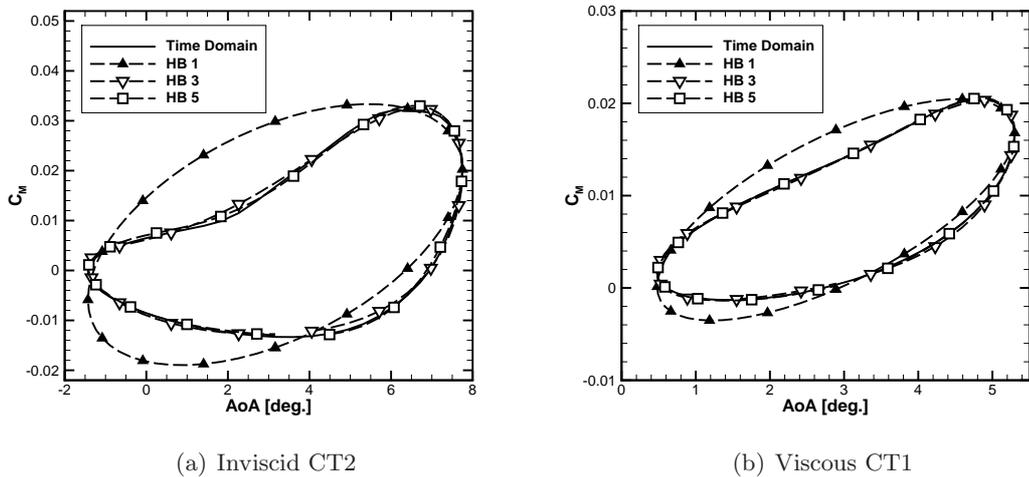


Figure 3.19: Pitching moment coefficient loops from PML HB

The results here are similar to that shown with TAU. The single harmonic solution has not matched the time-domain solution, and increasing the number of retained harmonics leads to better reconstruction of the time-domain signal. It is seen that again, three harmonics for these extreme motions are sufficient to provide a good approximation to the time-domain solution.

The purpose of using the frequency domain methods is to accelerate the time to convergence of a periodic problem. It is therefore useful to compare the solution times. The speed up is plotted in Figs. 3.20(a) and 3.20(b) where the line indicates the time for the time-accurate solver (i.e. 1.0). The time-accurate solves have been run for three cycles with 64 time steps per cycle so that a periodic state is obtained in the solution.

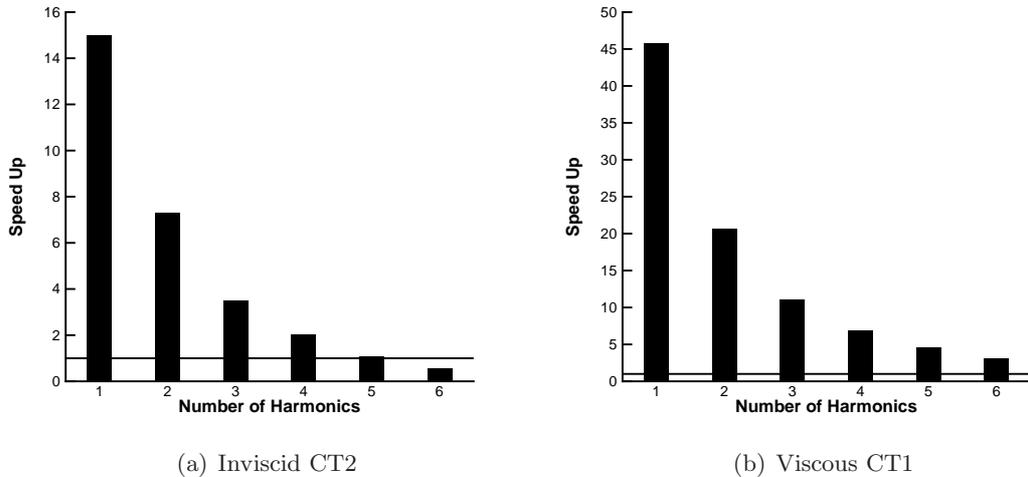


Figure 3.20: Speed up of PML HB

It is clear that increasing the number of harmonics reduces the speed up which can be achieved. However, when combined with the previous plot showing that three harmonics are sufficient for these cases, the HB method is significantly quicker than the time-domain solver, with a similar solution accuracy.

A final consideration is the memory requirement for the HB method which is shown in Table 3.10.

Solver	CT1	CT2
Time Domain	123	68.3
HB-1	383	214
HB-2	694	350
HB-3	1089	531
HB-4	1395	783
HB-5	1953	988
HB-6	2467	1262

Table 3.10: PML-HB memory requirement (MB)

For the fully-implicit formulation, the Jacobian matrix in Eq. (2.36) needs to be stored explicitly in memory and as such, for increasing harmonics, requires considerably more memory than the time-domain solver.

It has been shown that the LFD and HB methods can offer significant benefits in the reduction of computation time for cases involving periodic motions, such as for the calculation of dynamic derivatives. The solution of the LFD problem in TAU has been improved through the use of an implicit method with preconditioner. The preconditioning has been shown to greatly affect the convergence of the system, with a preconditioner matrix based on both the first and second order jacobian matrices offering further benefits in speed up. Although the Harmonic Balance is not as quick as the LFD method, it has shown to be able to capture nonlinearities, whilst having a reasonable speed up over the time-accurate solver.

Chapter 4

Tabular Aerodynamic Model

4.1 Background

During the design phase of an aircraft, a complete flight dynamics assessment of the configuration must be carried out before the aircraft will be certified. As part of this assessment, it is necessary to use an aerodynamic model in order to determine the loads and moments for points through a simulated manoeuvre. The model used is often of tabular type. This is a large database of aerodynamic coefficients for flight parameter combinations. The data within these tables can be obtained from wind tunnel tests. The size of the tables requires tests in the order of millions to cover the flight envelope of a civil airliner. With the increasing use of CFD, computational simulation offers an alternative to empirical sources of data. A number of recent studies have looked into the use of CFD for tabular model generation, and methods to reduce the required number of simulations to populate the tables, as described in the introduction.

In using the tabular models for flight simulation, there are a number of key steps. The first requires construction of the static aerodynamic database or table. For example, an aerofoil with a flap will have three parameters which can be varied to describe the motion, Mach number, M , incidence, α , and control surface deflection, δ_{ele} . Each of these parameters will have upper and lower limits depending on the flight envelope and mechanical constraints of the model. Within this range, an interval must be described at which to measure the aerodynamic coefficients. The following table 4.1 is then formed.

M	α	δ_{ele}	C_L	C_D	C_Y	C_l	C_m	C_n
x	x	x	x	x	x	x	x	x
x	x	x	x	x	x	x	x	x
x	x	x	x	x	x	x	x	x
x	x	x	x	x	x	x	x	x

Table 4.1: Example Aerodynamic Table (x indicates non-zero entry)

Each of the data points in the table is obtained from a static calculation using either empirical or computational methods. In forming this table, the first assumption is made. This assumption is that the resolution defined within the chosen interval is sufficient to capture the flow dynamics through the parameter space. In this example, taking a number of intervals of 100, the total number of simulations required to populate the table is 100^n where n is the number of dimensions in the table. For three dimensions this is 1,000,000. Should CFD be chosen as the source of the data, this could prove to be computationally expensive. In order to reduce the number of required simulations, a further assumption is made. The second assumption is that of decoupling the parameters. In this example, it is assumed that there is no coupling between the incidence and control surface deflection. This leads to two tables 4.2 and 4.3.

M	α	C_L	C_D	C_Y	C_l	C_m	C_n
x	x	x	x	x	x	x	x
x	x	x	x	x	x	x	x
x	x	x	x	x	x	x	x
x	x	x	x	x	x	x	x

Table 4.2: Reduced table for Mach and incidence

M	δ_{ele}	C_L	C_D	C_Y	C_l	C_m	C_n
x	x	x	x	x	x	x	x
x	x	x	x	x	x	x	x
x	x	x	x	x	x	x	x
x	x	x	x	x	x	x	x

Table 4.3: Reduced table for Mach and control surface deflection

The contribution from each table is then summed to provide the total coefficient value at a given point in a manoeuvre. Using two tables, the 1,000,000 initial simulations is reduced to 20,000, a reduction of 50 times.

The second step in using the tabular models for flight simulation requires a method for accounting for the effect of unsteady motions on the loads and moments. This is done using dynamic derivatives. These are described in Section 2.3.1. Dynamic derivatives are a simple way of introducing the unsteady loads to the static values. The modification of the coefficients is as follows:

$$C_j = C_{j0} + C_{j\dot{\alpha}} \dot{\alpha} \quad (4.1)$$

The j subscript corresponds to the load or moment of interest, the zero subscript indicates the static term (from the tables), and the coefficient with a dotted subscript is the dynamic derivative. This equation breaks down the unsteady coefficient into a static

component and an unsteady component based on the instantaneous rate of change of incidence. There are no higher order terms. This leads to the third assumption. There is no accounting for history effects in the flow as a result of rapid motions. Neglecting these effects could prove to be important for certain manoeuvres.

The dynamic derivatives need to be calculated. This is done using forced periodic oscillations. In order to speed up this process, frequency domain methods, described in Section 3.1 and 3.2, are used. Depending on the manoeuvre to be simulated, the conditions for the calculation of the derivatives are chosen to be as similar as possible. This matching of conditions tries to ensure that flow features present in the manoeuvre are captured. As part of the simulation process, a single value for the derivative is used. This leads to the final assumption in this model. It is assumed that the dynamic derivative does not change significantly with the parameters to cause discrepancies in the simulated loads. For small amplitude manoeuvres this may be true, however, for larger amplitude manoeuvres with shocks and separation present, this will not be the case.

With assumptions being made in the formation of the model, it is necessary to determine how good the predictions are. In this chapter, a number of test cases of increasing complexity are used, through a number of flow regimes. The purpose is to demonstrate the performance of the tabular model and establish when it breaks down.

In order to assess the tabular model performance, the framework in [6] is used. This framework requires defining a manoeuvre, running this with a time-accurate CFD solver, the solution to which is used as the baseline comparator, and running the same manoeuvre with the tabular model. The simulated loads are then compared. In the following test cases, the tabular replays have been augmented with the relevant dynamic derivative values calculated at the conditions of the simulation. This is done due to the academic context of this work and a different approach is taken for industrial cases. The derivatives have been calculated using the Harmonic Balance solver, with three harmonics being retained.

4.2 CFD Validation

Before using CFD as a baseline solution, it is necessary to ensure that the model in use is sufficient for simulation of high-rate, high-incidence manoeuvres. The CFD predictions are first validated against experimental data. A motion ranging from the linear region to stall and back has been chosen for this purpose due to the complex flow with dynamic stall present. The motion is described in [68] as case 8 ($M = 0.3$, $Re = 4 \times 10^6$, $\alpha_0 = 10$, $\alpha_A = 10^\circ$, $k = 0.1$). The CFD solution is compared in Fig. 4.1 against the experimental data for both C_L and C_M .

This manoeuvre is the most extreme of those to be considered in this thesis, and as such, is a good upper limit with which to establish the ability of the CFD solver. It is

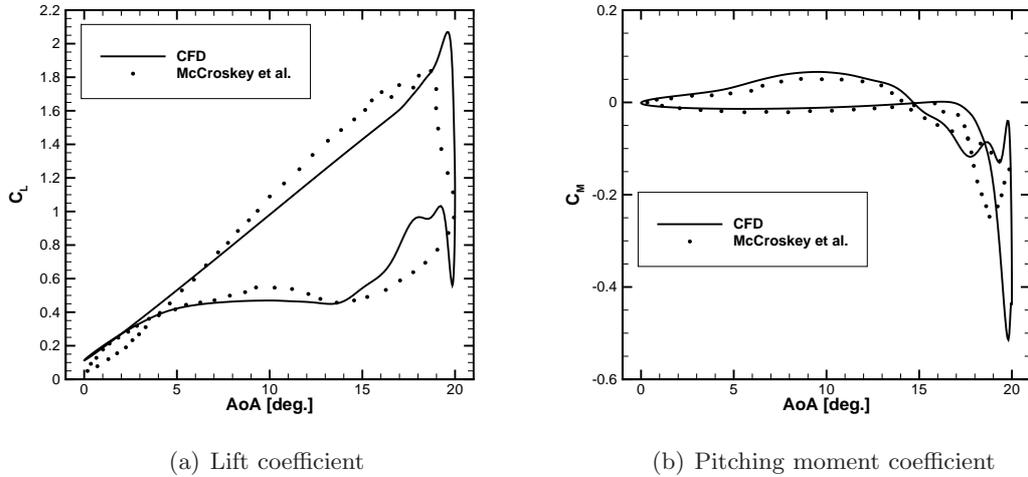


Figure 4.1: CFD validation (McCroskey case 8)

seen that the CFD solution is in good agreement with the experimental data through the majority of the motion, which due to the presence of substantial separation and shocks throughout, is satisfactory. There are discrepancies in the deep stall region, particularly for the pitching moment, which could be attributed to turbulence modelling. A large number of the manoeuvres in this thesis are assessed using the Euler equations rather than RANS to model the flow behaviour and have a corresponding low-rate and low-incidence in order not to push beyond the limits of the model.

4.3 Dynamic Derivatives

For each of the tabular-based simulations, the loads and moments have been modified using the dynamic derivative model described in Section 2.3.1. The dynamic derivative terms have been calculated using the Harmonic Balance solver in the PML code. Three harmonics have been used to capture the majority of the flow dynamics due to having proven to be sufficient in the previous chapter. The conditions used to calculate the dynamic derivatives are those of the manoeuvre. For example, an oscillatory manoeuvre at Mach 0.3, an amplitude of 5.0° and a frequency of 0.01 will have the Harmonic Balance solver run at the same conditions. The reason for this is to capture the flow features present during the manoeuvre. This allows the derivative value to best estimate the unsteady dynamics for the given conditions. The dynamic derivative model is linear, with the value determined from an averaging of the flow features about the mean point. Most of the manoeuvres used in this chapter are either low amplitude, or the flow varies linearly with the change in incidence, thus making the derivative model a good estimator.

4.4 2D Aerofoil

The first, and most simple case, is the NACA 0012 aerofoil with no control surface. This allows an assessment of the tables on a fundamental level, and in a more targeted manner, through removing the impact of assumptions used in forming the model, such as decoupling of parameters. The point distribution used is shown in Fig. 4.2, and has 25,509 points with an initial wall spacing of 1×10^{-5} .

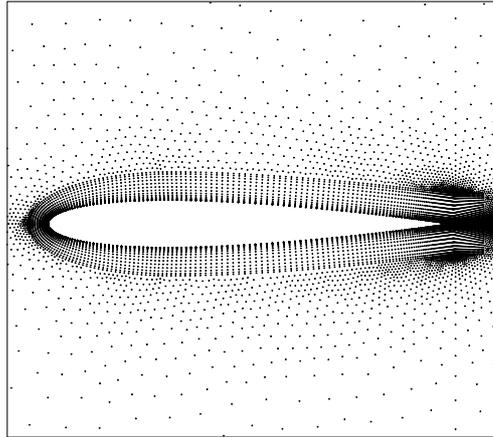


Figure 4.2: NACA 0012 RANS point distribution

4.4.1 Manoeuvres

To assess the tables, manoeuvres must be defined with various levels of complexity. The tables should work where there is linear behaviour of the flow without the presence of shocks or separation. The assessment needs to then progress to establish the tabular capability in the regions where nonlinear behaviour begins to dominate. These, for example, would include transonic Mach numbers or high-incidences, and even combining the two. In order to cover the range of flow regimes, two manoeuvre profiles of varying difficulty have been chosen. The first manoeuvre is a ramp motion. This consists of a constant-rate pitch-up, restricted to pitch only for the cases here. In varying the rate of pitch and beginning or final incidence, different flow phenomena can be present. For the aerofoil ramp case, rates ranging from $0.3^\circ/s$ to $10^\circ/s$ have been selected. This range of rates provides a scale of flow complexities to be studied. The incidence ranges from 0° to 10° at Mach 0.4 and $Re=4.8 \times 10^6$, remaining below stall throughout, to evaluate the adequacy of the tables where the forces and moments behave in a linear manner. Comparison between the CFD and Tabular replays at $2^\circ/s$ is shown in Fig. 4.3.

There is good agreement between the replays for the lift coefficient. It is seen that towards the end of the manoeuvre at the higher-incidence, the two replays begin to move away from each other. This is due to approaching the stall region where

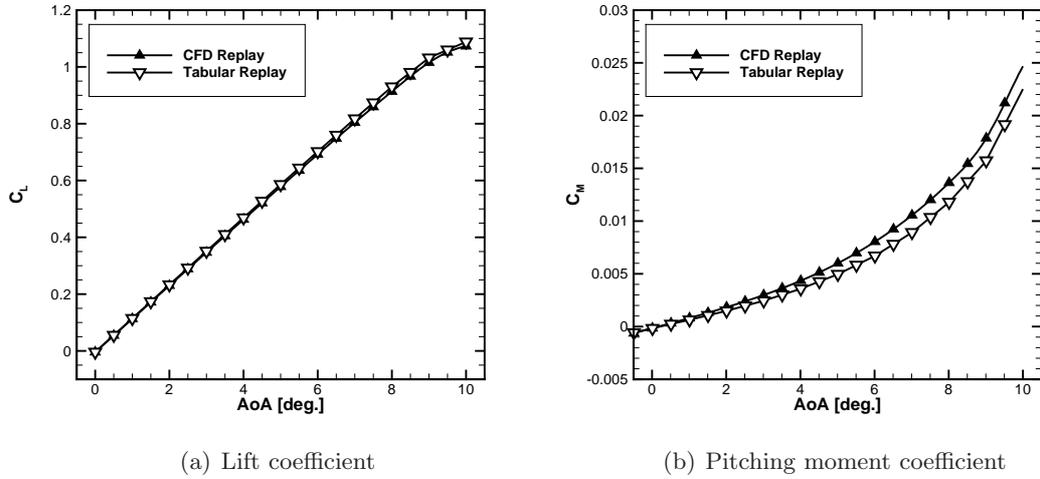


Figure 4.3: Ramp manoeuvre replay 2°/s

separation and shock waves will be starting to form, thus causing differences in the modelling methods. The differences are easier to view in the pitching moment plot. The pitching moment is more sensitive to changes across the aerofoil due to the lever arm amplifying differences at points toward the extremities of the aerofoil body. As the incidence increases, small shocks begin to form on the upper surface of the aerofoil. These causes corresponding changes in the pressure and thus the value of the pitching moment. Differences should be even greater for higher-rates. Comparison is shown in Fig. 4.4 for the 10°/s manoeuvre.

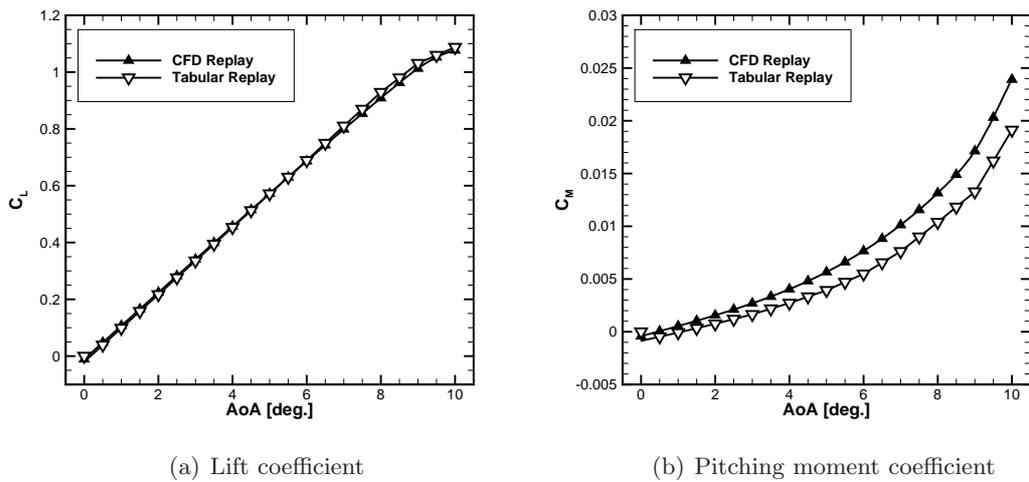


Figure 4.4: Ramp manoeuvre replay 10°/s

It is again seen that there is good agreement between the CFD and the tabular replay for the lift coefficient, with the same differences beginning to appear at the up-

per limits of the incidence. As suspected, the pitching moment coefficient is even more poorly matched than the lower-rate manoeuvre. With increasing the rate of the motion, a lag between the motion and fluid updating is introduced. This effect will cause separation to begin, and combined with the shocks due to the high-incidence, discrepancies between the modelling approaches are seen. These discrepancies are caused by the lack of accounting for history dependent flow features in the tabular model.

The second manoeuvre profile consists of a sinusoidal motion in the stall region. The case used for CFD validation, McCroskey case 8, is again chosen due to the high-incidence and high-rate nature of the motion, along with passing in and out of the stall region, where both linear and nonlinear effects are present. The replays for this are shown in Fig. 4.5 for both the lift and pitching moment coefficients.

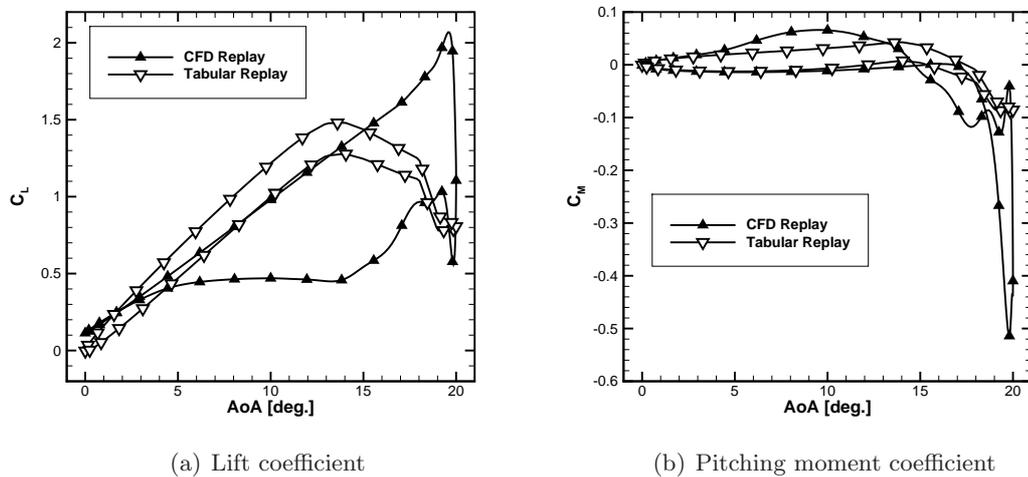


Figure 4.5: Oscillatory manoeuvre replays

It is seen that for this case, where there is large nonlinearity due to dynamic stall, the agreement between the CFD and tabular replays is poor. The lift coefficient has a similar slope through the lower-incidence part of the manoeuvre, although for the rest there is a complete mismatch between the model replays. This is particularly prevalent at the high end of the manoeuvre where the flow is almost completely separated. The tabular replay has a profile similar to a static stall. This is to be expected due to the data being obtained from static calculations. The pitching moment coefficient is surprisingly well matched through more than half of the manoeuvre. However, as with the lift coefficient, at the high incidence, there is substantial difference between the two replays. Again this is due to the separation through dynamic stall not being captured by the static calculations used in the formation of the tabular model. From both plots, it is clear that even the addition of the dynamic derivatives is not suitable. The dynamic derivative model has the effect of adding hysteresis to the loops with the mean being the values taken from the tables. Based on this, there is no possibility that the

tabular model could replicate the response seen in the CFD replay for this manoeuvre. Although this type of manoeuvre is outside the flight envelope of a civil airliner, it is necessary to consider in order to establish the limits of the tabular model. Even for this simple two-dimensional case, with only two tabular parameters, the model is not adequate in this regime. This adds to the motivation to better understand why the model breaks down.

4.5 2D Aerofoil with Control Surface

In order to increase the complexity of the problem, the aerofoil has been modified to include a control surface. This modification increases the dimensionality of the tables, as well as introducing the possibility to have more complex flow features as a result of two moving bodies. Due to the use of two different flap modelling approaches, two aerofoil bodies have been defined. The first defines the aerofoil and control surface as two separate bodies as point distributions, to be used with the PML preprocessor overlap functionality. The two component point distributions have been defined: the body, which is cut at $0.75c$ and has 14,088 points with the farfield at $50c$; and a flap section of length $0.25c$ with 10,339 points and the farfield at $25c$, where c is the chord length. The respective distributions are shown in Figs. 4.6(a) and 4.6(b). All cases where this method has been used have been solved for the Euler equations.

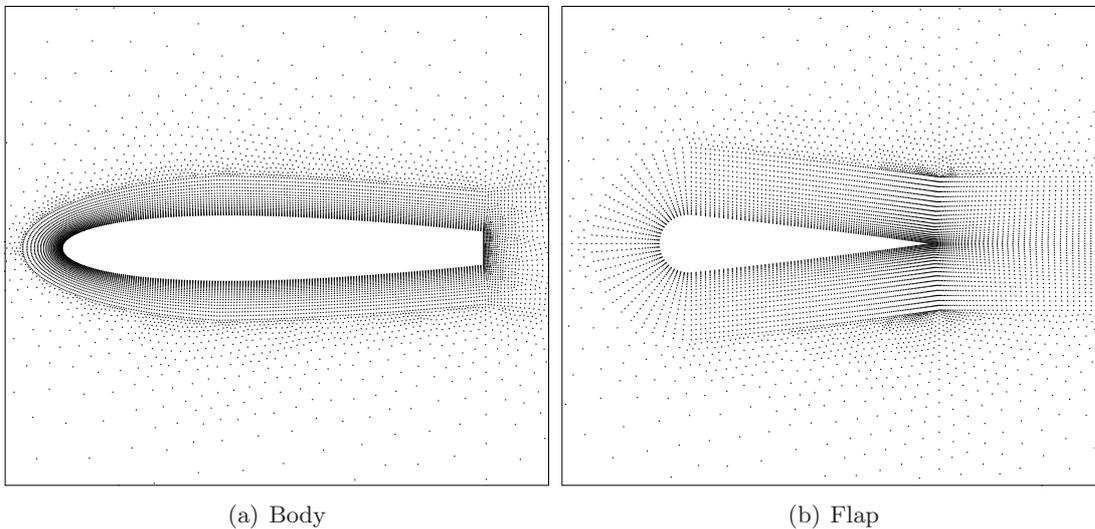


Figure 4.6: NACA 0012 point distributions

The second approach, using the deformation tool (Section 2.2), requires only a single body to be defined. This case makes use of a finer point distribution of 33,393, with a wall spacing of $1 \times 10^{-5}c$, in order to solve the RANS equations. Although the deformation approach requires a single body to be defined, in order to model a control surface deflection, a structure also has to be defined. The points in the

computational domain are mapped to this structure. The structure is then deformed as per the deflection specified by the control input. Using the mapping of fluid points to structural points, the deflected structure causes a deflection in the computational domain. The structure for this case is dense around the hinge point in order to get better definition in the deformation, this is particularly important due to the use of the RANS equations to model the flow. Any sharp changes in the geometry could artificially induce flow phenomena such as separation which would not otherwise be present. The point distribution and underlying structure are shown in Fig. 4.7(a). The structure is deformed with the prescribed deflection as shown in Fig. 4.7(b) for a deflection of $+10^\circ$.

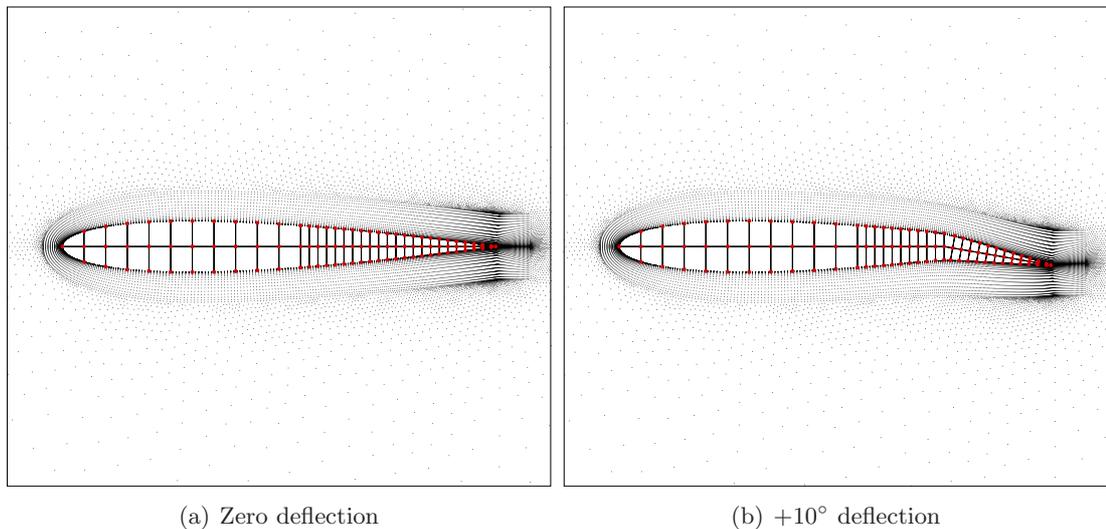


Figure 4.7: NACA 0012 point distributions with underlying structure

4.5.1 Manoeuvres

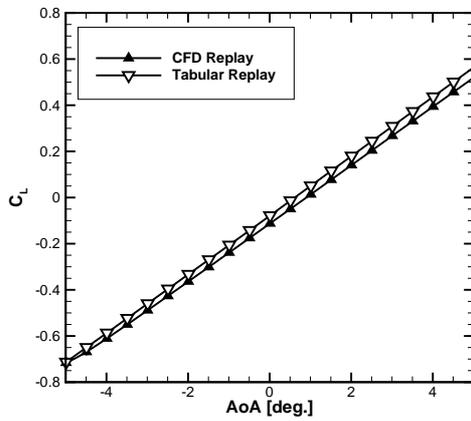
A number of manoeuvres have been chosen in order to cover the range of flow regimes presented in the Introduction in Fig. 1.1. The manoeuvres are also representative of those that are within the flight envelope of a civil airliner. Two types of manoeuvre have been chosen with increasing complexity to allow for a systematic study of the tabular model.

As for the previous case, the most simple manoeuvre is that of a ramp. This involves a pitch up of the aerofoil at a constant rate. Depending on the rate chosen, the aerodynamics for this manoeuvre remain largely in the linear regime. In order to increase the problem complexity through the presence of shock waves, two Mach numbers have been used. Computational methods are of particular use in modelling problems in the transonic regime, where a civil airliner spends most of its time. The ramp manoeuvre is run for the conditions shown in Table 4.4.

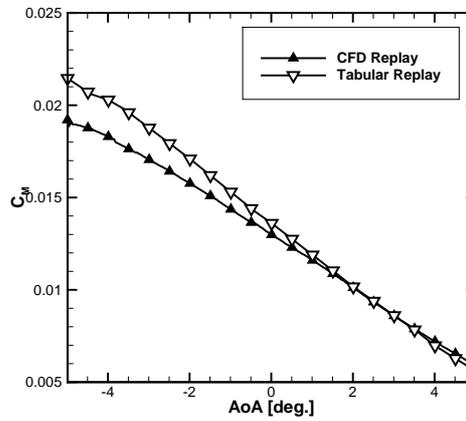
M	α start	α end	$\dot{\alpha}$	δ_{ele}
0.3	-5.0°	5.0°	10.0°/s	-1.0°
0.8	-5.0°	5.0°	10.0°/s	-1.0°

Table 4.4: Control surface ramp manoeuvre parameters

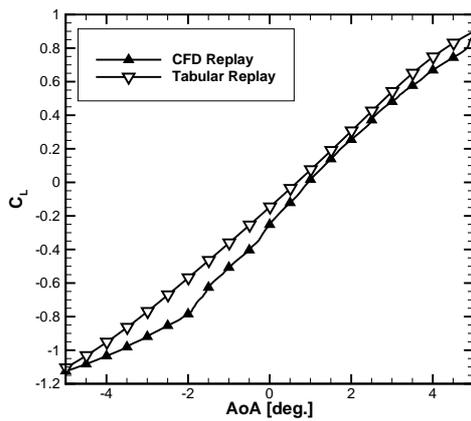
The control surface is held at a constant deflection. In order to calculate dynamic derivative values as close to these conditions as possible, the Mach number, amplitude and mean incidence are the same with a reduced frequency of 0.01. A comparison of the lift and pitching moment coefficients between the CFD and tabular replays is shown for Mach 0.3 and Mach 0.8 in Fig. 4.8.



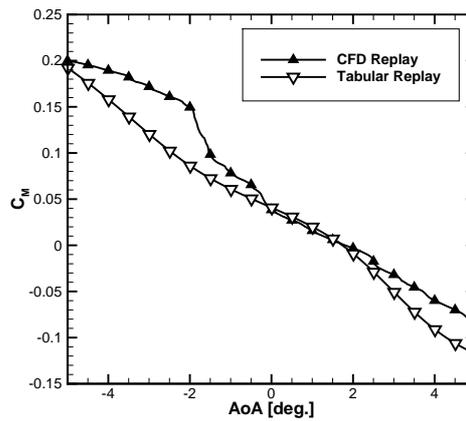
(a) Lift coefficient at M=0.3



(b) Pitching moment coefficient at M=0.3



(c) Lift coefficient at M=0.8



(d) Pitching moment coefficient at M=0.8

Figure 4.8: Ramp motion at 10°/s

The lift coefficient at Mach 0.3 shows good agreement throughout, with the slope being very similar. The small difference begins to increase throughout the manoeuvre. A slope difference is the result of history dependent effects causing a lag between the motion and flow updating. This causes a lower lift coefficient due to the body effectively seeing the flow from the previous timestep. The pitching moment coefficient has a greater discrepancy at the start of the manoeuvre, which reduces as the incidence increases. The initial disagreement could be due to the high incidence of the flap section that is deflected upward. This high incidence will cause unsteadiness in this region, particularly off the trailing edge. The moment arm accentuates the effects, leading to the larger discrepancy. The tabular model with dynamic derivatives has no accounting for these effects and as such are based on a slope for the static calculations. The rate of this manoeuvre is at the upper end of the flight envelope for a civil aircraft, particularly at this Mach number, and will rarely be required in service. The small discrepancy is consistent with what was seen in the simpler single body case at Mach 0.3.

For the Mach 0.8 case however, there is considerable disagreement in both the lift and pitching moment coefficients. For the lift coefficient, the tabular replay shows a consistent slope due to the source of the data. There is some curvature at the extremes of the motion as a result of the high Mach flow causing shocks to form. A steady change in the pitching moment slope is also seen, with a symmetric movement at the upper and lower ends of the manoeuvre. The CFD replay through the first half of the manoeuvre is very different. The shallow slope in both the lift and pitching moment coefficients at the start suggests the presence of stronger shocks than those seen for the static case. At -2.0° there is a sudden change in the gradient of both slopes. Due to this not appearing in the tabular replay, it suggests that it is a history dependent effect that causes this change. Further discrepancy is seen after 2.0° , although it is not as pronounced as at the start. In order to determine if this is due to history dependent effects, the pressure distributions at each side of the slope change can be plotted, for both the steady and unsteady simulations, with a view to determine if it is a result of shock motion. The pressure plots are taken at $\alpha = -2.0^\circ$ and -1.5° , and are shown in Fig 4.9(a) for the steady state cases, and in Fig. 4.9(b) for the unsteady case for the CFD replay.

In the steady-state case, the pressure distributions remain similar between the two incidences. The shock moves by around 2% of the chord and remains on the body of the aerofoil, i.e. upstream of the hinge point. The unsteady pressure plots show something very different. At an incidence of -2.0° , the shock is stronger than that of the steady case, as well as being located on the flap. At this point, the flap has an increased incidence compared to the body and as such will result in greater differences in the lift coefficient. The pressure distribution at an incidence of -1.5° shows a weaker shock which has been displaced by around 10% of the chord. In one-twentieth of the manoeuvre, the shock has moved five times further than for the steady case. This

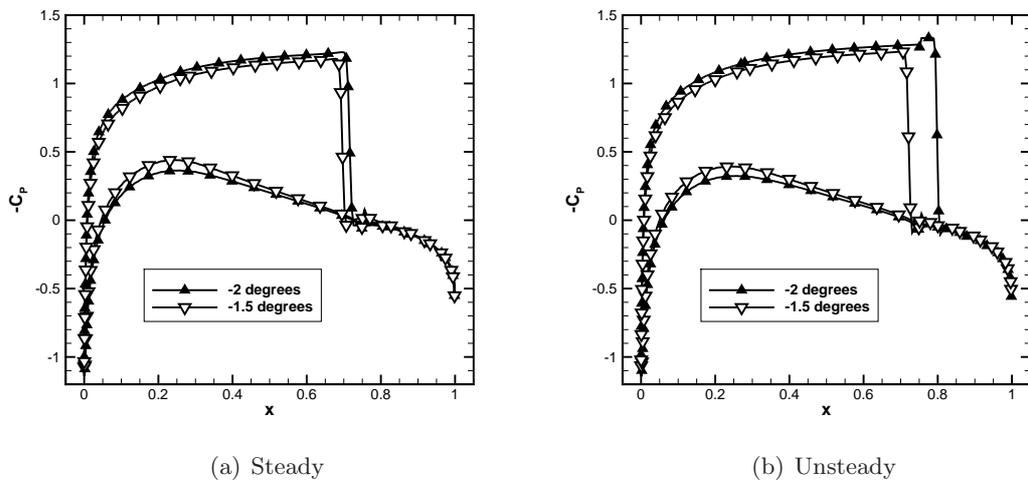


Figure 4.9: Pressure coefficient distribution at two locations in the manoeuvre at Mach 0.8

rapid change in shock location is the reason for the slope change in the lift coefficient plot. The most interesting aspect of these plots is the location of the shock before the slope change. The shock being on the flap is further downstream than where it is for the tabular replay. This suggests there is a lag effect in the flow updating. This is consistent with the other tests at this rate. Again it is shown that the lack of accounting for history effects in the tabular model has led to discrepancies in the replays. It is also possible to view the shocks in the field plots shown for the steady-state and unsteady simulations in Figs. 4.10(a) and 4.10(b).

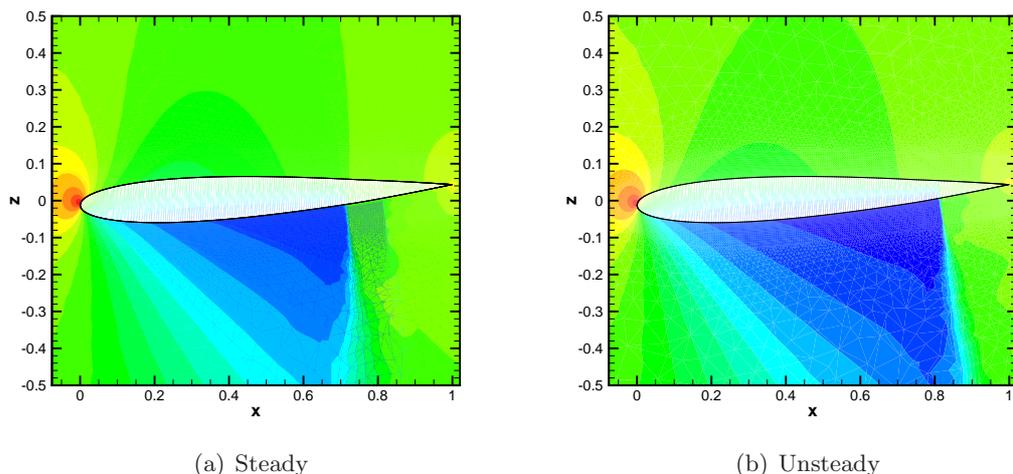


Figure 4.10: Pressure coefficient distribution at an incidence of -2.0° in the manoeuvre at Mach 0.8

From the field plots, it is even clearer that there is a substantial difference in the size and location of the shock. Given that the tabular replay uses the steady-state values, the effect of the shock movement is not captured as it should be, even after accounting for unsteady rate effects through the use of dynamic derivatives.

Adding to the discrepancies is the lack of coupling in the stability and control derivatives. The pitch damping and flap damping derivatives are calculated individually. This assumes that there is no effect between the flap motion and the effects over the body of the aerofoil. This will however not be the case, particularly close to the hinge point where there is a constantly changing surface profile. It may be more beneficial to calculate a value that accounts for combined body motion and flap motion effects. This will be looked at in the next chapter. This case certainly illustrates the limitations of the tabular model for a manoeuvre that is within the flight envelope of a civil airliner.

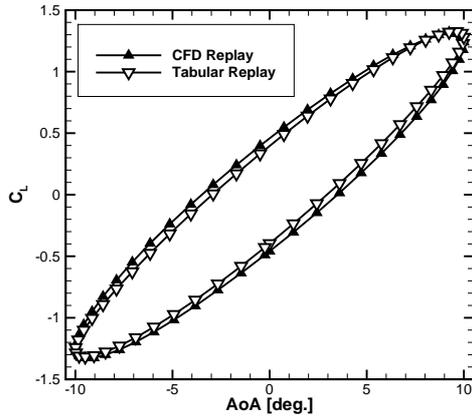
The second manoeuvre type is that of an obstacle avoidance. This manoeuvre introduces a variable rate of pitch, and is set up to have the aerodynamics passing through the linear and nonlinear regimes. The conditions for this manoeuvre are shown in Table 4.5.

M	α_0	α_A	k_α	δ_{ele0}	δ_{eleA}	$k_{\delta_{ele}}$
0.3	0.0°	10.0°	0.01	0.0°	5.0°	0.01
0.8	0.0°	5.0°	0.01	0.0°	2.0°	0.01

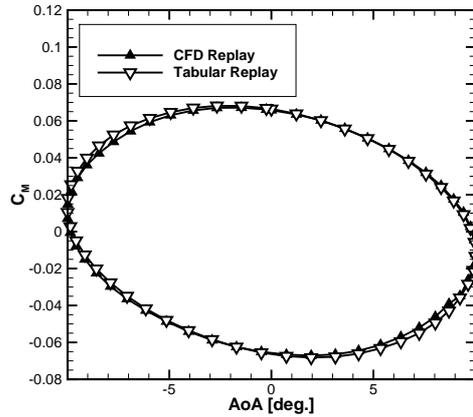
Table 4.5: Control surface obstacle manoeuvre parameters

Rather than being a standard forced oscillatory motion, the obstacle avoidance approach comes from having the flap deflection 90° out-of-phase from the body deflection. This feature has the potential to introduce some interesting flow dynamics, particularly if the speed of the motion is increased to introduce a lag between the motion of the body and updating of aerodynamics. Comparisons for lift and pitching moment coefficient are shown for the two Mach numbers in Fig. 4.11.

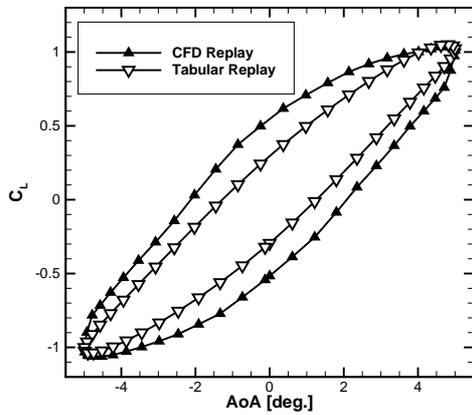
The manoeuvre at Mach 0.3 is the simplest of the two, although it does have a large amplitude in order to introduce some nonlinear behaviour. It is seen that the tabular replay with the dynamic contribution is able to estimate the time accurate replay well. There is a small persistent offset throughout the loop particularly around the +5.0° on the upstroke and -5.0° on the downstroke in the lift coefficient plot, although this is not present for the pitching moment. This difference in hysteresis could be attributed to the dynamic derivative value used. Using a higher value will lead to a greater width in the tabular replay loop and get better agreement with the CFD replay. The derivative value may be slightly off due to the averaging process used in its calculation. The Mach 0.8 manoeuvre is less well predicted. It has been designed to create complex aerodynamics, such as strong moving shocks, so that differences can



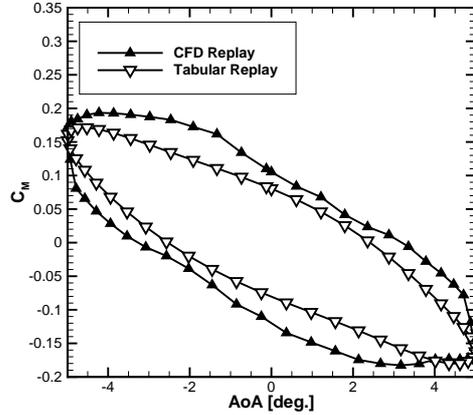
(a) Lift coefficient at $M=0.3$



(b) Pitching moment coefficient at $M=0.3$



(c) Lift coefficient at $M=0.8$



(d) Pitching moment coefficient at $M=0.8$

Figure 4.11: Obstacle avoidance manoeuvre(Euler)

be seen between the CFD and tabular replays. The differences are clear mainly in the width of the loops. The mean of each is fairly similar suggesting there is again an issue with the dynamic derivative used. The tabular model is not sufficient to predict the time-accurate replay for this manoeuvre in either of the lift or pitching moment coefficients. There are large discrepancies in the loop and changes in the curvature of the CFD replay, which are not captured. The changes in slope are consistent with that seen in the ramp manoeuvre. These changes are due to the shock location moving from the flap to the body. Although this is certainly a history effect due to the time lag in the flow and motion updating, the individual calculation of the dynamic derivatives will again play its part. The rapid movement of the shock across the flap hinge also introduces large history effects which are not captured in the tabular model. These two cases show the need to assess the assumptions, in order to determine when it is fit for

purpose.

The complexity of this problem can be increased further by the inclusion of viscous effects in the simulation. The use of the RANS equations to model the aerodynamics allows for separation to increase the number of nonlinearities in the flow, along with causing complex flow features in the boundary layer around the hinge point. The same inputs have been used as for the Euler case. For the RANS simulations, a different point distribution has been used, along with the deformation technique for modelling the flap deflection. The lift and pitching moment coefficient comparisons are shown in Fig. 4.12.

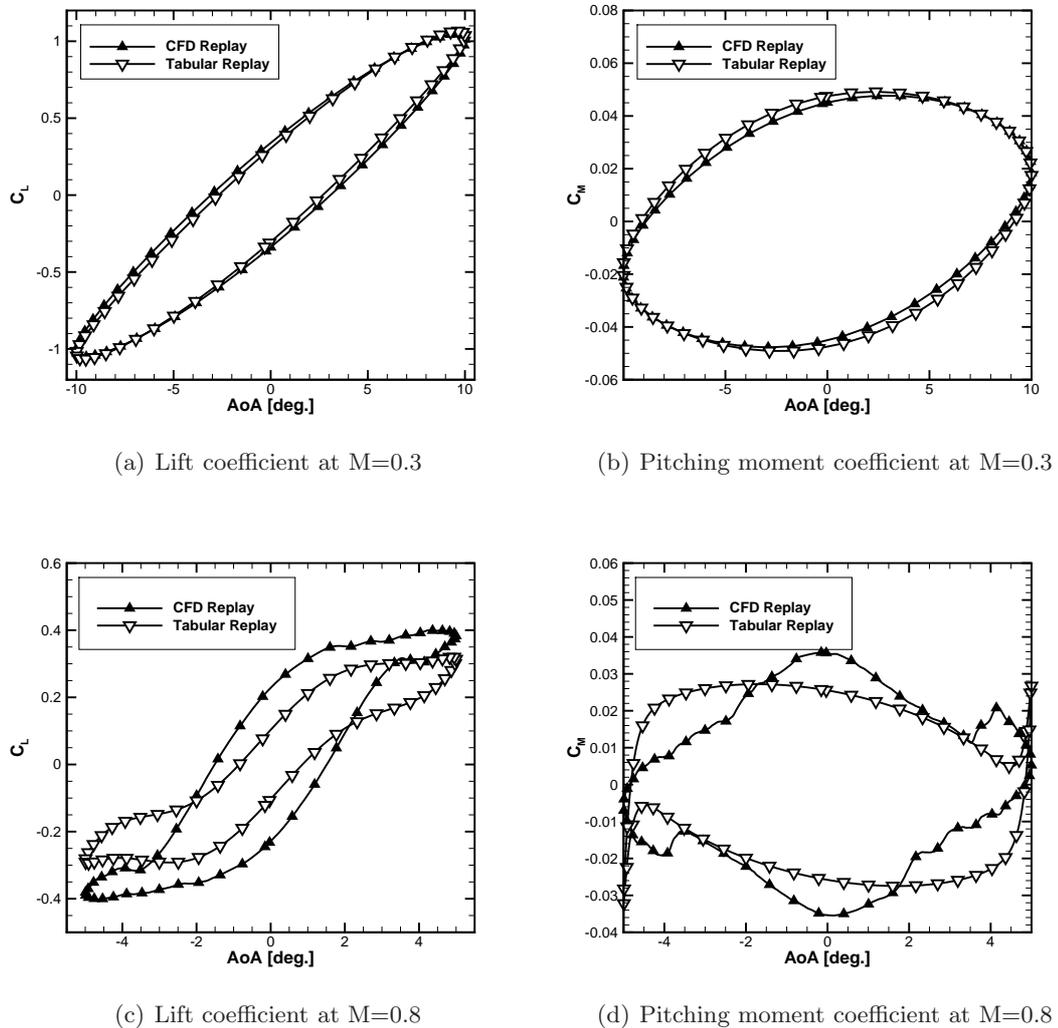


Figure 4.12: Obstacle avoidance manoeuvre (RANS)

As with the Euler simulations, the subsonic case shows excellent agreement between the CFD and tabular replays. There is very little difference between the mean values in the replays and the hysteresis in the loops. However, the transonic case shows

significant discrepancies between the two replays. Unlike the Euler simulations, there is now disagreement in both the mean values at each incidence and the hysteresis. The differences in the mean values suggests there is a delayed stall due to history effects which is not captured in the static-based tabular replay. The lack of matching in the hysteresis is consistent with the inviscid case, and is indicative of the nonlinearities due to shock motion not being captured. The shock motion is particularly noticeable in the pitching moment loop, where at the extremes of the manoeuvre, there is sharp change in the slope. These sharp changes suggest sudden variations in the flow around the flap region due to the moment arm effect again accentuating flow changes. This case is a very good example of when the tables are no longer suitable. The reasons for this will be studied further in the next chapter.

The aerofoil test case has shown on a very basic level that the tabular model is suitable for a number of flow regimes within a typical airliner flight envelope, however, when there are nonlinear flow features present, the model begins to break down. It has also been shown that the dynamic derivative model may not be suitable for these more demanding manoeuvres and should be used when there are limited history effects.

4.6 LANN Wing

The final case is the LANN wing. This is a transport aircraft configuration. The three dimensional nature allows a sideslip parameter to be introduced to increase the dimensionality of the tables. It further adds the possibility to assess the assumption of coupling. For this case the TAU code was used. The computational grid is shown in Fig. 4.13 with 267,463 points. An important note for this case is the use of a symmetry plane during sideslip simulation. Although this is not true sideslip, it is representative. This grid has been used for both the CFD and tabular replays, with the consistency between data sources the most important factor. This treatment is solely a cause of time constraints during the assessment.

The increase in complexity of the geometry allows further comparison between replays for different types of manoeuvres and testing of the tabular model with three-dimensional tables ($[M, \alpha, \beta]$). An oscillatory manoeuvre has been chosen for simplicity simulated at Mach 0.3, with sinusoidal variations in incidence and sideslip. The conditions for this manoeuvre are shown in Table 4.6.

M	α_0	α_A	k_α	β_0	β_A	k_β
0.3	0.0°	5.0°	0.001	0.0°	5.0°	0.001

Table 4.6: LANN wing manoeuvre parameters

As for the control surface obstacle avoidance, in order to create interesting flow dynamics, the incidence and sideslip have been set to be 90.0° out-of-phase. The

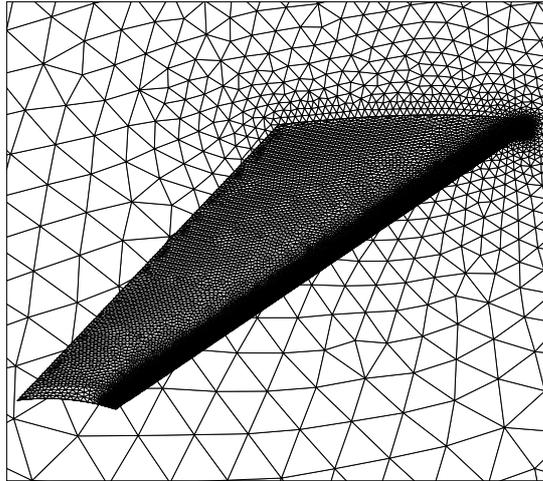


Figure 4.13: LANN wing grid

resulting path traced is a circular motion. A comparison of the CFD and tabular replays is shown in Fig. 4.14 for both the lift and pitching moment coefficient.

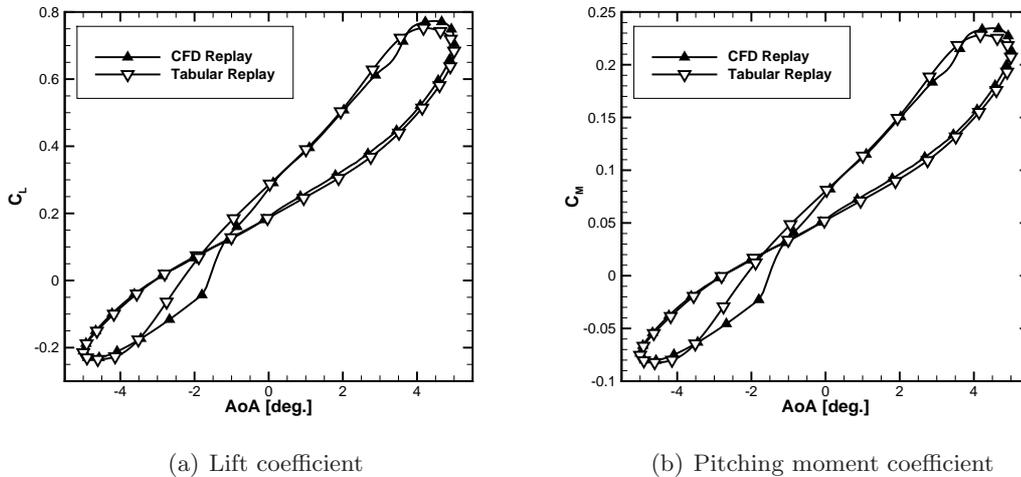


Figure 4.14: LANN wing replay comparison

The low amplitude and low rate motion has been chosen to be representative of that experienced by a transport aircraft wing. As a result, the performance of the model for this configuration should be closer to that in industrial practice on full aircraft. It is seen that throughout most of the motion there is good agreement between the two replays. Yet, despite the low rate and low amplitude, there are still discrepancies between $\alpha = -3.0^\circ$ and $\alpha = 0.0^\circ$. This occurs when the wing is on the upstroke for the incidence and is experiencing an ever increasing sideslip. At this point, crossflow will be increasing across the wing surface. These effects can start to cause an interaction with the oncoming flow and result in nonlinear flow dynamics. Any cross flow

effects, particularly with any history effects will not be captured during steady-state simulations. Further, the coupling of the two motions in incidence and sideslip will not be captured using the dynamic derivative model, again due to individual calculation of the values. Although there is not a large difference between the two replays, it is clear that more complex manoeuvres using this model, such as increasing the amplitude and motions rates, will certainly cause large discrepancies. The flow features present when using a three-dimensional model lead to difficulties in the simulation of the loads using the tabular-based model. This example further shows the need to assess the tabular model in greater detail, particularly with this being a typical civil domain configuration and manoeuvre.

In each of the three test cases presented here, through all flow regimes of interest for a civil aircraft, the comparison between the tabular and CFD replays is consistent with previous studies. For the low rate manoeuvres, where the aerodynamics remains largely linear, the model is able to predict the loads and moments well. However, when the manoeuvres are extended to higher rates of incidence or sideslip, the tabular model breaks down. A number of possible sources of error have been mentioned where this has been the case. Each of the assumptions in forming the table will play their part in errors between the modelling approaches, although to varying intensities. The next chapter will look to assess each assumption in a systematic way, using the examples in this chapter as a basis for forming the test cases and manoeuvre profiles for the study.

Chapter 5

Assessment of Assumptions

5.1 Tabular Model Assumptions

Having established in the previous chapter that the tabular model is not adequate for all manoeuvres of interest, it is necessary to assess the model in greater depth. In the formulation of the tabular aerodynamic model, four key assumptions are made.

The first assumption is a decoupling of certain flight parameters. For example, table 5.1 for $[M, \alpha, \beta, \delta_{ele}]$ can be broken down into two smaller tables for $[M, \alpha, \beta]$ and $[M, \alpha, \delta_{ele}]$ as given in tables 5.2 and 5.3.

M	α	β	δ_{ele}	C_L	C_D	C_Y	C_l	C_m	C_n
x	x	x	x	x	x	x	x	x	x

Table 5.1: Coupled Aerodynamic Table (x indicates non-zero entry)

M	α	β	C_L	C_D	C_Y	C_l	C_m	C_n
x	x	x	x	x	x	x	x	x

Table 5.2: Decoupled Aerodynamic Table for sideslip (x indicates non-zero entry)

M	α	δ_{ele}	C_L	C_D	C_Y	C_l	C_m	C_n
x	x	x	x	x	x	x	x	x

Table 5.3: Decoupled Aerodynamic Table for control surface deflection (x indicates non-zero entry)

The forces and moments are calculated at each step in the manoeuvre from the decoupled tables which are then summed to give the global values. Provided this assumption is valid, the combined loads and moments should be equal to those from the coupled table. The benefit of the decoupling allows the fully coupled table to be

broken down into smaller dimensions, thus reducing the required number of solves to populate the entries.

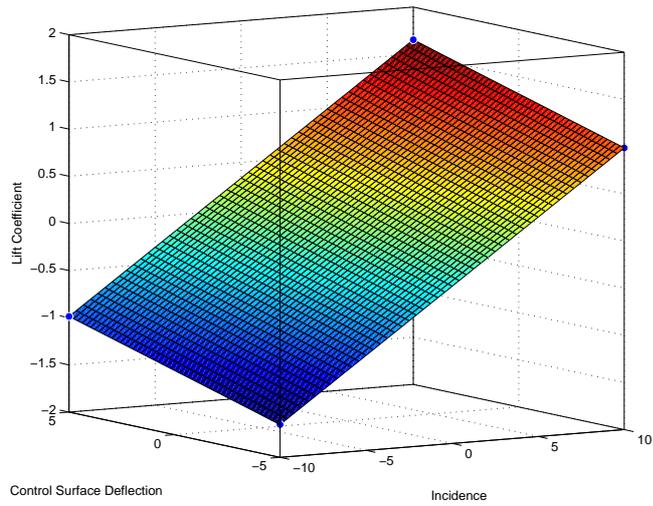
The second assumption is that of adequate resolution within the parameter space. Given the infinite number of possible parameter combinations, it is impossible to calculate the loads and moments for each point. In order to get around this problem, the range of values for each parameter is discretised. At each discrete point, a calculation is made. However, the number of discrete points must be chosen to ensure that the variation of flow conditions within the parameter space is effectively captured. In between the discrete points, sampling techniques are used where a calculation does not exist. This assumption can have implications where there is significant nonlinear behaviour of the flow dynamics. This can be seen through plotting the lift coefficient in the parameter space as a surface for the aerofoil with control surface case at Mach 0.8, as in Figs. 5.1(a) and 5.1(b).

For the above figures, two resolutions have been used across the range $-10.0^\circ < \alpha < 10.0^\circ$ and $-5.0^\circ < \delta_{ele} < 5.0^\circ$. At this Mach number, at the higher incidence there are significant shocks. This leads to a nonlinear variation of the lift coefficient with each of the parameters. This is seen in the curved surface, captured by the 33 point resolution. This however causes a problem for the four point resolution, where the surface is assumed to be bilinear. Using a four point resolution is unrealistic in practice, but it demonstrates the principle. This is an assumption which could impact the simulated loads and moments from the tabular model, particularly in the areas of interest at the extremes of the flight envelope.

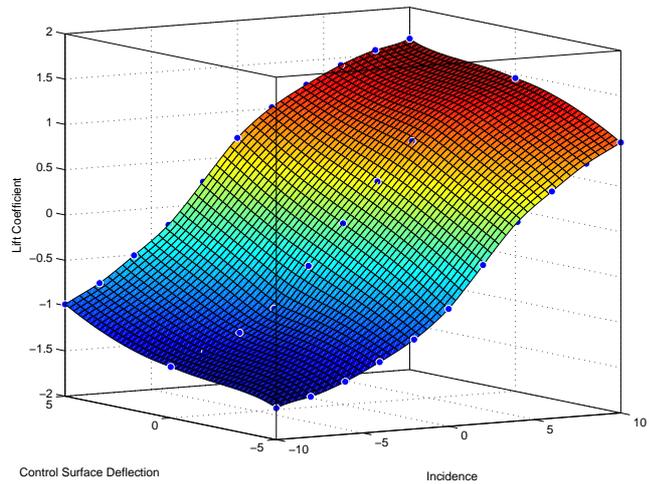
The third assumption comes from the dynamics modelling. The dynamic derivatives have been shown to vary with the conditions for which they are calculated. For example, there may be a variation with reduced frequency or Mach number. The variation in the pitch damping derivative for an aerofoil is shown in Figs. 5.2(a) and 5.2(b) respectively.

In practice, the tabular model is used with derivative values held constant throughout the simulations. This assumes that the value is considered as invariable with flight conditions, which is not the case. Dynamic derivatives are calculated using small amplitude oscillations. This allows the linearisation about the mean point to be considered valid. However, for a large amplitude manoeuvre, the aerodynamics could no longer be varying linearly. Further to this, calculating dynamic derivatives for a small amplitude oscillation, then using it for a high amplitude manoeuvre, could introduce errors through not capturing shocks or separation. This could be overcome through the use of a number of derivative values through the manoeuvre. This will be looked at later in this chapter.

Finally, there is the assumption of neglecting history effects. The values in the tables are obtained from steady-state simulations, these values are then augmented with a dynamic derivative. The dynamic derivative model, as shown in Eq. (2.9), consists of a steady component and an unsteady component based on the instantaneous rates of



(a) 4 points

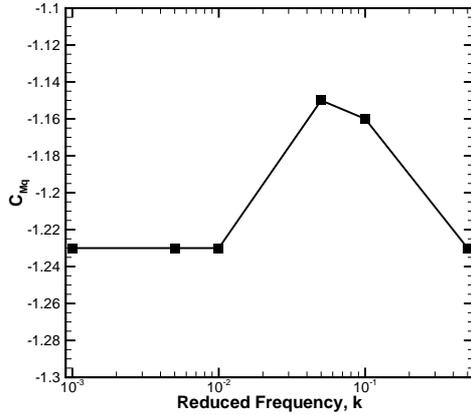


(b) 33 points

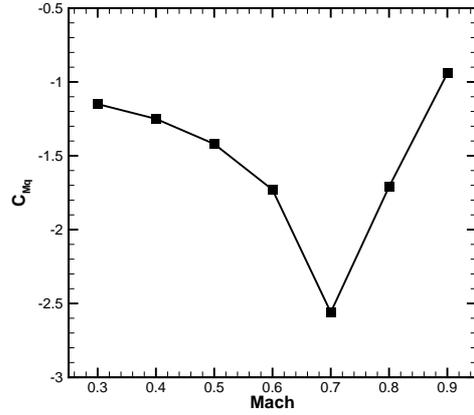
Figure 5.1: Effect of resolution on lift coefficient prediction for aerofoil with control surface at Mach 0.8

the motion. There are no extra terms that account for historical influences as in the unsteady dual-time CFD model. This therefore renders this model as quasi-steady. It is the case, for certain manoeuvres, that history effects become significant. A number of examples were shown in the previous chapter.

There is no previous literature on the impact of each of these assumptions on the modelling of loads and moments for flight dynamics analysis. In this chapter, an assessment is carried out, building on the work in [8]. This chapter continues with an assessment based on forced motions, before moving to the applicability to control applications using free-response manoeuvres. The chapter finishes with an assessment of



(a) Variation with reduced frequency



(b) Variation with Mach number

Figure 5.2: Effect of input parameters in pitch damping derivative

the dynamic derivative model and using multiple values versus a single derivative. The error introduced for a number of manoeuvres is shown.

5.2 Forced

As part of this assessment, two simulation approaches have been taken. The first is the use of forced motions to isolate the impact of each assumption on the loads and moment prediction of the tabular model. The test cases used in the previous chapter are used. The focus is on the aerofoil with trailing edge flap and the LANN wing due to the increased dimensionality in the tables increasing the problem complexity. The manoeuvre profiles used are also those from the previous chapter and are described where relevant.

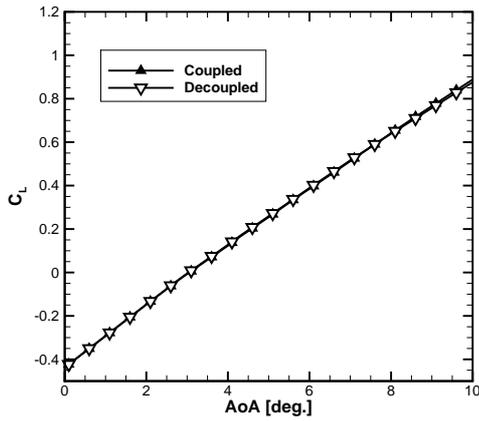
Coupling

The first assumption to consider is that of the decoupling of parameters to reduce the size of the tables. This is first tested using the aerofoil with control surface. The table of variables $[M, \alpha, \delta_{ele}]$ is reduced to $[M, \alpha]$ and $[M, \delta_{ele}]$, in order to decouple the effects between the incidence of the aerofoil and the flap deflection. The parameters have the ranges in table 5.4.

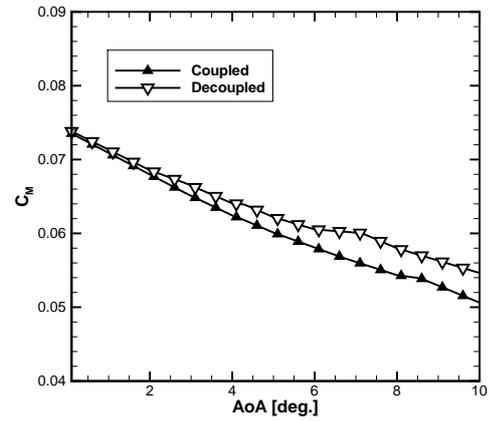
	Mach	α	δ_{ele}
Range	0.3 to 0.8	-10.0° to 10.0°	-10.0° to 10.0°
Interval	0.1	2.0°	5.0°

Table 5.4: Coupled and decoupled table parameter ranges

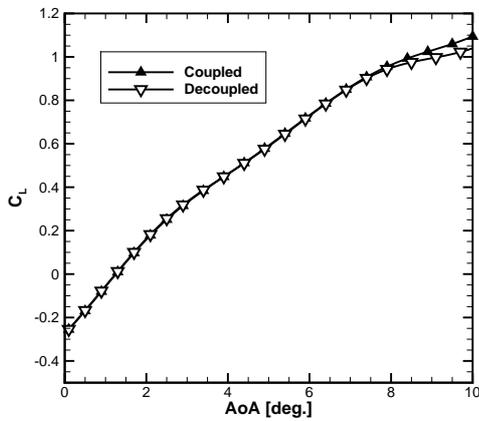
The assumption is assessed using replays with the tables, and comparing the decoupled solution with that of the coupled solution. Two manoeuvres have been selected. The first is a ramp profile starting at an incidence $\alpha = 0.0^\circ$ and pulling up to 10.0° at a rate of $10.0^\circ/s$, with a constant control surface deflection $\delta_{ele} = -5.0^\circ$. In order to assess the coupling, it is necessary to have a varying third dimension. This is done by varying the Mach number which starts at 0.5 and decreases through to 0.3 as the incidence increases. The second manoeuvre is made more complicated by increasing the starting Mach number. Again, there is a ramp profile from $\alpha = 0.0^\circ$ to 10.0° at a rate of $10.0^\circ/s$, but with a constant control surface deflection $\delta_{ele} = -2.0^\circ$. The Mach number is increased to start at 0.8 decreasing to 0.5. This leads to passing through the transonic regime and thus introduces nonlinear behaviour of the flow. The two manoeuvres have been run with a coupled and a decoupled table formulation. A comparison of the simulated lift and pitching moment coefficients is shown in Fig. 5.3.



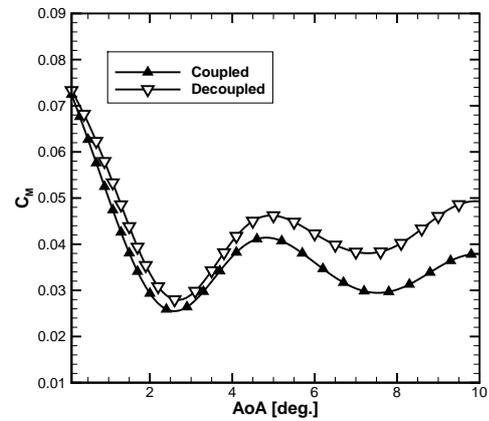
(a) Lift coefficient at Mach 0.5 to 0.3



(b) Pitching moment coefficient at Mach 0.5 to 0.3



(c) Lift coefficient at Mach 0.8 to 0.5



(d) Pitching moment coefficient at Mach 0.8 to 0.5

Figure 5.3: Aerofoil with control surface ramp replays with and without coupling

With regard to the lift coefficient, it is seen for the lower Mach case that there is little difference throughout much of the manoeuvre up until around $\alpha = 9.0^\circ$ where there are shocks forming on the upper surface of the aerofoil. In the linear region, the agreement is almost perfect, with the assumption of decoupling having a negligible effect on the simulated lift for this manoeuvre. For the higher Mach case, a similar result is seen with good agreement up until $\alpha = 7.0^\circ$. At this Mach number the shocks are much stronger, thus increasing the nonlinearity in the lift. The shocks are present from around half way through the manoeuvre and become more significant as the angle increases. This increase in complexity shows where this assumption becomes a problem. The discrepancies are as much as 5% to 10% toward the end of the manoeuvre and would likely increase as the incidence increases further. For the pitching moment coefficients, the discrepancy builds as the manoeuvre progresses. For both the low and high Mach cases, there is good agreement at the start, which is to be expected considering the benign conditions. However, the discrepancy appears to be compounding as the incidence increases. This is likely caused by a coupling influence between the reduced flap incidence due to deflection. The error in the high Mach case is around 25%. In this instance the assumption is not of great concern, although the errors could become more significant for some manoeuvres. This is because at higher Mach numbers in the civil domain, manoeuvres are typically low incidence and very low rate, leading to the conclusion that the discrepancies shown here will not be relevant.

Although the main focus of this assumption is on the decoupling in the tables, it is possible to extend this to an assessment of coupling in the dynamic derivative used. Typically, a stability derivative is calculated for the body motion with $\delta_{ele} = 0.0^\circ$, and a control derivative is calculated with a moving flap and stationary body. The two are then multiplied by their respective rates in the manoeuvre and summed to obtain the unsteady contribution. However, for high Mach, high incidence manoeuvres, the coupling between body and flap motion could become significant as was seen in Fig. 4.11. When the derivatives are calculated individually, the simulations will not capture the influence of the bodies moving in relative motion and the subsequent impact on the flow dynamics. For example, the stability derivative is calculated by an oscillation in the pitch with the control surface at a deflection of 0.0° . Likewise the control derivative is calculated by oscillating the flap whilst holding the body at an incidence of 0.0° . It is possible to run a simulation with the body and the flap moving, then calculating a combined derivative. This leads to coupling effects being captured, although linearised about the mean state. In order to assess this, the obstacle avoidance manoeuvre from the previous chapter is taken, and is simulated for a subsonic case at Mach 0.3, and a transonic case at Mach 0.8. The individual dynamic derivative values have been calculated using the decoupled approach, and then also a coupled approach. The coupled approach used is a direct simulation of the manoeuvre using CFD, with the derivative calculated relative to the pitch. The tables can be used with the two types of derivative

to simulate the obstacle manoeuvre. The simulated lift and pitching moment coefficient responses are shown in Fig. 5.4.

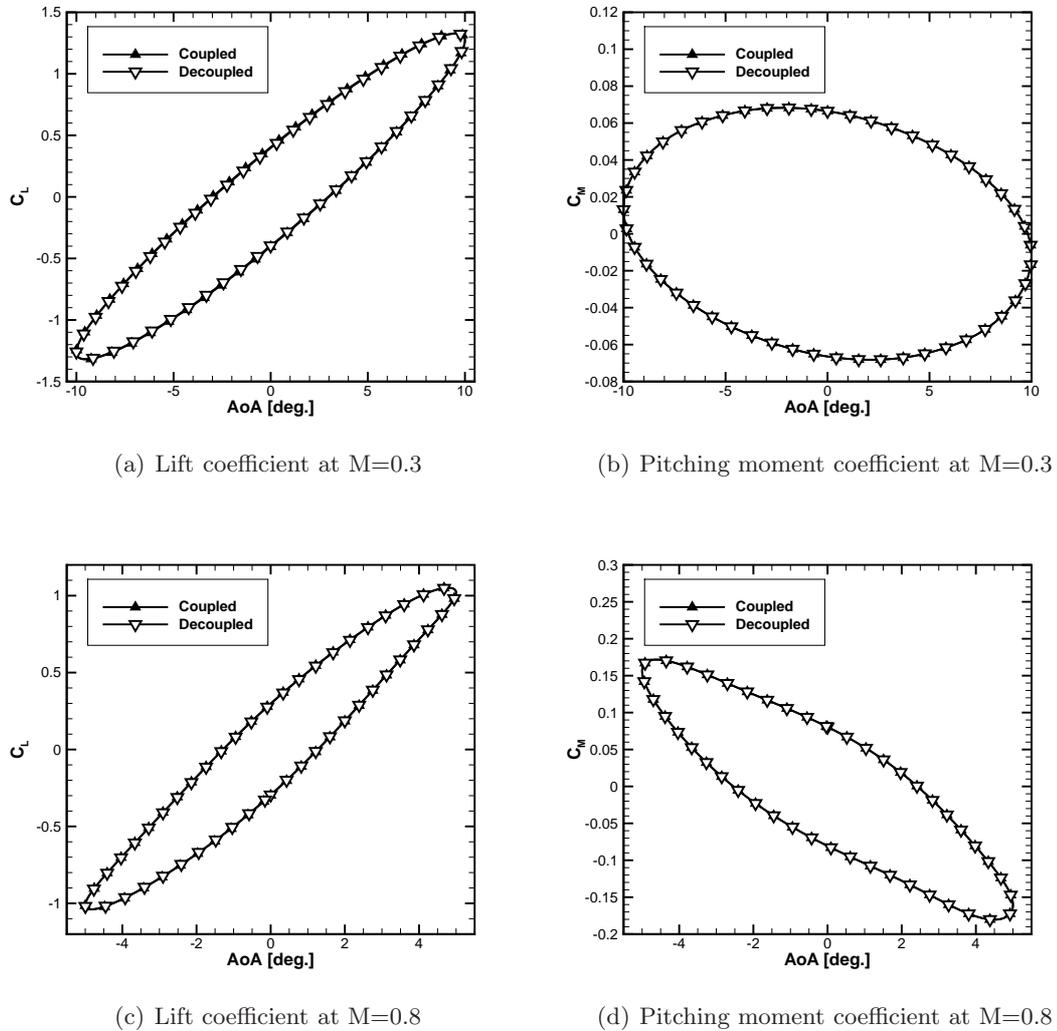


Figure 5.4: Aerofoil with control surface obstacle replays with and without coupling in the derivative

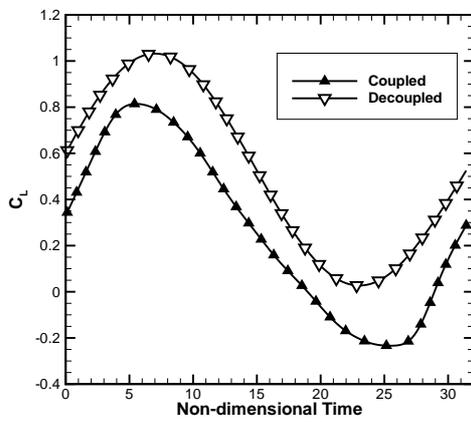
This case has been chosen due to the presence of discrepancies in the simulation in the previous chapter. Despite this, it is seen that for both the subsonic and transonic Mach cases, the difference between the coupled and decoupled approaches is negligible. The transonic case does contain some significant nonlinear behaviour, but this does not appear to have affected the replays. For this case it can be concluded that the assumption of decoupling of the dynamic derivatives is valid.

A coupling assessment can also be carried out for the LANN wing. The coupled table $[M, \alpha, \beta]$ can be broken down into two, two dimensional tables in $[M, \alpha]$ and $[M, \beta]$. The manoeuvre parameters are shown in table 5.5.

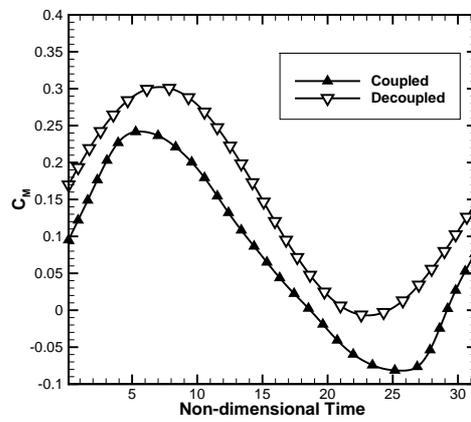
α_0	α_A	k_α	β_0	β_A	k_β
0.0°	5.0°	0.001	0.0°	5.0°	0.001

Table 5.5: LANN coupling manoeuvre parameters

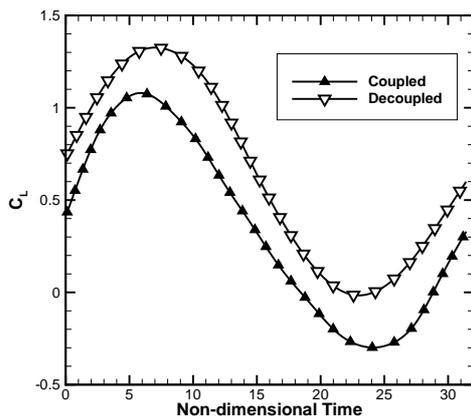
Again, in order to introduce variation in all three dimensions, two Mach number regimes have been chosen. Firstly, an initial Mach number of 0.5 decreasing to 0.3, and an initial Mach number of 0.8 decreasing to 0.5. The simulated lift and pitching moment coefficients using the coupled and decoupled approaches are shown in Fig. 5.5 for the two Mach regimes.



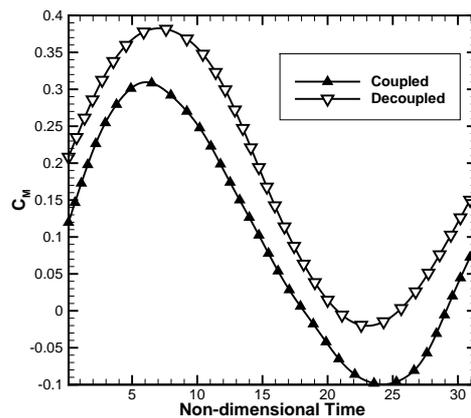
(a) Lift coefficient at Mach 0.5 to 0.3



(b) Pitching moment coefficient at Mach 0.5 to 0.3



(c) Lift coefficient at Mach 0.8 to 0.5



(d) Pitching moment coefficient at Mach 0.8 to 0.5

Figure 5.5: LANN Wing: coupled and decoupled replays

It is seen that there is a substantial difference between the coupled and decoupled tabular replays. This is to be expected when considering what happens at each point in

the manoeuvre. For example, at the start point in the coupled replay, $M=0.5$, $\alpha = 0.0^\circ$ and $\beta = -5.0^\circ$. This is also true of the decoupled contribution from the $[M, \beta]$ table. Due to the wing not having a symmetric section, there will be a non-zero contribution from the $[M, \alpha]$ table. The sum of the decoupled parts will always be offset from the coupled replay. This is not seen in the control surface comparisons due to the deflections considered to be zero, therefore no contribution from the $[M, \delta]$ table. This could be improved by calculating the $[M, \beta]$ table for an α value where the contribution is zero for all coefficients, if this exists. However, when the dimensions of the tables increase, it would be increasingly difficult to find a point where the contribution is zero.

Resolution

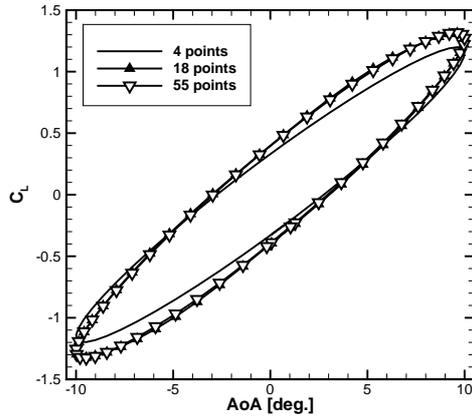
The resolution of the tables is critical to the efficacy of the model, particularly where the aerodynamics are rapidly changing. In order to assess this, the obstacle case has been run at Mach 0.3 and 0.8 with various resolutions in the tables. A number of resolutions have been chosen. The parameter range and intervals are shown in table 5.6 per Mach number.

α Range	α Interval	δ_{ele} Range	δ_{ele} Interval	Number of Points
-10.0° to 10.0°	2.0°	-10.0° to 10.0°	5.0°	55
-10.0° to 10.0°	4.0°	-10.0° to 10.0°	10.0°	18
-10.0° to 10.0°	20.0°	-10.0° to 10.0°	20.0°	4

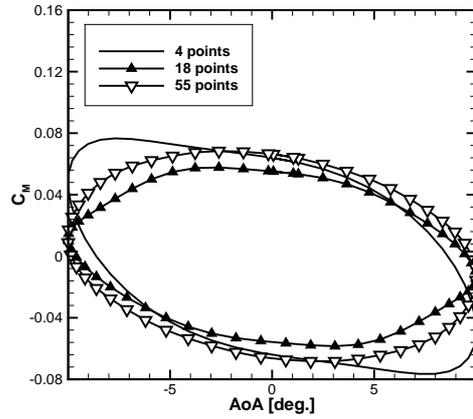
Table 5.6: Aerofoil with control surface table resolutions per Mach number

The finest resolution has 55 table entries for each Mach number. This is the resolution used in the previous chapter and was sufficient to simulate the CFD solution. The second resolution is reduced to 18 points and the final resolution has just 4 points. The coarsest is the 4 corners of the parameter space such as that seen in Fig. 5.1(a). A table can be formed for each of the resolutions, with the replays run for the obstacle manoeuvre at the two Mach numbers. The simulated lift and pitching moment coefficients are shown in Fig. 5.6.

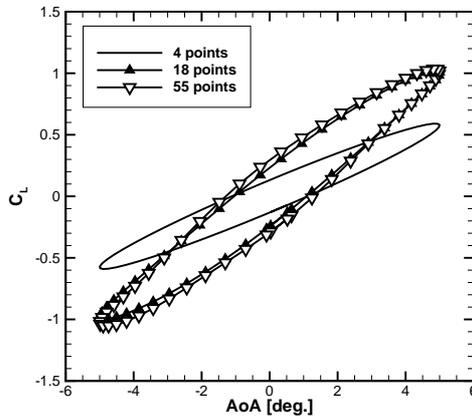
It is seen that for the subsonic case, there is little difference as a result of the different resolutions. This is to be expected due to the motion remaining in the linear regime. There is, however, a small difference using the coarsest table, which could be attributed to the parameter space no longer being bilinear at the extremes. For the Mach 0.8 case there is little difference in the two finest resolutions, where there is significant nonlinearity in the lift against incidence and control surface deflection. There is, however, a large difference when using the coarsest table, which is to be expected due to the nonlinearities previously mentioned. It is, however, highly unlikely that the tables used for a simulation in this regime will be as coarse as used here. As



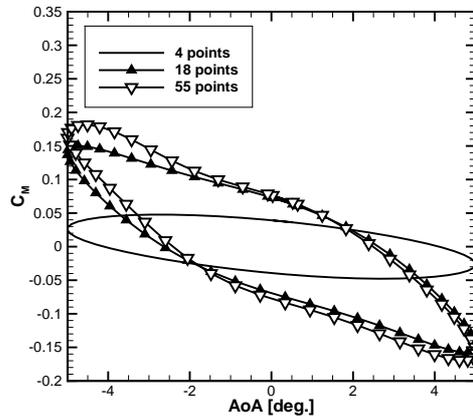
(a) Lift coefficient at $M=0.3$



(b) Pitching moment coefficient at $M=0.3$



(c) Lift coefficient at $M=0.8$



(d) Pitching moment coefficient at $M=0.8$

Figure 5.6: Aerofoil table resolution comparison

such, the resolution is not a significant factor in the ability to resolve variations in the parameter space for the case considered here.

The same comparison can be done for the LANN wing. The resolutions used for this case are shown in table 5.7.

α Range	α Interval	β Range	β Interval	Number of Points
-8.0° to 8.0°	2.0°	-6.0° to 6.0°	2.0°	63
-8.0° to 8.0°	4.0°	-6.0° to 6.0°	4.0°	20
-8.0° to 8.0°	16.0°	-6.0° to 6.0°	12.0°	4

Table 5.7: LANN wing table resolutions per Mach number

The finest resolution has 63 data points, which again is that used in the previous chapter where it was sufficient to capture the necessary flow phenomena. The second

resolution is reduced to 20 points, and the coarsest table has just 4 points. Again, tables are generated for each of the resolutions, and the oscillatory manoeuvre used earlier, is simulated using these tables. The simulated lift and pitching moment coefficients are shown for two Mach numbers in Fig. 5.7.

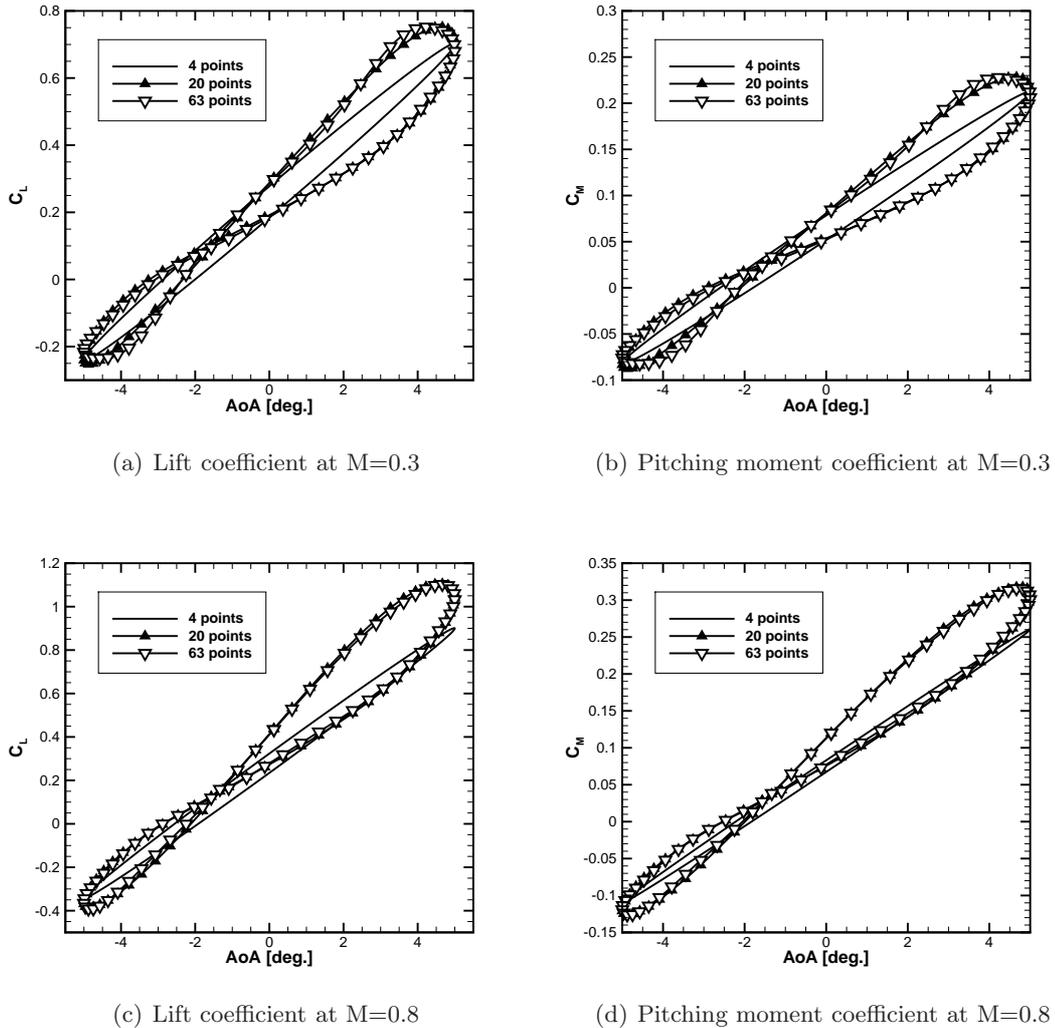


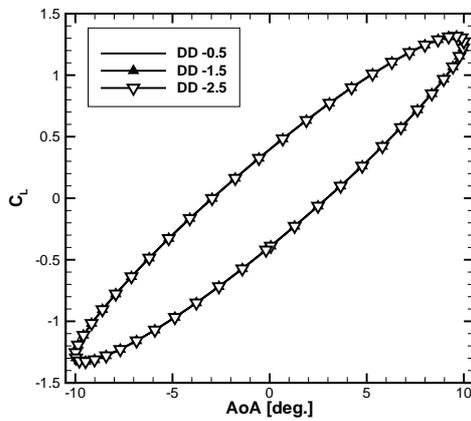
Figure 5.7: LANN table resolution comparison

For the LANN wing case, there is a cross-over point in the lift coefficient which is only captured by the two finest resolutions. However, as with the aerofoil test case, the coarsest resolution of just 4 points is unlikely to be used in practice, but is included here for comparison. The two finest resolutions show good agreement through most of the manoeuvre, with only small discrepancies at the extremes. Again, the tabular resolution for this case does not appear to present a problem in capturing the variations within the parameter space.

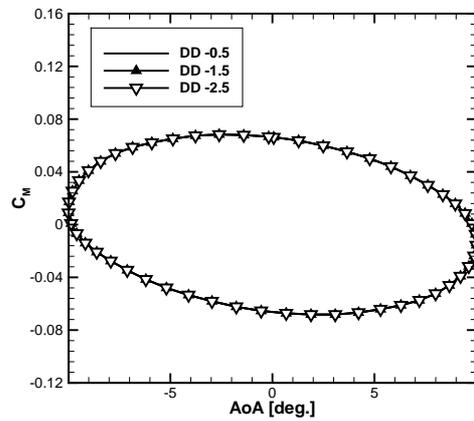
Dynamics

The effect of the unsteady model used on the aerofoil response can be significant. It has been shown in a number of papers [17, 18, 8] that the use of dynamic derivatives can be sensitive to the conditions for which they are calculated. This variation must be considered when running manoeuvres where the positions and rates of the aerofoil are outside of those for which the derivatives have been calculated.

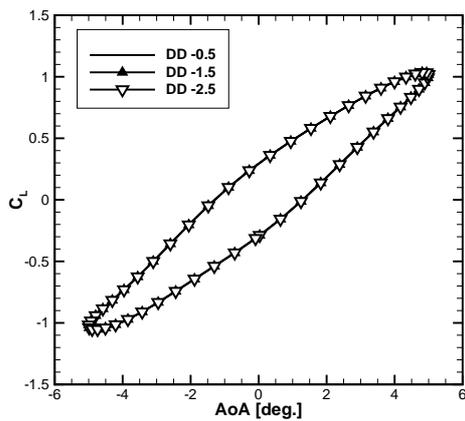
Taking the obstacle manoeuvre again, three different values have been used for the pitch damping derivative for both the Mach 0.3 and 0.8 cases. The tabular replays have been run with each value to determine the loads and moments throughout the manoeuvre. The variation in lift and pitching moment is shown in Fig. 5.8, where the tables used are from the finest resolution in the previous section.



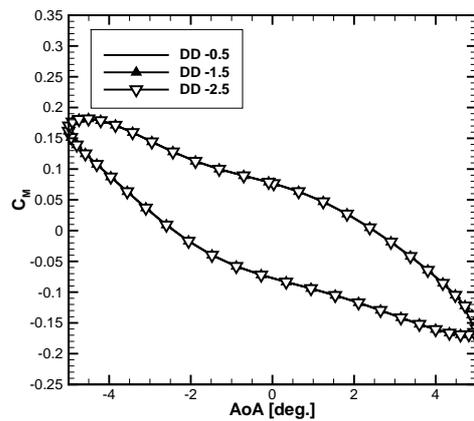
(a) Lift coefficient at M=0.3



(b) Pitching moment coefficient at M=0.3



(c) Lift coefficient at M=0.8



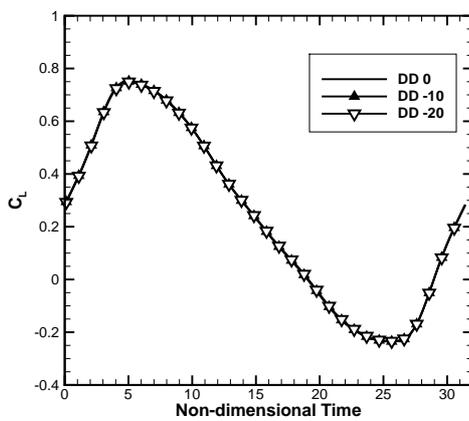
(d) Pitching moment coefficient at M=0.8

Figure 5.8: Aerofoil: Effect of dynamic derivative value on replay

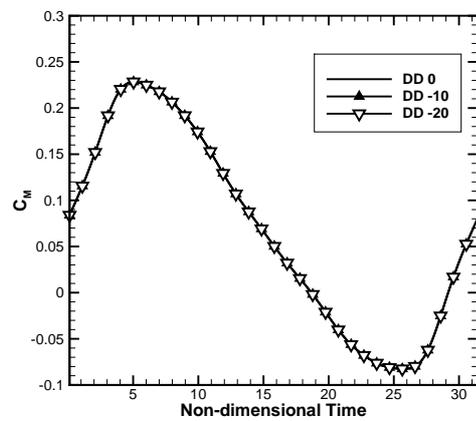
In both cases there is negligible difference for each of the derivative values used.

This is expected due to the small variation in the value used and the low rate for the manoeuvre. This may, however, become significant for certain manoeuvres where there are high rates, large rate changes, or large variation in the state of the aircraft.

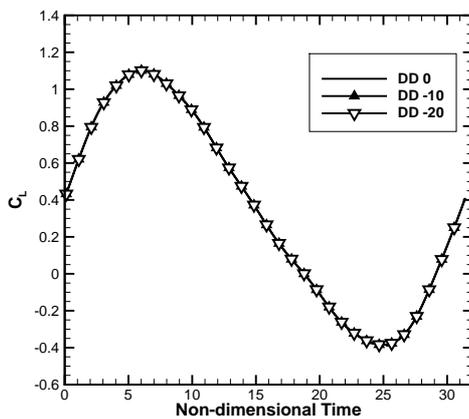
Again, the same comparison can be made for the LANN wing test case. For this assessment, both the $C_{L\dot{\alpha}}$ and $C_{L\dot{\beta}}$ and considered to be equal, as well as $C_{M\dot{\alpha}}$ and $C_{M\dot{\beta}}$. This is not likely to be the case, but the purpose is to see any variation with each of these values being modified. The comparison for two Mach numbers is shown in Figs. 5.9, with the finest resolution tables being used.



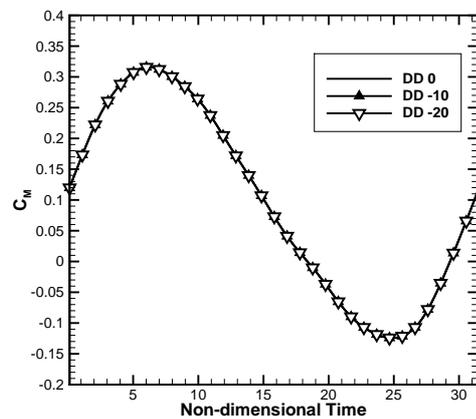
(a) Lift coefficient at M=0.3



(b) Pitching moment coefficient at M=0.3



(c) Lift coefficient at M=0.8



(d) Pitching moment coefficient at M=0.8

Figure 5.9: LANN Wing: Effect of dynamic derivative value on replay

It can be seen, as with the aerofoil assessment, that there is no discernable difference between the replays for the different dynamic derivative values. This is to be expected due to the small variations in the values. This is however representative of actual flight for a civil aircraft.

History

A final assumption is that of the quasi-steady model (i.e. no history effects), which is used to account for unsteadiness in the tabular replays. This leads to flow history being neglected, and as such, could impact the solution for certain manoeuvres. The flow adjusts rapidly with the motion of the body, however in some cases, the lag between the two can become significant. This history effect is present with highly nonlinear flow, such as through shocks and stall, particularly for dynamic stall cases. This will account for a very small part of the testing for commercial aircraft, and may be outside of the flight envelope, but should still be considered. In order to assess the impact of quasi-steady modelling, the unsteady CFD solver has been modified to remove history effects. This is done by modifying the dual-time residual term. The CFD residual:

$$\frac{\partial \mathbf{w}}{\partial t} = R(\mathbf{w}^{t+1}) + \frac{3\mathbf{w}^{t+1} - 4\mathbf{w}^t + \mathbf{w}^{t-1}}{2\Delta t}, \quad (5.1)$$

is modified by removing the dual-time terms to form:

$$\frac{\partial \mathbf{w}}{\partial t} = R(\mathbf{w}^{t+1}). \quad (5.2)$$

This quasi-steady residual takes account of the instantaneous rates applied as point velocities in the computational domain, but has nothing to account for history effects. This modification allows direct comparison between the unsteady CFD replay and the quasi-steady CFD replay. The differences between the two can only be attributed to neglecting history effects. For the single-body aerofoil case, the two CFD methods have been applied to the dynamic stall manoeuvre. A comparison against the fully-unsteady CFD and tabular replay is shown in Fig. 5.10.

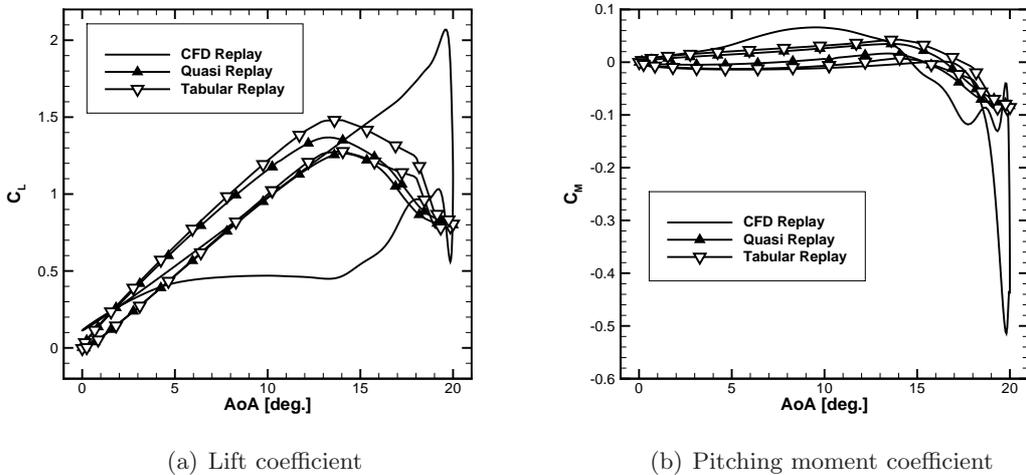
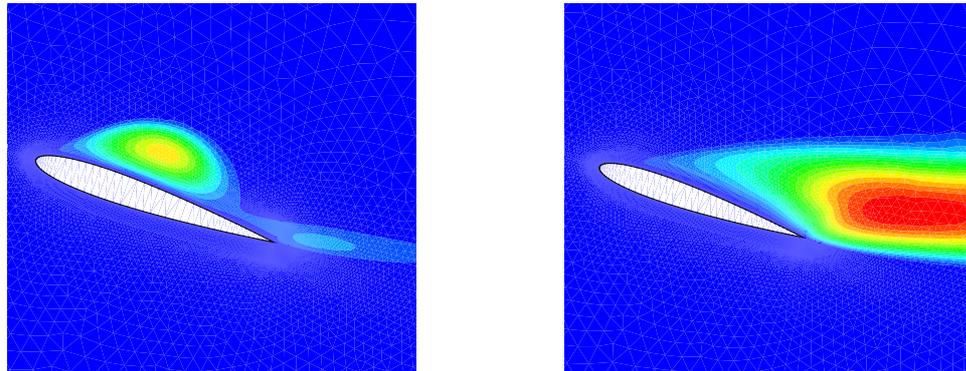


Figure 5.10: Oscillatory manoeuvre replays with quasi-steady

In both the lift and pitching moment coefficients there is significant discrepancy between the unsteady CFD and quasi-steady solutions. Dynamic stall has both strong shocks and separation, causing significant lag between the flow and motion. The discrepancy is solely caused by neglecting history effects. This insight is useful to determine the influence of this assumption on the performance of the tabular model. The best that can be achieved by the tabular model is the quasi-steady CFD replay. There is reasonable agreement between these in the above plots. There are still discrepancies at the higher angles of incidence, which is a result of the static calculations not being in line with the mean solution from the CFD replay. It is clear from this case that the effect of neglecting history is important and is certainly a shortcoming of the tabular model.

As an extension to the above plots, it is possible to view how the effect of history impacts the flow solution at points where there are large differences between the model and the unsteady CFD. Figs. 5.11 and 5.12 show the turbulent eddy viscosity for the dynamic stall case at two locations in the oscillation, which has been chosen due to the sensitivity of this term to the flow conditions.



(a) Full unsteady

(b) Quasi-steady

Figure 5.11: Turbulent eddy viscosity at 19.24° on the upstroke

There are significant differences in the simulated turbulent eddy viscosity, at both points in the manoeuvre, between the unsteady and quasi-steady simulations. In the high incidence case, the unsteady CFD predicts a recirculation bubble sitting on the upper surface of the aerofoil. In the quasi-steady simulation, this has already moved downstream. This is indicative of the substantial difference in lift coefficient at this point. At the lower incidence, the unsteady case shows a recirculation bubble that has recently detached and moved downstream. The quasi-steady however, shows very little turbulent eddy viscosity in comparison. Again, this is reflected in the difference in lift coefficient at this point. From these plots, it is clear that the introduction of viscous

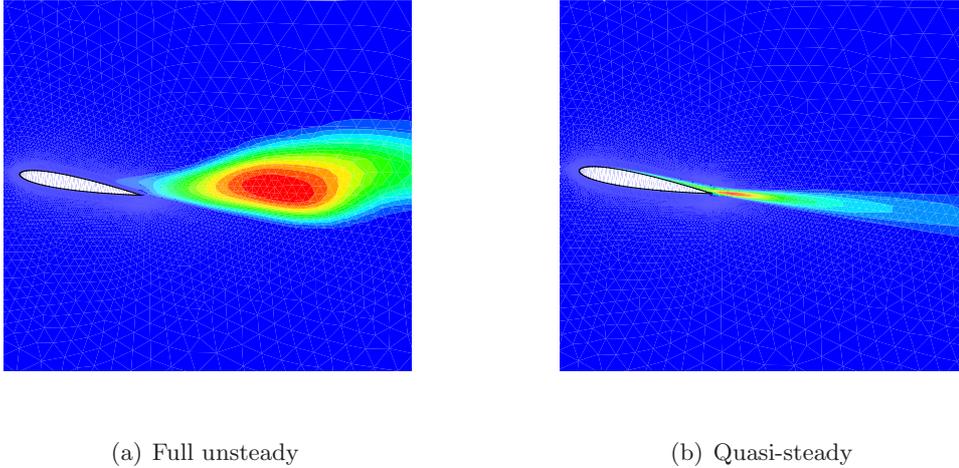


Figure 5.12: Turbulent eddy viscosity at 10.0° on the downstroke

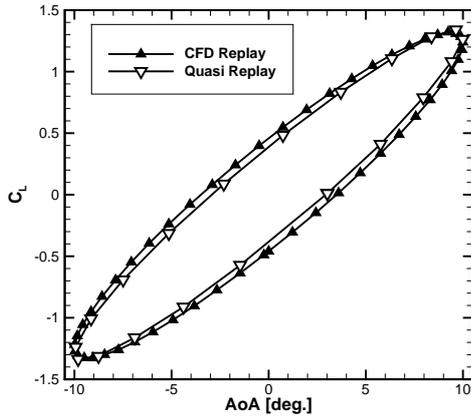
effects, causes even greater problems for the quasi-steady tabular model to accurately predict the unsteady solution.

In order to assess the effect of neglecting history for the aerofoil with the trailing edge flap, the obstacle manoeuvre is again used. An unsteady CFD replay and quasi-steady CFD replay are run. Cross-plotting the solutions enables the influence of history effects to be viewed for the given manoeuvre. A comparison is shown in Fig. 5.13.

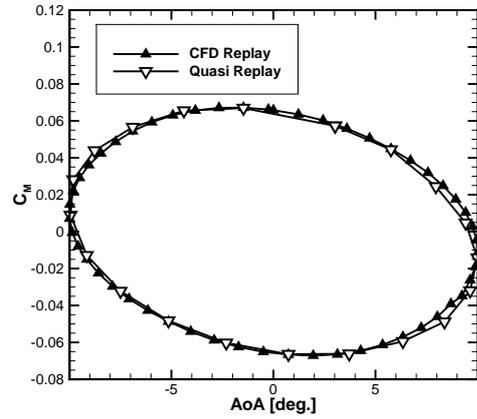
For the obstacle at Mach 0.3, there is little difference between the replays. This is expected due to the amount of unsteadiness for these conditions being low, and as such allows the flow history to be neglected. For the manoeuvre at Mach 0.8, the amount of unsteadiness is significant. This means that neglecting flow history has a significant impact on the solution. The largest differences are seen when the rates are the highest (i.e. when the flow will be changing most due to rapid displacement of the body). With the tabular model being quasi-steady, the best it can approximate is that of the quasi-steady CFD solution. In this instance, given that the quasi-steady replay cannot match that of the unsteady replay, the tabular model will have significant discrepancies, as was seen earlier. This comparison shows that the assumption of neglecting history effects can be valid, although there are certain conditions under which it is no longer the case.

5.3 Free Response

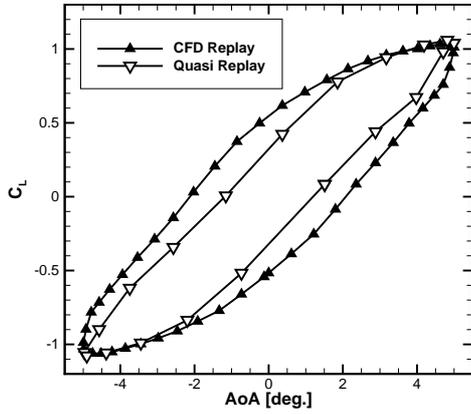
The second approach to the assessment is the use of free response simulations given a set of control inputs. The trajectories followed by the aerofoil are then compared. This method is different to the usual comparison of the loads and moments, and provides a different perspective on how the aerofoil responds, with a focus on flight control system



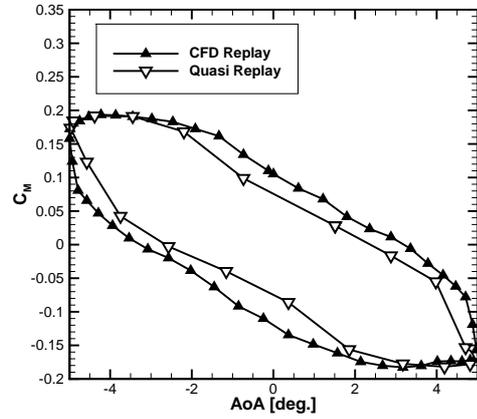
(a) Lift coefficient at $M=0.3$



(b) Pitching moment at $M=0.3$



(c) Lift coefficient at $M=0.8$



(d) Pitching moment coefficient at $M=0.8$

Figure 5.13: Effect of history on the replay

design. The test case used is that of the aerofoil with trailing edge flap. The aerofoil is free to pitch with flap inputs corresponding to those used for the obstacle case in the forced motion assessment. The value of I_y in each case is 0.5 kg.m^2 .

5.3.1 Simulation

With the addition of a control surface to the aerofoil and the use of free-response simulation, as part of this it is necessary to establish the path of the body for a given control surface deflection. In order to do this, it is necessary to derive the equations of motion. To simplify the derivation, the aerofoil and flap are considered as a single fixed body at each step. This has the benefit of reducing the degrees of freedom of the system. For the aerofoil, there are only two degrees of freedom (pitch and plunge) as shown in Fig. 5.14

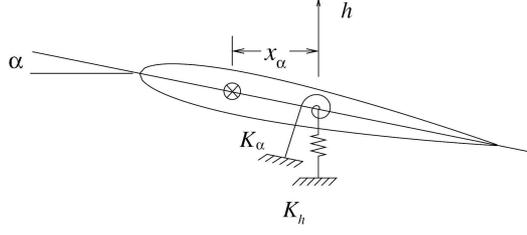


Figure 5.14: Two degree of freedom aerofoil [2]

For simplicity, the offset x_α is considered to be zero (i.e. the centre of gravity is coincident with the flexural axis). The equations of motion can be written in the following form:

$$M\ddot{x} + c\dot{x} + kx = F, \quad (5.3)$$

where x represents the displacements, c represents the damping and k is the stiffness. Each of these can be written using aerodynamic coefficients rather than the traditional structural dynamics counterpart. The damping of the system is described by the aerodynamic damping, in this case the dynamic derivatives. The stiffness terms are the linear variation of the loads with respect to the displacement, in other words the static aerodynamic coefficients. Using the aerodynamic terms, the equations of motion can then be written as

$$\begin{bmatrix} m & 0 \\ 0 & I_y \end{bmatrix} \begin{bmatrix} \ddot{h} \\ \ddot{\alpha} \end{bmatrix} + \begin{bmatrix} L_{\dot{h}} & L_{\dot{\alpha}} \\ M_{\dot{h}} & M_{\dot{\alpha}} \end{bmatrix} \begin{bmatrix} \dot{h} \\ \dot{\alpha} \end{bmatrix} + \begin{bmatrix} L_h & L_\alpha \\ M_h & M_\alpha \end{bmatrix} \begin{bmatrix} h \\ \alpha \end{bmatrix} = \begin{bmatrix} L \\ M \end{bmatrix} \quad (5.4)$$

where M here is the pitching moment and not the mass matrix as in Eq. (5.3). Depending on the simulation method, CFD or tabular, the equations of motion are solved as follows. For the unsteady CFD simulation, the stiffness and aerodynamic damping terms are included in the force and moment coefficients at each step. This allows Eq. (5.4) to be reduced to the following

$$\begin{bmatrix} m & 0 \\ 0 & I_y \end{bmatrix} \begin{bmatrix} \ddot{h} \\ \ddot{\alpha} \end{bmatrix} = \begin{bmatrix} L_{CFD} \\ M_{CFD} \end{bmatrix} \quad (5.5)$$

This is then a second order ODE which is solved to obtain the accelerations in pitch and plunge. When using the tabular model, the method is similar. As only the static values are known, the dynamic terms have to be added. This leads to solution of the following equation

$$\begin{bmatrix} m & 0 \\ 0 & I_y \end{bmatrix} \begin{bmatrix} \ddot{h} \\ \ddot{\alpha} \end{bmatrix} = \begin{bmatrix} L_{Tab} \\ M_{Tab} \end{bmatrix} + \begin{bmatrix} L_{\dot{h}} & L_{\dot{\alpha}} \\ M_{\dot{h}} & M_{\dot{\alpha}} \end{bmatrix} \begin{bmatrix} \dot{h} \\ \dot{\alpha} \end{bmatrix} \quad (5.6)$$

The right hand side values are known at the current time step, t . These are then used

to form a single right hand side and allow solution in the same manner as in eq. (5.5). Solution is carried out at each step in the manoeuvre until the entire trajectory can be traced.

5.3.2 Control Inputs

During the free-response simulations, it is necessary to apply control inputs. These have been predefined as control surface deflections at each point in time. The equations of motion are then solved at each step to determine the body location. The control inputs have been taken from the obstacle manoeuvre carried out in Section 4.5. Plots of the control surface deflections for the subsonic and transonic Mach number cases are shown in Figs. 5.15(a) and 5.15(b).

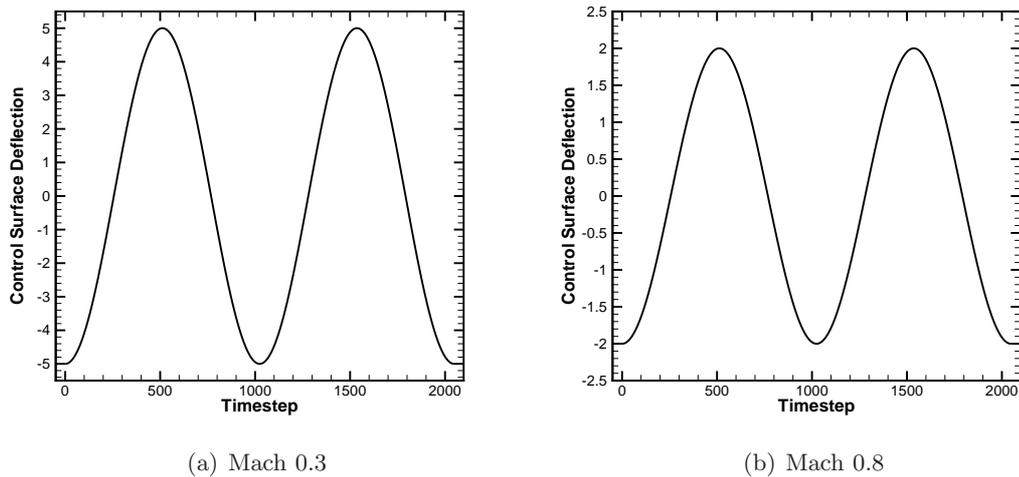
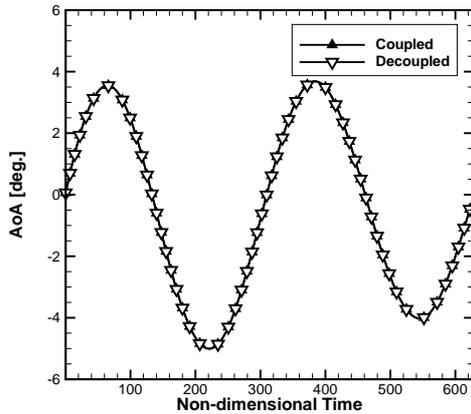


Figure 5.15: Free-response control inputs

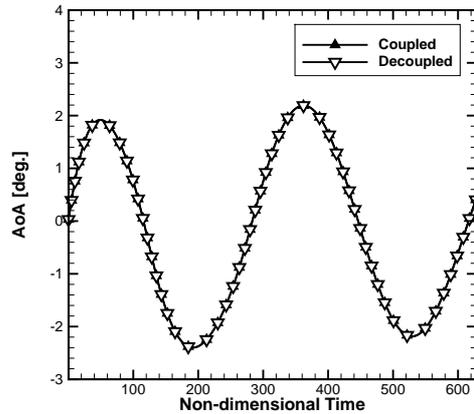
Coupling

The coupling has been assessed again using a coupled table $[M, \alpha, \delta_{ele}]$ which is then split to form two decoupled tables in $[M, \alpha]$ and $[M, \delta_{ele}]$. The incidence is plotted against time for the two methods, with the responses shown in Fig. 5.16 for the subsonic and transonic Mach numbers.

As with the forced motion assessment for the aerofoil and flap case, there is little to no difference between the coupled and decoupled solutions. This is true for both the subsonic and transonic cases. It is however possible that for more extreme manoeuvres, differences begin to appear. The conditions for this to happen are likely to lie outside of the flight envelope of a civil airliner.



(a) Mach 0.5 to 0.3

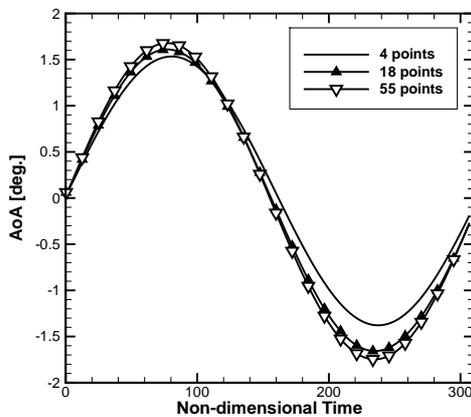


(b) Mach 0.8 to 0.5

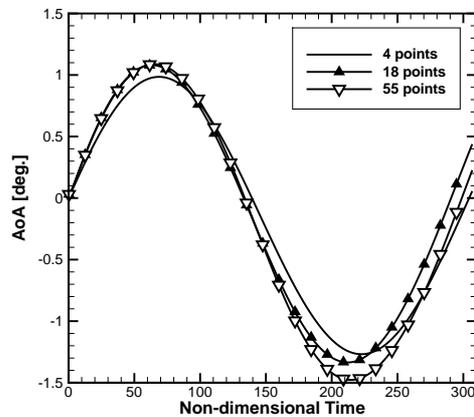
Figure 5.16: Tabular replays with and without coupling

Resolution

For the resolution assessment, the same resolutions as for the forced motion case, 55 points, 18 points and 4 points per Mach number, are taken. The resulting trajectories are then plotted with time. The responses are shown in Fig. 5.17.



(a) M=0.3



(b) M=0.8

Figure 5.17: Table resolution comparison

The coarsest resolution of 4 points shows results consistent with those from the forced motion assessment. For the Mach 0.3 case, there is good agreement for the 18 and 55 point resolutions, which again is consistent with the forced motion results. However, as the Mach number is increased to 0.8, differences begin to appear. There is initially little difference between the 18 and 55 point resolutions. This small error is however gradually compounded as the manoeuvre continues. At the end of one cycle

of the flap, there are noticeable differences between the two trajectories. This was not something that was seen in the forced motion assessment due to the prescribed motion. In this instance, the free-response simulation has shown up a problem which needs to be considered when running manoeuvres within a similar range to that presented here. Although the 18 and 55 point resolutions remain fairly close, it would be recommended in this instance to increase the resolution further. This may not however improve the solution accuracy if the discrepancy is due to an unsteady effect not captured as part of the dynamic derivative model.

Dynamics

The assessment of the dynamic derivative on the replay is carried out in the same way as that of the forced motion. Three values are chosen for the pitch damping derivative. The trajectories for each at subsonic and transonic Mach numbers are shown in Fig. 5.18.

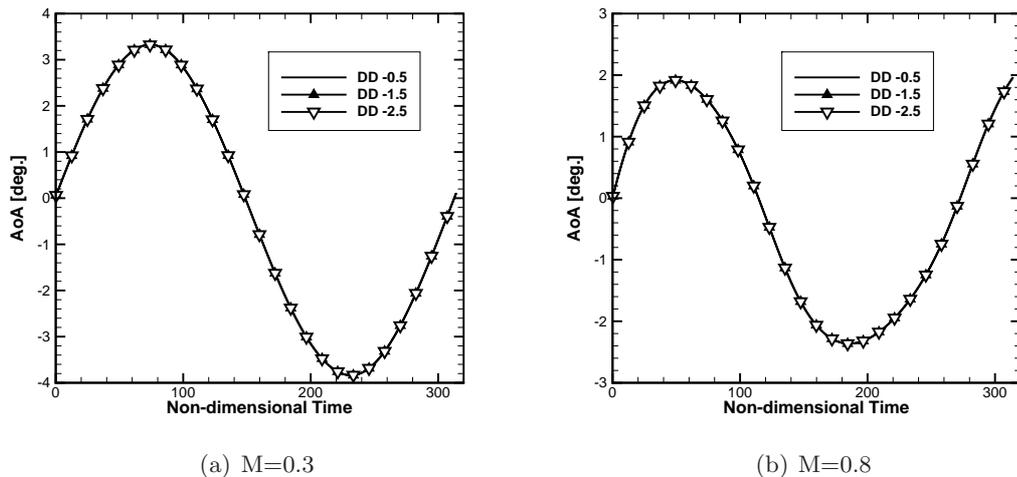


Figure 5.18: Effect of pitch damping derivative value on replay

Again, as with the forced motion assessment, there is little to no difference between the replays with the different values of the pitch damping derivative. It is possible that for this case, the small variation of the derivative value is not significant. There will certainly be extreme cases where this will prove to be an issue. Although again, this case is likely to be outside the flight envelope of interest. It is also possible that the free-response of a wing with control surfaces would see differences. The coupling of four parameters as opposed to three could lead to discrepancies.

Having assessed the tabular aerodynamic model on a fundamental level for a number of cases, and through a range of flow regimes, the assumptions are largely unimportant. It is however necessary to consider history effects when simulating manoeuvres at the

extremes of the flight envelope where nonlinear effects are present. Further to this, for three dimensional cases, decoupling of Mach, incidence and sideslip should not be done as this is not a suitable assumption to make. With this model being used on a daily basis for flight dynamics loads prediction, and flight control systems on board aircraft, there are instances where other models, particularly nonlinear models, should be considered in their place, in spite of the extra cost.

5.4 Dynamic Derivative Error Quantification

When a manoeuvre simulation is carried out, a single value for the dynamic derivative is usually taken, which is assumed to be independent of the flight parameters. As was described above, this is however not the case. If manoeuvres are simulated using a single value, errors are introduced. A method is proposed to understand the effect of this error on the simulated loads and moments. Aerodynamic tables of the dynamic stability derivatives are used.

As in previous studies, it is possible to carry out calculations of the dynamic derivative values with sweeps in the flight parameters. Some calculations have been run for the NACA 0012 aerofoil using the Harmonic Balance solver with three harmonics. Results are shown in Figs. 5.19(a) and 5.19(b) for variation of the lift damping derivative with reduced frequency and Mach number respectively.

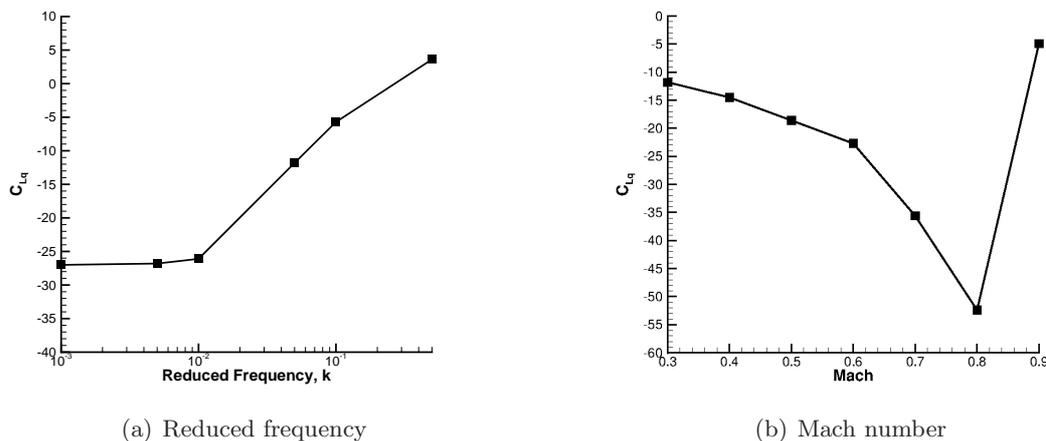


Figure 5.19: Lift damping derivative variation with flight parameters

As per the literature, there is significant variation with both flight parameters. The variation with reduced frequency is particularly important for manoeuvre simulation, as this is likely to vary considerably from one manoeuvre to another. The Mach number variation, although more extreme, is not as important due to the manoeuvre speed likely to remain fairly steady throughout. It does however become a problem if the

designer uses a dynamic derivative value calculated for Mach 0.3, which is used to assess a manoeuvre at Mach 0.8.

5.4.1 Error Estimation

As part of the flight dynamics assessment, aerodynamic tables of stability and control derivatives are calculated. These tables are typically less densely populated than the static aerodynamic tables, however they are particularly useful to quantify the error from the single derivative value approach in a rapid way.

The tables give information on what the derivative value should be for a particular flight parameter combination, or for a particular point within a manoeuvre. This information can be used to determine an error in the loads and moments compared to the single derivative value that is chosen for the simulation. In order to determine the error, the difference between the derivatives taken from the tables and the derivative chosen for the simulation is calculated and subsequently multiplied by the corresponding instantaneous rate. This error is then represented in the later plots as error bars. This procedure can be broken down as below.

-
1. Aerodynamic tables of stability and control derivatives are calculated
 2. Derivative values are chosen for the simulation and the manoeuvre is defined
 3. Manoeuvre data is used with the stability and control derivative tables to compute the correct derivative values for each step
 4. A delta is calculated by subtracting the derivative from step 3 from the single defined derivative of step 2.
 5. The delta is multiplied by the relevant rates to obtain the error in the corresponding loads and moments.
-

5.4.2 Application

A demonstration of the proposed method is now shown. The test case is a NACA 0012 aerofoil through a range of manoeuvres. The point cloud used is shown in Fig. 4.2. The CFD solver used is the PML code.

The aerodynamic table of dynamic derivatives has been calculated for a range of flight parameters as described in Table 5.8. All entries in the tables have been obtained with the PML implementation of Harmonic Balance with three retained harmonics.

As with the static tabular aerodynamic model, it is possible to use the dynamic derivative table with a coupled or decoupled approach, i.e. a coupling between different flight parameters when calculating the derivatives. In the cases here, a coupled

Flight Parameter	Range	Interval	Data Points
Mach Number	0.3 to 0.8	0.1	6
Mean Incidence	-10° to +10°	2°	11
Amplitude of Oscillation	1° to 5°	1°	5
Reduced Frequency	1×10^{-4} to 5×10^{-1}	log scale	8

Table 5.8: NACA 0012 dynamic derivative table parameters

approach is taken where the the table is $[M, k, \alpha_0, \alpha_A]$. This approach requires many more calculations, but it ensures that errors are from the derivative modelling approach and not from the assessment method.

A number of manoeuvres have been simulated, with the error from the dynamic derivative model expressed with error bars. In each case, an arbitrary value of -1/rad. has been taken for each of the derivatives. The first manoeuvre is an oscillatory motion at Mach 0.3, with a mean incidence of 0.0° , an amplitude of 5.0° and a reduced frequency of 0.001. The lift and moment coefficient response with time step is shown in Figs. 5.20(a) and 5.20(b).

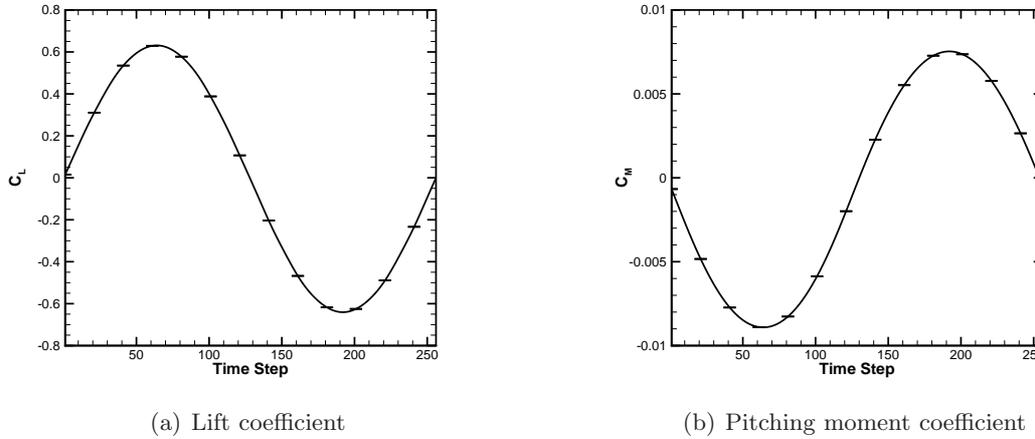
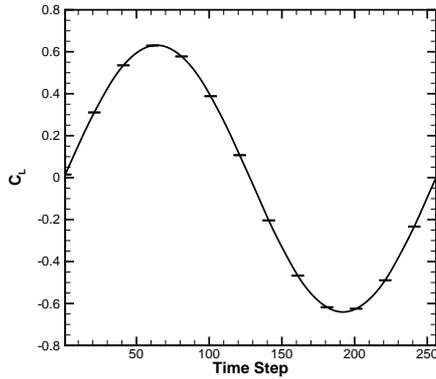


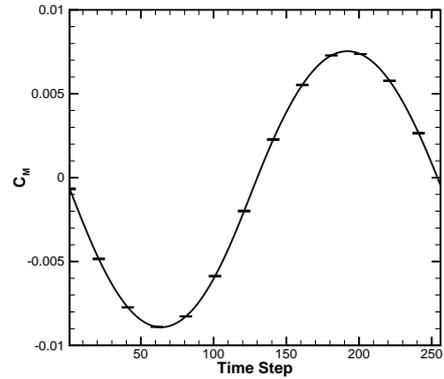
Figure 5.20: $M = 0.3, \alpha_0 = 0.0^\circ, \alpha_A = 5.0^\circ, k = 0.001$

For this first manoeuvre, the behaviour of the aerodynamics remains linear. This leads to little variation with the flight parameters of the dynamic derivative, and as such the error bars are near zero, i.e. there is negligible error introduced using the assumption of a single derivative. In order to increase the complexity of the problem, the second manoeuvre increases the reduced frequency from 0.001 to 0.1. This should begin to introduce some history effects through lag in the flow. Results for this are shown in Figs. 5.21(a) and 5.21(b).

In this second manoeuvre, the error bars are starting to open up, although very little. This suggests that the assumption of a single derivative is more important for



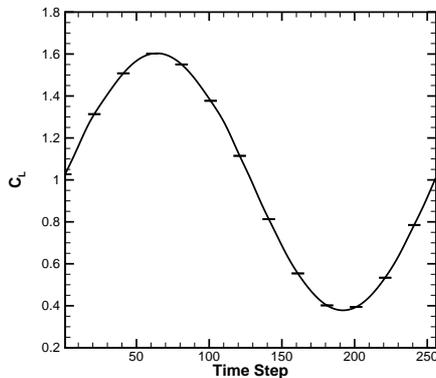
(a) Lift coefficient



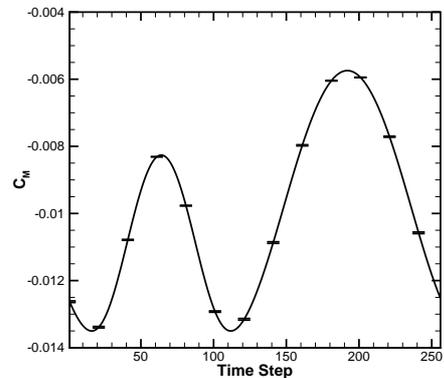
(b) Pitching moment coefficient

Figure 5.21: $M = 0.3$, $\alpha_0 = 0.0^\circ$, $\alpha_A = 5.0^\circ$, $k = 0.1$

this test case. The pitching moment coefficient is more sensitive to changes in flow conditions, and as such, the error bars are wider in this plot than for the lift coefficient. A second way to introduce nonlinearity is through using high incidences. A third manoeuvre has been simulated where the mean incidence has been increased from 0.0° to 8.0° . The reduced frequency has been reduced back to 0.001. Results for this are shown in Figs. 5.22(a) and 5.22(b).



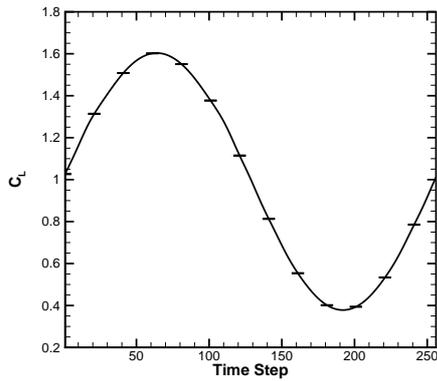
(a) Lift coefficient



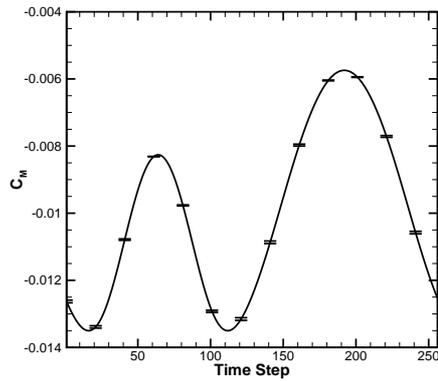
(b) Pitching moment coefficient

Figure 5.22: $M = 0.3$, $\alpha_0 = 8.0^\circ$, $\alpha_A = 5.0^\circ$, $k = 0.001$

The effect of increasing the mean incidence has had a similar effect to increasing the reduced frequency. The error bars are widening, although again very little. As with the previous manoeuvre, the pitching moment is more sensitive to errors than the lift coefficient. A final manoeuvre for the aerofoil case is to combine the higher incidence with the increased reduced frequency. This combined change should lead to greater nonlinear behaviour, and larger errors. Results are shown in Figs. 5.23(a) and 5.23(b).



(a) Lift coefficient



(b) Pitching moment coefficient

Figure 5.23: $M = 0.3$, $\alpha_0 = 8.0^\circ$, $\alpha_A = 5.0^\circ$, $k = 0.1$

In this final manoeuvre, it is clear that the error is beginning to build. The lift coefficient is again largely error free, however, the pitching moment has noticeable gaps in the error bars. It is clear from this plot that as the edge of the flight envelope is approached, the nonlinear behaviour of the flow causes the assumption of a single derivative to break down. The errors introduced become important, and for certain applications of this model, these errors could certainly start to accumulate, such as in free-response simulations.

From this assessment, it is seen that, even though the manoeuvre conditions are pushed toward the upper end of the civil domain flight envelope, the error introduced by using a single dynamic derivative is not significant. This could be taken further for more complex manoeuvres and test cases and may form part of future work.

Chapter 6

Conclusions

Computational models for flight dynamics analysis have been the focus of this work. The tabular aerodynamic model has been studied in this thesis and assessed for adequacy through a number of manoeuvres. The focus of the assessment has been within the flight envelope of a an aircraft in the civil domain. The model has been assessed through a number of key flow regimes using several test cases. The most interesting, from the point of view of this study, are at the extremes of the flight envelope where the aerodynamic loads vary nonlinearly with the flight parameters. As part of the assessment, the manoeuvres were simulated using the tabular models, with the solution compared to the same replay from a time-accurate CFD solver. The CFD solution was used as the baseline comparator. It was shown that for the low complexity manoeuvres, i.e. low Mach, low incidence, the tabular model was able to accurately simulate the loads and moments through all manoeuvres, with respect to the CFD solution. When the Mach number and/or incidence were increased, shocks and separation began to cause problems with the accuracy of the tabular replays. During these manoeuvres, the tabular model was largely unable to accurately match the solution obtained from the time-accurate CFD solver. This initial assessment provided an insight as to when the tabular model was no longer sufficient.

The assessment was extended to explore the model on a more fundamental level. The assumptions in forming the tabular model were established, and a number of test cases and manoeuvres were chosen to understand the effect of each on the accuracy of the simulation. The four assumptions included: removing coupling effects between some flight parameters; having a resolution in the tables that is sufficient to capture changes in the parameter space; using a dynamics model that captures all effects of interest; and finally neglecting history effects. Each assumption was taken individually and assessed using both forced and free-response manoeuvres. It was first shown that the effect of decoupling certain flight parameters was negligible, except between incidence and sideslip for the wing test case. Secondly, the tabular resolution was considered as not being significant, provided a reasonable number of data points in the

parameter space, as determined by experience. Thirdly, the use of a single dynamic derivative value for the dynamics modelling was considered as not significant for the test cases studied. It was noted that, should the manoeuvres increase in rate or incidence, i.e. outside of the assumptions used to develop the dynamic derivative model, this assumption may become important. Finally, the most significant assumption was that of neglecting history effects. The tabular model, inclusive of the dynamic derivative modelling, is a quasi-steady model i.e. does not include terms that account for history dependent flow effects. In neglecting history effects, the performance of the tabular model for simulating the loads and moments throughout manoeuvres in this work, was significantly affected. This was particularly noticeable during high Mach number manoeuvres with the presence of strong moving shock waves. The largest differences seen in this work were for a dynamic stall case. Throughout this manoeuvre there was substantial separation and shock waves which made history effects dominant in the aerodynamics. From this assessment, it can be concluded that the tabular model is only suitable for manoeuvre simulation where the aerodynamics remain mostly linear with respect to the flight parameters. This renders them unsuitable for simulations where computational simulation is most useful, at the edges of the flight envelope.

The dynamic derivative model has also been studied. This model considers the forces to consist of a static mean component plus an unsteady component related to the instantaneous motion rates. This linearisation introduces its own problems. Further to this, as mentioned above, the derivative model is typically used in practice with a single value. However, the derivative varies with the parameters at which it is calculated. A method has been proposed to determine the effect of this assumption for an arbitrary manoeuvre. The approach allows direct plotting of error bars with the simulated forces in order to view the discrepancies. For the cases considered in this work, there was little error seen due to the low rates and incidence.

The dynamic derivatives can be calculated using forced oscillatory motions. Given this, frequency domain methods were used in this work. Frequency domain methods have the benefit of being able to directly compute the periodic state of a periodic motion. The two methods used were the Linear Frequency Domain and the Harmonic Balance. The Harmonic Balance technique has a formulation that allows capturing of nonlinearities. It was implemented, as part of this work, into two different CFD solvers, each taking different solution approaches. It was shown that the implementations were able to speed up solution of the fully unsteady CFD calculation by at least a factor of three, whilst maintaining sufficient accuracy with respect to the CFD solution. The Linear Frequency Domain was the focus of more in depth work in this thesis. Firstly, the semi-implicit implementation within the DLR TAU code was extended to make use of a fully implicit GCR Krylov solution method. This new solution approach led to a speed up in simulation time of around a factor of 10 across all test cases. A comparison of the GCR solver with the previously available methods was done. It was shown that

on top of the speed up, in comparison to the other implicit solver, PETSc, there was a significant benefit in terms of the memory requirement.

As part of the implicit solution method, a preconditioner was used to improve the performance of the solver. The system matrix is for the Jacobian of second order discretisation. The preconditioner is usually based on the first order accurate Jacobian due to stability issues. In this thesis, an alternative preconditioning method has been proposed based on both the first and second order Jacobian matrices. This approach requires forming a new matrix with weighting α applied to the second order Jacobian terms. This weighted matrix was then used to form the ILU preconditioner. It was shown that the influence of the parameter α was significant and just a small change can make the difference between a converged solution being achieved or not. It was also shown that there was an optimum value at which the best convergence rate was achieved. Interestingly, this value was the same across all test cases in 2D and 3D, and for all flight conditions studied. This suggests applicability to the general case. The effect of the weighting was also studied in terms of how well the preconditioner approximated the exact inverse by viewing variations in the flow field. It was seen that adding just a small amount of the first order Jacobian improves the approximation of the preconditioner, which is seen as a smoothing in the flow field values. The presence of the second order spatial discretisation Jacobian terms improves the accuracy of the preconditioner matrix, thus leading to best convergence.

6.1 Future Work

Although a fundamental assessment has been carried out, there are other areas that can be explored. Control systems design is one such area that makes use of tabular aerodynamic models, particularly for onboard control systems. A method to determine the feasibility would be using the models for inverse simulation. For example, prescribing the loads and moments for each step in the manoeuvre, the tables would then be used to determine the flap deflection and body incidence required. A comparison can then be made between the tabular and CFD replays. Following on from the aerofoil case in this thesis with a control surface, a large scale simulation could be carried out using a wing or aircraft model with control surfaces.

Following on from the above, further work could look at an extension of the assumption assessment for higher dimensional tabular models. Again, these could include wings and then full aircraft with control surfaces. The same methodology of using forced motions would allow for individual assessment of the assumptions, with the possibility of the increased dimensionality leading to noticeable differences in the tabular and CFD replays. It may prove that the assumptions become significant even for low speed and low incidence cases.

Finally, a mathematical analysis of the preconditioner approach proposed in this work should be carried out. This would provide a much better understanding as to why the convergence is so greatly improved by making a small change to the values in the matrix.

Bibliography

- [1] Cebeci, T., Shao, J., Kafyeke, F., and Laurendeau, E., *Computational fluid dynamics for engineers : from panel to Navier-Stokes methods with computer programs*, Horizons Publishing, Long beach, CA, 2005.
- [2] Beran, P. S., Pettit, C. L., and Millman, D. R., “Uncertainty quantification of limit-cycle oscillations,” *Journal of Computational Physics*, Vol. 217, No. 1, September 2006, pp. 217–247.
- [3] Ghoreyshi, M., Badcock, K., and Woodgate, M., “Accelerating the Numerical Generation of Aerodynamic Models for Flight Simulation,” *Journal of Aircraft*, Vol. 46, No. 3, May-June 2009, pp. 972–980.
- [4] Da Ronch, A., Ghoreyshi, M., and Badcock, K. J., “On the Generation of Flight Dynamics Aerodynamic Tables by Computational Fluid Dynamics,” *Progress in Aerospace Sciences*, Vol. 47, 2011, pp. 597–620.
- [5] Mackman, T. J., Allen, C. B., Ghoreyshi, M., and Badcock, K. J., “Comparison of Adaptive Sampling Methods for Generation of Surrogate Aerodynamic Models,” *AIAA Journal*, Vol. 51, No. 4, April 2013, pp. 797–808.
- [6] Ghoreyshi, M., Badcock, K., Ronch, A. D., Marques, S., Swift, A., and Ames, N., “Framework for Establishing Limits of Tabular Aerodynamic Models for Flight Dynamics Analysis,” *Journal of Aircraft*, Vol. 48, No. 1, 2011, pp. 42–55.
- [7] Vallespin, D., Badcock, K., Ronch, A. D., White, M., Perfect, P., and Ghoreyshi, M., “Computational Fluid Dynamics Framework for Aerodynamic Model Assessment,” *Progress in Aerospace Sciences*, Vol. 52, July 2012, pp. 2–18.
- [8] McCracken, A., Akram, U., Da Ronch, A., and Badcock, K. J., “Requirements for Computer Generated Aerodynamic Models for Aircraft Stability and Control Analysis,” *5th Symposium on Integrating CFD and Experiments in Aerodynamics*, Tokyo, Japan, 3-5 October 2012.
- [9] Greenwell, D. I., “A Review of Unsteady Aerodynamic Modelling for Flight Dynamics of Manoeuvrable Aircraft,” *AIAA Atmospheric Flight Mechanics Conference and Exhibit*, Providence, Rhode Island, 16-19 August 2004.
- [10] Kyle, H., Lowenberg, M., and Greenwell, D. I., “Comparative Evaluation of Unsteady Aerodynamic Modelling Approaches,” *AIAA Atmospheric Flight Mechanics Conference and Exhibit*, Providence, Rhode Island, 16–19 August 2004.

- [11] Silva, W. A., “Application of Nonlinear Systems Theory to Transonic Unsteady Aerodynamic Responses,” *Journal of Aircraft*, Vol. 30, No. 5, 1993, pp. 660–668.
- [12] Tobak, M., Chapman, G. T., and Schiff, L. B., “Mathematical Modelling of the Aerodynamic Characteristics in Flight Dynamics,” *NASA TM-85880*, 1984.
- [13] Tobak, M. and Chapman, G. T., “Nonlinear Problems in Flight Dynamics Involving Aerodynamic Bifurcations,” *NASA TM-86706*, 1985.
- [14] Da Ronch, A., McCracken, A., Badcock, K. J., Ghoreyshi, M., and Cummings, R. M., “Modeling of Unsteady Aerodynamic Loads,” *AIAA Atmospheric Flight Mechanics Conference*, Portland, Oregon, 8-11 August 2011.
- [15] Glaz, B., Liu, L., and Friedmann, P. P., “Reduced-Order Nonlinear Unsteady Aerodynamic Modelling Using a Surrogate-Based Recurrence Framework,” *AIAA Journal*, Vol. 48, No. 10, 2010, pp. 2418–2429.
- [16] Bryan, G., *Stability in Aviation*, Macmillan and Co., London, 1911.
- [17] Bratt, J. B. and Wight, K. C., “The Effect of Mean Incidence, Amplitude of Oscillation, Profile and Aspect Ratio on Pitching Moment Derivatives,” Tech. Rep. 2064, National Aeronautical Establishment, June 1945.
- [18] Greenwell, D. I., “Frequency Effects on Dynamic Stability Derivatives Obtained from Small-Amplitude Oscillatory Testing,” *Journal of Aircraft*, Vol. 35, No. 5, 1998, pp. 776–783.
- [19] Hall, K. and Clark, W., “Linearized Euler Predictions of Unsteady Aerodynamic Loads in Cascades,” *AIAA Journal*, Vol. 31, No. 3, 1993, pp. 540–550.
- [20] Hall, K., Clark, W., and Lorence, C., “A Linearized Euler Analysis of Unsteady Transonic Flows in Turbomachinery,” *Journal of Turbomachinery*, Vol. 116, No. 3, 1994, pp. 477–488.
- [21] Pechloff, A. and Laschka, B., “Small Disturbance Navier-Stokes Method: Efficient Tool for Predicting Unsteady Air Loads,” *Journal of Aircraft*, Vol. 43, No. 1, 2006, pp. 17–29.
- [22] Widhalm, M., Dwight, R., Thormann, R., and Hübner, A., “Efficient Computation of Dynamic Stability Data with a Linearized Frequency Domain Solver,” *ECCOMAS CFD 2010*, Lisbon, Portugal, 14-17 June 2010.
- [23] Da Ronch, A., Ghoreyshi, M., Badcock, K., Gortz, S., Widhalm, M., Dwight, R., and Campobasso, M., “Linear Frequency Domain and Harmonic Balance Predictions of Dynamic Derivatives,” *28th AIAA Applied Aerodynamics Conference*, Chicago, Illinois, 28th June - 1st July 2010.
- [24] Hall, K. and Crawley, E., “Calculation of Unsteady Flows in Turbomachinery Using the Linearized Euler Equations,” *AIAA Journal*, Vol. 27, No. 6, June 1989, pp. 777–787.
- [25] Kreiselmaier, E. and Laschka, B., “Small Disturbance Euler Equations: Efficient and Accurate Tool for Unsteady Load Prediction,” *Journal of Aircraft*, Vol. 37, No. 5, 2000, pp. 770–778.

- [26] “AGARD-R-702 Compendium of Unsteady Aerodynamic Measurements,” Tech. rep., Advisory Group for Aerospace Research and Development - NATO, 1982.
- [27] Hall, K., Thomas, J., and Clark, W., “Computation of Unsteady Nonlinear Flows in Cascades Using a Harmonic Balance Technique,” *AIAA Journal*, Vol. 40, No. 5, May 2002, pp. 879–886.
- [28] Woodgate, M. and Badcock, K., “Implicit Harmonic Balance Solver for Transonic Flow with Forced Motions,” *AIAA Journal*, Vol. 47, No. 4, April 2009, pp. 893–901.
- [29] He, L. and Ning, W., “Efficient Approach for Analysis of Unsteady Viscous Flows in Turbomachines,” *AIAA Journal*, Vol. 36, No. 11, November 1998, pp. 2005–2012.
- [30] McMullen, M., *The Application of Non-Linear Frequency Domain Methods to the Euler and Navier-Stokes Methods*, Ph.D. thesis, Stanford University, 2003.
- [31] McMullen, M., Jameson, A., and Alonso, J., “Demonstration of Nonlinear Frequency Domain Methods,” *AIAA Journal*, Vol. 44, No. 7, July 2006, pp. 1428–1435.
- [32] Axelsson, O., “A Survey of Preconditioned Iterative Methods for Linear Systems of Algebraic Equations,” *BIT*, Vol. 25, 1985, pp. 166–187.
- [33] Benzi, M., “Preconditioning Techniques for Large Linear Systems: A Survey,” *Journal of Computational Physics*, Vol. 182, No. 2, 2002, pp. 418–477.
- [34] Saad, Y., *Iterative Methods for Sparse Linear Systems*, Self Published, 2nd ed., 2000.
- [35] Nejat, A. and Ollivier-Gooch, C., “On Preconditioning of Newton-GMRES Algorithm for a Higher-Order Accurate Unstructured Solver,” *14th Annual Conference of the Computational Fluid Dynamics Society of Canada*, Kingston, Canada, July 2006.
- [36] Dwight, R., *Efficiency Improvements of RANS-Based Analysis and Optimization using Implicit and Adjoint Methods on Unstructured Grids*, Ph.D. thesis, School of Mathematics, University of Manchester, 2006.
- [37] Duff, I. and Meurant, G., “The Effect of Ordering on Preconditioned Conjugate Gradients,” *BIT Numerical Mathematics*, Vol. 29, No. 4, 1989, pp. 635–657.
- [38] Chow, E. and Saad, Y., “Experimental Study of ILU Preconditioning for Indefinite Matrices,” *Journal of Computational and Applied Mathematics*, Vol. 86, No. 2, December 1997, pp. 387–414.
- [39] Elman, H., “Relaxed and Stabilized Incomplete Factorizations for Non-Self-Adjoint Linear Systems,” *BIT*, Vol. 29, 1989, pp. 890–915.
- [40] Elman, H., “A Stability Analysis of Incomplete LU Factorisations,” *Mathematics of Computation*, Vol. 47, No. 175, 1986, pp. 191–217.

- [41] Chapman, A., Saad, Y., and Wigton, L., “High Order ILU Preconditioners for CFD Problems,” *International Journal for Numerical Methods in Fluids*, Vol. 33, No. 6, 2000, pp. 767–788.
- [42] Duff, I. and Koster, J., “The Design and Use of Algorithms for Permuting Large Entries to the Diagonal of Sparse Matrices,” *SIAM Journal on Matrix Analysis and Applications*, Vol. 20, No. 4, 1999, pp. 889–901.
- [43] Duff, I. and Koster, J., “On Algorithms for Permuting Large Entries to the Diagonal of a Sparse Matrix,” *SIAM Journal on Matrix Analysis and Applications*, Vol. 22, No. 4, 2001, pp. 973–996.
- [44] Benzi, M., Haws, J., and Tuma, M., “Preconditioning Highly Indefinite and Non-symmetric Matrices,” *SIAM Journal on Scientific Computing*, Vol. 22, No. 4, 2000, pp. 1333–1353.
- [45] Pueyo, A. and Zingg, D., “Progress in Newton-Krylov Methods for Aerodynamic Calculations,” *AIAA 35th Aerospace Sciences Meeting and Exhibit*, No. AIAA-97-0877, Reno, NV, 6-9 January 1997.
- [46] Wong, P. and Zingg, D., “Three-Dimensional Aerodynamic Computations on Unstructured Grids Using a Newton-Krylov Approach,” *17th AIAA Computational Fluid Dynamics Conference*, No. AIAA-05-5231, Toronto, Canada, 6-9 June 2005.
- [47] Spalart, P. and Allmaras, S., “A One-Equation Turbulence Model for Aerodynamic Flows,” *30th Aerospace Sciences Meeting and Exhibit*, Reno, NV, 6-9 January 1992.
- [48] Jameson, A., “Time Dependent Calculations Using Multigrid, with Applications to Unsteady Flows Past Airfoils and Wings,” *AIAA 10th Computational Fluid Dynamics Conference*, Honolulu, 24-26 June 1991.
- [49] Gerhold, T., “Calculation of Complex Three-Dimensional Configurations Employing the DLR TAU-Code,” *35th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, NV, January 1997.
- [50] Schwamborn, D., Gerhold, T., and Heinrich, R., “The DLR TAU-Code: Recent Applications in Research and Industry,” *ECCOMAS CFD 2006*, TU Delft, The Netherlands, 2006.
- [51] Jameson, A., “Solution of the Euler Equations by a Multigrid Method,” *Applied Mathematics and Computation*, Vol. 13, 1983, pp. 327–356.
- [52] Kennett, D., Timme, S., Angulo, J., and Badcock, K., “An Implicit Meshless Method for Application in Computational Fluid Dynamics,” *International Journal for Numerical Method in Fluids*, 2012, Accepted for publication.
- [53] Kennett, D. J., Timme, S., Angulo, J., and Badcock, K. J., “Semi-Meshless Stencil Selection for Anisotropic Point Distributions,” *International Journal of Computational Fluid Dynamics*, Vol. 26, No. 9-10, 2012, pp. 463–487.
- [54] Rampurawala, A., *Aeroelastic Analysis of Aircraft with Control Surfaces using CFD*, Ph.D. thesis, Department of Aerospace Engineering, University of Glasgow, 2006.

- [55] Goura, L., *Time Marching Analysis of Flutter using Computational Fluid Dynamics*, Ph.D. thesis, Department of Aerospace Engineering, University of Glasgow, 2001.
- [56] Quaranta, G., Masarati, P., and Mantegazza, P., “A Conservative Mesh-Free Approach for Fluid-Structure Interface Problems,” *International Conference on Computational Methods for Coupled Problems in Science and Engineering*, Barcelona, 2005.
- [57] Witteveen, J. A. S. and Bijl, H., “Explicit Mesh Deformation Using Inverse Distance Weighting Interpolation,” *19th AIAA Computational Fluid Dynamics*, San Antonio, Texas, 22–25 June 2009.
- [58] Tang, C. and Gee, K., “Generation of Aerodynamic Data Using a Design of Experiment and Data Fusion Approach,” *43rd AIAA Aerospace Sciences Meeting and Exhibit*, No. AIAA-2005-1137, Reno, NV, 10-13 January 2005.
- [59] Williams, J. and Vukelich, S., “The USAF Stability and Control Digital DATCOM,” Tech. Rep. AFFDL-TR-79-3032, McDonnell Douglas Astronautics Company, St Louis, MO, 1979.
- [60] Da Ronch, A., Vallespin, D., Ghoreyshi, M., and Badcock, K. J., “Evaluation of Dynamic Derivatives Using Computational Fluid Dynamics,” *AIAA Journal*, Vol. 50, No. 2, 2012, pp. 470–484.
- [61] Saad, Y. and Schultz, M. H., “GMRes: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems,” *SIAM Journal on Scientific and Statistical Computing*, Vol. 7, No. 3, July 1986, pp. 856–869.
- [62] Balay, S., Brown, J., Buschelman, K., Gropp, W. D., Kaushik, D., Knepley, M. G., McInnes, L. C., Smith, B. F., and Zhang, H., “PETSc Web page,” 2011, <http://www.mcs.anl.gov/petsc>.
- [63] Balay, S., Buschelman, K., Eijkhout, V., Gropp, W. D., Kaushik, D., Knepley, M., McInnes, L. C., Smith, B. F., and Zhang, H., “PETSc Users Manual,” Tech. Rep. ANL-95/11 - Revision 2.3.3, Argonne National Laboratory, 2007.
- [64] Eisenstat, S., Elman, H., and Schultz, M., “Variational Iterative Methods for Nonsymmetric Systems of Linear Equations,” *SIAM, Journal of Numerical Analysis*, Vol. 20, No. 2, April 1983, pp. 345–357.
- [65] Ghoreyshi, M., Badcock, K., Da Ronch, A., Marques, S., Swift, A., and Ames, N., “Framework for Establishing the Limits of Tabular Aerodynamic Models for Flight Dynamics Analysis,” *AIAA Guidance, Navigation, and Control Conference*, Chicago, Illinois, 10 - 13 August 2009.
- [66] Bruaset, A. M., Tveito, A., and Winther, R., “On the Stability of Relaxed Incomplete LU Factorizations,” *Mathematics of Computation*, Vol. 54, 1990, pp. 701–719.
- [67] Saad, Y., “Krylov Subspace Techniques, Conjugate Gradients, Preconditioning and Sparse Matrix Solvers,” *Von Karman Institute for Fluid Dynamics Lecture Series 1994-05*, March 1994.

- [68] McCroskey, W. J. and Pucci, S. L., “Viscous-Inviscid Interaction on Oscillating Airfoils in Subsonic Flows,” *19th AIAA Aerospace Sciences Meeting*, St Louis, Missouri, 12-15 Jan 1981.
- [69] Goumas, G., Kourtis, K., Anastopoulos, N., Karakasis, V., and Koziris, N., “Understanding the Performance of Sparse Matrix-Vector Multiplication,” *Proceedings of the 16th Euromicro Conference on Parallel, Distributed and Network-Based Processing*, Toulouse, France, 13 - 15 February 2008.

Chapter S

Supplementary Information – Implicit Implementation of LFD

The implementation of the methods in TAU is vital for the efficient running of the code and to maximise the performance. Described here is how the GCR solver with ILU preconditioning has been implemented.

S.1 Matrix Storage

The Jacobian matrices are stored explicitly in memory for both the PETSc and GCR solvers. This requires an efficient storage scheme to ensure that the memory requirement does not become so large as to make the methods unfeasible. The Compressed Sparse Row (CSR) [34] format is used in this report and is constructed as follows. Three arrays are stored, *row* which has one term for each row in the matrix, where the entries indicate which element starts a new row, *column* which has one term for each element in the matrix, where the entries indicate which matrix column the elements are in, and *value* which also has one term for each element in the matrix. An example to illustrate is now shown:

$$\begin{bmatrix} 1 & 0 & 2 & 0 \\ 3 & 4 & 0 & 5 \\ 0 & 6 & 0 & 0 \\ 7 & 8 & 0 & 9 \end{bmatrix}$$

$$row = [1, 3, 6, 7, 10]$$

$$column = [1, 3, 1, 2, 4, 2, 1, 2, 4]$$

$$value = [1, 2, 3, 4, 5, 6, 7, 8, 9]$$

This method makes effective use of the sparse nature of the matrices however, when

these matrices become very large the arrays can require a lot of memory. A blocked format is therefore used where blocks of elements are stored in the same manner as if each block was an element as described above. The blocked form is shown below for the previous matrix with a block size of 2, where the elements inside each block have column major ordering:

$$\begin{aligned} row &= [1, 3, 5] \\ column &= [1, 2, 1, 2] \\ value &= [1, 3, 0, 4, 2, 0, 0, 5, 0, 7, 6, 8, 0, 0, 0, 9] \end{aligned}$$

It can be seen that the blocked format has shorter arrays for *row* and *column* requiring less memory to store an equivalent matrix. This becomes significant when the matrices become very large and for the typical block sizes of 5, 6 and 7 encountered in CFD, the *column* array is at least twenty-five times shorter. The shorter arrays have the added benefit of speeding up the time to access the data as more matrix elements can be stored in the cache leading to a reduction in the solver CPU time.

S.2 Setting up the System Matrix

A key consideration is how to carry out the augmentation of the system matrix. Two approaches to augmentation are possible. The first is creating a large real matrix with four quadrants as shown in Eq.(2.21). This approach is taken with both the PETSc and GCR solvers. This is implemented by taking the Jacobian matrix from TAU which is stored in Block Compressed Sparse Row format (BCSR). The off-diagonal frequency matrices are then inserted into the correct position in the BCSR arrays. The top half of the matrix is created first followed by the addition of the bottom half to this. The functions used to carry out this operation have been optimised to be as efficient as possible and take very little time to run compared to the solution itself.

The second option is to set-up a complex system where the frequency terms are added to the diagonal of the Jacobian matrix. This is slightly more complicated to implement as it involves using complex data structures and complex arithmetic. For the purposes of testing, the complex solver is only implemented as a stand-alone solver which reads the TAU matrices from a file rather than communicating directly with the code. The complex approach has the benefit that adding the frequency terms to the diagonal of the Jacobian matrix does not change the sparsity of the system whereas the real augmentation expands the matrix bandwidth, leading to possible issues in terms of the diagonal dominance and as such the convergence of the solver. A further benefit is the lower memory requirement to run than the real system due to it only having to store the sparsity pattern once.

S.3 Parallelisation

The methods already implemented in TAU (MG LU-SGS and PETSc) run in parallel using MPI commands. For the GCR solver, there are three main considerations. The first is how to implement the preconditioner. ILU preconditioners are global methods and require the whole matrix in order to obtain a good approximation, this would require a lot of parallel communication and for very large systems, this communication could inhibit the performance of the solver. For the method presented in this report, the factorisation is carried out locally and assumes there is no overlap with other processors. This will cause the approximation of the preconditioner to be less accurate but it allows for a more straight forward implementation.

The second consideration is how to carry out the matrix vector products in parallel. When the matrix is decomposed into the required number of domains, some of the off-diagonal terms will be stored on other processors which are required by the matrix-vector product. To obtain these terms, the vector for the multiplication has extra memory assigned to it and a parallel call is made to fill this memory with the correct values from the other processors. Once all the terms are on the correct processors, the matrix-vector product is carried out and the solution is returned to the Krylov solver.

The final consideration is the calculation of the residuals in the Krylov subspace. This is carried out using a global sum function which gathers all the terms in the relevant residual vector and sums these returning a number to all processors to then use. Having addressed all three steps, the preconditioned Krylov solver works efficiently in parallel.

S.4 Memory Reduction

Storing the Jacobian matrix explicitly can require large amounts of memory, especially when the system is augmented and the preconditioner is formed. The main approach that has been taken for the GCR method to minimise the memory requirement is to make use of the preconditioner being an approximation. Initially the preconditioner was implemented to store the value array as doubles which require 8 bytes of memory per entry and ensures the values are accurate to 15 decimal places. This precision however is not needed and instead floats are used which require half the memory and are accurate to 7 decimal places. This switch from doubles to floats is a further approximation, and has a large effect on the run-time memory particularly for preconditioners with high levels of fill-in, without affecting the accuracy of the solution. This approach was first taken by Goumas et al. [69] who looked to minimise the memory through the use of float and short int wherever possible.

A further method was used in the implementation of the GCR solver to remove the need to augment the system matrix. The augmented second order Jacobian is only

needed for the matrix-vector product within the GCR algorithm. Using this knowledge, instructions are given to find the correct elements when carrying out the multiplication so that the Jacobian is only stored once along with one array for all the frequency terms.