

University of Southampton Research Repository  
ePrints Soton

Copyright © and Moral Rights for this thesis are retained by the author and/or other copyright owners. A copy can be downloaded for personal non-commercial research or study, without prior permission or charge. This thesis cannot be reproduced or quoted extensively from without first obtaining permission in writing from the copyright holder/s. The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the copyright holders.

When referring to this work, full bibliographic details including the author, title, awarding institution and date of the thesis must be given e.g.

AUTHOR (year of submission) "Full thesis title", University of Southampton, name of the University School or Department, PhD Thesis, pagination

# Elements of computational flight dynamics for complete aircraft

Thesis submitted in accordance with the requirements of  
Politecnico di Torino for the degree of Dottore Magistrale in Ingegneria  
by  
Marco Cristofaro  
B.E. [Aerospace], Università degli studi di Padova (Italy), 2011

October 2014



Copyright © 2014 by Marco Cristofaro

---

All rights reserved.



# Abstract

This work summarizes the effort to introduce high-fidelity methods in aircraft design process. Flight dynamic characteristics are crucial in aircraft design. Simulation tools evaluate the aircraft statical and dynamical stability and manoeuvrability usually are based on a previously computed tabular aerodynamic model.

During the conceptual design phase, the aerodynamic database is usually computed with semi-empirical methods. These tools rely on existing configurations databases (statistical methods) or linear aerodynamic hypothesis (e.g. vortex lattice method), and so are not suitable for innovative designs. The exploitation of such methods may lead to evaluation errors in the design process, which can be found only in the following steps and so may be very expensive to rectify via additional work, wind tunnel and flight testing, enlarging the time-to-market and increasing the whole life cycle product cost.

The adoption of high-fidelity, physical based aerodynamic models starting from the very first steps of the aircraft design would reduce the uncertainty of current design procedures and prevent costly aircraft retrofitting. Computational fluid dynamics may be utilized to achieve the required high-fidelity, but, because of the substantial computational cost, it is currently used only during ensuing design steps. In this thesis the steps towards an autonomous high-fidelity flight dynamics analysis are presented.

A tool for generating the aerodynamic tables with the semi-empirical United States air force stability and control digital compendium with the common parametric aircraft configuration schema is developed. The function for the flow solver Edge is updated and both scripts are implemented and validated inside the computerised environment for aircraft synthesis and integrated optimisation methods.

Reduced order models to overcome computational fluid dynamics limitations for automated generation of aerodynamic tables are then presented. Two methods are developed in order to obtain a more efficient approach for samples positioning inside the flight envelope domain. Emphasis is given on the ability to capture nonlinearities appearance in the flow field with only a few computations over the whole flight envelope. The methods rely on Kriging interpolation, and are validated for semi-analytical functions and for real test cases. This may permit to reduce the number of required

computational fluid dynamics solutions to use the flight simulator of a factor of some tens, without compromising the main aircraft statical and dynamical behaviour results.

A test case is then presented, showing the statical and dynamical aircraft stability comparison between different geometry configurations, by use of reduced order models and with a low computational budget.

The limitations of a derivatives based aerodynamic model are then presented for a test case, highlighting the differences with a computational fluid dynamics and flight dynamics full-coupled model. A blocks software architecture is used to obtain a tool open and customizable.

A computational fluid dynamics based optimization loop is then used to analyse the longitudinal trim conditions of a test case, presenting the derivatives aerodynamic model limitations. The geometry optimization feasibility, considering the aircraft stability as objective, is assessed. A model based on aerodynamic derivatives is assumed for the representation of the aerodynamic loads, because traditionally used by flight dynamics tools. Advances in this direction are discussed.

# Acknowledgements

I would like to acknowledge my supervisors Dr. A. Da Ronch and Dr. D. D'Ambrosio. I extend my thanks to the colleagues V. Commissio, T. Dechelle, M. Gianfrancesco, E. Picouet, and Y. Wang at the Computational Aeroseervoelastic and Flight Dynamics Lab of the University of Southampton for fruitful discussions and for creating a stimulating and pleasant international working environment over the past six months.

I would like to thank A.V. Jungo and all the CEASIOM developers team for the constructive collaboration.

I am particularly grateful to Dr. A. Da Ronch for being a careful mentor and a supporting friend. His passion and ideas were a continuous stimulus that allowed the development of this work and my personal growth.

The financial support from Politecnico di Torino for the "Tesi all'estero" grant is appreciated.

I ringraziamenti ai miei genitori, Liliana e Giovanni, per avermi cresciuto, supportato e permesso di vivere tutte le esperienze che desideravo, non saranno mai sufficienti per esprimere la mia gratitudine. Inoltre ricordo con gioia e ringrazio per l'affetto mia sorella, Ilaria, e tutta la mia famiglia. Un ringraziamento speciale è dedicato a mio zio, Pino, per l'attenzione dedicatami sin dai primi anni scuola. Ringrazio Tiziana e Roberto per avermi accolto nella loro famiglia. Mando un bacio a Stefania per aver sopportato la continua lontananza, sicuro del prospero futuro che ci attende insieme. Infine dedico il lavoro di questa tesi a mio nonno, Elio, per aver sempre risposto alle mie domande e per rappresentare, anche ora che ci ha lasciato, un grande esempio di vita.



# 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.

Marco Cristofaro  
October 2014



# List of Publications

## Papers in Conference Proceedings

Cristofaro, M., Wang, Y., and Da Ronch, A., "Towards computational flight dynamics of a passenger jet aircraft," *ICAS International Council of Aeronautical Sciences Conference, St Petersburg, Russia*, Vol. 0462, September 2014, DOI: 10.13140/2.1.4988.5445.

Zhang, M., Cristofaro, M., Wang, Y., Da Ronch, A., Rizzi, A., "Investigating the Piaggio Avanti design using CEASIOM," *ICAS International Council of Aeronautical Sciences Conference, St Petersburg, Russia*, Vol. 0464, September 2014, DOI: 10.13140/2.1.2891.3923

Cristofaro, M., Da Ronch, A., Vos, J., and Rizzi, A., "Recent developments in the CEASIOM framework using the common language CPACS," *4th EASN Association International Workshop on Flight Physics and Aircraft Design, Aachen, Germany*, October 2014.



# Table of Contents

<b>Abstract</b>	<b>5</b>
<b>Acknowledgements</b>	<b>7</b>
<b>Declaration</b>	<b>9</b>
<b>List of Publications</b>	<b>11</b>
<b>List of Figures</b>	<b>19</b>
<b>List of Tables</b>	<b>23</b>
<b>List of Symbols</b>	<b>25</b>
<b>1 Introduction</b>	<b>31</b>
1.1 Introduction to computational flight dynamics . . . . .	32
1.2 Computational fluid dynamics . . . . .	33
1.3 CFD progress . . . . .	34
1.4 Aerodynamic coefficients table generation . . . . .	37
1.5 Alternative methods for the generation of aerodynamic table . . . . .	38
1.6 Thesis outline . . . . .	39
<b>2 Aircraft design tools</b>	<b>43</b>
2.1 CEASIOM modules . . . . .	45
2.1.1 Geometry module AcBuilder and SUMO . . . . .	45
2.1.2 Aerodynamic module AMB . . . . .	46
2.1.3 Stability and control module SDSA . . . . .	49
2.1.4 Aeroelastic module NeoCASS . . . . .	50
2.1.5 Test cases . . . . .	52
2.2 Limitation of data handling . . . . .	53
2.3 CPACS file scheme . . . . .	54
2.4 CEASIOM upgrade . . . . .	56
2.4.1 DATCOM for CPACS . . . . .	56
2.4.1.1 Introduction . . . . .	56

2.4.1.2	DATCOM limitations . . . . .	57
2.4.1.3	Software description . . . . .	57
2.4.1.4	Differences with v100 and unsolved issues . . . . .	61
2.4.1.5	Wrapper DATCOM . . . . .	61
2.4.2	Edge . . . . .	63
2.5	Validation . . . . .	65
2.5.1	DATCOM . . . . .	65
2.5.2	Edge . . . . .	67
2.5.3	Conclusion and further development . . . . .	70
<b>3</b>	<b>Surrogate models</b>	<b>71</b>
3.1	Cognitive approach for CFD sampling . . . . .	72
3.2	CEASIOM tools availability . . . . .	74
3.3	Demonstration . . . . .	74
3.3.1	Methods developed on analytical aerodynamic similar functions .	74
3.3.1.1	Local maxima and minima based sampling method . .	77
3.3.1.2	Second derivative based sampling method . . . . .	78
3.3.1.3	Priority functions method . . . . .	81
3.3.1.4	Choice of correlation model . . . . .	82
3.3.2	Multi dimensions domain methods extension . . . . .	83
3.3.2.1	Maximum <i>mse</i> close to local maxima and minima . .	83
3.3.2.2	Hessian based sampling method . . . . .	84
3.4	Validation with realistic cases . . . . .	88
3.4.1	UCAV experimental data . . . . .	88
3.4.2	TCR experimental data . . . . .	91
3.5	Data fusion . . . . .	94
3.5.1	Validation . . . . .	94
3.5.2	Iterative sampling with data fusion . . . . .	96
3.6	Conclusions and future development . . . . .	97
<b>4</b>	<b>Regional jet test case</b>	<b>99</b>
4.1	Test case . . . . .	100
4.2	Generation of Aerodynamic Tables . . . . .	101
4.2.1	Validation . . . . .	102
4.2.2	Aerodynamic Tables . . . . .	102
4.2.3	Kriging Interpolation . . . . .	104
4.2.4	Data Fusion . . . . .	106
4.3	Results . . . . .	107
4.3.1	Baseline Geometry . . . . .	107
4.3.1.1	Trim and Stability . . . . .	107
4.3.1.2	Longitudinal handling qualities . . . . .	109

4.3.1.3	Lateral–directional handling qualities . . . . .	109
4.3.2	New Wing Geometry . . . . .	111
4.3.2.1	Aerodynamic characteristics . . . . .	112
4.3.2.2	Trim and Handling Qualities . . . . .	113
4.3.3	New Horizontal Tail Geometry . . . . .	114
4.4	Conclusions . . . . .	115
<b>5</b>	<b>Alternative aerodynamic models for flight dynamics</b>	<b>117</b>
5.1	Architecture . . . . .	118
5.2	Validation of the CFD module PML . . . . .	119
5.2.1	Two dimensional steady state . . . . .	119
5.2.1.1	Stand alone results comparison for one or two processors computations . . . . .	120
5.2.1.2	Stand alone and FlexFlight results comparison for one processor computation . . . . .	121
5.2.1.3	Computing time . . . . .	121
5.2.1.4	Conclusion . . . . .	122
5.2.2	Two dimensional unsteady . . . . .	123
5.2.2.1	Stand alone and FlexFlight interface results comparison for one processor computation . . . . .	123
5.2.2.2	Conclusion . . . . .	124
5.2.3	Three dimensional steady . . . . .	124
5.2.3.1	Stand alone results comparison for one or two processors computations . . . . .	124
5.2.3.2	Stand alone and FlexFlight interface results comparison for one processor computations . . . . .	125
5.2.3.3	Computing time . . . . .	125
5.2.4	Conclusion . . . . .	126
5.3	Flight dynamics and CFD coupling . . . . .	127
5.3.1	Formulation . . . . .	128
5.3.1.1	Two-dimensional thin aerofoil Theodorsen theory . . .	129
5.3.1.2	Aerodynamic stability derivatives theory . . . . .	129
5.3.2	Results . . . . .	130
5.3.2.1	Time step validation . . . . .	130
5.3.2.2	Initial condition influence . . . . .	131
5.3.2.3	Hinge position effect . . . . .	132
5.3.2.4	Fluid–dynamic method comparison . . . . .	133
5.3.2.5	Free to forced motion comparison . . . . .	134
5.3.3	Conclusions . . . . .	137

<b>6 Design optimization with CFD</b>	<b>139</b>
6.1 Optimizer formulation . . . . .	140
6.1.1 Active set algorithm . . . . .	140
6.1.2 Interior point algorithm . . . . .	140
6.1.3 Response surface reconstruction technique . . . . .	141
6.2 CFD integration in the optimization loop . . . . .	141
6.3 Rae2822 aerofoil test case . . . . .	142
6.3.1 Formulation . . . . .	142
6.3.2 Results . . . . .	142
6.3.2.1 Active set . . . . .	142
6.3.2.2 Interior point . . . . .	143
6.3.2.3 Response surface reconstruction . . . . .	144
6.3.2.4 Comparison . . . . .	145
6.4 Piaggio Avanti test case . . . . .	147
6.4.1 Formulation . . . . .	148
6.4.2 Results at fixed speed . . . . .	149
6.4.3 Results without rotational equilibrium . . . . .	151
6.5 Geometry optimization problem definition . . . . .	152
6.6 Conclusions and future work . . . . .	154
<b>7 Conclusions and Outlook</b>	<b>155</b>
<b>Bibliography</b>	<b>159</b>
<b>A The Kriging interpolation toolbox DACE</b>	<b>163</b>
A.1 Theoretical basis . . . . .	163
A.2 Practical application . . . . .	164
A.3 Numerical examples over the influence of the input parameters . . . . .	165
<b>B Controller design</b>	<b>169</b>
B.1 Model predictive control . . . . .	169
B.1.1 Application to a mass spring system . . . . .	170
B.1.2 Application to sampling problem . . . . .	171
B.2 Game theory: Q–learning . . . . .	173
B.2.1 Application to sampling decision . . . . .	174
<b>C Investigating the Piaggio Avanti design using CEASIOM</b>	<b>175</b>
C.1 Abstract . . . . .	175
C.2 Introduction . . . . .	175
C.3 Conceptual Design Tool AAA . . . . .	176
C.4 Preliminary Design Toolset CEASIOM . . . . .	177
C.5 Interfaces and Wrappers . . . . .	178

C.6	CEASIOM Down-Select Configuration & Pitch Control Study	179
C.7	Piaggio Avanti	181
C.7.1	Aerodynamics using Vortex Lattice Method	182
C.7.2	Euler solutions	183
C.7.2.1	Innovative design	185
C.7.2.2	Steady pull-up	185
C.8	Conclusions	186



# List of Figures

1.1	Design process graphical trends representation [1] . . . . .	31
1.2	NASA Technology Development Roadmap for CFD [2] . . . . .	36
2.1	CEASIOM scheme [3] . . . . .	44
2.2	AMB scheme [3] . . . . .	48
2.3	NeoCASS layout [4] . . . . .	50
2.4	Old <code>xml</code> CEASIOM file parameters for the wing . . . . .	53
2.5	<code>CPACS.xml</code> file scheme, main structure and wing definition . . . . .	55
2.6	Geometry definition differences within the old and new CEASIOM . . . . .	55
2.7	DATCOM for CPACS script structure . . . . .	60
2.8	DATCOM wing parameter definition [5] . . . . .	62
2.9	Boeing 747-8 real model . . . . .	65
2.10	Lift, polar and pitching moment dependence on angle of attack comparison at Mach number 0.75 and altitude of 10,000 m . . . . .	66
2.11	Piaggio P180 II real model . . . . .	67
2.12	Piaggio P180 II surface mesh, highlighting the control surfaces . . . . .	67
2.13	Lift, drag and pitching moment computed at the nose Edge results comparison; Mach numbers 0.35 and 0.7 at the altitude of 10,000 m . . . . .	69
3.1	Kriging <i>mse</i> -based sampling for lift similar function at iteration 10 . . . . .	76
3.2	Maxima or minima based method for lift-similar function at iteration 10 . . . . .	77
3.3	Comparing two samplings methods for a non-continuously differentiable function with gauss correlation model . . . . .	78
3.4	Comparing the five samplings methods for a non-continuously differentiable function with exponential correlation model . . . . .	80
3.5	Giving priority to the firsts function for choosing the next sampling after 10 iterations . . . . .	81
3.6	Differences between exponential and spherical correlation models . . . . .	82
3.7	Generic 2-dimensional sub-space grid to compute mixed differences, for any $n$ -dimensional domain . . . . .	85
3.8	UCAV predicted flow topology for different angles of attack (AoA) [6] . . . . .	88

3.9	Comparing the results obtained with the two developed methods on the UCAV data after 10 iterations (2 initial samples at the extrema) . . . . .	90
3.10	Transonic cruiser CFD pressure distribution, at $\alpha = 0, 8, 20$ deg; $V_\infty = 40$ m/s, $\beta = 0$ deg and $\delta_{can} = 10$ deg . . . . .	91
3.11	Comparing the results obtained with the two developed methods on the TCR data after 10 iterations (with 4 initial samples at the domain vertices) . . . . .	93
3.12	Data fusion applied to the UCAV wind tunnel high fidelity data with low two different fidelity data . . . . .	95
3.13	Iterative data fusion applied to the UCAV wind tunnel high fidelity data after 3 iterations different low-fidelity databases . . . . .	96
4.1	Wind tunnel model of the regional jet aircraft in scale 1:23 . . . . .	101
4.2	Surface grid with the control surfaces in red and pressure coefficient distribution resulting from Euler solution at $M = 0.78$ and $\alpha = 3$ deg . . . . .	101
4.3	Lift, drag and pitch moment coefficients at Mach number 0.5 and Reynolds number $2.2 \cdot 10^7$ ; the reference point is taken at the centre of gravity . . . . .	103
4.4	Samples distribution over the $\alpha - M$ domain . . . . .	104
4.5	Dependency of pitch moment coefficient on angle of attack and Mach number; the surface represents the Kriging interpolation of Edge samples (red cubes) . . . . .	105
4.6	Dependency of pitch moment coefficient on angle of attack and Mach number; the surface represents data fusion between DATCOM and Edge . . . . .	106
4.7	Trim conditions at different cruise altitudes; see Table 4.2 for tables definition . . . . .	108
4.8	Phugoid and short period characteristics according to ICAO criteria at Mach number of 0.78 and altitude of 11,887 m; see Table 4.2 for definition . . . . .	109
4.9	Dutch roll and spiral characteristics according to MIL-F-8785C criteria at Phase B at Mach number of 0.78 and altitude of 11,887 m; see Table 4.2 for definition . . . . .	110
4.10	Roll characteristics according to Cooper-Harper criterion at Mach number of 0.78 and altitude of 11,887 m; see Table 4.2 for definition . . . . .	110
4.11	Configurations 1 and 8 geometry compared with the baseline . . . . .	111
4.12	Pitching moment coefficient comparison at Mach number of 0.78 and altitude of 11,887 m; see Table 4.4 for definition; arrows indicate increasing values of wing sweep angle . . . . .	112
4.13	Trim conditions at different cruise altitudes; see Table 4.4 for definition; arrows indicate increasing values of wing sweep angle . . . . .	113

4.14	Short period characteristics according to ICAO criteria at Mach number of 0.78 and altitude of 11,887 m; arrows indicate increasing values of wing sweep angle . . . . .	113
4.15	Short period characteristics according to ICAO criteria at Mach number of 0.78 and altitude of 11,887 m; arrows indicate increasing values of horizontal tail volume . . . . .	115
5.1	FlexFlight architecture scheme . . . . .	119
5.2	2 dimensional NACA 0012 two processors grid division . . . . .	120
5.3	$C_p$ distribution for the CFD solver with one and two processors . . . . .	120
5.4	$C_p$ distribution on the aerofoil for the stand alone CFD solver and via Python . . . . .	121
5.5	Residual trend for the stand alone solver and with the FlexFlight interface	122
5.6	Residual trends for one and two processors computations . . . . .	122
5.7	Lift and pitching moment for a pitching sinusoidal motion with amplitude $\alpha_A = 4.59$ deg and reduced frequency $k = 0.0811$ . . . . .	123
5.8	Two processors Goland wing surface mesh division . . . . .	124
5.9	Pressure coefficient on the chord at different location along the wing span	125
5.10	Residual trend for the stand alone solver and with the FlexFlight interface; computation with 4 processors . . . . .	126
5.11	Residual temporal trend for 1,2 and 4 processors computations; the stand alone solver is used . . . . .	127
5.12	NACA 0012 with the reference frame and the aerodynamic external forces	128
5.13	Time step effect on the flight-dynamic simulation; free stream velocity at 50 m/s and hinge position at 5% of the chord . . . . .	131
5.14	Adimensionalized pitch angle time history; free stream velocity of 50 m/s	132
5.15	Hinge position effect over the free to pitch aerofoil motion; free stream velocity at 50 m/s . . . . .	133
5.16	Fluid-dynamic method effect over the free to pitch aerofoil motion; free stream velocity at 50 m/s and hinge position at 5% of the chord . . . . .	134
5.17	Euler free motion applied to forced motion computations . . . . .	135
5.18	Euler free motion applied to forced motion Euler and RANS CFD computations, highlighting the pressure coefficient difference at the initial state . . . . .	136
5.19	Pressure field comparison between Euler and RANS steady simulations at $\alpha = 1$ deg; only the upper part of the flow field is shown with upper part presenting not viscous solution and lower viscous . . . . .	136
6.1	Active set algorithm iteration history . . . . .	143
6.2	Interior point algorithm iteration history . . . . .	144

6.3	Sampling algorithm iteration history . . . . .	145
6.4	Drag coefficient samples with the different algorithms . . . . .	146
6.5	Pressure coefficient at the active set minimum computed drag ( $C_D = 0.002215$ at $\alpha = -2.87$ deg) . . . . .	146
6.6	Mesh deformation detail for a deflection of -12 deg (down) . . . . .	147
6.7	Drag, lift and pitching moment with reference to the estimated CG coefficients; Mach 0.83 at 39000ft ( $v_\infty=250$ m/s) . . . . .	150
6.8	Drag coefficient and drag force trends changing the velocity; with vertical equilibrium imposed . . . . .	152
6.9	Optimization loop graphical representation; geometry analysis for aircraft stability characteristics optimization . . . . .	153
A.1	Comparing the three regression models in DACE . . . . .	165
A.2	Comparing the correlation models in DACE . . . . .	166
A.3	$\theta$ parameter effect on Gaussian correlation models . . . . .	167
A.4	Problem with two unknowns . . . . .	167
B.1	MPC working scheme [7] . . . . .	169
B.2	Model predictive controller applied to a mass spring system . . . . .	171
B.3	MPC definiton scheme for sampling problem . . . . .	171
B.4	MPC test on $C_L-\alpha$ function sampling problem presented in Chapter 3 .	172
B.5	Q-learning theory application applied to the sampling decision method .	174
C.1	The two design loops in the <i>conceptual design phase</i> process and the <i>down-select</i> to project study in <i>preliminary design</i> . . . . .	177
C.2	Different conceptual design tools linked to preliminary design tool-chain CEASIOM by CPACS . . . . .	179
C.3	3-view drawing of Piaggio–Avanti in AAA . . . . .	180
C.4	Control surface illustrated on 3-view drawing [8] . . . . .	182
C.5	Tornado VLM solutions from CEASIOM . . . . .	182
C.6	Euler solutions from CEASIOM . . . . .	183
C.7	Euler solutions at Mach number of 0.62, canard ON & OFF cases; error bars represent maximum deviations during the last 500 CFD iterations .	184
C.8	$C_p$ from Euler solutions for trimmed flight at Mach number of 0.62 ( $U = 356$ kts at 39,000 ft), full & canard–off configurations . . . . .	184
C.9	Longitudinal trim calculated and interpolated/extrapolated from Euler solutions for full (solid line) & canard–off (dashed line) configurations .	185
C.10	Economy cruise trimmed conditions at altitude of 39,000 ft, calculated from CEASIOM–Euler . . . . .	185
C.11	Mach contour from CEASIOM–Euler solutions for Piaggio Avanti 3g steady pull–up manoeuvre . . . . .	186

# List of Tables

1.1	Structure of the aerodynamic table; the x represents that those variables are combined. . . . .	37
3.1	Structure of the aerodynamic table database constructed in CEASIOM; the x represents that all the values are considered and combined. . . . .	72
3.2	Number of initial sample points for which the regression model is satisfied and bring to a zero <i>mse</i> everywhere . . . . .	76
3.3	Mean real relative errors comparison for the UCAV case using second derivative and local maxima or minima based methods . . . . .	89
3.4	Mean predicted relative errors comparison for the UCAV case using second derivative and local maxima or minima based methods . . . . .	89
3.5	Mean real relative errors comparison for the multi-dimensional TCR case using Hessian and local maxima or minima based methods . . . . .	92
3.6	Mean predicted relative errors comparison for the multi-dimensional TCR case using Hessian and local maxima or minima based methods . . . . .	92
3.7	Mean real relative errors comparison for iterative data fusion applied to the UCAV wind tunnel high fidelity data after 3 and 10 iterations different low-fidelity databases . . . . .	96
4.1	Mission specifications and dimensions of the regional jet aircraft . . . . .	100
4.2	Generated full tables and reference numbers . . . . .	107
4.3	Baseline mass and inertia computed properties . . . . .	108
4.4	Configurations with different leading edge sweep angle ( $\Lambda_{LE}$ ) and aspect ratio ( $AR$ ) . . . . .	111
4.5	Configuration with different horizontal tail . . . . .	114
5.1	Resulting external aerodynamic forces and moments . . . . .	120
5.2	Computation time results for a mean residual of $10^{-8}$ . . . . .	121
5.3	Half wing resulting force and moment aerodynamic coefficients . . . . .	125
5.4	Computation time in seconds for a mean residual of $10^{-3}$ . . . . .	125
5.5	First oscillation extrema for the NACA0012 with different time steps, starting from $\alpha_0 = 1$ deg . . . . .	131

5.6	First oscillation extrema for the NACA0012 with different initial conditions	132
6.1	RAE2822 drag optimization algorithm results comparison after 20 iterations	145
6.2	Piaggio aerodynamic derivatives	151
C.1	Structure of the aerodynamic database constructed in CEASIOM for use in the flight simulation SDSA module.	179
C.2	Cruise speed at maximum cruise power [8]	181

# List of Symbols

$a$	= speed of sound, [m/s]
$AR$	= aspect ratio, $b^2/S$
$b$	= reference wing span, [m]
$c$	= mean aerodynamic chord, [m]
$d$	= fuselage total length, [m]
$C_L$	= lift coefficient
$C_D$	= drag coefficient
$C_\ell$	= rolling moment coefficient
$C_m$	= pitching moment coefficient
$C_n$	= yawing moment coefficient
$C_p$	= pressure coefficient, $2(p - p_\infty)/\rho V_\infty^2$
$c_r$	= root chord, [m]
$c_t$	= tip chord, [m]
$C_Y$	= lateral force coefficient
$\mathbf{C}_{nl}$	= nonlinear constraints
$D$	= drag force, [N]
$f$	= dimensional frequency, [Hz]
$g$	= gravitational acceleration, [m/s <sup>2</sup> ]
$h$	= vertical displacement, [m]
$\mathbf{H}$	= Hessian matrix, $\partial^2(\cdot)/\partial x_i \partial x_j$
$\mathbf{I}$	= matrix of inertia, [kg m <sup>2</sup> ]
$L$	= lift force, [N]
$k$	= reduced oscillation frequency, $\omega c/(2 V_\infty)$
$M$	= Mach number, $V_\infty/a$
$m$	= mass, [kg]
$p$	= local pressure, [Pa]
$p_\infty$	= free stream pressure, [Pa]
$p_0$	= total pressure, $p + \rho V_\infty^2/2$ , [Pa]
$q$	= pitch rotational speed, [deg/s]
$Re$	= Reynolds number, $V_\infty c/\nu$
$S$	= reference wing area, [m <sup>2</sup> ]

<b>T</b>	= trasformation matrix
$T$	= thrust, [N]
$t$	= physical time, [s]
$V_\infty$	= freestream speed, [m/s]
$x$	= horizontal displacement, [m]
$x_{a.c.}$	= longitudinal aerodynamic center position, [m]
$x_{a.e.}$	= longitudinal elastic axes position, [m]
$x_{c.g.}$	= longitudinal center of gravity position, [m]
$x_h/c$	= adimensional hinge position along the aerfoil chord
$W$	= weight, $m \cdot g$ , [N]

## Greek Symbols

$\alpha$	= angle of attack (aka incidence), [deg]
$\alpha_0$	= initial incidence, [deg]
$\alpha_A$	= amplitude of oscillatory motion in incidence, [deg]
$\alpha_r$	= root chord aerfoil rotation, [deg]
$\alpha_t$	= tip chord aerfoil rotation, [deg]
$\beta$	= angle of sideslip, [deg]
$\beta_t$	= twist angle, [deg]
$\beta(x)$	= increment function
$\Gamma$	= dihedral angle, [deg]
$\Delta T$	= thrust increment, [N]
$\delta_a$	= ailerons deflection, [deg]
$\delta_{can}$	= canard deflection, [deg]
$\delta_e$	= elevator deflection, [deg]
$\delta_F$	= flap deflection, [deg]
$\delta_r$	= rudder deflection, [deg]
$\varepsilon_R$	= integral mean real error
$\varepsilon(\cdot)$	= integral mean real error for the $(\cdot)$ function
$\nu$	= kinematic viscosity of air, [m <sup>2</sup> /s]
$\Theta$	= pitch angle, [deg]
$\theta$	= correlation function parameters for Kriging interpolation
$\Lambda$	= sweep angle at the quarter chord, [deg]
$\mu$	= weight coefficient, $2W/\rho V_\infty^2 S$
$\Phi_w$	= Wagner function
$\Psi_1, \Psi_2, \varepsilon_1, \varepsilon_2$	= Jones coefficients for the Wagner function approximation
$\tau$	= non-dimensional time, $t 2V_\infty/c$
$\omega$	= oscillation frequency, $2 \pi f$ , [rad/s]

## Acronyms

AAA	= advanced aircraft analysis
aka	= also known as
CAD	= computer-aided design
CEASIOM	= computerized environment for aircraft synthesis and integrated optimization methods
CFD	= computational fluid dynamics
CFL	= Courant–Friedrichs–Lewy number
CG	= center of gravity
CPACS	= common parametric aircraft configuration schema
DACE	= design and analysis of computer experiments
DATCOM	= data compendium
DLR	= German aerospace center
DoF	= degrees of freedom
EDGE	= CFD solver from FOI
EIF	= expected improvement function
FCS	= flight control system
FOI	= Swedish defence agency
GUESS	= generic unknowns estimator in structural sizing
<i>hf</i>	= high fidelity
ICAO	= international civil aviation organization
IGES	= graphics exchange specification
<i>lf</i>	= low fidelity
MDO	= multi-disciplinary optimization
<i>mse</i>	= mean square error
NeoCASS	= next generation aero structural sizing suite
PML	= parallel meshless CFD solver
RANS	= Reynolds-averaged Navier–Stokes
ROM	= reduced order model
SDSA	= simulation and dynamic stability analysis
SimSAC	= simulating aircraft stability and control
SMARTCAD	= simplified models for aeroelasticity in conceptual aircraft design
SBRF	= surrogate-based recurrence-framework
S&C	= stability and control
SUMO	= surface modeler
VLM	= vortex lattice method
TCR	= transonic cruiser
UVLM	= unsteady vortex lattice method
UCAV	= unmanned combat air vehicle



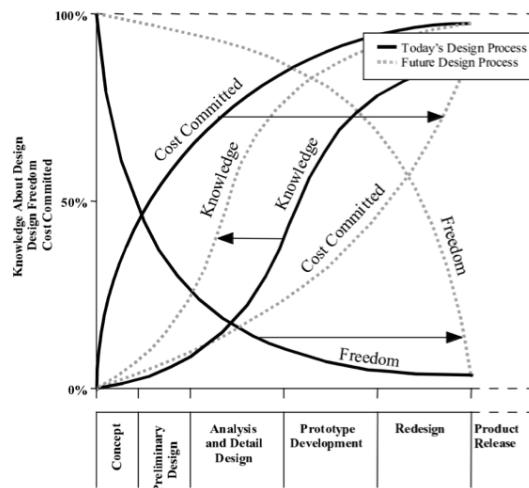
*La coda ha moti e se 'l destro stremo della coda bassa sar più basso che 'l sinistro,  
allora si volterà inverso il lato sinistro.*  
Leonardo da Vinci, Codice del volo, 1505



# Chapter 1

## Introduction

The aircraft design process is a complex multi-disciplinary optimization problem. The project development is usually divided into steps, characterized by the exploitation of different tools with increasing models fidelity. The design is an iterative process, but bad choice taken in the first steps may lead to expensive retrofitting or, at the limit, to the end of the project. The conceptual and preliminary design phases are then the most influencing over the overall project cost and quality. Figure 1.1 presents a graphical representation of the design process, in terms of cost committed, knowledge, and freedom [1].



**Figure 1.1:** Design process graphical trends representation [1]

During the first design steps cheap semi-empirical models are usually employed. These methods are based on strong approximation or statistical data, but their low cost allows the analyses of various configurations. Structural beam model, aerodynamic linear panel method and Bryan's flight dynamic method can give the gist of the main aircraft design parameters influence and an approximated final design overview.

The target of the presented study is the understanding of the quality and number of data required to perform better and faster designs. Good aircraft projects can be more easily generated if higher fidelity models are used earlier during the design process.

The solution can be found in the generation of an open, collaborative and multi-fidelity software. Different disciplines should interact between each other, with a free data exchanging. The software should be able to include different fidelity and cost model for every discipline, allowing the user to combine all of them.

The computerized environment for aircraft synthesis and integrated optimization methods (CEASIOM) is used in this work as base platform for the object study. Enhancements, improvement and tests are mainly run by the exploitation of this environment. However, the developments that are obtained in this software, should not be considered confined in it, since every method and model can be applied in different design environments or any external analyses.

## 1.1 Introduction to computational flight dynamics

The flight simulation is usually empowered to a tool that requires a tabular aerodynamic model as input. Since every point of the flight envelope should be computed, the generation of this aerodynamic table with viscous flow simulations can be very expensive.

The employment of high fidelity computational fluid dynamics, CFD, simulations during the conceptual phase would reduce the design uncertainty and avoid costly retrofitting during the next steps. It would become a step toward the future design process and so move the trends of Fig. 1.1 as presented with the dotted lines.

In order to perform flight simulation with Navier–Stokes CFD results, reduced order models may be exploited to reduce the overall cost but still maintaining a high fidelity. These models should reduce the number of required simulations for a full order aerodynamic model but still characterize possible nonlinearities. The advantages taken using a viscous flow model must not be lost during the creation of a reduced order model. The application of these methods would permit the computation of CFD based aerodynamic tables since the conceptual design phase.

Initial experimental validation of such methods can be performed in wind tunnel, with the final validation coming in full-scale testing, e.g. flight tests. After that the validation is completed, the methods can be fully exploited during the aircraft design process.

Generally speaking, aerodynamic forces and moments depend on the whole time history of motion in a nonlinear way [9]. However the state of art flight simulator tools are based on linearized or tabular aerodynamic models. These models applies linearization hypothesis to the flow response, and approximate the whole flow field time history with small independent steps. For example, the evaluation of the dynamic

derivatives (e.g. lift with different pitch rotational speed) can be approximated with a sinusoidal time-dependent CFD simulation at a defined oscillation frequency. This represents a strong approximation, since the real states time-history spectrum may contain different frequencies. These models are computationally low-cost, and real time simulations can be easily run.

Today, higher speed processors and bigger quantitative of available memory, allow to surpass the traditional flight simulator approach. Unsteady CFD computation can be used to calculate time accurate aerodynamic forces and moments. These values can be used for the motion equation integration, in order to obtain a CFD time accurate flight simulation. Real time simulations may be achieved with today high performances computers, or probably tomorrow' common personal computers. This method permits to introduce time accurate structural deformation, obtaining a full aeroelastic model. Furthermore the generation of the aerodynamic table is no more necessary, although the flight simulation cost is much higher.

## 1.2 Computational fluid dynamics

The study of performance, stability, and control of a flying vehicle is a challenging problem. The flow field around the aircraft is governed by nonlinear partial differential equations, and so the interaction analyses between resulting forces and moments and aircraft attitude and velocity is highly complex.

The fast technical evolution in the computer manufacturing, supplies everyday faster processors and wider memory quantities. These calculating tools can be employed for the study on highly computational demanding problems.

The resolution of a generic fluid dynamic problem can be considered one of them. The motion of fluid substances is described by the Navier–Stokes equations, named after Claude–Louis Navier and George Gabriel Stokes. Their nonlinear nature makes most problems difficult or impossible to solve analytically. The computational fluid dynamics (CFD) aims to discretize a continuum domain, generating a solvable problem. The discretization is obtained by dividing the domain in small parts (mesh) and the solution is a discrete approximation of the real flow field. CFD simulates the interaction of liquids and gases with surfaces defined by boundary conditions. With higher-speed supercomputers, more accurate and efficient solutions can be achieved.

Historically, methods were first developed to solve the linearized potential equations. Two-dimensional methods, using conformal transformations of the flow about a cylinder to the flow about an aerofoil were developed in the 1930s [10]. During the 60's the first three-dimensional analyses were based on a discretization of the geometry surface with panels, giving rise to panel methods programs class. In the two-dimensional realm, a various number of panel codes have been developed for aerofoil analysis and design. The codes typically have a boundary layer analysis included, so that viscous

effects can be modelled [11]. The next step during the 1980's was the Euler equations, which promised to provide more accurate solutions of transonic flows. The Navier-Stokes viscous equations are the ultimate developers target.

Three-dimensional methods are important tools for the aircraft stability and handling characteristics evaluation. The first models were based only on statistical or semi-empirical considerations. Although the resulting data were usually well representative for traditional configurations, these methods are not usable for new aircraft concepts and might fail for usual configuration as well. Physical based models, exploiting CFD for the flow field solution, were then introduced simultaneously to their development. The resulting pressure distribution over the body surface is integrated to obtain aerodynamic forces and moments and their values are then used to integrate the motion equation. With the prevailing Navier-Stokes solvers development, viscous flows solutions are today introduced for stability and control evaluation.

### 1.3 CFD progress

The fast progression of computational fluid dynamics technology has fundamentally changed the aerospace design process compared to the pre-computer era. The use of CFD permits a reduction in wind tunnel time for aircraft development, as well as lower numbers of experimental tests in gas turbine engine design. CFD enabled the design of high speed re-entry vehicles in the absence of any reliable testing facilities. Furthermore physics-based simulation technologies such as CFD offer the possibility of understanding and insight into the critical physical phenomena limiting component performance, thus opening new frontiers in aerospace vehicle design.

The research and development of computational aerodynamics has seen the birth of higher fidelity methods from the 1970s to the 1990s. Starting with panel methods, proceeding to linearized and nonlinear potential flow models, inviscid flow (Euler equations based) methods until the Reynolds-averaged Navier-Stokes (RANS) equations solution.

These advances were, and still are, beneficial to the progression in High performance computing hardware. Moores law states:

*"Processor speed doubles approximately every two years"*

It holds very well about the increase in computational power during the last 20 years, and a comparable development of advanced algorithms was carried during the same time.

Unfortunately the last decade has seen a stop in the capabilities used in aerodynamic simulation within the aerospace sector, with RANS methods considered the high-fidelity method and improvement adopting larger meshes, more complex geometries, and more numerous runs afforded by the decreasing hardware costs. However the

well-known limitations of RANS methods for separated flows have confined reliable use of CFD to a small region of the flight envelope or operating design space. For this reason the current CFD has become a mature technology but trustworthy only for cases for which a previous experience base exists.

The required ICAO Climate Change Action Plan <sup>1</sup> in fuel burn, noise emissions, pollution, and climate impact will only be obtained with more sophisticated analysis of future configurations. New conspicuous investments in the resulting engineering design process would overall decrease risk during the design phase of new configurations, reduce time-to-market, improve products, and facilitate revolutionary aerospace vehicles through the ability to consider novel designs. The improvement of a simulation-based engineering design process along which CFD plays a critical role is a multifaceted problem. Having until now relied on mature algorithms and exploited the decreasing computer hardware costs, the CFD development community is now unable to capitalise on the rapidly changing computer architectures, which include massive parallelism and heterogeneous architectures. The scale and diversity of problems inside the aerospace engineering are such that the computational power has to be put next to new algorithms, solvers, physical models, and techniques with better mathematical and numerical properties. Software complexity is increasing exponentially, slowing adoption of novel techniques into production codes and shutting out production of new software development efforts, while simultaneously complicating the coupling of various disciplinary codes for multi-disciplinary analysis and design. In [2] is presented the NASA plan for the adoption of a strategy to achieve the goals of the Vision 2030 CFD capability presented in Fig. 1.2.

---

<sup>1</sup>[http://www.icao.int/environmental-protection/Pages/ClimateChange\\_ActionPlan.aspx](http://www.icao.int/environmental-protection/Pages/ClimateChange_ActionPlan.aspx) [retrieved October 10, 2014]

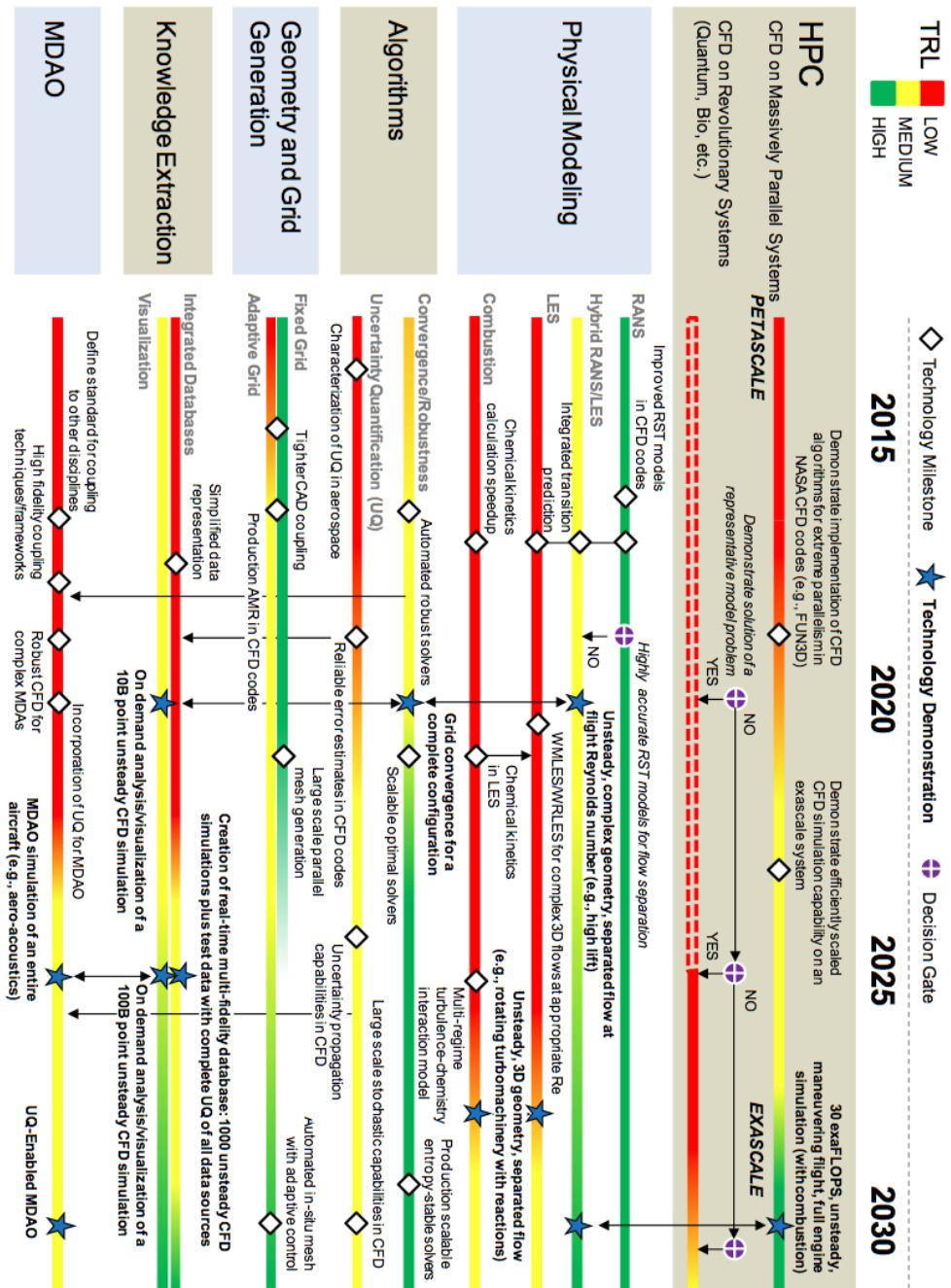


Figure 1.2: NASA Technology Development Roadmap for CFD [2]

## 1.4 Aerodynamic coefficients table generation

In order to solve equations of motion, aerodynamic forces and moments must be expressed as a function of the aircraft mass and inertia, the state variables and command variables. In general aerodynamic forces and moments depend on the whole time history of motion in a non-linear way, but some simplifying assumptions have to be taken into account to solve equations of motion. Forces and moments are assumed to have a linear dependence on their parameters and the other is small perturbation hypothesis concerning the analysis of motion.

Bryans method is based on the variations evaluation of forces and moments with reference to an initial condition (usually equilibrium). Considering the Taylor expansion and neglecting higher order terms, one can rewrite forces and moments differences as sum of aerodynamic derivatives multiplied by the corresponding variables variations. Then, experimental results demonstrate that some of the aerodynamic derivatives have a very small influence and so they can be neglected.

The typical problem in flight dynamic studies is faithfully describing the relationship between the motion variables and the aerodynamic loads, inside the inertial equations of motion.

The practical computation of the forces and moments acting on a flying aircraft is traditionally empowered to the aerodynamic coefficients table previously computed. So the generation of a well representative aerodynamic coefficients table is crucial to obtain a good simulation of the flight dynamic. The external aerodynamic forces and moments acting on the aircraft are tabulated for every different combination of statical state variables (e.g.  $M, \alpha, \beta, \dots$ ), dynamical state variables (e.g.  $p, q, \dots$ ) and control variables (e.g.  $\delta_e, \delta_r, \dots$ ). For this reason the aerodynamic table may contain many thousands of entries. Table 1.1 represents a possible form, with the variables of the left side and the resulting aerodynamic coefficients on the right.

$\alpha$	$M$	$\beta$	$\delta_{ele}$	$\delta_{rud}$	$\delta_{ail}$	...	$p$	$q$	$r$	$C_L$	$C_D$	$C_m$	$C_Y$	$C_\ell$	$C_n$
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	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	x	—	—	—	—	—	—	—	x	x	x	x	x	x	x

**Table 1.1:** Structure of the aerodynamic table; the x represents that those variables are combined.

A flight simulator uses then these values to compute the aerodynamic forces acting on an aircraft and solve the motion equation to obtain the following states.

The CEASIOM software aims to help the design engineers for the stability and control project evaluation during the conceptual design phase. In [12] the generation of CFD grids starting from the CEASIOM geometry is presented. The generation process of a tabular aerodynamic model is then showed by Da Ronch et al. in [13] and by Ghoreyshi et al. in [14]. The included flight simulator tool is then discussed in [15]. The evaluation of the dynamic derivatives with unsteady time-accurate simulations are extensively analysed in [16–18]. Some test cases are then used to validate the CEASIOM software. The existing Boeing 747 is investigated in [19] and a new transonic passenger aircraft configuration is analysed in [20].

## 1.5 Alternative methods for the generation of aerodynamic table

During the conceptual phase of the aircraft design, empirical methods are usually used to obtain these tables, with no possibility in prediction of the flow non-linearity that might happen in the flight envelope. In order to have a good quality aerodynamic table starting from the early design phase some methods may be used during the computations. Methods to automate and reduce the high-fidelity analyses required for the generation of aerodynamic tables are presented in [13, 14]. It uses a Kriging-based surrogate model [21] and allows fusing two available databases. It stated the needs to address further research at the development of a methodology to efficiently identify local minima/maxima and changes in curvature in the aerodynamic loads. Mackman et al. in [22] presented a surrogate modelling strategy, using effective interpolation and sampling methods. Despite the intention to reduce the number of calculations, they used an unrealistic number of samples for any CFD-based application. In [23] Santini developed a data fusion iterative program to obtain  $N$  new sample point positions at every iteration, adopting every time data fusion with a low fidelity database and considering various criteria. An efficient creation of the aerodynamic database for the X-31 experimental aircraft is presented in [24] and the generation of reduced-order models for computing the unsteady and nonlinear aerodynamic loads from pitching motions in the transonic speed range is described in [25]

## 1.6 Thesis outline

The work in this thesis is partly developed within the CEASIOM project <sup>2</sup>. The main driver of the project is to overtake the inadequacy of standard semi-empirical methods currently used during the aircraft conceptual design [3]. The prevailing tools are not suitable for innovative configurations and their exploitation may lead to evaluation errors in the design process, which can be found only in the following steps. Early errors during the design process may be very expensive to rectify via additional work, wind tunnel and flight testing, enlarging the time-to-market and increasing the whole life cycle product cost. The augmented fidelity methods would reduce the uncertainty of current design procedures and prevent costly aircraft retrofitting.

In this thesis, the exploitation of CFD in a routine manner is investigated for flight mechanics. The use of reduced order models for automated generation of aerodynamic tables is discussed, so that high fidelity stability and control results can be obtained with a reasonable computational cost. The use of CFD methods for complex configurations, coupled with an efficient and smart algorithm for reduced order models, allows the generation of feasible and realistic tables with 100,000 flight points. Furthermore it may be integrated inside multi-disciplinary optimization problems. Emphasis is given on developing a method able to capture nonlinearities appearance in the flow field with only a few CFD samples over the whole flight envelope. A test case is presented, showing a comparison between configurations with geometry variations using a low computational cost. The geometry optimization loop feasibility from a very early design phase is then assessed. A model based on stability or aerodynamic derivatives is assumed for the representation of the aerodynamic loads, because traditionally used by flight dynamics tools. The limitations of a derivatives based aerodynamic model are then presented for a test case, highlighting the differences with a coupled CFD-flight dynamics model. A CFD based optimization loop is then used to assess the trim conditions of a test case, presenting the differences with the derivatives aerodynamic model results. Advances in this direction are discussed.

Chapter 2 introduces the CEASIOM framework for the multi-disciplinary aircraft design process. The old CEASIOM file format limitations are overtaken by the new CPACS file format <sup>3</sup>. This step makes the geometry completely customizable, giving to designers full control of it and allowing non-traditional configurations. The counterpart of such freedom is the loss of data meaning, so a conversion is needed in order to lead back the problem to the traditional geometry variables. Matlab scripts for the digital semi-empirical method Datcom and the CFD solver Edge with the CPACS file format are implemented and validated with the old CEASIOM file format.

---

<sup>2</sup>More details at <http://www.ceasiom.com/> [retrieved October 10, 2014]

<sup>3</sup>More details at <https://code.google.com/p/cpacs/> [retrieved October 10, 2014]

Chapter 3 discusses the needs of a rational approach for CFD sampling in the flight envelope domain. The aerodynamics for traditional flight conditions (e.g. small angle of attack, subsonic speed) generally follows linear trends, but the appearance of an unexpected nonlinearity in the flow field may lead to evaluation errors. Reduce order models have to be considered to limit the overall computational cost, to introduce CFD in the early design steps. A Kriging interpolation model can be generated to compute the aerodynamic function over the whole flight envelope starting from a few CFD samples. A tool able to focus on the nonlinearities appearance and not waste computational resources for the useless linear part is then developed and discussed. A ten of samples per domain dimension are considered as target for a good evaluation and two strategies are adopted and compared. The tool is then validated with both semi-analytical and realistic functions.

Chapter 4 presents the study on a real regional jet test case. An efficient generation of an aerodynamic database for flight simulations is investigated. The aim of this analysis is to find how geometry variations influence the flight dynamics, by using high fidelity CFD generated aerodynamic tables. The influence of aerodynamic model on the trim and flight dynamics is first presented. Euler CFD computations are used with a Kriging interpolation model to obtain a high-fidelity full order aerodynamic model at low computational cost. The weight and balance characteristics were computed via empirical methods. The study shows then how a full aero table can be generated for any new configuration, from the baseline model via data fusion. Different geometrical configurations are considered and 14 of them are then analysed with low-cost CFD based aerodynamic models, presenting the differences about the static and dynamic stability obtained from a flight simulator.

Chapter 5 introduces an alternative model formulation to be used in the representation of flight dynamics. Blocks architecture is used to integrate different programmes dealing with all the aspects of the aircraft flight dynamics. The Python interface between the blocks passes the resulting aerodynamic forces and moments from the CFD code to a structural and flight dynamics integrator code, that compute the structural deformation and state variables at the next step. This closed loop can be used to solve both a steady and unsteady problem. A rigid body, two-dimensional aerofoil free to pitch is considered as test case. The CFD single block is first validated, and the results passing through the Python code are verified. The coupling effect of aerodynamic and flight mechanics is then investigated. The results from Theodorsen thin aerofoil linear formulation, conventional model based on aerodynamic derivatives, and Euler and RANS CFD computation coupled with a flight dynamic solver are compared.

Chapter 6 discusses the introduction of CFD in a routine manner for an optimization process. The tool described in Chapter 3 can be applied to efficiently reduce the required CFD computation to  $\simeq 5\%$  of the total number of entries, to obtain a full order aerodynamic database. The regional jet test case sets an example of a single loop

for the external geometry optimization. A two dimensional aerofoil is first analysed. The angle of attack is computed with an optimization coupled to a CFD solver, to minimize the drag coefficient. The analysis is presented to verify some minimizer routines, and their suitability to the problem. Afterwards a three-dimensional test case is presented to demonstrate the optimization feasibility and to assess the limitations of the aerodynamic derivatives based method. The optimum combination of angle of attack, elevator deflection and speed at fixed altitude is computed to obtain the minimum drag coefficient and satisfy the longitudinal trim constraints. The elevator deflection is obtained via mesh deformation.

Chapter 7 concludes the thesis and offers an outlook and suggestions for a future work.

Appendix A illustrates the capability of the Matlab toolbox for Kriging interpolation DACE. A short overview is given on the effect of the required input over the output interpolation model. Some considerations on the interpolation method for the aerodynamic database generation case are then disputed.

Appendix B discloses the application of the multi predictive control and Q-learning game theory to the problem presented in Chapter 3. A short review and an example are first presented for both the methods. A direct application to the problem is finally performed.

The framework for creating CFD-derived stability and control databases described in Chapter 2 was exercised for several aircraft configurations. The application to the Piaggio Avanti P180 II case led to the publication presented in Appendix C.



## Chapter 2

# Aircraft design tools

Advanced aircraft analysis (AAA) provides a powerful framework to support the iterative and non-unique process of aircraft preliminary design. The AAA program allows design and preliminary design engineers to take an aircraft configuration from early weight sizing through open loop and closed loop dynamic stability and sensitivity analysis, while working within regulatory and cost constraints.

The current version of AAA is based on the methods of Airplane Design Parts I–VIII by Jan Roskam, Airplane Flight Dynamics Parts I–II by Jan Roskam, Airplane Aerodynamics and Performance by Jan Roskam and Eddie Lan and methods developed for airplane design by DARcorporation engineers. Since 1991, when DARcorporation acquired the rights for AAA and continued the development as a commercial venture, AAA has been upgraded several times.

AAA enables a fully functioning three-dimensional aircraft drafting tool Shark/AP [26]. More information about AAA geometry format and description can be found in [27] or on the website <sup>1</sup>.

The SimSAC Project, simulating aircraft stability and control characteristics for use in conceptual design, started with three major tasks: development of design software, validating the software on benchmark tests and applying the software to design exercises. The computerized environment for aircraft synthesis and integrated optimization methods, CEASIOM <sup>2</sup>, is a open-source framework tool that integrates discipline-specific tools for conceptual design. The software is the outcome of a European project, and Dr. Da Ronch has been involved from the conception till the most recent development. New tools and scripts are implemented continuously, making it a open development enviroment and an updated usable software.

Why CEASIOM? The 80% of the life–cycle cost of an aircraft depends on decisions taken during the first steps of the design: the conceptual design phase. A very important issue is the interaction of aerodynamics with structures and controls that are

---

<sup>1</sup><http://www.darcorp.com/Software/AAA/> [retrieved October 10, 2014]

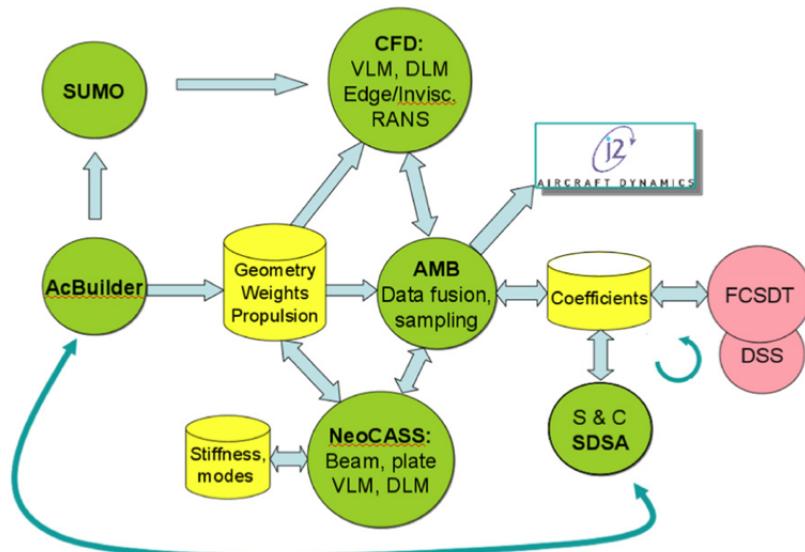
<sup>2</sup>Freely available at <http://www.ceasiom.com/> [retrieved October 10, 2014]

usually analysed with low fidelity methods. For this reason it would be cheaper in term of time and money, to increase the knowledge about stability and control (S&C) as early as possible in the aircraft development process: usually the first aerodynamic models considered are based on semi-empirical methods, and the errors that may appear can be discovered only later with wind tunnel or flight test data.

Furthermore current trends in aircraft design are for unconventional designs with augmented-stability and expanded flight envelopes, and for those the handbook methods are no more valid and it is required an accurate description of the nonlinear flight dynamic behaviour of the aircraft.

CEASIOM aims to support engineers in the conceptual design process of the aircraft, with emphasis on the improved prediction of stability and control properties achieved by higher-fidelity methods. Moreover CEASIOM assimilates in just itself the main design branches: aerodynamics, structures and flight dynamics.

Figure 2.1 represents the CEASIOM working process.



**Figure 2.1:** CEASIOM scheme [3]

In the presented study new implementation are added to the CEASIOM version 100 (v100), upgrading the software to the new version called CPACS CEASIOM.

The CEASIOM v100 geometry defintion was limited by the small number of input parameters. The upgraded version is based on a new design file definition, that allows the user to generate any custom aircraft geometry.

## 2.1 CEASIOM modules

### 2.1.1 Geometry module AcBuilder and SUMO

This is a customized geometry construction system to define the aircraft configuration coupled to surface and volume grid generators. AcBuilder provides basic geometry parameterization first, and then the surface modeller SUMO creates surface and volume grids for Euler CFD simulations.

CEASIOM approach for the mesh generation is to build a customized geometry modeller for aircraft design, which supports export of meshable surface. The "in-house" SUMO path has become the main approach for automatic Euler flow model meshing and for the experiments with automated meshing for Reynolds-averaged Navier-Stokes, RANS, simulations. CFD computations to estimate aerodynamic forces and moments early in the design stage can provide a head start on the controls design. For this strategy to succeed, so that changes in the aircraft configuration can be assessed at acceptable costs, the simulation methods must be fast, reasonably accurate, and easy to use. These three requirements can be addressed by adaptive fidelity CFD. Low order methods are used in the low speed linear region, and higher order solvers in the high speed and non-linear region: this is called adaptive-fidelity CFD and it uses DATCOM, vortex lattice modelling, CFD in inviscid Euler or full RANS.

An important challenge is to approach automatic volume mesh generation for geometries including control surface deflections. It generates a `geo.xml` file that defines the geometry with sufficient details. CEASIOM geometry is easily interrogated for lifting surfaces such as wing, vertical tail, and stabilizer for vortex lattice method, VLM, analysis.

Panel methods and Euler simulations require much higher fidelity geometry, in particular a closed surface, smooth enough to support a surface grid with proper refinements at critical places like leading and trailing wing edges, wing tips, etc. SUMO package builds an aircraft model from a set of closed spline surfaces and provides a proper graphical interface for designing the shapes from cross sections and control points. It refines the CEASIOM model into a more detailed geometry based on a moderate number (often less than 30) spline surfaces. Starting from the spline surface geometry, SUMO generates a closed unstructured triangular surface mesh by a modified version of Chews algorithm. The mesh quality is controlled by local geometric quantities such as element dihedral angles and aspect ratio and it yields to better mesh quality than Delaunay methods for strongly skewed surfaces. Following the surface meshing procedure, the volume between aircraft and farfield boundary is filled by Hang Si's TetGen [28]<sup>3</sup> quality Delaunay tetrahedral mesh generator. The resulting unstructured volume mesh is suitable for inviscid flow solutions as long as the quality of

---

<sup>3</sup><http://wias-berlin.de/software/tetgen/> [retrieved October 10, 2014]

the underlying surface mesh suffices. The accuracy of the numerical solution of the flow problem is affected by the quality of the surface discretization. As most current CFD methods use local low-order polynomial solution representation based on nodal or cell-average values, the surface mesh must approximate the geometric surface sufficiently accurately and resolve inviscid flow features such as pressure peaks and shocks.

When viscous flow models are used the SUMO geometry can be exported to mesh generation software by means of graphics exchange specification, IGES. Whereas the Euler mesh generation is quite robust, RANS meshing is much more challenging [12].

### 2.1.2 Aerodynamic module AMB

It is a replacement of and complement to current handbook aerodynamic methods with new adaptable fidelity modules:

1. Handbook classical approach is exploited by digital data compendium, DATCOM, a United States air force Fortran code that integrates statistical data and empirical methods <sup>4</sup>.
2. Steady and unsteady TORNADO vortex-lattice code (VLM) for low-speed aerodynamics and aero-elasticity <sup>5</sup>.
3. Inviscid CFD code, Edge, for high-speed aerodynamics and aero-elasticity <sup>6</sup>.
4. RANS flow solver for high-fidelity analysis of extreme flight conditions.

In order to compute a good prediction of the S&C behaviour and sizing of the flight control system is important the availability of a complete and accurate aerodata. This is represented by a multidimensional array of dimensionless coefficients of aero-dynamic forces and moments, stored as a function of the state vector  $(\alpha, \beta, V_\infty, p, q, r, \theta)$  and controls vector  $(\Delta T, \delta_e, \delta_F, \delta_a, \delta_r)$ . The total number of components of the aerodata can be some tens of thousands, or even more in late design stages. Data fusion can be exploited to merge results from different sources, low fidelity/cost data indicating trends and some high fidelity/cost simulations correcting the values. The cheap samples are considered to provide information at least about the trend of the target function, whereas the expensive samples give quantitative data. Moreover interpolation methods can significantly reduce the number of data points, which actually need to be computed to fill the table, and it is possible to find the regions where the aerodynamics is nonlinear and use the high fidelity/cost tools to compute only these regions.

In order to obtain a more accurate result with less computational cost various alternatives are available:

---

<sup>4</sup><http://www.pdas.com/DATCOM.html> [retrieved October 10, 2014]

<sup>5</sup><http://www.redhammer.se/tornado/> [retrieved October 10, 2014]

<sup>6</sup><http://www.foi.se/en/Customer--Partners/Projects/Edge1/Edge/> [retrieved October 10, 2014]

1. Data Fusion method: using increment function  $\beta(x)$  it is possible to fuse low fidelity data base with a few from a higher fidelity to estimate.
2. Kriging method: regression and correlation models are used to interpolate and extrapolate a value over the domain in order to have a specific response given few samples.

These methods guarantee a computational cost reduced by around 95% and large flexibility [23].

Stability and control characteristics of aircraft at the edge of the flight envelope is one of the most difficult and expensive aspects of the aircraft development process because of the non linearities and unsteadiness in the flow and CFD represents the state of art in predicting nonlinear flow physics.

Linearized aerodynamic models based on a functional representation for the indicial aerodynamic force and moment responses in terms of blade motion and gust functions can be used in subsonic flow. It was found that these methods are sufficiently accurate to be used as a practical design tool. However, simplifying assumptions pertaining to the flow physics limit the generality of the linear indicial approach. When nonlinear effects are significant or hysteresis exhibit memory-effects the indicial approach becomes inaccurate. A more accurate method would be a nonlinear indicial model to predict time-dependent unsteady aerodynamic loads associated with flight manoeuvres at high angles of attack and high pitch rate. The model extends the usual flight dynamics equations by introducing a first order delay differential equation for an additional internal state variable that accounts for unsteady effects associated with separated and vortical flow. A model with constant coefficient aerodynamic derivatives, retaining a quadratic term in angle of attack and expressing the damping derivative as a function of the angle of attack, can be considered, as well as a second model using a look-up table for static aerodynamics, augmented with alpha-dependent damping derivatives. It was demonstrated that the indicial method was significantly more accurate. Volterra functions have been successfully applied for aeroelastic studies of the limit cycle oscillations. The identification of kernels up to fourth order demonstrated a feasible undertaking and a good agreement compared to the time-accurate CFD solution was achieved.

A computational saving of several orders of magnitude compared to full-order CFD simulations was achieved and, furthermore, CFD can be used to establish the limits of tabular models.

Two criteria to automate the selection of candidate sample points to strengthen and verify the readiness of the surrogate model were implemented. However, they can capture only global nonlinear features. This motivates the need to address further research at the development of a methodology to efficiently identify local minima/maxima and changes in curvature in the aerodynamic loads. The major computational cost is the computation of CFD analyses at points in nonlinear regions of the flight envelope.

Once constructed, the surrogate-based model is used in place of the expensive simulation process to calculate, at a negligible cost, the aerodynamic loads at any flight point. The tabular model is consistent with a nonlinear quasi-steady representation of the aerodynamics and can be used in real-time to fly an aircraft through the database. This gives the opportunity to establish the limitations of the tabular model due to the neglect of time history and unsteady effects [13]. Figure 2.2 represents the AMB available path for the generation of aerodynamic data.

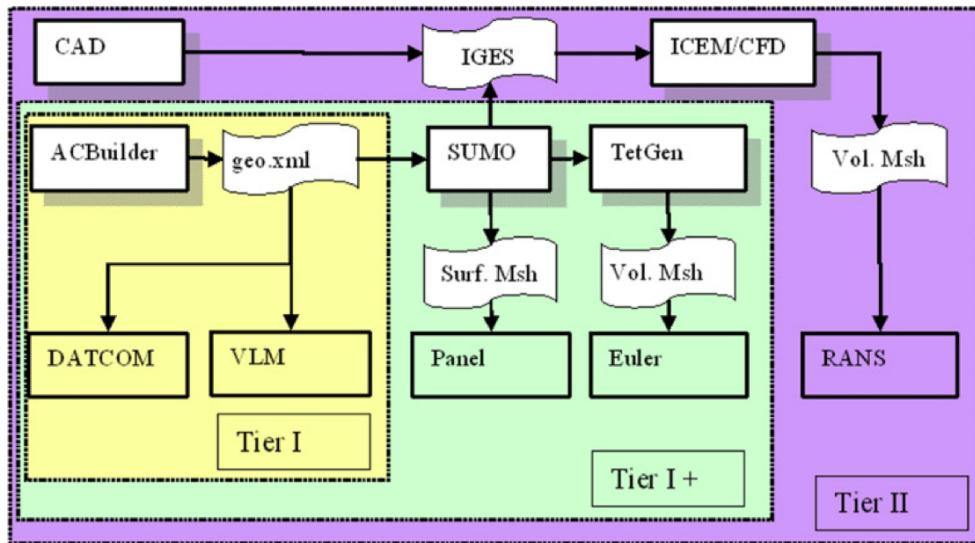


Figure 2.2: AMB scheme [3]

### 2.1.3 Stability and control module SDSA

The SDSA (simulation and dynamic stability analysis) is a tool for analysing the dynamic characteristics (using linear and nonlinear analysis), testing simple flight control systems and computing selected performance characteristics of the aircraft just in the conceptual design stage. It is a flight simulator, dynamic stability and control analyser and flying-quality assessor: six degrees of freedom test flight simulation, performance prediction, including human pilot model, stability augmentation system and a linear quadratic regulator based flight control system (FCS). For eigenvalue analysis, the nonlinear model is linearized numerically by computing the Jacobian matrix at selected equilibrium (trim) point. The solution of the eigenvalue problem gives directly the frequency and damping coefficients. The eigenvectors are also computed to identify the motion modes.

1. The stability analysis gives:
  - (a) eigenvalue analysis of linearized model,
  - (b) time history identification (nonlinear model). Six degrees of freedom flight simulation: test flights, including trim response,
  - (c) turbulence.
2. Flight control system:
  - (a) human pilot model,
  - (b) actuators model,
  - (c) stability augmentation system,
  - (d) FCS based on linear quadratic regulator (LQR) theory.
3. Performance prediction across the flight envelope including  $V_{min}$  and  $V_{max}$ , selected manoeuvres parameters, range and endurance characteristics,
4. Miscellaneous (data review, results review, cross plots, etc.).

The results showed that SDSA is an excellent tool for computing stability characteristics, however, the results generated using SDSA can strongly depend on the quality of the aerodynamic data. The flight simulation applied within SDSA is a powerful tool to investigate the dynamic characteristics of designed aircraft. It supports and complements linear analysis. However, obtaining the precise values of period and damping coefficients from simulated flight (and from recorded real flight as well) for typical modes of motion, is sometimes difficult, in opposite to eigenvalue analysis. The principal advantage of flight simulation is that it can go beyond the area of linear analysis and allows simulating effects not available from eigenvalues (i.e. adverse yaw) [15].

#### 2.1.4 Aeroelastic module NeoCASS

The NeoCASS (Next generation aero structural sizing suite) module integrates the state of the art geometry construction, aerodynamic and structural analysis codes that combine depictive, computational, analytical, and semi-empirical methods, validated in an aircraft design environment, in order to tackle the various aspects of the aerostructural analysis of a design layout at the conceptual design stage.

The preliminary analysis is focused on determining a reasonable structural/non-structural mass and stiffness distribution. To introduce geometry nonlinear effect a linear equivalent plate and a linear/nonlinear equivalent beam are used. These models lead to low-order algebraic system, medium fidelity models particularly suitable for structural sizing, aeroelastic analysis and optimization at the conceptual design level, maintaining the computational cost low and allowing a quick study of various configurations. This aims at including the airframe and its effect from the very beginning of the conceptual design. In most cases, very simplified formulas and datasheets are adopted, which implies a low level of detail and a poor accuracy. Through NeoCASS, a preliminary distribution of stiffness and inertias can be determined, given the initial layout. The adoption of empirical formulas is reduced to the minimum in favour of simple numerical methods.

In this module two classic lifting surface methods are implemented: the vortex lattice method (VLM) is used for subsonic steady aero-dynamic and aeroelastic calculations, and the doublet lattice method (DLM) for subsonic flutter analysis and prediction. For higher fidelity and higher Mach number CEASIOM may use the inviscid version of the CFD code Edge [4]. Figure 2.3 shows the NeoCASS main steps.

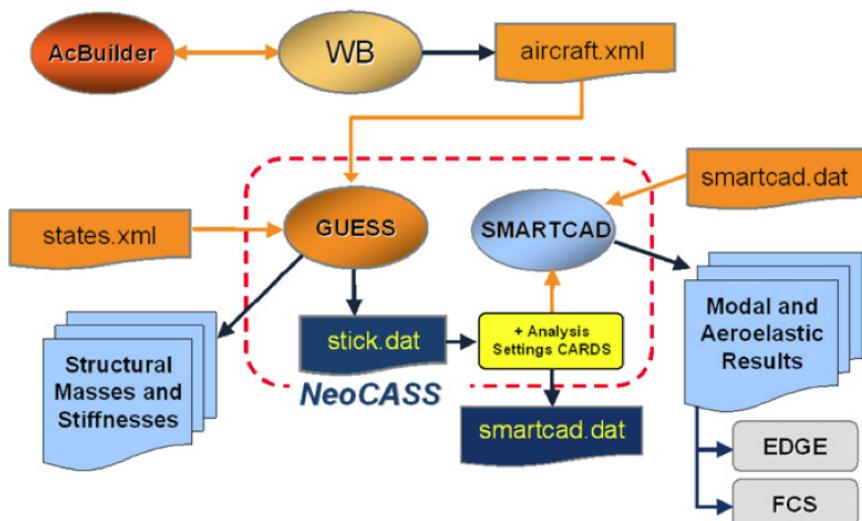


Figure 2.3: NeoCASS layout [4]

Generic unknowns estimator in structural sizing (GUESS) module is a compromise between empirical methods and detailed but time-consuming finite-element analysis for the weight estimation. Two different approaches can be used: one creates detailed analysis models in correspondence of few critical locations and extrapolates the results to the entire aircraft but it can be misleading because of the great variety of structural, load and geometric characteristics. The second approach aims instead to creating a coarse model of the aircraft, but this scheme may miss key loading and stress concentrations. Two different modes are present in GUESS:

1. Standard Mode (GUESS SM) when predefined basic manoeuvres are used.
2. Modified Mode (GUESS MM) when the designer can specify a generic manoeuvre.

Once the loads are defined by GUESS, the sizing is performed for each station under the constraints of ultimate compressive and tensile strength, local and global buckling and minimum gage. Principles of minimum weight are used such that, given an applied load and the limitations on the outside dimensions, the most efficient type of construction, with geometry and material are determined. GUESS automatically generates a stick beam model for the simplified aeroelasticity models, SMARTCAD. About the aerodynamic mesh created, it can be used for classic lifting surface panel methods such as the VLM. The aerodynamic shape is represented with trapezoidal surfaces and control surfaces are considered only for the aerodynamic contribution.

SMARTCAD is the module dedicated to the aero-structural analysis. It can perform:

- static analysis, linear buckling;
- vibration modes calculations;
- linearized flutter analysis;
- linear/nonlinear static aeroelastic analysis, trimmed calculation for a free-flying rigid or deformable aircraft;
- steady and unsteady aerodynamic analysis to extract derivatives;
- structural optimization.

A dedicated multi-disciplinary optimization (MDO) tool is also available in NeoCASS, so that the initial structural sizing can be efficiently refined in order to satisfy the aeroelastic constraints. The structural parameters are the only design variables used during this process (geometry is fixed).

Once vibration modes are available from NeoCASS, the CFD code allows the creation of a reduced order model (ROM) for the generalized aerodynamic forces. The ROM can then be used by NeoCASS to correct aerodynamic derivatives with deformability effects or to assess flutter run solving for a non-canonical eigenvalues problem. The modal model can also be used by Edge to determine a free-flight trimmed condition for the deformable aircraft [4].

### 2.1.5 Test cases

Two very different aircraft were considered: a conventional T-tail based on the existing Eclipse Aviation EA500 very light jet and the second, a novel Z-wing configuration known as the GAV or general aviation vehicle. The first aircraft serves as a baseline comparison for the second, and the cruise case is considered as a benchmark for identifying potential drag reductions and aircraft stability characteristics. CEASIOM allows linear analysis and control system design to be used as tools at a very early stage and offer the potential for design optimisation, and the ability to be able to consider highly novel aircraft concepts, creating a full 6 degree of freedom aircraft model based on simple initial design concepts.

Analysis of the EA500 has shown that the weights and balance and aerodynamic predictions from CEASIOM are sufficiently accurate to provide useful feedback on the response of classical aircraft concepts. Sample results have also been given for the aircraft modal analysis. About the novel asymmetric aircraft configuration the results indicate that the proposed performance benefits have not been realised but it demonstrated the capability that CEASIOM has for modelling asymmetric aircraft. It is possible if this initial GAV design were revisited that the potential performance benefits could be realised. Linear analysis of the asymmetric GAV demonstrated coupling between the longitudinal and lateral-directional aircraft dynamics. This is clearly highly undesirable and combined with the associated positive real eigenvalues required the application of a stability augmentation system. The CEASIOM environment has therefore been shown to be not only flexible enough to deal with highly unconventional asymmetric aircraft, but results have been shown that with these tools it is now possible to bring stability analysis and control system design earlier into the aircraft conceptual design process [29].

Furthermore the flying qualities predicted with CEASIOM are compared with the available flight-test data of the Boeing B747 aircraft in order to verify the goodness of the overall approach in [19].

## 2.2 Limitation of data handling

The file format used inside the CEASIOM environment is based on the eXtensible Markup Language (.xml extension). The file tags have been defined a priori and every module can read and update any of the fields. The communication between the modules is guaranteed by the usage of the same file format. The CEASIOM environment operates inside Matlab<sup>©</sup> and so the xml file structure is equivalent to the variable structure under which all the data are stored. The xml file is loaded any time a module is started and saved whenever required by the user. The xml file scheme of the old version of CEASIOM (from now on called CEASIOM 100) was defined a priori by the developers and the tree reflects the main disciplines in the aircraft design process. An example of the old file scheme tree is presented in Fig. 2.4. This file scheme presents a limited

Parameter	Unit	Value
<b>area</b>	$m^2$	155.0
<b>Span</b>	m	34.1
<b>AR</b>		<b>7.502</b>
<b>spanwise_kink1</b>	[0-1]	0.3414
<b>spanwise_kink2</b>	[0-1]	0.75
<b>taper_kink1</b>	[0-1]	0.58
<b>taper_kink2</b>	[0-1]	0.32
<b>taper_tip</b>	[0-1]	0.176
<b>root_incidence</b>	deg.	0.6605
<b>Kink1_incidence</b>	deg.	0.1152
<b>Kink2_incidence</b>	deg.	-1.193
<b>tip_incidence</b>	deg.	-1.193
<b>LE_sweep_inboard</b>	deg.	32.0
<b>LE_sweep_midboard</b>	deg.	32.0
<b>LE_sweep_outboard</b>	deg.	32.0
<b>dihedral_inboard</b>	deg.	5.681
<b>dihedral_midboard</b>	deg.	0.294
<b>dihedral_outboard</b>	deg.	0.294
<b>airfoilRoot</b>	name	0012
<b>airfoilKink1</b>	name	0012
<b>airfoilKink2</b>	name	0012

**Figure 2.4:** Old xml CEASIOM file parameters for the wing

number of previously determined parameters to define the external geometry. The aim of such simplification was the ability to define a complete aircraft geometrical configuration with only around 100 parameters, which is consistent with the preliminary aircraft design definition. The required values to define the whole aircraft geometry are easily understandable bulks for the designer; since they are traditionally used in the aircraft design books. This allows having an immediate understanding of the aircraft design from the xml file. Unfortunately this simplification leads to the inability to define more complex and precise geometries. Unconventional configurations, e.g. a strut-braced wing aircraft presented in [30], are almost impossible to define since the file scheme does not have the ability to adapt to custom user input. AcBuilder, that is the module for the geometry generation, always requires the definition of one fuselage, one main wing and a tail or canard. Furthermore the component geometry is identified by the definition of a limited number of sections, so that a smooth continuously changing number of parameters is impossible to define. For more flexibility and capability during the geometry definition and the benefit of communication with other aircraft design tools,

the new common parametric aircraft configuration schema, CPACS, is introduced as reference file scheme for the CEASIOM environment.

## 2.3 CPACS file scheme

The common parametric aircraft configuration schema (CPACS) is a data definition for the air transportation system [31]. CPACS enables engineers to use the same file scheme to exchange information between different tools. CPACS can describe the characteristics of aircraft, rotorcraft, engines, climate impact, fleets and mission in a structured, hierarchical manner <sup>7</sup>. This new file scheme guarantees a higher flexibility for unconventional configurations generation and the total control on the external geometry generation. The aircraft components shape is assessed by the definition of various components sections; the base shape, three rotations and three scaling factors for the main axis direction characterize each of them. The number of sections can be increased depending on the design requirements and the needs. The file format does not present limitations about the geometry generation, and although wings (including any type of lifting surface) and fuselage are explicitly defined, any geometry is potentially definable. The freedom about the geometry definition subsequently causes the loss of meaning of the numerical values. The main traditional values in the aircraft design are not directly identifiable but have to be computed, e.g. wingspan computed as sum of distances between all the wing sections. Anyway, the reference surface and lengths are collected externally of the geometry definition, so that they can be found easily.

The main structure is composed by:

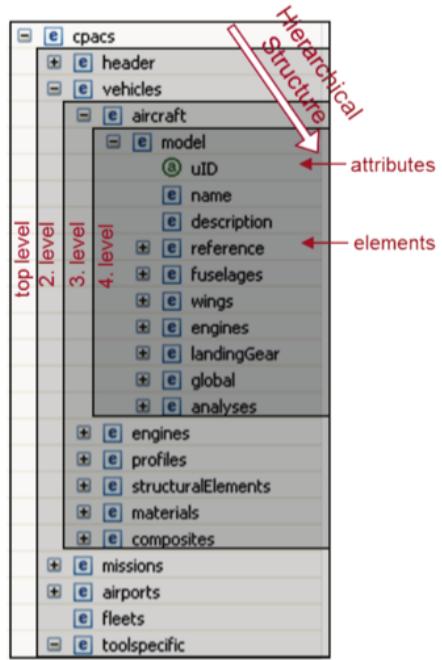
- airports
- fleets
- header
- missions
- toolspecific
- vehicles.

Figure 2.5 shows the level characterization by attributes and elements and the wing element definition.

Figure 2.6 clearly shows the simplifications of the external geometry that were adopted for the geometry definition by the CEASIOM 100. The presented geometry comparison is generated for a traditional airliner configuration, and it is evident how the new file scheme CPACS permits more accurate, and so smoother, external surfaces.

---

<sup>7</sup><https://code.google.com/p/cpacs/> [retrieved October 10, 2014]

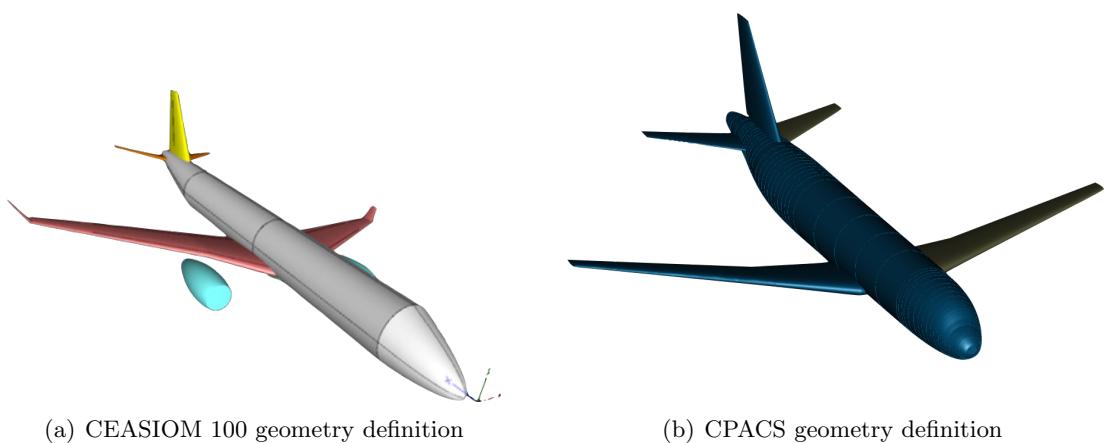


(a) CPACS scheme main structure

▪ [e] wing
⑧ symmetry
⑧ uID
[e] name
[e] parentUID
[e] description
▷ [e] transformation
▷ [e] sections
▷ [e] positionings
▷ [e] segments
▷ [e] componentSegments
▷ [e] componentSegment
⑧ uID
[e] name
[e] fromElementUID
[e] toElementUID
▷ [e] structure
▷ [e] controlSurfaces
▷ [e] wingFuselageAttachments
▷ [e] wingFuelTanks

(b) CPACS wing element definition

**Figure 2.5:** CPACS.xml file scheme, main structure and wing definition



(a) CEASIOM 100 geometry definition

(b) CPACS geometry definition

**Figure 2.6:** Geometry definition differences within the old and new CEASIOM

## 2.4 CEASIM upgrade

The author upgraded the AMB CEASIM module for the generation of aerodynamic models. In the following sections, the implementation to upgrade to CPACS design definition scheme work is presented and validated. Furthermore, in Chapter 3, two methods for a more efficient reduce order model generation are developed and implemented in AMB. Other developers work is extensively presented in Jungo [30] and Martinez [32] master thesis. The recent development steps are summarized in the conference paper [33].

### 2.4.1 DATCOM for CPACS

#### 2.4.1.1 Introduction

DATCOM (Digital compendium) is a Fortran code that implements the methods contained in the United States air force stability and control to calculate the static stability, control and dynamic derivative characteristics of fixed-wing aircraft. It requires an input file containing a geometric description of an aircraft and gives as outputs the stability derivatives according to the specified flight conditions. The generated aerodynamic model is based on statistical and semi-analytical assumptions.

The McDonnell Douglas Corporation under contract with the United States air force initially developed it in 1976, but it is now open source.

The DATCOM+ package is used in this project. This is a package developed by Holy Cows Inc.<sup>8</sup> that allows the user additional functions. The input file can be named differently than `for005.dat` and can contain comments. Furthermore it generates various additional outputs to generate a graphical representation of the aircraft in `ac3dview`, a data table in XML format, as well as a free-format linear function interpolation data table file that can be used to plot the derivatives as function of the different parameters. Although DATCOM is almost 40 years old software, it is still extensively used in the aircraft conceptual design process to compute preliminary data about the static stability and handling qualities for a standard configuration aircraft. The reasons of this success are the very few input parameters required (in order of tens) and the short computational time to have a full preliminary outlook of the aerodynamic tables. The input file is however arcane and difficult to write, so an interface between the project file format and DATCOM is usually required.

This work is about generating the interface between the CPACS file scheme and DATCOM in the Matlab<sup>©</sup> environment. The output file is then read and converted in a Matlab<sup>©</sup> variable ready to be used for further analysis, e.g. data sampling with others results. The developed tool is then implemented within the CEASIM environment.

---

<sup>8</sup><http://www.holycows.net/DATCOM/> [retrieved October 10, 2014]

#### 2.4.1.2 DATCOM limitations

DATCOM limitations are first shortly presented [5]:

- Inlets, external stores, and other protuberances cannot be input. The simplification affects the drag coefficient of the aircraft.
- Dynamic derivatives are not an output for aircraft that have wings that are not straight-tapered or have leading edge extensions.
- There is no method to input twin vertical tails mounted on the fuselage, although there is a method for H-Tails. This problem can be addressed by approximating the twin vertical tails as a single equivalent vertical tail mounted to the fuselage.
- Digital DATCOM cannot provide outputs for the control derivatives with regard to the rudder control surface.
- It cannot analyse three lifting surfaces at once, such as a canard-wing-horizontal tail configuration.
- It cannot analyse more than one control surface at once and it always consider it as part of the most after lifting surface. For this reason the horizontal tail has to be removed for flap or aileron analysis.
- Control surfaces dimensions are not taken as input, and the corresponding derivatives are evaluated with empirical statistical-based methods.

Hence, the limitations summarized above restrict DATCOM validity to conventional aircraft configurations and a multi-fidelity aerodynamic model is required to design and explore unconventional configurations.

#### 2.4.1.3 Software description

The target of this work is to create an interface that enables the user to run a DATCOM analyses starting from a `aircraft.xml` file formatted in the CPACS schema. The developed code is based on the DATCOM script used in the AMB module of the CEASIOM v100 and on the `CPACSwrapper` function developed by Till Pfeiffer. A standalone version is developed and then integrated in the upgraded CEASIOM version.

From now on this *font* is used to identify computer folders, *italics font* for Matlab<sup>©</sup> variables and functions and the *type font* for computer files.

The main function read the user defined state variables (defined through AMB in CEASIOM and given as external input in the standalone mode) and the `CPACS.xml` file and gives back two outputs: a `datcomoutput.txt` file that is the original DATCOM output and the `datcomresults.mat` that is a Matlab<sup>©</sup> variable and contains tabulated states and aerodynamic forces and moments coefficients, both in the *Output* folder. The version integrated in CEASIOM stores the results for further analyses. The following list describes shortly the folder structure:

<b>Main.m</b>	
<i>Input</i>	<code>-state.mat</code>
	<code>-CPACS.xml</code>
<i>Output</i>	<code>-datcomoutput.txt</code>
	<code>-datcomresults.mat</code>
<i>lib</i>	<code>-functions ... .m</code>

Depending on the project data availability, the user can choose the test case to analyse between a no control analyses, or one with ailerons, elevator or flaps. It must be considered that all the analyses contain basic analyses too. The `datcomresults.mat` output file is generated in order to carry out more analyses, e.g. data sampling with a proper CFD method.

The structure of the code will be now presented to give a full outlook of the proceeding path. Figure 2.7 presents the functions flowchart, in order to have a visualization of all the steps that are performed.

The main script works using the global structure variable called *acproject* in which all the required parameters are written. The working path needs to call the following function:

- *CPACS wrapper*

The input is the `CPACS.xml` file, while the outputs are *acproject* Matlab<sup>©</sup> variable, in which some geometry common parameters are computed from the input file, and `CPACSxml` variable that is exactly the input file converted in a Matlab<sup>©</sup> structure.

- *wrapper DATCOM*

This is the core of the code because it reads the *acproject* as input and deduce all the DATCOM required variables, writing then in *acproject.AMB.datgeo* sub-structure.

- *load state.mat*

It loads the flight states input conditions and writes them in *acproject.AMB.state* sub-structure.

- *read flap elevator rudder and ailerons type*

In this script the type of flap, rudder, elevator and aileron is read, converted for DATCOM and written in *acproject.AMB.datgeo* sub-structure.

- *AcView input file creator*

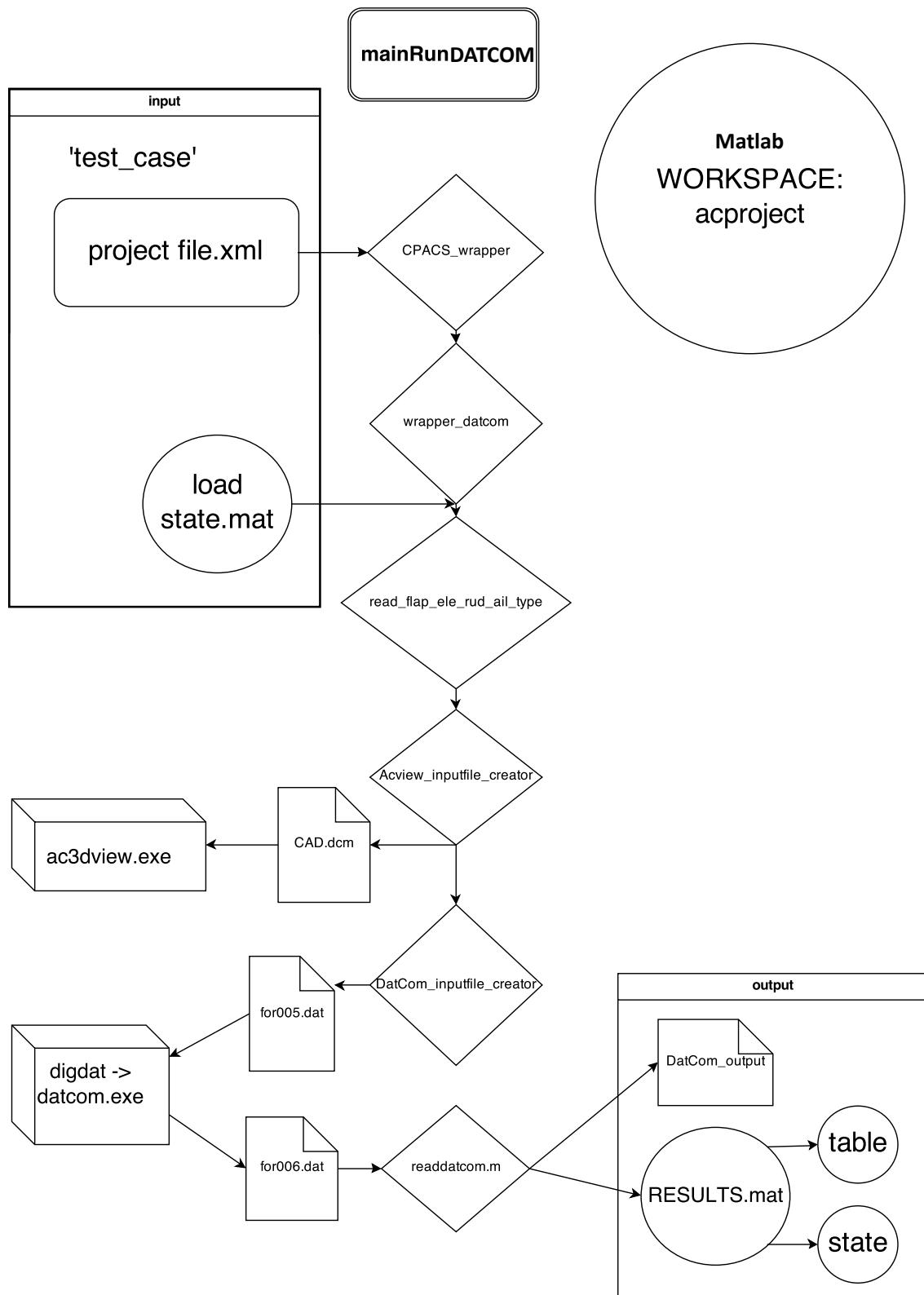
This script write the `CAD.dcm` text file in DATCOM subfolder with the required data. Just after the batch file is execute in the command prompt that gives back some interesting output files as `CAD.out`, `CAD.xml`, `CADAero.xml`, `CAD.lfi`, `CAD.csv` and `CAD.ac` that is then opened with *ac3dview* executable that gives back a visualization of the aircraft.

- *DATCOM input file creator*

This script write the `for005.txt` text file in DATCOM subfolder that contains all the required data for the selected test case. Then the original DATCOM runs, executing `digdat` batch file, and it gives as output a `for006.txt` text file.

- *read DATCOM*

It opens the file containing the results and reads all the output values saving them in `acproject.AMB.datcomrun.table` sub-structure. Finally the resulting aerodynamic tables Matlab<sup>©</sup> variable and the output `for006.txt` are moved into the output folder.



**Figure 2.7:** DATCOM for CPACS script structure

#### 2.4.1.4 Differences with v100 and unsolved issues

The main differences of this code with respect to DATCOM of CEASIOM v100 are presented in the following table:

<input checked="" type="checkbox"/>	CPACS compatibility added
<input checked="" type="checkbox"/>	no more aerofoil information needed (already in <code>CPACS.xml</code> )
<input checked="" type="checkbox"/>	simplification of the analyses steps (see flowchart Fig.2.7)
<input checked="" type="checkbox"/>	possibility to choose between analyses type with no need to compute all of them
<input checked="" type="checkbox"/>	log file introduced

The founded unsolved issues are then shortly described:

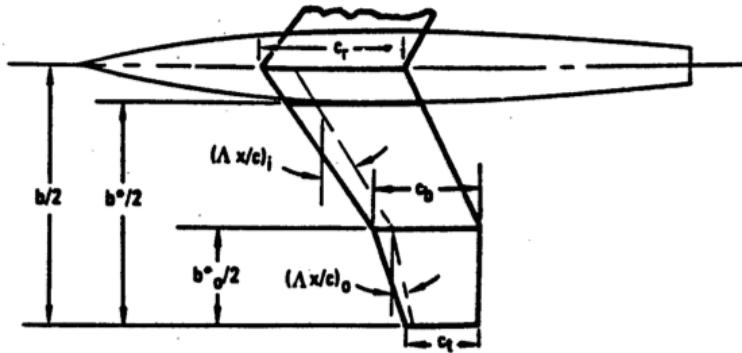
<input type="checkbox"/>	the visualization of the vertical tail doesn't match with fuselage (adding upper fuselage coordinate the problem does not solve)
<input type="checkbox"/>	PHETE parameter is unknown and, as in CEASIOM, it is taken from the manual examples PHETE=0.05288
<input type="checkbox"/>	CPACS has no information about control surface type and so a function had to be introduced
<input type="checkbox"/>	no ac3dview Linux compatibility
<input type="checkbox"/>	no winglets implemented because no CPACS example available

#### 2.4.1.5 Wrapper DATCOM

The core of the entire programme is the wrapper. It uses the results about the geometry from *CPACSwrapper* and computes the required DATCOM input variables. A short overview of the various translation will be now displayed:

- The reference wing surface, reference chord length and reference span are explicitly defined in the wrapper output *acproject*.
- The center of gravity location for the moment reference point, is taken to be at the maximum take-off mass location, if not present is taken at main reference frame origin (0,0,0) that correspond with the fuselage nose.
- Maximum 20 longitudinal body stations are extracted, for which position and section radius is specified.
- Since in CPACS main wing, horizontal tail and vertical tail are all stored as "wings", a method to distinguish between them had to be created. In order to do so some hypothesis were made: the main wing is considered having the longest

span, the vertical tail is the only one not symmetric with reference to the plane  $x$   $y$  and so the remaining one is the horizontal tail. Figure 2.8 shows some DATCOM wing input parameters. The following data were then extracted:



**Figure 2.8:** DATCOM wing parameter definition [5]

– the root section is the one with reference frame origin in  $y = 0$ , so it was possible to find:

- \* location of root wing apex
- \* root chord aerofoil rotation with respect to the horizontal plane  $\alpha_r$  (with reference to the reference plane)

$$\alpha_r = \arcsin \left( \frac{T_{1,3}}{c_r} \right)$$

where  $T_{1,3}$  is the first line, third columns element of the root section transformation matrix  $\mathbf{T}$

- \* root chord  $c_r$  (considering the scale factor for the aerofoil)

– tip section is found as the one whose reference frame origin has the largest  $y$  (span), so it was possible find:

- \* tip chord  $c_t$  (considering the scale factor for the aerofoil)
- \* semi-span theoretical panel from theoretical root chord  $b/2$  (reference frame  $y$  position)
- \* semi-span of exposed panel  $b'/2$  (semi-span theoretical minus the corresponding fuselage radius)
- \* sweep angle at the quarter chord of the inboard panel

$$\Lambda = \arctan \left( \frac{(x_t + c_t/4) - (x_r + c_r/4)}{b} \right)$$

- \* dihedral angle of inboard panel

$$\Gamma = \arctan \left( \frac{z_t - z_r}{b} \right)$$

- \* twist angle, negative leading edge rotated down

$$\beta_t = \arcsin \left( \frac{T_{1,3}}{c_r} \right) - \alpha_r$$

where  $\mathbf{T}$  is the transformation matrix of the tip section

- \* unitary chord aerofoil points, no more than 50 values.  $x$  starting from 0 end ending with 1 and both the upper and lower  $y$  starting and ending with 0.

- Then it is necessary to find which control surfaces are ailerons and which are flaps. Because of the impossibility to know a priori the kind of control surface, very restrictive hypothesis were made to obtain an automated process. Ailerons were defined as the outer trailing edge control surfaces and the flaps as the inner. For both of them and for the horizontal tail elevator the following parameters were computed:

- Inner and outer span position.
- Chord length at inner and outer positions.

#### 2.4.2 Edge

Edge is a CFD solver for unstructured grids with arbitrary elements developed by the Swedish defence agency, FOI<sup>9</sup>. CEASIOM includes the executable files of Edge for non viscous flows using one processor computations, and the AMB module permits to compute the aerodynamic tables with this CFD solver, generating a physics based aerodynamic model of the aircraft. CEASIOM in the Matlab<sup>©</sup> enviroment just generates the required input file (`.ainp`), requiring already existing mesh (`.bmesh`) and boundary conditions (`.aboc`) files. In case of deformed control surface analyses, the deflection is defined by choosing the deflected surface, and axis and magnitude of the rotation (`.amot`). The deflection is not really created but the effect is simulated by transpiration boundary conditions. For further information about the CFD solver Edge please see the user guide [34] and the theoretical guides [35, 36].

The AMB module inside CEASIOM takes advantage of this CFD solver by permitting the user to run CFD computations from the Matlab<sup>©</sup> enviroment. The five options available to run Edge are presented in the following Table:

---

<sup>9</sup>A limited version of the program is freely available online at <http://www.foi.se/en/Customer--Partners/Projects/Edge1/Edge/> [retrieved October 10, 2014]

<input type="checkbox"/>	Single simulation
<input type="checkbox"/>	Brute-force simulations combining some of the states and control variables
<input type="checkbox"/>	Read pre-defined state-control input file
<input type="checkbox"/>	Kriging interpolation model, with defined number of simulations sampled on the maximum predicted error (standard)
<input type="checkbox"/>	Newly implemented Kriging interpolation model, with defined number of simulations, sampled on the most significant predicted state, see Chapter 3 (cognitive).

These options allow the user to run as many CFD simulations as required autonomously, with no need of further interventions. Since the flow solver is available with executable files, the simulations are run by use of some bash files that are called inside Matlab<sup>©</sup>.

Edge for AMB requires already existing mesh and boundary conditions files, which must be previously generated. A tool for mesh generation, SUMO, is integrated with both the new and the old CEASIOM, so that the generation of a mesh can be easily obtained from the geometry defined in the project .xml file. Such simplification allows the user to easily modify the design geometry in CEASIOM and compute a new mesh with no need of additional work. Anyway the user can generate the unstructured mesh with any external tools, but it needs more time since the geometry CAD file is generally required.

Since the geometry is taken as external input from the mesh, the translation of the old CEASIOM script in the new CPACS based version did not required much programming effort. There is no need of reading the CPACS file, since all the required information to start the simulations are given as user input in the AMB module. Shortly the Edge sub-module:

- asks the required simulation type (e.g. single, full table, Kriging, etc.)
- reads the states and controls values as input or from the table
- requires to individuate mesh and boundary conditions files
- writes the input file
- starts all the required CFD simulations serially

## 2.5 Validation

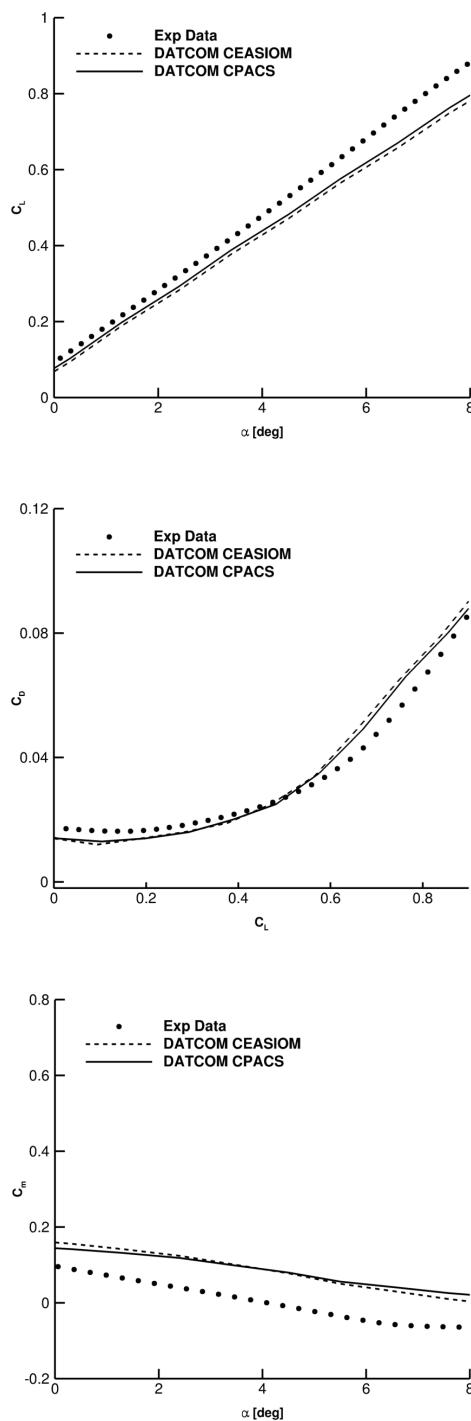
### 2.5.1 DATCOM

In order to validate the new DATCOM code a test case was taken from a previous study about a B747 (Fig. 2.9) with the old CEASIOM environment [19]. In that analyses various aerodynamic models were used but for this validation only the DATCOM and experimental data are taken into account. The CPACS project file was generated



**Figure 2.9:** Boeing 747-8 real model

by wrapping it from a SUMO project. The project file needed some improvement to make it more accurate: the aerofoil coordinates were increased and reference lengths and center of gravity location were introduced. The three most common trends that characterize the aerodynamic effect on an aircraft are showed. Figure 2.10 shows the lift compared with the angle of attack, the polar curve and the pitching moment computed respect to the same estimated center of gravity location trend changing the angle of attack. The CPACS version of DATCOM is really similar to the old CEASIOM results. Furthermore both of them are rather similar to the experimental data extracted from [19], assessing the good quality of DATCOM results for traditional configurations and flight conditions. The computational time is a few seconds for computing the entire flight envelope, considering only the longitudinal plane, and so it is a very efficient aerodynamic model generation tool.



**Figure 2.10:** Lift, polar and pitching moment dependence on angle of attack comparison at Mach number 0.75 and altitude of 10,000 m

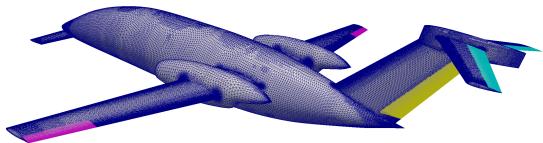
### 2.5.2 Edge

The validation of the CFD solver Edge implementation inside AMB in the new CEASIOM version is less prone to errors compared to previously presented work for DATCOM. The flow solution is dependent only on the input, volume mesh, and boundary conditions files. Using the same input files leads to the same output solution. Once the input files are generated, the flow solver is started via bash file outside Matlab<sup>©</sup> and so it is no more dependent on the Matlab<sup>©</sup> scripts. The test case used in this section is the propeller business jet Piaggio Avanti P180 II (Fig. 2.11).



**Figure 2.11:** Piaggio P180 II real model

The mesh is generated with the commercial tool Pointwise<sup>©</sup>. First a surface grid is created from the geometrical model, separating the control surfaces in order to deflect them and so carry out controllability analyses as showed in Fig. 2.12. Then a volume mesh is generated filling the volume between the aircraft surface and the farfield. The current study only considers longitudinally symmetric states and so the volume mesh is generated only on half model, imposing the longitudinal plane as symmetry surface. The overall mesh for non-viscous flows symmetric with respect the aircraft longitudinal symmetry plane contains 232,334 nodes.



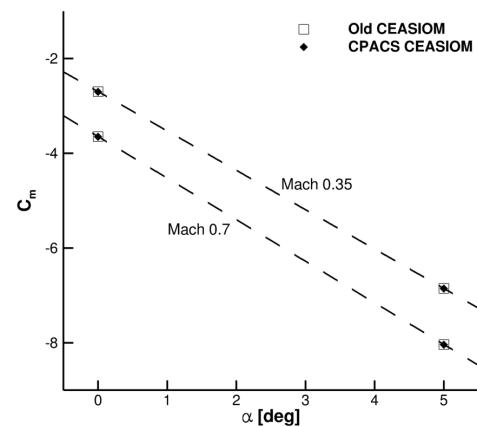
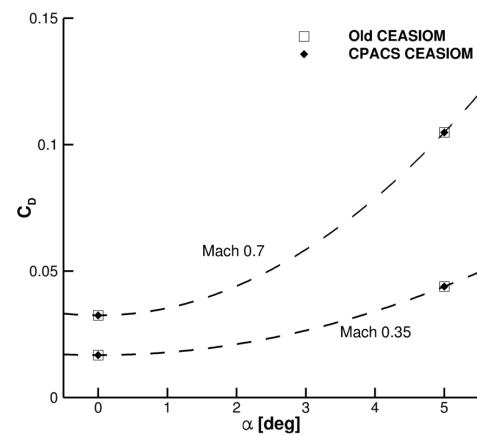
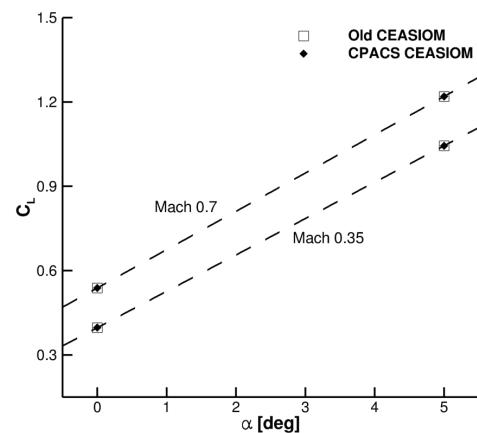
**Figure 2.12:** Piaggio P180 II surface mesh, highlighting the control surfaces

The blade effect is not modelled and the same volume mesh and boundary conditions are used for both the CEASIOM versions. Since Euler simulations are run, slip boundary conditions are imposed on the surface. This study does not aim to validate the mesh generation from the geometry with the two CEASIOM versions and so it con-

siders the exact same mesh as input. The CPACS geometrical model might be more precise compared to the old CEASIOM one, but this is not of interest of this analyses since the mesh can be created with any external tools starting from a CAD design file (avoiding the CEASIOM geometry shaping simplifications, that are imported with the internal tool SUMO).

Figure 2.13 presents the dependence from angle of attack of lift, drag and pitching moment, computed in the frame of reference origin located at the aircraft nose. The non-viscous CFD solutions are obtained for 4 flight envelope points: at  $\alpha = 0$  and 5 deg and for Mach numbers of 0.35 and 0.7 with the same altitude of 10,000 m. The rough expected trends are obtained through polynomial fitting of the two samples per Mach number, first order for lift and pitching moment and second order for the drag (with horizontal slope at  $\alpha = 0$  deg). Such approximation is used to reduce the overall computational time, since every state requires a few hours computations (single processor), and since the exact same results are obtained, there is no interest in obtaining more results. All the simulations converged to machine accuracy of  $10^{-14}$ . The actual flow field CFD solution samples are showed with the 4 symbols.

The four Euler CFD solutions with the flow solver Edge in single processor mode are computed within both the old and new CEASIOM environments. The obtained results are fully coherent for the two CEASIOM versions (100 and CPACS based), with a negligible relative difference of less than 0.2% for the integrated values of forces and moments coefficients. The very good results fully validate the Edge CFD solver tool integrated in the new CEASIOM version based on CPACS file scheme.



**Figure 2.13:** Lift, drag and pitching moment computed at the nose Edge results comparison; Mach numbers 0.35 and 0.7 at the altitude of 10,000 m

### 2.5.3 Conclusion and further development

The semi-empirical method DATCOM and the non viscous flow solver Edge allow the user to generate different aerodynamic models in the AMB module inside CEASIOM. These models are then given as input to a flight simulator that computes stability and handling qualities. The results from these tools are key factors for obtaining good quality models.

The newly developed DATCOM code, although it might need some further improvement during the testing process, supplies aerodynamic forces and moments coefficients corresponding with the old results. The data obtained in the presented validation for the B747 fully correspond to the CEASIOM 100 version. Other test cases project in both the formats would be however useful to carry further analyses. An important consideration is about the geometrical project definition. The old and new file schemes define different geometrical characteristics, and so if a not perfect coherence exists between the two geometries, a different input file would be created for DATCOM. If the defined geometry is the same, identical output should be generated. Because of the very adaptable but complex CPACS file scheme nature, it is really difficult to extract general data about the aircraft. CPACS, differently to the CEASIOM 100 project format, does not contain the usual geometrical parameters and the evaluation process might lead to some error. This computation is done by *CPACS wrapper* and *wrapper DATCOM* routines, and so the may be the origin of any wrong output. An important issue about the CPACS: a field with the type of control surfaces should be introduced and some space should be dedicated the aerodynamic table output data.

The CFD flow solver Edge integration in the new CEASIOM is more straightforward compared to DATCOM. The flow solver computation is dependent only on external input files (mesh and boundary condition) and some user inputs. The project values are not read at all inside the computation, and only during the post processing the reference lengths and surface are used for generating the non-dimensional aerodynamic coefficients from the output forces and moments. These values are explicitly defined in the CPACS file scheme, and so no further computation is required. The results obtained about the Piaggio Avanti P180 II demonstrate the equivalence of Edge in the CEASIOM 100 and the new CPACS based version.

## Chapter 3

# Surrogate models

The computation of the complete aerodynamic database over the whole flight envelope may be really computationally expensive. A discrete domain contains a lot of points because of the multi-dimensional nature of the problem. Every state and control variable can change independently from the others causing an ever-increasing number of required analyses. The traditional approach to compute every single point of the domain with a CFD simulation may bring to prohibitive computational cost. Every single CFD computation may need several hours, depending on the CFD methods and computation facilities, but if every point needs to be computed this single time increase exponentially with the required resolution, width and number of dimensions of the domain. With this prospective, the full-order aerodynamic model generation for a single case may be really expensive, and any modification in the design geometry requires a fully new aerodynamic database, leading to very high comparison cost for CFD based aerodynamic models.

Considering these aspects an efficient way to produce a high quality full aerodynamic database with a fewer CFD computation is to use interpolation. This would lead to some approximation errors, but the general trend may be acquired. The choice of the samples is then crucial to obtain a high fidelity model. The flight envelope points characterized by linear aerodynamics may be spared because of the good quality of the interpolation for linear trends. When a nonlinearity appears in the flow field, e.g. a shock-wave or a sudden detachment of the boundary layer, the points close this state need a deeper analysis to obtain a better aerodynamic reduced order model. The following section presents some developed methods to try to drive the sampling choice where the nonlinearities appear in the flow field. At every step of the process a Kriging interpolation model is generated via the DACE Matlab<sup>©</sup> toolbox [21].

### 3.1 Cognitive approach for CFD sampling

During the conceptual design phase of an aircraft, the adopted aerodynamic model is usually based on statistical data or strong linear aerodynamic assumptions. These methods may lead to evaluation errors for the aerodynamic characteristics and for the overall aircraft flight dynamics properties. The generation of a full order aerodynamic model with CFD is usually obtained later in the design process, because of the high computational cost and the not yet frozen geometry. The exploitation of physical based models, as CFD, in the first steps of the design would reduce the uncertainty of current methods, leading to a cheaper and faster design process. In order to evaluate with a flight simulator the aircraft stability and handling qualities, a full aerodynamic table is usually necessary. The external aerodynamic forces and moments acting on the aircraft are tabulated for every different combination of state (e.g.  $M, \alpha, \beta, q, \dots$ ) and control surface deflection (e.g.  $\delta_e, \delta_r, \dots$ ). For this reason the aerodynamic table may contain many thousands of entries, for any of which a single CFD simulation should be performed. The CEASIOM aerodynamic table format is presented in Table C.1, dividing between aerodynamic model inputs, on the left, and outputs, on the right. The considered inputs are the traditional state and control variables, and this form of the table may reasonably contain  $\simeq 100,000$  entries.

$\alpha$	$M$	$\beta$	$\delta_{ele}$	$\delta_{rud}$	$\delta_{ail}$	$\dots$	$p$	$q$	$r$	$C_L$	$C_D$	$C_m$	$C_Y$	$C_\ell$	$C_n$
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	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	x	—	—	—	—	—	—	—	x	x	x	x	x	x	x

**Table 3.1:** Structure of the aerodynamic table database constructed in CEASIOM; the x represents that all the values are considered and combined.

Reduced order models may be adopted in order to decrease the computational cost for filling the table. The interpolation of some states allows the evaluation of the whole table with a reduced cost. The analytical formulation of such method is based in interpolating  $m$  functions, representing aerodynamic forces and moments, over a  $n$  dimension domain, consisting of state and control surfaces deflection combination. Any function is interpolated independently from the others, since the knowledge of the link between the different aerodynamic forces is unpredictable a priori (e.g. the stall of a wing causes a sudden trend variation in both lift and pitching moment, but the

correlation between these changes is difficult to predict). The main complexity of this problem is the number of state and control variables that define the size of the domain.

The traditionally used Bryan's theory usually approximates aerodynamic forces and moments trends as linearly dependent on the state and control variables, close to a steady state straight flight condition. This approximation allows good evaluation of the aerodynamic forces and moments for flow fields characterized by linear aerodynamic [37], so only for a narrow flight envelope. When any nonlinearity appears in the flow field, the approximation is no more effective, leading to evaluation errors. The aerodynamic table overcomes such limitations, allowing the user to compute different forces and moments for any point of the flight envelope. Nevertheless the linear approximation for states close to steady state straight flight condition is generally valid and may be exploited to reduce the cost of such points. Although including nonlinear dependence on static parameters, Bryan variants assume a linear dependence on control and dynamic terms, e.g. it seems linearise around a nonlinear equilibrium which rules out capturing nonlinear phenomena during manoeuvres. Interpolation is then essential to keep a low computational cost for the full table generation and the choice of state and control variables for the CFD computations is crucial to obtain well representative interpolation models.

The target of the study presented in this Chapter is to develop a free-toolbox for cognitive sampling. The choice for placing a new sample inside the domain should be dependent on the previously obtained results. First the domain vertices should be computed to avoid extrapolation problems, then some points are sampled via Latin hypercube sampling and finally the points for which the interpolation has the higher error should be analysed. Since the target function is unknown apart from the few computed samples, the interpolation error is unknown. Anyway the functions can be evaluated by interpolation, for which a predicted error is computed as function of the adopted correlation model. This interpolation model and the correlated predicted error are then used to evaluate the best choice for the next function evaluation. This method is called cognitive because it is based on a cognitive approach that learns from the past and predicts the best choice from the until now acquired knowledge.

The generation process of reduced interpolation model for the generation of full aerodynamic table with CFD, may take advantages from these methods. The cognitive method may lead a more efficient approach to the problem, leading to better results considering the same budget. The aerodynamic forces and moments trends are highly nonlinear functions for the intrinsically nonlinear nature of the flow governing equations. Nevertheless for a certain domain interval these functions are well approximated as linear, validating Bryan's theory, and for these points CFD computations results are not necessary since good evaluation can be obtained via interpolation model. As soon as aerodynamic nonlinearities appear in the flow field, the functions trends may become nonlinear, causing errors in the interpolation model. These points need then to be

discovered and then analysed with CFD samples, increasing then the reliability of the interpolation model.

## 3.2 CEASIOM tools availability

In the old CEASIOM versions a reduce order model for the generation of full aerodynamic tables was available coupled with the CFD flow solver Edge, in one processor and not viscous mode. This method is based on the generation of a Kriging interpolation model from a few CFD computations, from which the full table is evaluated. The samples are first positioned at the border of the flight envelope, and then a Kriging interpolation model is generated. The new samples are chosen where the predicted interpolation error is maximum. The new point is then a function only of the samples distribution and the adopted correlation model and it is totally independent from the obtained function evaluation. The chosen point is found just as the farthest from all the other samples. The maximum number of steps, and so of samples for CFD computations, is defined by the user and the process is stopped if a certain tolerance over the predicted interpolation error is reached.

This method permits to obtain a full order aerodynamic model with only a few CFD computation but the samples location are generally not efficiently distributed since they are totally independent from the functions evaluations. The interpolation model may bring high evaluation errors, and nonlinear behaviours may be not deepened enough or not even identified.

## 3.3 Demonstration

### 3.3.1 Methods developed on analytical aerodynamic similar functions

The Kriging interpolation Matlab<sup>©</sup> toolbox DACE is a really robust and efficient tool. Some theoretical basis and the influence of the inputs for the creation of the Kriging interpolation model, is extensively investigated in Appendix A. In this Section it is extensively exploited to generate reduced order model in order to restrict the computational cost of CFD analyses to fill the aerodynamic table. Unfortunately the Kriging interpolation method does not consider the nature of the phenomena and eventual nonlinearities of the flow (e.g. stall, vortex separation or appearance/disappearance of a shock wave) may be not sufficiently analysed. Four analytical functions of the usual and well known aerodynamic trends were created in order to study further methods:

1. A  $C_L-\alpha$  similar function. The initial trend is linear and then a sudden parabolic decrease appears representing the stall effect over the lift.
2. A  $C_D-\alpha$  similar function. Initially constant and then quadratic.

3. A  $C_m - \alpha$  similar function. This represents the experimental results of the transonic cruiser (TCR) developed during the SimSAC (Simulating aircraft stability and control characteristics for use in conceptual design) project [13]. It is formed by an initial and ending longitudinally-static unstable part with  $C_{m\alpha} > 0$  and a central stable part with  $C_{m\alpha} < 0$
4. A non-continuously differentiable function that might represent the effect of a nonlinear perturbation (e.g. high tapered body at high angle of attack might presents asymmetric vortex separation and so the resulting  $C_Y - \alpha$  or  $C_n - \alpha$  may have this trend).

A linear regression model, because of the generally linear behaviour of the main aerodynamic forces in the flight envelope, are adopted in the presented analyses. This should be optimized depending on the analysed function trend and on the engineers experience. The correlation model used is probably the most important decision about the successful of the Kriging interpolation and it will be further discuss later. The regression model parameter is taken as  $\theta = 0.5$  because the weight of the samples over the interpolation should be big enough to consider the samplings points more influent than the least square, see Fig. A.3. This choice causes smaller values of the predicted error close to the sampled points, but it will not be an issue for the developed methods.

The domain and co-domain are both unitary in order to simplify the computations and have an easier comparison between resulting errors values. For operative analysis this should be obtained by normalization. The one dimension domain is then discretised with 20 points, which is a reasonable value for a typical aircraft aerodynamic table. The initialization of the computation is given by the border points (in this case  $[0, 1]$ ) and two inner points given by Latin hypercube sampling function included in the DACE toolbox. The initial points number is really important because the regression model might be fully satisfied and may bring to a zero interpolation predicted error  $mse$  everywhere in the first step. So the constant regression model needs at least 2 initial points, 3 for the linear and 4 for the quadratic. In case of multi-dimensional domain,  $IS$  initial samples number is necessary to start the Kriging computation for  $poly2$  regression model as shown in Eq. (3.1),

$$IS = \sum_{i=1}^{n+1} i + 1 = \frac{(n+1)^2 + (n+1)}{2} + 1 = \frac{n^2 + 3 \cdot n + 4}{2} \quad (3.1)$$

but there are  $2^n$  edges and since Eq. (3.2),

$$2^n \geq \sum_{i=1}^{n+1} i + 1 \quad \forall \quad n \geq 4 \quad (3.2)$$

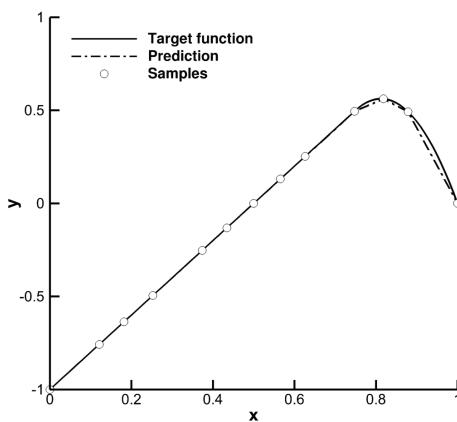
for a  $n > 4$  domain is preferred to place initial samples only at the edges of the domain, to avoid extrapolation as well. For  $n \leq 4$  it is decided to use Latin hypercube sampling (include in DACE toolbox) for the required points. The number of points for which the regression model is surely satisfied and the number of edges are presented in Table 3.2 for any dimension.

	constant	linear	parabolic	edges number
$n = 1$	1	2	3	2
$n = 2$	1	3	6	4
$n = 3$	1	4	10	8
$n = 4$	1	5	15	16
$n$	1	$n + 1$	$\frac{n^2 + 3 \cdot n + 2}{2}$	$2^n$

**Table 3.2:** Number of initial sample points for which the regression model is satisfied and bring to a zero *mse* everywhere

Nevertheless it is always possible that the initial sample points are positioned so that the regression model would find a zero mean square error (*mse*) everywhere (e.g. aligned for `poly1`), so it is advised to force the first iteration to run also if the predicted error is zero.

The given predicted error is dependent only from the correlation model and the samples distribution, so, if the next sample is taken where it is maximum, this may lead to useless computations. Using this method for the lift–similar function, it collects a lot of points in the linear part, almost ignoring the more important similar–stall part (see Fig. 3.1). The predicted function, compared with the target function, gives the real error. The value of this error integrated over the domain is  $4.04 \cdot 10^{-3}$ .

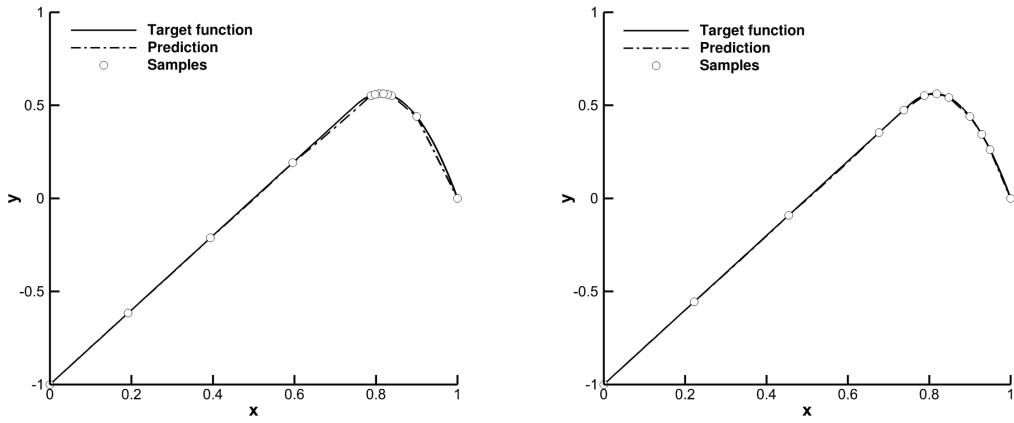


**Figure 3.1:** Kriging *mse*–based sampling for lift similar function at iteration 10

### 3.3.1.1 Local maxima and minima based sampling method

A new approach was then developed trying to optimize this method. It is usually of interest a deeper analysis of the solution in presence of a local maximum or minimum in the solution. This is usually the case if an aerodynamic nonlinearity appears. A better analysis around this state is generally of interest, because of the unpredictability of the trend in the proximity. Considering the interpolating Kriging model a simple first method is then developed: the position of local maxima or minima are computed comparing any function value with all the points within a sphere (segment for 1 dimension domain problems) centered in it, with radius defined by an input value. If a value is bigger or smaller than all the others in the sphere, the point is considered as local maximum or minimum respectively.

The first method places the next sample at the found maximum or minimum location. A second, more efficient method, is then developed. After finding the position of local maxima and minima, the local predicted integration error  $mse$  is augmented and compared with the maximum value of it in the whole domain. A value is used to scale the errors at the maxima or minima positions, and they are compared with the global maximum error. The next sample is obtained as position of the maximum value obtained from the comparison.



(a) Local maxima or minima based method (integrated real error =  $6.81 \cdot 10^{-3}$ ) (b) Maximum of  $mse$  close to local maxima or minima based method (integrated real error =  $2.65 \cdot 10^{-3}$ )

**Figure 3.2:** Maxima or minima based method for lift-similar function at iteration 10

Figure 3.2 presents an example of the solution for the two developed methods based on the location of local maxima and minima (using a division of the domain in 10 and a scale for the error of 100). The integrated real error is computed via integral mean as presented in Eq. (3.3), where  $|D|$  is the domain dimension,  $\|\cdot\|$  represents the absolute

value,  $y(x)$  is the original function and  $\hat{y}(x)$  the interpolation approximated function.

$$\varepsilon_R = \frac{\int_D \|y(x) - \hat{y}(x)\| dx}{|D|} \quad \text{with } x \in D \quad (3.3)$$

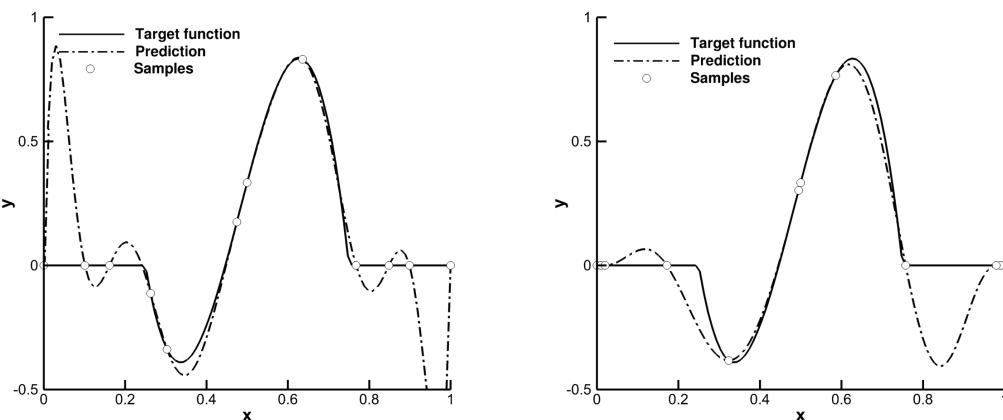
For the fist method the maxima and minima locations are better investigated but the rest of the function is mainly neglected, that can lead to some error far from local maxima or minima. About the second method, all the first steps are close to the stall and then the Kriging error start to be much bigger so that some samples are taken in the linear part. It can be seen that the integral of the actual error decreases.

### 3.3.1.2 Second derivative based sampling method

Unfortunately, the methods previously described are not fully good working for a non-continuously differentiable function. Another method was then developed, that may integrate the previous one: this is based on a second derivative computation with a central difference (second order of accuracy) of the approximated function evaluated from the interpolation:

$$\left( \frac{\partial^2 y}{\partial x^2} \right)_i \simeq \frac{y_{i-1} - 2 \cdot y_i + y_{i+1}}{\Delta x^2} \quad (3.4)$$

This method does not consider the predicted error but decides the next sample point as the one with the biggest second derivative. It permits to find where the biggest curvature is and so the local places where the trend of the unknown changes the most. This method seems to work well mainly if discontinuities are present or it may act as a damper if oscillations caused by the interpolation are generated.



(a) Local maxima or minima based method ( $\varepsilon_R = 1.23 \cdot 10^{-1}$ ) (b) Second derivative based method ( $\varepsilon_R = 8.15 \cdot 10^{-2}$ )

**Figure 3.3:** Comparing two samplings methods for a non-continuously differentiable function with gauss correlation model

Figure 3.3 compares this method with the first local maxima or minima based method, for a perturbation shape function.

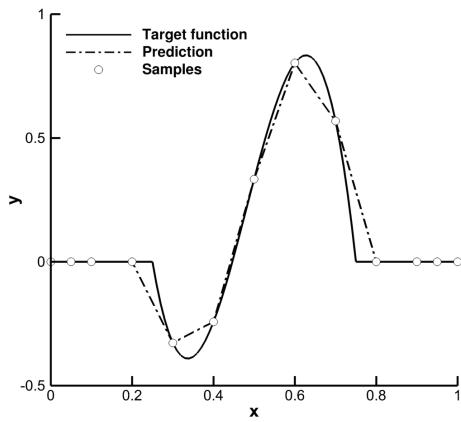
The second derivative based method is not fully working, because the maximum second derivative is close to the already sampled points (see Fig. 3.4). In order to be more efficient another method was developed: it searches for the maximum value of the second derivative over the domain and then it finds the closest maximum *mse* in direction of the consecutive farthest sample point. This method is fully working and giving a very good choice for the sampling choice. The Achilles' heel of the second order derivative based methods are the points with non-continuous first derivative (cusps), in which the central difference returns very high values.

In [22] the second derivative value was used in combination with the interpolation predicted error. The criterion was given by

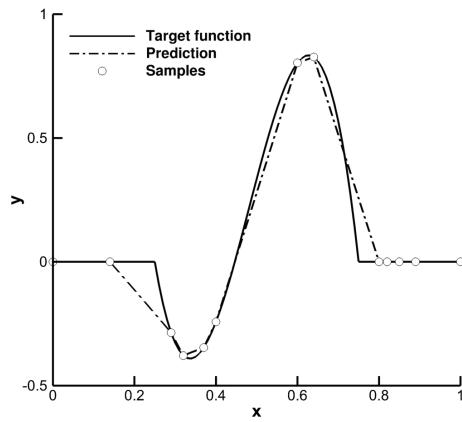
$$C = (|\nabla^2 y| + \epsilon) P_{\phi,\chi}^2(x) \|y_{y,\chi}\|_N \quad (3.5)$$

where  $\epsilon$  is an offset parameter to ensure a nonzero value when  $|\nabla^2 y| = 0$ .  $P_{\phi,\chi}^2(x) \|y_{y,\chi}\|_N$  represents the predicted error at iteration  $N$  (given by the power function  $P_{\phi,\chi}^2(x)$ ). The largest values of  $C$  indicated promising new sample locations, and that formulation was said to achieve a balance between adding points in locations where the data are nonlinear and adding points in unsampled regions of the domain. However this method may be good if the number of available sample is big and the ending sample point concentration is high. In [22] around 100 samples were used, but the target of the present study is to obtain good values for much less samples (a reference of around 10 points per dimension is used). In this case the previous method may present some negative aspects: the sample points decision is carried by a parameter that combines two values that are not related and comparable and at any point there is no real meaning in error and second derivative weights over the  $C$  value. The fact that in a point this value is bigger than the others is empirical, and may bring to good results although that has no mathematical meaning. Instead the methods previously described compute the position of the interesting point (maximum second derivative or local maxima–minima position) and from those points it finds the maximum close error. Then it computes the global maximum error and scales it with a tolerance, comparing it with the value previously found. The last presented function does not mix two different values, but use the first only to decide where to give priority to the research.

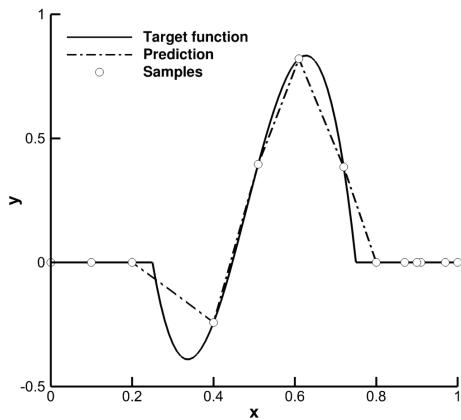
Figure 3.4 shows a comparison between all the methods until here presented.



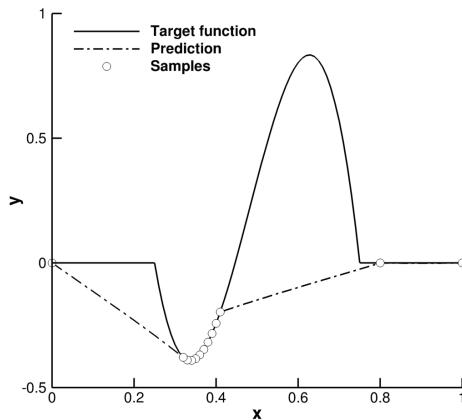
(a) Kriging error based method ( $\varepsilon_R = 3.96 \cdot 10^{-2}$ )



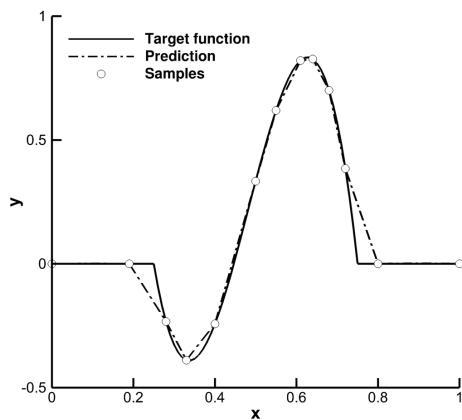
(b) Local maxima or minima based method ( $\varepsilon_R = 3.92 \cdot 10^{-2}$ )



(c) Maximum of  $mse$  close to local maxima or minima based method ( $\varepsilon_R = 5.12 \cdot 10^{-2}$ )



(d) Second derivative based method ( $\varepsilon_R = 2.38 \cdot 10^{-2}$ )

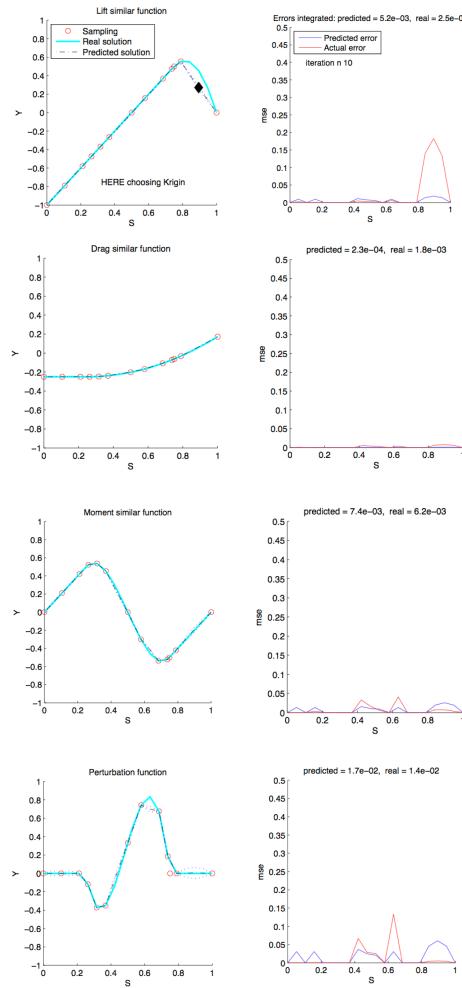


(e) Maximum of  $mse$  close to maximum second derivative based method ( $\varepsilon_R = 2.36 \cdot 10^{-2}$ )

**Figure 3.4:** Comparing the five samplings methods for a non-continuously differentiable function with exponential correlation model

### 3.3.1.3 Priority functions method

In case of multi unknown functions a method was developed in order to give a priority to some function about the choice of the next sample point. In case of a very fast and efficient computation is needed, the user may prefer to focus on the main aerodynamic loads as lift, drag or pitching moment and decide to have a more precise result about them and neglects the others. The result is presented in Fig. 3.5. In this case an exponential correlation model is used, and priority is given to the functions from upward to downward (the priority is given by multiplying the errors by a factor according to the given priority, and then comparing them to find the biggest one). During the process steps the script first decide to sample according to the lift–similar function, and then moves to the others when the predicted interpolation error is reduced.



**Figure 3.5:** Giving priority to the firsts function for choosing the next sampling after 10 iterations

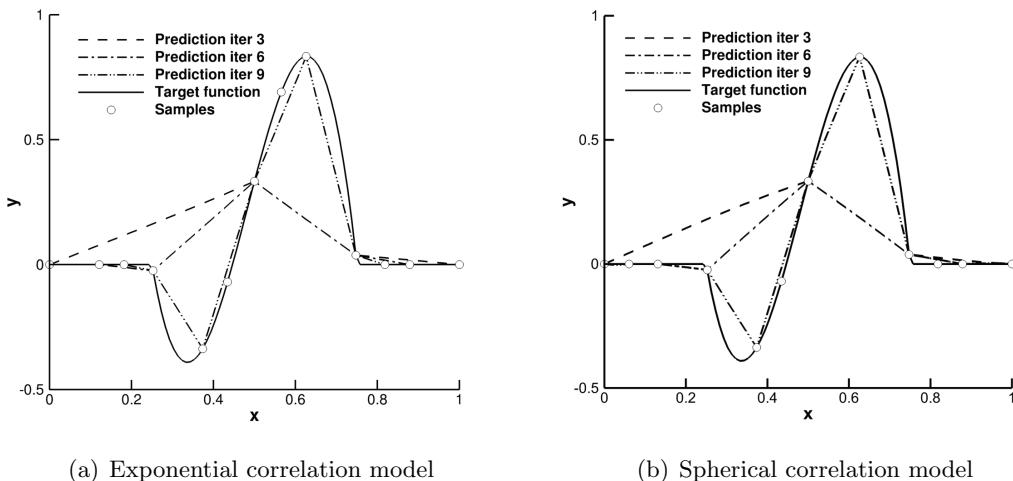
### 3.3.1.4 Choice of correlation model

The aim of this section is to find which is the best correlation model between the ones available.

- As can be seen in Fig. 3.5, a Gaussian correlation model suffers of high unstable oscillations for a non-continuously-derivable function.
- The cubic regression model is highly oscillating for all the created models at the increasing of the sampling points. This is exactly the behaviour we do not want, and so it may work well only for a very few sample points.
- The spline regression model reacts to the cusp (where right and left incremental ratios are different) with oscillations.
- The linear regression model operates very similarly to the unwanted behaviour of the spline model.
- The general exponential  $exp g$  can have both shapes, depending on the  $\theta$  parameter:  $\theta_{n+1} = 2$  and  $\theta_{n+1} = 1$  gives the Gaussian and the exponential function respectively [21].

The remaining exponential and spherical correlation models are so more deeply analysed.

The behaviour of the two correlation model at iterations number 3,6 and 9 are presented in Fig. 3.6, focusing on the close to the cusp for iteration number 9. It can be seen that the cusp does not cause any oscillation in both cases. The only difference between the two models that was found is the slightly bigger difference between real and predicted error for the spherical model.



**Figure 3.6:** Differences between exponential and spherical correlation models

### 3.3.2 Multi dimensions domain methods extension

Multi dimension domain problems are now investigated.

#### 3.3.2.1 Maximum $mse$ close to local maxima and minima

This method is developed starting from the script previously described for one dimension domain problems. In this case the local maxima and minima are searched looking in a multi dimensional sphere, whose radius is initially computed as minimum of the Euclidean norms of any two points with non-equal coordinates. If the value of the function at the considered point results bigger or smaller than all the close values, it is marked as local maximum or minimum respectively. After that all the local maxima and minima are found, the mean square predicted interpolation error,  $mse$ , trend close to them is considered. For every maximum and minimum, the point with maximum  $mse$ , belonging to a sphere centered in the found local maximum and with radius equal to the distance from the nearest sample point, is extracted. Then the new sample point is chosen between them as the one with the maximum  $mse$ .

Finally the  $mse$  of the found point is compared with the global maximum of the  $mse$  divided by a custom scale and so the method decides where to locate the next sample as the biggest between them.

This method might be very similar to the easy search of the maximum  $mse$  over the whole domain because of the presence of various local maxima or minima due to the intrinsic nonlinear nature of the aerodynamic equations, and so of the resulting forces. Anyway if a  $n$ -dimensions domain is considered the number of maxima and minima may decrease because the condition of maximum or minimum has to be satisfied for all dimensions.

The choice of looking only in sphere of radius equal to the maximum distance from the closest sample is not the best, because in other directions the error may still increase. However if the samples budget is small, the method may not look over the whole domain, neglecting the most important nonlinearities. This issue is partly avoided by always comparing the found point and the global maximum error scaled by a custom factor. In this way the computation does not blindly persist on the same location.

### 3.3.2.2 Hessian based sampling method

The second derivative based method could not be applied as easily as the previous one. For  $n$ -dimension domain problems, a unique second derivative value is not available, but it depends on the direction with respect to which is calculated. In order to consider all the local second derivative values and not to be bounded to any frame of reference, an easy but reasonable way is then found. The global problem is to find where a generic discrete function  $f_j$  manifests the biggest curvature in space, with respect to any considerable direction. The Hessian matrix elements are defined as partial derivatives as shown in Eq. (3.6a). If the second derivatives of  $f_j$  are all continuous, then the Hessian is a symmetric matrix (for the symmetry property of second derivatives known as Schwarz's or Clairaut's theorem).

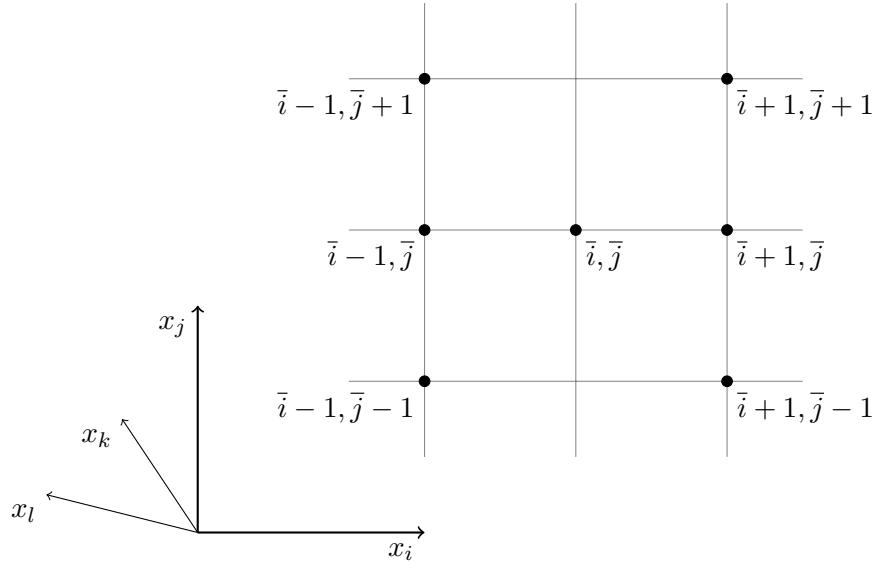
$$\mathbf{H} = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_n} \end{bmatrix} \quad (3.6a)$$

$$H_{i,j} = \frac{\partial^2 f}{\partial x_i \partial x_j} = \frac{\partial}{\partial x_i} \left( \frac{\partial f}{\partial x_j} \right) \quad (3.6b)$$

A finite difference approach is adopted, because of the discrete nature of the domain, in order to compute the Hessian matrix. About the diagonal terms, pure second derivative, the approximation with central difference presented in Eq. (3.4) is used, obtaining a second order of accuracy. The procedure of approximating the mixed second derivatives is based on central difference (second order of accuracy as well) and it is fully illustrated in Eq. (3.7b) with reference to Fig. 3.7. The  $\bar{i}$  and  $\bar{j}$  stand for the indices of the discrete domain indicating the  $x_i$  and  $x_j$  coordinates of the analysed point.

$$\left( \frac{\partial^2 f}{\partial x_i \partial x_j} \right)_{\bar{i},\bar{j}} = \left( \frac{\partial}{\partial x_i} \left( \frac{\partial f}{\partial x_j} \right) \right)_{\bar{i},\bar{j}} \simeq \left( \frac{\partial}{\partial x_i} \left( \frac{f_{\bar{j}+1} - f_{\bar{j}-1}}{|x_{\bar{j}+1} - x_{\bar{j}-1}|} \right) \right)_{\bar{i}} \quad (3.7a)$$

$$\simeq \frac{\left( \frac{f_{\bar{j}+1} - f_{\bar{j}-1}}{|x_{\bar{j}+1} - x_{\bar{j}-1}|} \right)_{\bar{i}+1} - \left( \frac{f_{\bar{j}+1} - f_{\bar{j}-1}}{|x_{\bar{j}+1} - x_{\bar{j}-1}|} \right)_{\bar{i}-1}}{|x_{\bar{i}+1} - x_{\bar{i}-1}|} \quad (3.7b)$$



**Figure 3.7:** Generic 2-dimensional sub-space grid to compute mixed differences, for any  $n$ -dimensional domain

The problem is now to choose the frame of reference directions to compute the Hessian matrix.

The mixed derivatives with respect to a generic direction indicated with a unit vector  $\hat{\mathbf{n}}$  are defined as  $\mathbf{H} \cdot \hat{\mathbf{n}}$ . Any element of the obtained vector represents  $\frac{\partial^2 f}{\partial x_i \partial \hat{\mathbf{n}}}$  where  $x_i$  is the  $i$ -th direction used to compute  $\mathbf{H}$ .

The aerodynamic functions usually manifest the biggest curvature with respect to some main direction of the domain (e.g.  $\alpha, \beta, M, \dots$ ). The aerofoil pitching moment computed with respect to the angle of attack and Mach number may be considered as a two-dimensional problem example. Increasing one the two variables in proximity of a shock appearance, the pitching moment may suddenly change leading to a big curvature in that point. Increasing the angle of attack, the flow on the upper surface is accelerated more and so it might arrive to the sound speed, causing the formation of a shock. At the same time increasing the speed, so the Mach number considering the same speed of sound (function of the altitude), may lead to the same effect. The formation of a shock causes a sudden pressure drop behind it, and so an additional pitch-down moment appears. Beyond this static effect the unstable nature of a shock and the non-static boundary layer separation after that, causes the shock to oscillate, causing oscillating forces (phenomenon-known as transonic buffeting) but this only manifests for a non-stationary study. For a time independent analyses, the biggest value of second derivative for the pitching moment may lie between angle of attack and Mach number directions.

Finding the maximum of the second derivative looking at every available direction, for every point of the domain, might be an option. Nevertheless this may be compu-

tationally expensive and not lead to a real improvement in the overall computation. In fact the biggest findable curvature is in the same location where the average of all the curvatures is maximum. In order to obtain an averaged value of the curvatures, the curvatures vectors with respect to the main directions are computed (Eq. (3.8a)) and the square summed (Eq. (3.8b)), so considering both second and mixed derivatives. Then the sum of all the resulting values is calculated in order to obtain a usable value (see Eq. (3.8c)). This value is of immediate computation considering the main directions as shown in Eq. (3.8d) and represents the square of the Forbenius norm of the Hessian matrix.

$$dd_i = \left( \frac{\partial^2 f}{\partial x_i \partial x_1}, \dots, \frac{\partial^2 f}{\partial x_i \partial x_j}, \dots, \frac{\partial^2 f}{\partial x_i^2}, \dots, \frac{\partial^2 f}{\partial x_i \partial x_n} \right) = \mathbf{H} \cdot x_i \quad (3.8a)$$

$$\|dd_i\|^2 = \cdot \sum_{j=1}^n dd_i_{(j)}^2 = \cdot \sum_{j=1}^n (\mathbf{H} \cdot x_i)_{(j)}^2 \quad (3.8b)$$

$$\|\mathbf{H}\|_F^2 = \cdot \sum_{i=1}^n \|dd_i\|^2 = \sum_{i=1}^n \sum_{j=1}^n (\mathbf{H} \cdot x_i)_{(j)}^2 \quad (3.8c)$$

$$\text{if using } \begin{cases} \hat{x}_1 = (1, 0, \dots, 0)' \\ \vdots \\ \hat{x}_i = (0, \dots, 0, 1, 0, \dots, 0)' \quad \Rightarrow \|\mathbf{H}\|_F^2 = \sum_{i=1}^n \sum_{j=1}^n H_{i,j}^2 \\ \vdots \\ \hat{x}_n = (0, \dots, 0, 1)' \end{cases} \quad (3.8d)$$

The obtained value is independent of the reference frame considered, being the same for any usable frame. In the following equation a short demonstration is given about the previous statement. Consider a 3-dimensional domain, the gradient of a generic function  $f$  is a vector for which the components are dependent on the used reference frame but the length is not (computed as the Euclidean norm). The Hessian matrix can be interpreted as the combination of the gradient of three different functions  $g_1, g_2$  and  $g_3$ , that are the three components of the gradient of the original function  $f$ . The Euclidian norm of the gradient of these three functions is, as previously described, independent from the considered reference frame. So we obtained three values independent from any reference frame, and if we sum the square of them we obtain a value independent on the used reference frames as well. This is the same value previously obtained with

the main directions of the domain and it is the Euclidian norm of the Hessian matrix.

$$\text{hyp. 3 dimensional space} \quad (3.9a)$$

$$\nabla f = \left( \frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \frac{\partial f}{\partial x_3} \right) = \bar{d}(x_1, x_2, x_3) \quad (3.9b)$$

$$\|\nabla f\|^2 = \|\bar{d}(x_1, x_2, x_3)\| = \mathbf{d} = \text{cost} \quad \forall \langle x_1, x_2, x_3 \rangle \quad (3.9c)$$

$$\mathbf{H} = \nabla(\nabla f) = \nabla \left( \frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \frac{\partial f}{\partial x_3} \right) = \nabla(g_1, g_2, g_3) \quad (3.9d)$$

$$= \left( \frac{\partial}{\partial y_1}, \frac{\partial}{\partial y_2}, \frac{\partial}{\partial y_3} \right) \bar{d}(x_1, x_2, x_3) = \bar{h}(x_1, x_2, x_3, y_1, y_2, y_3) \quad (3.9e)$$

$$\|\mathbf{H}\|_F^2 = \|\bar{h}(x_1, x_2, x_3, y_1, y_2, y_3)\| = \mathbf{h} = \text{cost} \quad \forall \langle y_1, y_2, y_3 \rangle \quad \wedge \quad \forall \langle x_1, x_2, x_3 \rangle \quad (3.9f)$$

$$\Rightarrow \|\mathbf{H}\|_F^2 = \sum_{i=1}^n \sum_{j=1}^n H_{i,j}^2 = \text{cost} \quad (3.9g)$$

The previous demonstration may be reduced to the assertion that the Frobenius norm is invariant under a unitary transformation, as a reference frame rotation. This is easily demonstrated in Eq. (3.10) using the orthogonal nature of  $\mathbf{P} : \mathbf{P}^T \cdot \mathbf{P} = \mathbf{I}$  and the cyclic nature of the trace:  $\text{trace}(\mathbf{XYZ}) = \text{trace}(\mathbf{ZXY})$ .

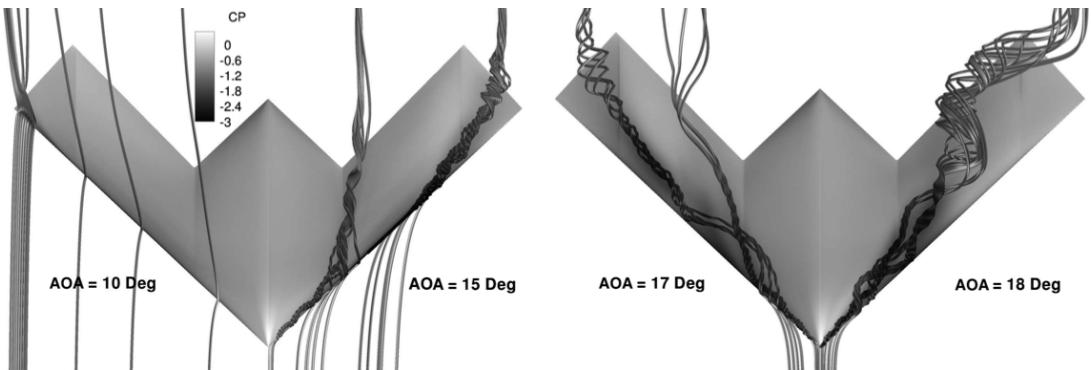
$$\|\mathbf{A}\|_F^2 = \|\mathbf{P}^T \cdot \mathbf{B} \cdot \mathbf{P}\|_F^2 = \text{trace}((\mathbf{P}^T \cdot \mathbf{B} \cdot \mathbf{P})^T \cdot (\mathbf{P}^T \cdot \mathbf{B} \cdot \mathbf{P})) = \text{trace}(\mathbf{B}^T \cdot \mathbf{B}) = \|\mathbf{B}\|_F^2 \quad (3.10)$$

### 3.4 Validation with realistic cases

The data until now acquired are based on a mathematical model and are not significant of any real case. The functions developed in the previous section are similar to the aerodynamic trends but not representative of real computations. Two real cases are considered in order to obtain a validation on real data sets.

#### 3.4.1 UCAV experimental data

The first is taken from an article about vortical flow prediction applied to a UCAV (unmanned combat air vehicle) [6]. The geometry consists of a lambda wing with a sweep angle of 53 deg and a wing washout of 5 deg. In the presented study the experimental data campaign took place in the  $3.25 \times 2.8$  m NWB wind tunnel at Braunschweig for the rounded leading edge model are considered. A dual vortex structure is present around 17 deg of angle of attack, resulting in highly nonlinear aerodynamic behaviour.



**Figure 3.8:** UCAV predicted flow topology for different angles of attack (AoA) [6]

Figure 3.8 presents the flow field topology obtained with a RANS CFD solver with  $k-\epsilon$  turbulence model. At 10 deg incidence a small outboard vortex seems emanating from the tip section. As the incidence increases to 12 deg, an apex vortex starts to form and a dual vortical structure is present until 18 deg angle of attack. In this range, the vortices get stronger and the onset of the outer shear layer separation travels inboard. At a 16 deg angle of attack, the tip vortex is seen to jump as it travels inboard. From here to around a 19 deg incidence, a slow merging of the two vortices occurs. The dual vortical flow topology becomes clearer from the stream traces. At 17 deg incidence, an image during the vortex merging process is shown. Vortex breakdown is present at 17 and 18 deg of incidence where the stream traces change in colour as the spiral increases in size.

The aerodynamic coefficients considered for this validation are the lift, drag and pitching moment with respect the angle of attack. The analysis conducted is then one-dimensional. The two sampling methods that were adopted are the ones looking

for the maximum of the predicted  $mse$  close to the maximum of second derivative and local maxima or minima, respectively.

The DACE toolbox input were:

- linear regression model
- exponential correlation model with  $\theta = 0.5$

While the used custom input values were:

- 5 for the error ratio to compare the maximum Kriging error and the error found with the developed method
- 100 for the weight of the functions priority
- (3,1,2) for the order of priority given to the functions, in order to have high priority for the pitching moment and low for the drag.

The two methods presented in Fig. 3.9 computed a final interpolation model for every considered function with 12 samples, that are chosen during the process steps. The obtained relative difference between target and prediction functions (real relative errors) is presented in Table 3.3, while the Kriging model predicted interpolation relative errors are shown in Table 3.4.

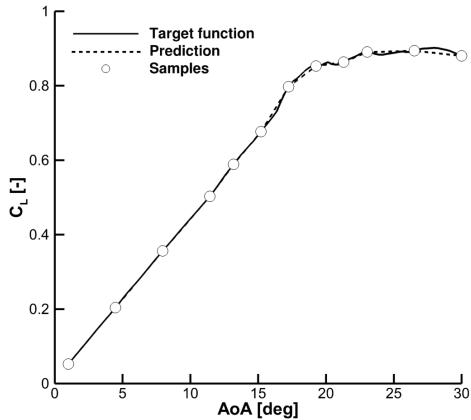
Mean real relative errors [%]	$\varepsilon_{C_L}$	$\varepsilon_{C_D}$	$\varepsilon_{C_m}$
Local maxima or minima based method	0.41	3.63	2.40
Second derivative based method	0.39	13.33	1.92

**Table 3.3:** Mean real relative errors comparison for the UCAV case using second derivative and local maxima or minima based methods

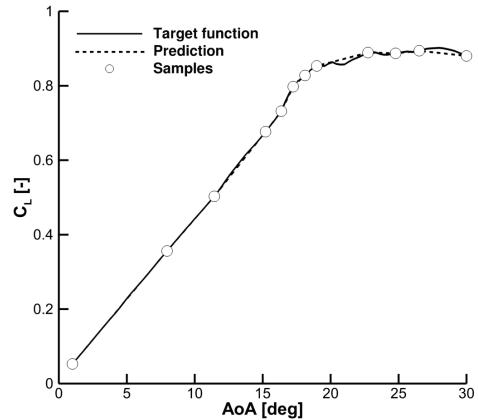
Mean predicted relative errors [%]	$\varepsilon_{C_L}$	$\varepsilon_{C_D}$	$\varepsilon_{C_m}$
Local maxima or minima based method	0.16	0.88	0.06
Second derivative based method	0.24	1.06	0.10

**Table 3.4:** Mean predicted relative errors comparison for the UCAV case using second derivative and local maxima or minima based methods

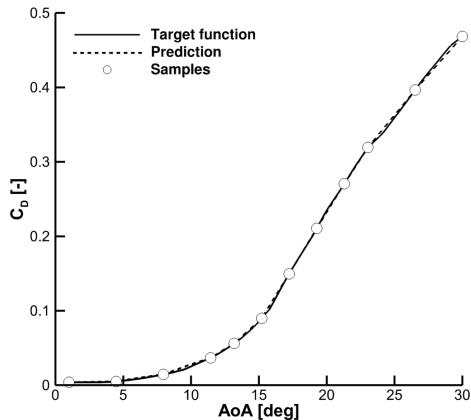
The solutions gives very good results, finding the sudden drop of the pitching moment with only 12 sample points for both methods (iteration number 10). For the maxima–minima method this is achieved because of the presence of a local maximum close to the nonlinearity in the pitching moment and for the second derivative based method the lift function has a high second derivative at the same angle of attack of the pitching moment drop and so some sampled are positioned there, permitting to have a further analyses close to this nonlinearity.



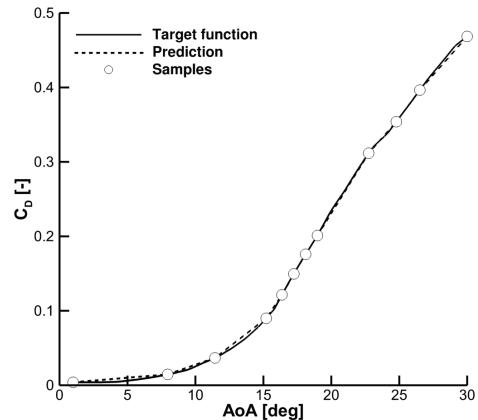
(a) Lift with local maxima–minima method



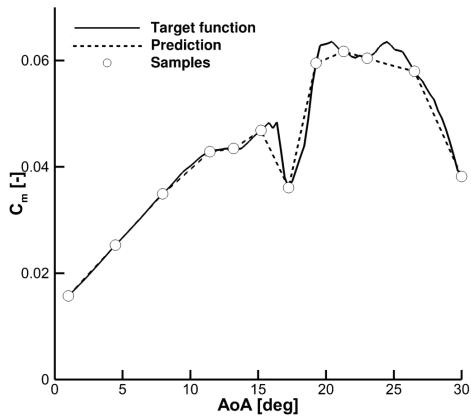
(b) Lift with second derivative method



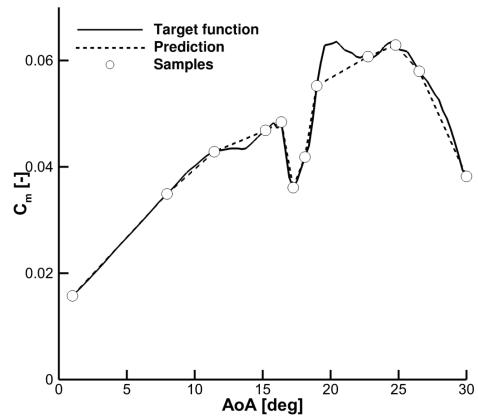
(c) Drag with local maxima–minima method



(d) Drag with second derivative method



(e) Pitching moment with local maxima–minima method (f) Pitching moment with second derivative method

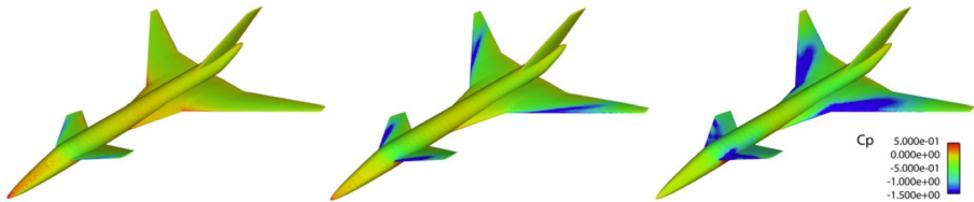


(f) Pitching moment with second derivative method

**Figure 3.9:** Comparing the results obtained with the two developed methods on the UCAV data after 10 iterations (2 initial samples at the extrema)

### 3.4.2 TCR experimental data

The Transonic cruiser (TCR) is a design concept proposed by SAAB for a civil passenger aircraft flying in the transonic regime. This test case has been analysed in the CEASIOM environment in [20].



**Figure 3.10:** Transonic cruiser CFD pressure distribution, at  $\alpha = 0, 8, 20$  deg;  $V_\infty = 40$  m/s,  $\beta = 0$  deg and  $\delta_{can} = 10$  deg

Figure 3.10 presents the surface pressure distribution over the aircraft obtained from CFD simulations. A data fusion process of different data sources obtained the resulting aerodynamic data. Hundreds of CFD calculations were carried out, of which 67 were considering Euler equations and 187 were with Navier-Stokes viscous fluid model. The considered data for this analysis are the lateral force ( $C_Y$ ) and pitching moment coefficients with respect to the angle of attack and the sideslip angle ( $\beta$ ). The choice is driven by the strong nonlinear nature of these trends, which are a good test case for the developed method. The analysis conducted has then a two-dimensional domain.

The two used sampling methods are the ones looking for the maximum of the predicted  $mse$  close to the maximum of second derivative and local maxima or minima, respectively. The DACE toolbox input were:

- linear regression model
- exponential correlation model
- $\theta = [1, 0.5]$  for  $\alpha$  and  $\beta$  axis respectively

While the used custom input values were:

- 1.5 for the error ratio to compare the maximum Kriging error and the error found with the developed method
- 1 for the weight of the functions priority
- (5,2) for the order of priority given to the functions, in order to have high priority for the pitching moment and low for the drag.
- 10 maximum iterations (equals to 14 sampled points)

In the two following tables the resulting mean errors are presented, respectively the mean real error in Table 3.5 and mean predicted error in Table 3.6.

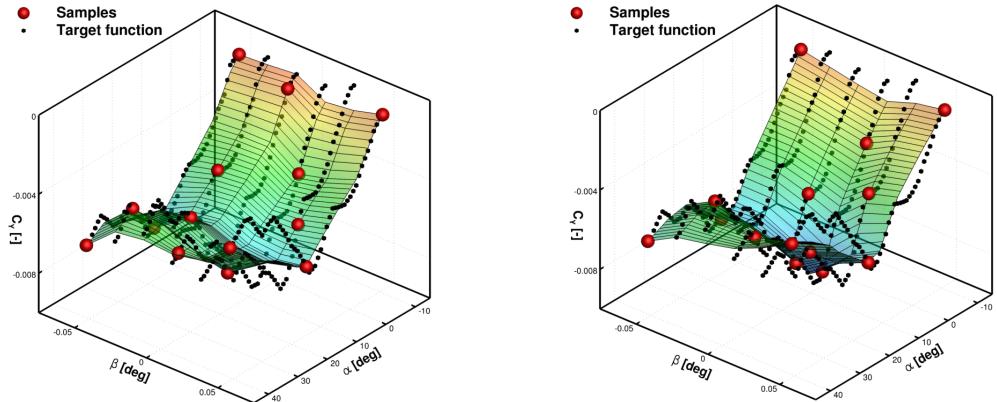
Mean real relative errors [%]	$\varepsilon_{C_Y}$	$\varepsilon_{C_m}$
Local maxima or minima based method	10.63	15.90
Hessian based method	11.05	13.12

**Table 3.5:** Mean real relative errors comparison for the multi-dimensional TCR case using Hessian and local maxima or minima based methods

Mean predicted relative errors [%]	$\varepsilon_{C_Y}$	$\varepsilon_{C_m}$
Local maxima or minima based method	3.26	2.57
Hessian based method	3.47	3.11

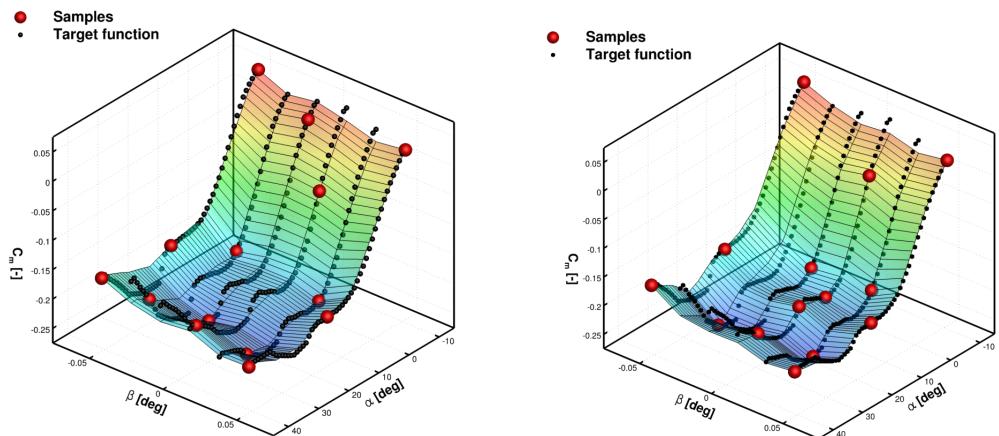
**Table 3.6:** Mean predicted relative errors comparison for the multi-dimensional TCR case using Hessian and local maxima or minima based methods

In Fig. 3.11 the coloured surface indicates the model prediction with the red sample points data and the black cubes are the discrete target functions. It can be seen how the two methods catch well the general trend of the function with only 14 samples (4 starting and 10 iterations). Unfortunately there is a big nonlinearity in the lateral force at high AoA and  $\beta = 0.05$  that is not captured by both the methods. This is a bad result for the two methods also if only 14 points over 318 total points of the domain were analysed (4.4 % of the usual required computations). The Hessian based method anyway has a better behaviour because it focus the solution close to the basin created in the lateral force function for angles of attack between 10 deg and 20 deg.



(a) Lateral force with local maxima–minima method

(b) Lateral force with Hessian method



(c) Pitching moment with local maxima–minima method

(d) Pitching moment with Hessian method

**Figure 3.11:** Comparing the results obtained with the two developed methods on the TCR data after 10 iterations (with 4 initial samples at the domain vertices)

## 3.5 Data fusion

The aerodynamic coefficients can be generally obtained using different sources. If more data sets are available, data fusion can combine them in order to obtain a more accurate model. In the case of two data sets available, usually one can be considered low-fidelity ( $lf$ , cheaper and usually more populated) and the other high-fidelity ( $hf$ , more expensive and usually less populated). During the data fusion process the cheap samples provide information about the trend of the target function, while the expensive samples give quantitative information of the model. Two methods are considered to apply data fusion:

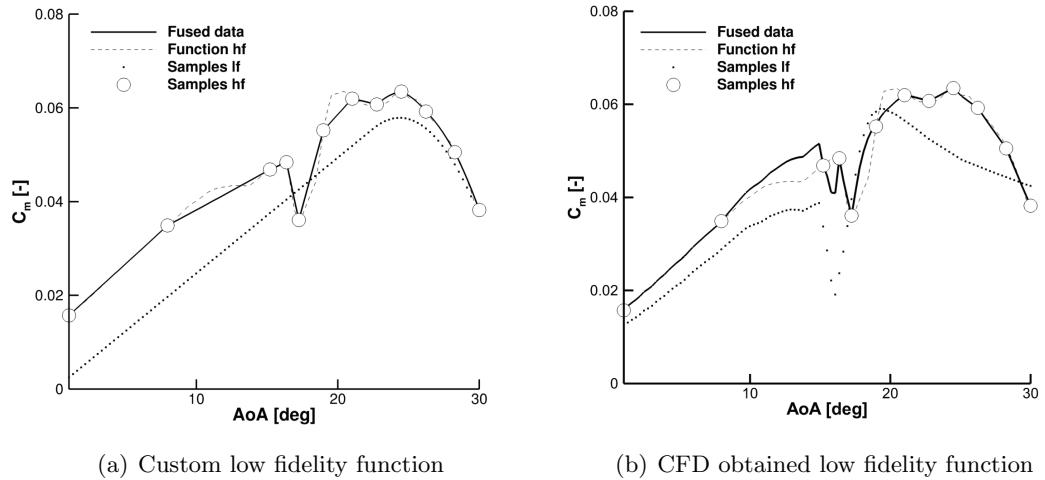
1. One is explained by Da Ronch et al. in [13]. A Kriging interpolation model  $\hat{\eta}_{(\mathbf{x})}$  is calculated from the samples of the cheap aerodynamic evaluations and it is evaluated at the locations at which expensive predictions are available,  $\hat{\eta}_{(\mathbf{x}_i)}$ . The vector of the input parameters at the expensive samples,  $\mathbf{x}_i$ , is then augmented by the evaluation of the Kriging function for the cheap samples:  $\mathbf{x}_i^{aug} = [\mathbf{x}_i \ \hat{\eta}_{(\mathbf{x}_i)}]$ . A Kriging interpolation model is finally calculated for the augmented samples and the data fused evaluation is given by the evaluation of such function in the continuous vector  $\mathbf{x}^{aug} = [\mathbf{x} \ \hat{\eta}_{(\mathbf{x})}]$ .
2. Another method is described in [23] by Santini. As the previous method a Kriging interpolation model  $\hat{\eta}_{(\mathbf{x})}$  is calculated from the samples of the cheap aerodynamic evaluations. This method is then based on an increment function  $\hat{\beta}_{(\mathbf{x})}$  obtained with the interpolation of  $\beta_{(\mathbf{x}_i)} = f_{hf}(\mathbf{x}_i) - \hat{\eta}_{(\mathbf{x}_i)}$  where  $f_{hf}(\mathbf{x}_i)$  are the high-fidelity sampled points. From this the data fusion approximation is easily derived as  
$$\hat{f}_{(\mathbf{x})} \simeq \hat{\eta}_{(\mathbf{x})} + \hat{\beta}_{(\mathbf{x})}.$$

The first method presents some oscillating problems dealing with the interpolation for a second order regression model because of the aligned nature of  $\mathbf{x}_i^{aug}$  and in the case of a linear low-fidelity database, the resulting Kriging interpolation matrix is ill-conditioned. For these reasons the implemented function try to use this method, but it does not always succeed. So the user is advised to use the second method and be very careful about the first.

### 3.5.1 Validation

Two similar data fusion cases are investigated using the second of the methods previously described. The high-fidelity data are taken from results of the wind tunnel test data about a UCAV reported in [6] already used in the one-dimensional validation. The sample point choice is taken from the results of the previously described Hessian iterative sampling method. About the low-fidelity data, in the first case a custom function presenting an initially linear behaviour and then a sudden drop, is used. This

function is slightly vertically shifted with respect to the high-fidelity data. The second low-fidelity case is taken from [6] and represents the resulting data of a CFD computations solving RANS equations with a  $k-\epsilon$  turbulence model. This case is more realistic and represents a real possible situation.

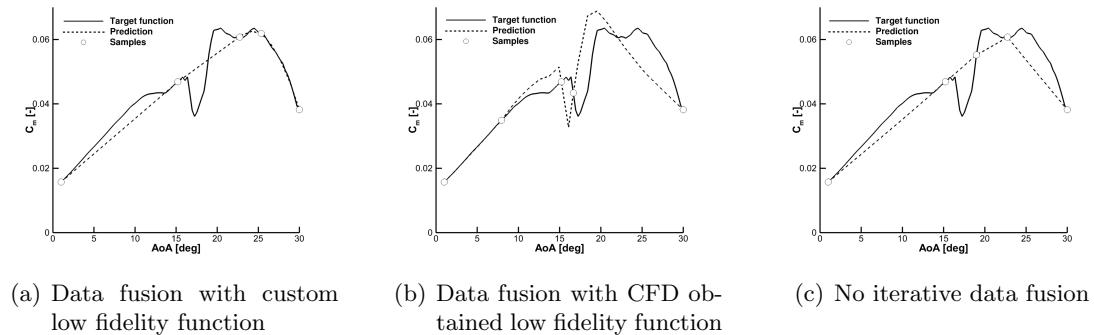


**Figure 3.12:** Data fusion applied to the UCAV wind tunnel high fidelity data with two different low-fidelity data

The first result presented in Fig. 3.12 shows that, in case of a linear low-fidelity data trend, the data fusion looks really similar to a linear interpolation of the high-fidelity data. A small difference may appear close to the high curvature of the low-fidelity function, for which the fused data try to represent the same bend. The second resulting fused data are really different. The two data sets may appear similar but, because of the horizontal shifting of the curves, the fused data displays a high oscillation close to the presence of the nonlinearity. Theoretically we cannot know which is the exact position of that, but the low-fidelity data information may lead astray. In conclusion data fusion may bring some improvement, but the user is highly advised to take a hard look at the results.

### 3.5.2 Iterative sampling with data fusion

Data fusion can be inserted during the iterative sampling method, obtaining a better model of the function starting from the first iterations. In Table 3.7 and in Fig. 3.13 are presented the results after 3 iterations not using data fusion and using data fusion with the two previously described low-fidelity database.



**Figure 3.13:** Iterative data fusion applied to the UCAV wind tunnel high fidelity data after 3 iterations different low-fidelity databases

Mean real relative errors [%]	$\varepsilon_{C_m}$ at iter=3	$\varepsilon_{C_m}$ at iter=10
NO iterative data fusion	9.4	3.7
Custom <i>lf</i> function	6.6	3.6
CFD <i>lf</i> function	10.1	3.1

**Table 3.7:** Mean real relative errors comparison for iterative data fusion applied to the UCAV wind tunnel high fidelity data after 3 and 10 iterations different low-fidelity databases

Figure 3.13 shows that the prediction function follows the low-fidelity database, and whether this is good depending on the compatibility of the two databases. About the first case the ease of the cheap available results is that the resulting fused data are really well approximated, whereas the nonlinearity is not found. The second case instead sample the nonlinearity part, because of its presence in the cheap available results, but the overall approximated function is a worse approximation. For this reason the user is advised to use data fusion during iterative sampling, but only for the first iterations, otherwise the low-fidelity database might bring to some inaccurate prediction.

The best result after 10 iterations is obtained using the CFD low-fidelity database and the iterative data fusion only for the first 5 iterations: the obtained mean real relative error was  $\varepsilon_{C_m} = 2.1\%$ .

### 3.6 Conclusions and future development

The DACE toolbox allows generating a Kriging interpolation model that can be used to have a complete filled interpolated aerodynamic data table. The previously described methods can be used as tools to decide the best position of the next needed samples. This would make the overall CFD computation more efficient, obtaining a more accurate solution and focusing on the appearing of nonlinearities in the flow field. The study until now conducted leaded to the DACE parameters that seem to best fit our purpose. These are a linear regression model `regpoly1`, an exponential correlation model `correxp` and values of the parameter  $\theta$  that should be bigger for a stronger relation between function and variable (e.g. for  $C_m(\alpha, \beta) \rightarrow \theta = [\theta_\alpha, \theta_\beta] = [1, 0.5]$ ). About the sampling method, the one that appeared to deal the most efficiently with the problem is the local maximum *mse* close to the global maximum value of the Hessian (function curvature). Nevertheless the maxima and minima search may be useful as well.

Furthermore an analysis about the choice of the two custom parameters is essential:

1. **tolerance** is for comparing the errors found close to the maximum of the second derivative or the local maxima or minima and the maximum error found all over the domain
2. **priority** exists to give priority to some functions with respect to the others, whilst the priority order should be given base on the experience and on the functions of interest.

In case of availability of data from two sources, data fusion permits to have a better approximation of the aerodynamic coefficients. During data fusion the cheap samples provide information about the trend of the target function and the expensive samples give quantitative information. The developed functions gave good results, although a check on the fused function may be required because of the possible incompatibility of the two fused databases that might bring some bad results.

The data fusion used for iterative sampling seems to work properly. The best results were obtained if it is used only for a few first iterations. It creates a more representative interpolation model of the function when no data are available yet. After that some samples are computed, iterative sampling can work autonomously, so that the cheap database does not cause wrong function prediction.



## Chapter 4

### Regional jet test case

The aircraft design process is very expensive and the largest part of the life–cycle cost is directly dependent on decisions taken during the conceptual design phase. Obtaining accurate and reliable information about aircraft stability and performance characteristics during the first steps of the design is then highly critical. The engineering tools for aircraft design are generally based on empirical handbook methods or linear fluid mechanics hypothesis [3]. These provide low–cost aerodynamic data predictions reliable for benign flow conditions, e.g. ruling out the most critical points of the flight envelope. The data points at the borders of the flight envelope do not consider the nonlinear flight dynamic behaviour of the aircraft. As discussed in Ref. [14], a solution may be found in using computational fluid dynamics (CFD) techniques in the conceptual design phase in order to predict the nonlinear effects. However the computation of the complete aerodynamic database over the whole flight envelope with CFD is very expensive and requires a long computation time. High–fidelity simulations are still expensive despite today high performance computing facilities are available Ref. [25]. In aircraft design where the geometry changes, the use of CFD is prohibitively expensive and unrealistic for computing the entire aerodynamic database. The computation of dynamic derivatives requires an unsteady time accurate CFD analysis, which needs a very long computational time [16–18]. Hence, acceleration techniques based on CFD allows retain the fidelity at a reduced cost. An efficient way to produce a good full aerodynamic database with a fewer computations is to use a Kriging–based surrogate model and fusion of two available databases as described in Ref. [13]. A hierarchy of aerodynamic models is first computed, from semi–empirical approaches up to computational fluid dynamics analyses, and validated at both low and high speeds against wind tunnel measurements. The CFD based aerodynamic model is efficiently generated exploiting Kriging interpolation. The trim characteristics and longitudinal and lateral handling qualities are first investigated for the regional jet aircraft. Geometry variations are then applied. For every considered configuration a full–order CFD aerodynamic model was computed fusing a few new computations with the full model for

the baseline geometry, maintaining the high-fidelity characteristics of CFD over the whole flight envelope. The handling qualities are then evaluated and the influence of the new configuration design parameters over the performances and stability are presented. It is found that improved aircraft characteristics could be achieved for relatively small changes relative to the baseline geometry. This study aims to show the feasible introduction of CFD from early aircraft design phases and to study the influence of the aerodynamic model and some geometrical parameters over the resulting aircraft statical and dynamical behaviour.

## 4.1 Test case

The aircraft model used in this work is a conceptual design of a regional jet originally designed using traditional hand-book methods. The aircraft model belongs to the 100–120 passengers class. The existing similar models considered during the design process were the new Bombardier CS100, the Embraer E–190, the Boeing 737–600 and the Airbus A318–100. Table 4.1 shows the mission specifications and the main dimensions <sup>1</sup>. To preform flight simulation, a model of the aerodynamic forces and

Parameter	Value
Range	2,600 km
Cruise Mach	0.78
Cruise altitude	10,668–11,887 m
Number of engines	2
Number of passengers	110
Landing field length	1,450 m
Take off field length	1,550 m
Long. ref. length	3.6 m
Lat. ref. length	30 m
Wing area	105 m <sup>2</sup>

**Table 4.1:** Mission specifications and dimensions of the regional jet aircraft

moments is required throughout the entire flight envelope. Figure 4.1 presents scaled 1:23 model used in the German–Dutch Wind Tunnels (DNW) at Reynolds number  $4 \cdot 10^6$ .

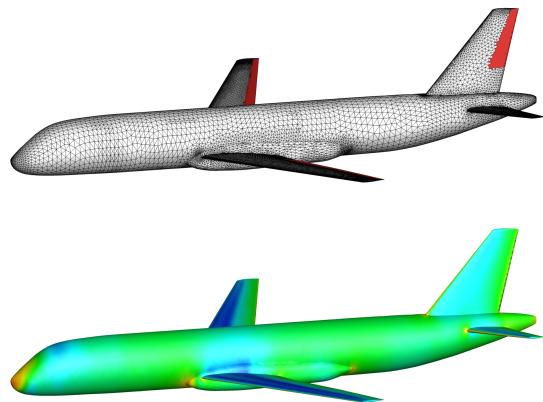
The preformed analyses is obtained using the CEASIOM 100 software presented in Chapter 2. Inside CEASIOM, AcBuilder is a customized geometry construction system to define the aircraft configuration. It requires only  $\sim 100$  geometrical parameters because the design is still considered in the conceptual phase. With a simplified set of parameters, the CEASIOM model approximates the nose geometry and the tail.

---

<sup>1</sup>The presented model that is stored in a CEASIOM-compatible format is open source and can be requested to Andrea Da Ronch A.Da-Ronch@soton.ac.uk.



**Figure 4.1:** Wind tunnel model of the regional jet aircraft in scale 1:23



**Figure 4.2:** Surface grid with the control surfaces in red and pressure coefficient distribution resulting from Euler solution at  $M = 0.78$  and  $\alpha = 3$  deg

Once the geometry is built, an automated grid generator is used to create within few seconds an initial mesh for CFD calculations. The Surface Modeller SUMO was used. It is a graphical tool for a rapid aircraft geometry creation coupled to high efficient unstructured surface and volume grid generators. It takes as input the AcBuilder basic parameterization and uses it to produce surface and volume grids for Euler CFD simulations. The surface unstructured grid with the control surfaces and the Euler CFD result about the pressure coefficient distribution for a Mach number of 0.78 and an angle of attack of 3 deg is shown in Fig. 4.2. This reference geometry is referred as baseline geometry in the remaining of the Chapter.

## 4.2 Generation of Aerodynamic Tables

In this work the aircraft stability characteristics are investigated. Aerodynamic tables are needed to study the aircraft static and dynamic properties. The following sections illustrate the models used for the aerodynamic predictions, their validation against wind tunnel measurements, and the approaches used to reduce the number of CFD calculations to a manageable computational cost.

### 4.2.1 Validation

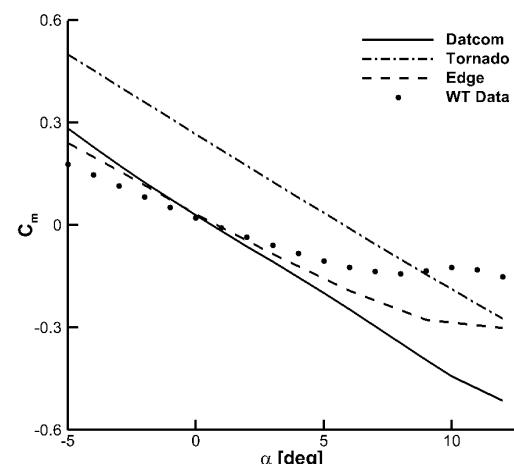
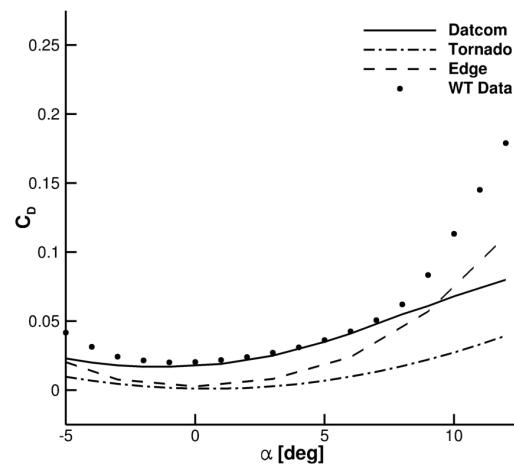
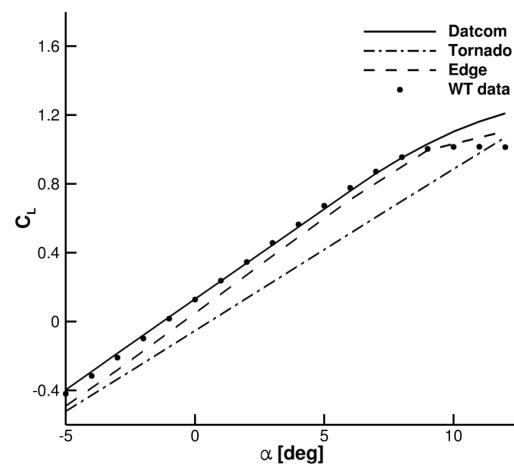
The methods used for the aerodynamic predictions are an empirical method (Stability and Control Digital Data Compendium – DATCOM [5]), a linear panel method based on Vortex Lattice Method (VLM) (Tornado <sup>2</sup>) and an unstructured CFD solver (Edge <sup>3</sup>). The lift, drag and pitch moment coefficients ( $C_L$ ,  $C_D$ , and  $C_m$ ) with the angle of attack at Mach number of 0.5 are shown in Fig. 4.3. The good agreement with wind tunnel data found for DATCOM is expected since it relies on a database of similar aircraft configurations to the one presented herein. Tornado provides the correct global trends but the quantitative differences are due to the limiting underlying assumptions (linear panel method) and the neglect of non-lifting surfaces that contribute significantly to the pitch moment. The flow solver Edge is used in Euler mode in this work. For this reason it does not consider the viscous term and so the resulting drag values appear smaller than the coefficients computed during the wind tunnel test. For high angles of attack, Edge shows the most similar behaviour to the wind tunnel data. At the cruise Mach number of 0.78, similar considerations were found, with Edge showing a better correlation to experimental data. The results are not shown for brevity.

### 4.2.2 Aerodynamic Tables

The aerodynamic table is divided in static, control and dynamic derivatives as shown in Table C.1. The static part of the table is based on a  $(\alpha, M, \beta)$  three-dimensional domain. The  $\alpha$  values were considered between -5 and 12 deg with a 1 deg step,  $M$  between 0.1 and 0.9 with 0.1 step and  $\beta$  values between -10 and 10 deg with a 1 deg step. The total number of static flight points is 3,402. The control derivatives part of the table considers the elevator, rudder and ailerons deflections separately. The elevator deflection was studied for values between -20 and 20 deg with 5 deg steps and the rudder and ailerons deflections for values between -15 and 15 deg with 5 deg steps. About the dynamic derivatives, the yaw, pitch and roll angular velocities are considered. They are taken from -80 to 80 deg/s with a step of 20 deg/s. Every considered control surface deflection and angular speed was studied singularly for any combination of  $\alpha, M$ , so adding a total of 7,128 more entries to the table. The total size of the aerodynamic table was 10,530 flight states. The reference point considered for computing the moments and the angular velocities was the approximated centre of gravity position. The CFD was not used to compute the dynamic derivatives because of the very high cost for computing an unsteady time accurate CFD solution. DATCOM results were used for both the control and dynamic derivatives. Furthermore, for any geometry modification a new table needs to be generated, and so an efficient method to reduce the computational time is highly required. In order to fill in the aerodynamic

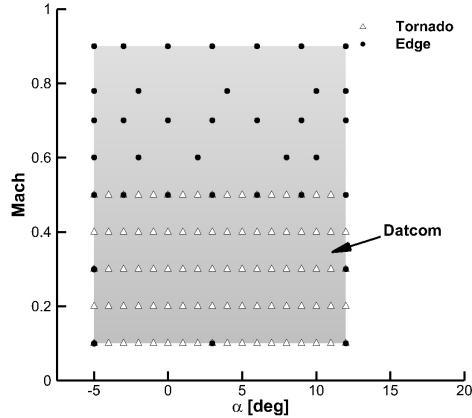
<sup>2</sup><http://www.redhammer.se/tornado/> [retrieved October 10, 2014]

<sup>3</sup><http://www.foi.se/edge/> [retrieved October 10, 2014]



**Figure 4.3:** Lift, drag and pitch moment coefficients at Mach number 0.5 and Reynolds number  $2.2 \cdot 10^7$ ; the reference point is taken at the centre of gravity

table, the data were taken from the three aerodynamic models and experimental results. Figure 4.4 shows the flight states that were computed for each aerodynamic model.



**Figure 4.4:** Samples distribution over the  $\alpha - M$  domain

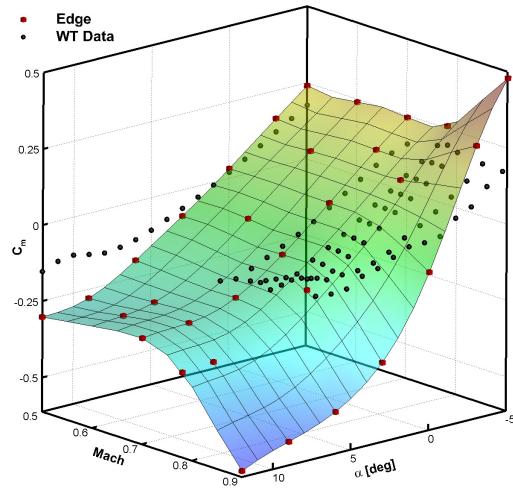
The semi-empirical method DATCOM was used all over the domain to provide a quick overview of the aerodynamic data since it has good accuracy for traditional aircraft and very low computational cost. Tornado was used to compute the results for Mach number from 0.1 to 0.5, which is the appropriate range for the vortex lattice method. Tornado computational cost is relative low and it does not consider any compressibility correction. Edge was used for higher Mach numbers, from 0.5 to 0.9, in order to capture the nonlinear aerodynamic characteristics. For Mach numbers from 0.1 to 0.3 only 5 samples from Edge were computed on the edges of the domain to calibrate the low fidelity results. For all the combinations of these parameters, the results of different sideslip angle ( $\beta$ ) were investigated for angles of 0, 5 and 10 deg apart from DATCOM that does not provide a sideslip angle input option. Full tables were obtained for both DATCOM and Tornado. About the table computed using CFD, Kriging interpolation and data fusion approaches were used to reduce the cost to a manageable requirement. A total of 97 non-viscous CFD calculations using Edge were performed in the  $(\alpha, M, \beta)$  parameters space.

#### 4.2.3 Kriging Interpolation

Kriging method generates an interpolation model for nonlinear and multi-dimensional deterministic functions. In Ref. [22, 24] the Kriging interpolation is efficiently used to reduce the computational cost for generating a full aerodynamic model. Once the Kriging model is created, it becomes a computationally cheap model for prediction of the function at untried locations.

The available samples that were computed by Edge covered Mach numbers from 0.5 to 0.9, angles of attack from -5 to 12 deg and beta of 0, 5 and 10 deg. In Fig. 4.5

the pitching moment resulting surface in the  $(\alpha, M)$  domain, obtained with the Kriging interpolation of the Edge data is compared with the experimental results from the wind tunnel testing. The linear trend shows a smaller slope for the wind tunnel results and for high Mach number the Edge pitch moment slope is much higher. This nonlinearity does not appear from the wind tunnel tests, for which the pitch moment slope is very similar for every Mach number. Only the pitch moment coefficient results are presented for brevity. The Kriging interpolating surface have the same trends with wind tunnel results for all the computed force and moment coefficients. The lift coefficient at low Mach numbers is very close to experimental results from every angle of attack. Close to the transonic field, for high Mach numbers, the values of lift coefficient are higher than the wind tunnel results. This also results in the differences between pitch moment coefficients on  $C_m$  Kriging surface and experimental results. About the drag coefficient predicted by Kriging model, it shows a good correlation with experiment results. The values are lower than wind tunnel because the solutions are achieved solving the Euler equations.

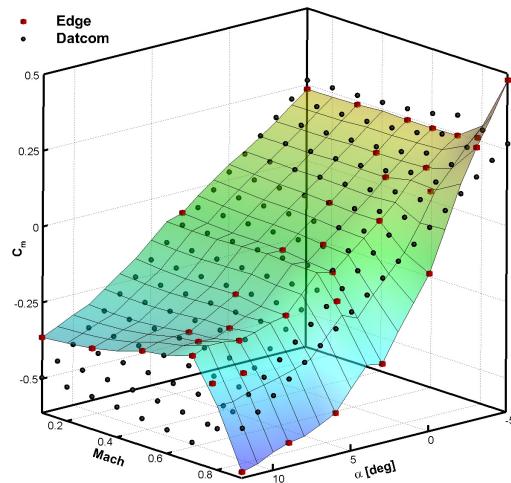


**Figure 4.5:** Dependency of pitch moment coefficient on angle of attack and Mach number; the surface represents the Kriging interpolation of Edge samples (red cubes)

In conclusion the Kriging model shows very efficient prediction capability and could considerably reduce the computational cost. In this study only the 5.45% of the total number of flight states of the full table were computed by CFD. This method extends a few calculations fidelity to the entire flight envelope at a reduced cost, enabling the use of CFD for aerodynamic predictions. Hence, the aerodynamic table accuracy is improved for a more accurate flight simulation. Furthermore Kriging interpolation allows studying different configurations with a good level of accuracy and without an excessively high computational cost as shown in the continuation of this study.

#### 4.2.4 Data Fusion

The aerodynamic force and moment coefficients can be obtained from test or computed by using various methods predictions. Data fusion can combine different aerodynamic model results into a more accurate database. Considering the available data coming from the empirical method DATCOM (cheap and low-fidelity) and a CFD solver Edge (expensive and high-fidelity), it is possible to fuse them considering the new function with the same trend as the cheap samples but with quantitative values given by the expensive samples. Usually the cheap estimates are more densely distributed over the domain compared to the expensive ones. The method extensively employs the Kriging interpolation function, and it is based on the creation of an interpolated function whose domain is increased by one dimension with the values of the cheap computation at the expensive locations. The function is then evaluated over the entire domain as explained in Reference [13]. About this study DATCOM was considered to be the low fidelity method, which was used to compute all the domain points. Expensive high fidelity samples were taken from Edge, and they were densely computed only in transonic field. Furthermore 5 samples were taken on the border at low speed (Mach 0.1) to avoid extrapolation problems. The data fusion results about the pitch moment are shown in Fig. 4.6.



**Figure 4.6:** Dependency of pitch moment coefficient on angle of attack and Mach number; the surface represents data fusion between DATCOM and Edge

The trend of data fusion model was influenced by the DATCOM data, but the values were corrected by the Edge samples. The data fusion model exhibits good capabilities at high Mach number and high angle of attack. Nonlinear aerodynamic effects can be still seen for the contributions of Edge samples. Data fusion, combined with Kriging interpolation, allows computing a full aerodynamic model with an overall low computational cost. The contribution of DATCOM is not very evident because of the linear trend, but if a nonlinearity appeared in it, it would be transposed in

the fused data, adding important information to the new model. During this study different fidelity aerodynamic methods, Kriging interpolations and data fusion models were used here to investigate their impacts on aircraft trim properties. The models that will be used in the following section are shown in Table 4.2. In the next section the trim and handling qualities resulting from the tables are compared. The proximity of the Kriging interpolation of the Edge samples database compared to the one using the wind tunnel data is assessed. During the conceptual design phase wind tunnel tests are usually too expensive to be integrated into the design cycle and so the T3 database can be a cheaper but well representative alternative.

Source of data	Table
DATCOM	T1
Tornado	T2
Edge with Kriging interpolation	T3
Data fusion of T3 and wind tunnel data	T4

**Table 4.2:** Generated full tables and reference numbers

## 4.3 Results

### 4.3.1 Baseline Geometry

The equations of motion, governing the static and dynamic behaviour of the aircraft, are composed by the aerodynamic forces and moments acting on the aircraft, geometrical and mass features, the command variables, notably throttle and elevator, rudder and aileron deflections and the state variables [38]. During the conceptual design phase, usually there is no accurate mass and inertia properties and so some approximated empirical methods are adopted. In this work, weight and balance data were predicted by Howe's empirical method integrated in the CEASIOM Weight & Balance module. It is important to consider that the mass properties heavily influence the trim and stability results, and so these must be only considered as trend indications. Table 4.3 shows the main mass and inertia values of the baseline configuration. The resulting maximum take-off weight is similar to the values of the existing aircraft used as reference for this work.

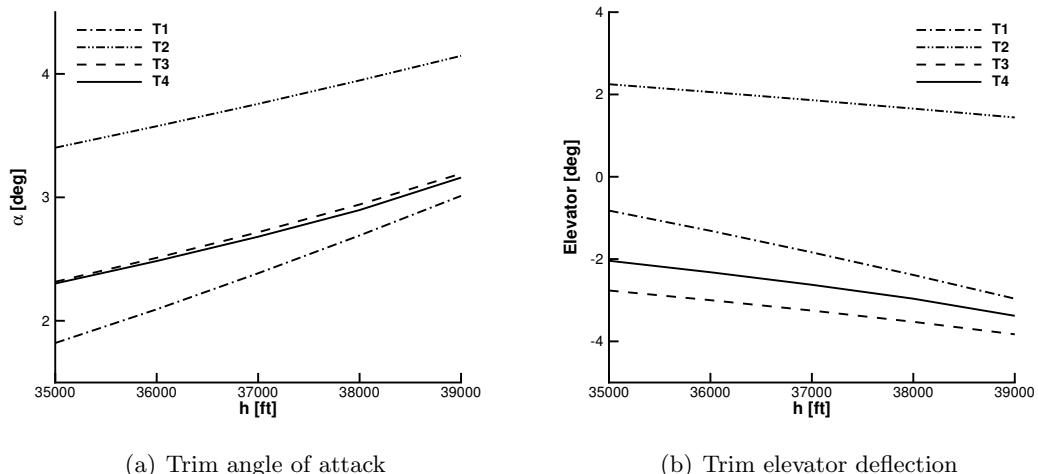
#### 4.3.1.1 Trim and Stability

In order to longitudinally trim the aircraft in the whole flight envelope, e.g. for different values of angle of attack, the traditional way to obtain that is by a variation in pitching moment values by regulating the incidence angle of the horizontal tail plane. This may be achieved by use of the control surface or by rotating the whole horizontal tail.

Parameter	Value
MTOW	$4.97 \cdot 10^4$ kg
CG location	(16.13, 0, 0.0236) m
$I_{xx}$	$1.11 \cdot 10^6$ kg·m <sup>2</sup>
$I_{yy}$	$1.90 \cdot 10^6$ kg·m <sup>2</sup>
$I_{zz}$	$2.87 \cdot 10^6$ kg·m <sup>2</sup>
$I_{xz}$	$6.72 \cdot 10^4$ kg·m <sup>2</sup>
$I_{xy}, I_{yz}$	0 kg·m <sup>2</sup>

**Table 4.3:** Baseline mass and inertia computed properties

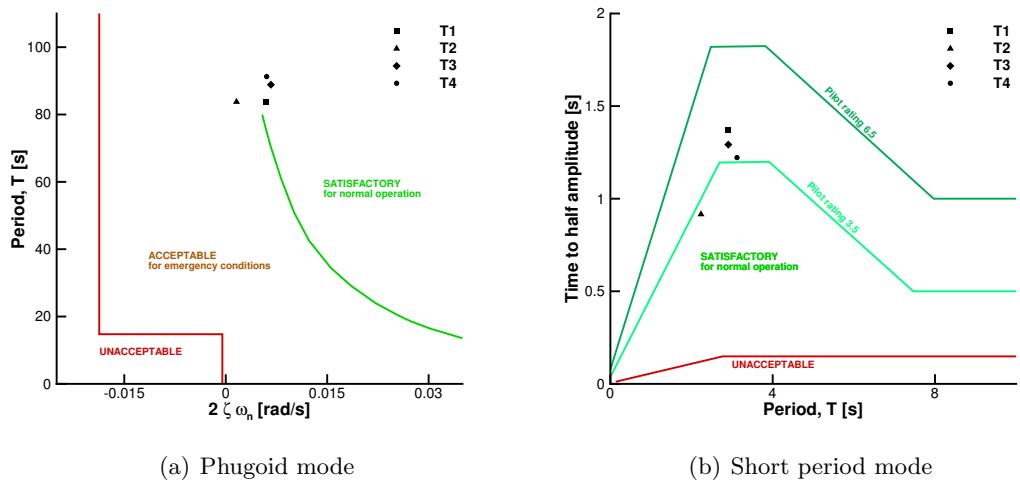
The Mach number was fixed at 0.78 and the flight altitude was varied between 10,668 and 11,887 m, which is the step cruise height. Figure 4.7 shows the trim angles of attack and deflection of elevator, respectively, at different altitudes during the cruise mission phase. The aerodynamic model considering the wind tunnel experimental data (T4), which is expected to have the highest fidelity, is taken as reference solution for all methods. Tornado model (T2) is the only one with a positive elevator deflection for trimming the aircraft. DATCOM table (T1) identifies a larger slope of the trim angle of attack increasing the altitude. The results obtained from the Edge database (T3) are very similar to the values computed with wind tunnel data (T4), obtaining an almost identical trimming angle of attack and a slightly smaller elevator deflection but with the same trend.



**Figure 4.7:** Trim conditions at different cruise altitudes; see Table 4.2 for tables definition

#### 4.3.1.2 Longitudinal handling qualities

Aircraft handling qualities at Mach cruise of 0.78 and altitude of 11,887 m were evaluated using the resulting aerodynamic force and moment coefficients given by different aerodynamic models. Figure 4.8 shows the phugoid and short period characteristics, respectively, according to International Civil Aviation Organization (ICAO) criteria. About the phugoid longitudinal dynamic mode, the models show similar results, with the lowest rating and the farthest being Tornado (with a smaller damping).



**Figure 4.8:** Phugoid and short period characteristics according to ICAO criteria at Mach number of 0.78 and altitude of 11,887 m; see Table 4.2 for definition

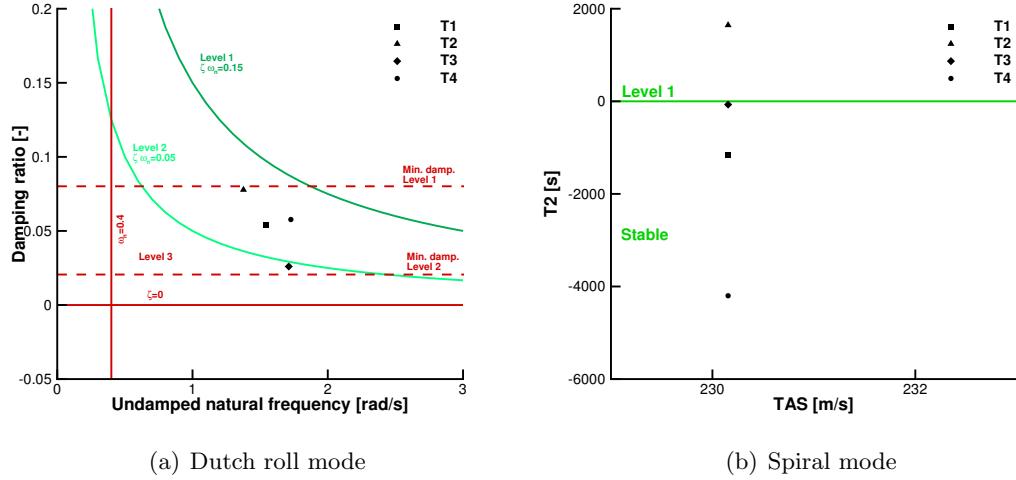
The short period mode for the 4 considered methods has almost the same period, but different times to half amplitude. The worst rating case is Tornado and the best is DATCOM, with the other two in the middle. According to ICAO, both the phugoid and short period are acceptable for all the tables.

As for trim, linear panel method is suspected in transonic regime, unless corrections are made. However, corrections require experience and database of existing aircraft and are inadequate to support engineers to design innovative configurations.

#### 4.3.1.3 Lateral-directional handling qualities

As previously done with the longitudinal handling qualities of aircraft, in this section the lateral-directional handling qualities are investigated. The flight conditions are Mach cruise of 0.78 and altitude of 11,887 m. Figure 4.9 shows the dutch roll and spiral modes characteristics, according to the United States military standards MIL-F-8785C criteria at Phase B. The dutch roll mode shows different results for every considered table. The less damped mode is obtained using Edge database (T3) that may be caused by the neglected viscosity term in the fluid dynamic equations. The Tornado database

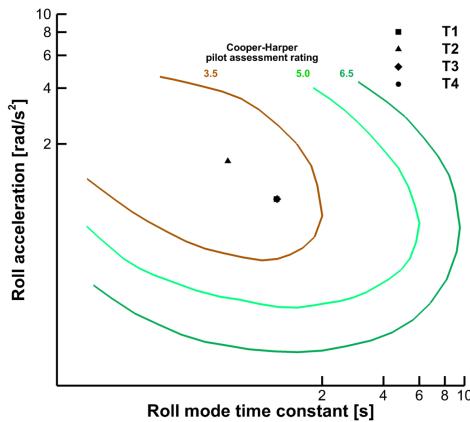
(T2) is the only one to present an unstable non-conservative behaviour in damping. The natural frequency does not change consistently. According to MIL-F-8785C, the results are acceptable for all the tables.



**Figure 4.9:** Dutch roll and spiral characteristics according to MIL-F-8785C criteria at Phase B at Mach number of 0.78 and altitude of 11,887 m; see Table 4.2 for definition

For the spiral mode, Tornado (T2) is the only one to lead to an unstable mode. This means that for Tornado the static directional stability has a relatively higher level compared to lateral stability and dihedral effect [38].

The roll characteristics according to Cooper–Harper criterion are then showed in Fig. 4.10. According to Cooper–Harper criterion the rating of the roll mode characteristics are good (Cooper–Harper pilot assessment rating under 3.5). All the results are almost coincident apart from Tornado (T2). The presented trim and handling qual-



**Figure 4.10:** Roll characteristics according to Cooper–Harper criterion at Mach number of 0.78 and altitude of 11,887 m; see Table 4.2 for definition

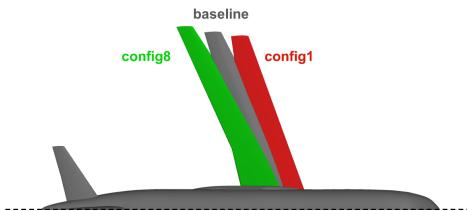
ties analysis show that the Kriging interpolation of the Edge computed samples (T3) are a good approximation of the table based on the wind tunnel test data (T4). The trend is very similar, and the off-set is always small. For some analysis, e.g. short period or phugoid mode, the resulting values are almost identical. For this reason the table obtained with Kriging interpolation of the CFD samples can be considered a good alternative to wind tunnel tests during the conceptual design phase.

### 4.3.2 New Wing Geometry

Having investigated the stability characteristics of the baseline geometry and the influence of aerodynamic models on the predicted static and dynamic behaviour, the next step is to explore whether aircraft characteristics may be improved by a better geometry design. For every new configuration a new aerodynamic table needs to be generated. With the tools and methods described above, the cost of this task is manageable even using CFD as source of the aerodynamic data. Using Kriging interpolation and data fusion was possible to compute a high-fidelity aerodynamic full-database with a few high-fidelity computation for every case. To study the geometry impact on aerodynamics and handling qualities, 8 configurations shown in Table 4.4 were considered starting from the baseline. The parameters include leading edge sweep angle ( $\Lambda_{LE}$ ) and aspect ratio ( $AR$ ). Figure 4.11 show the configuration geometry differences between baseline and Configuration 1 and 8 respectively.

Configuration	$\Lambda_{LE}$ [deg]	$AR$ [-]
Baseline	27	9.4
Configuration 1	20	9.0
Configuration 2	20	9.4
Configuration 3	20	10.0
Configuration 4	27	9.0
Configuration 5	27	10.0
Configuration 6	33	9.0
Configuration 7	33	9.4
Configuration 8	33	10.0

**Table 4.4:** Configurations with different leading edge sweep angle ( $\Lambda_{LE}$ ) and aspect ratio ( $AR$ )

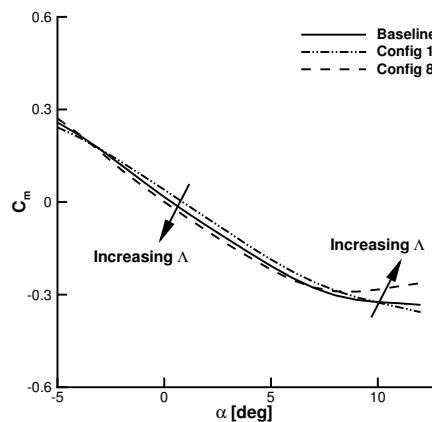


**Figure 4.11:** Configurations 1 and 8 geometry compared with the baseline

For every configuration the mass and inertia properties were computed with Howe's method and a new aerodynamic table was computed. Since the differences between new configurations and baseline are not large, the same flow topology assumption was used here. Only 18 CFD samples at the border of the flight envelope were computed for every new configuration. Using data fusion between the baseline configuration table obtained with the Kriging interpolation of the Edge data (T3), and the new samples at the border, a full table was computed for every new considered configuration. The aim of such modifications was to simulate a real design cycle, where different geometries are screened to find the best one.

#### 4.3.2.1 Aerodynamic characteristics

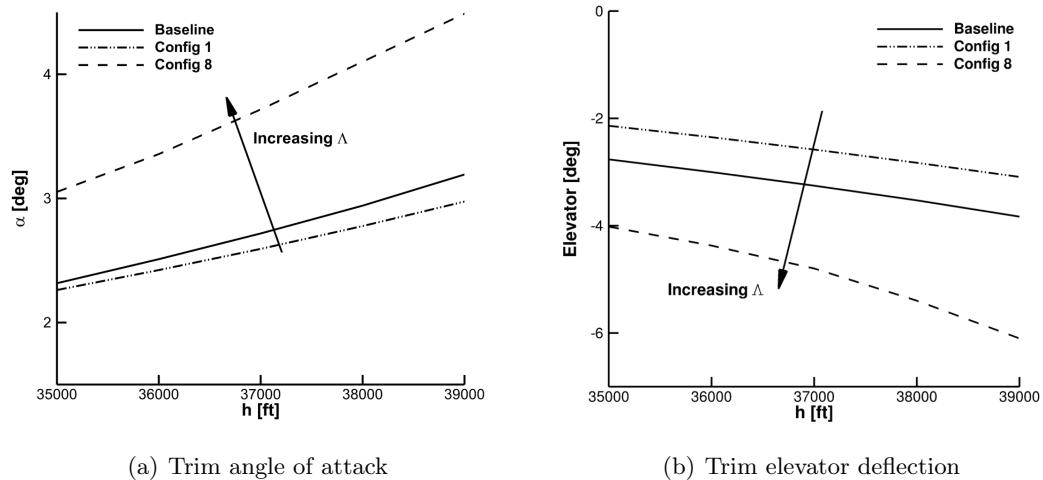
The considered databases are now based on the Edge predictions for all the configurations. About the baseline the table is generated by using Kriging interpolation of the 97 computed samples (T3), while for the new configurations it is generated by data fusion between the baseline table (T3) and the 18 samples computed for each geometry. Since all the results are located between the values obtained for Configuration 1 and 8, in this section only the results of these two configurations are shown and compared with the baseline. Figure 4.12 shows the pitch moment coefficient trends comparison between baseline and new configurations. The lift coefficient, drag coefficient and pitch moment coefficient of new configurations are close to the baseline results. The most influencing parameter is the sweep angle, increasing it the pitching moment decreases at lower angle of attack and increases faster when close to the stall.



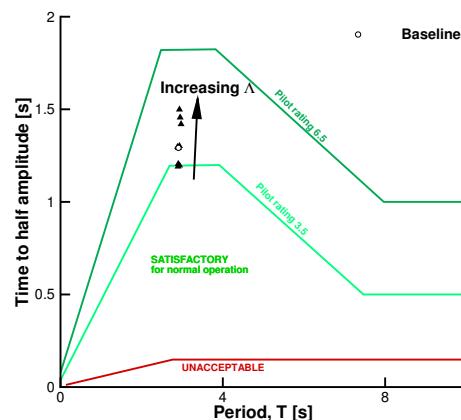
**Figure 4.12:** Pitching moment coefficient comparison at Mach number of 0.78 and altitude of 11,887 m; see Table 4.4 for definition; arrows indicate increasing values of wing sweep angle

#### 4.3.2.2 Trim and Handling Qualities

After computing a full-order aerodynamic model for all the configurations with the aerodynamic tables previously presented, it was possible to carry out an analysis on the impact of the modified design parameters over the handling qualities. The trim angle of attack and elevator deflection results at step cruise stage for all the configurations and baseline are shown in Fig. 4.13. The most influencing parameter over the trim conditions is the sweep angle. Increasing it with respect to the baseline configuration, both the angle of attack and elevator



**Figure 4.13:** Trim conditions at different cruise altitudes; see Table 4.4 for definition; arrows indicate increasing values of wing sweep angle



**Figure 4.14:** Short period characteristics according to ICAO criteria at Mach number of 0.78 and altitude of 11,887 m; arrows indicate increasing values of wing sweep angle

For the 8 configurations presented in Table 4.4, the phugoid characteristics do not change considerably with respect to the baseline configuration. The period  $T$  is  $90 \pm 2$  seconds and the damping ratio  $2\zeta\omega_s$  is  $0.045 \pm 0.01$  rad/s. The short period characteristics improve to some extent when increasing the sweep angle ( $\Lambda_{LE}$ ). The trend of short period characteristics between baseline and 8 configurations are shown in Fig. 4.14. Increasing the sweep angle, the short period mode shows better characteristics increasing the time to half amplitude. At the same time the angle of attack and elevator deflection to trim the aircraft increase in module, causing an higher drag. So increasing the baseline sweep angle seems to lead to better handling qualities but decreases the aerodynamic efficiency.

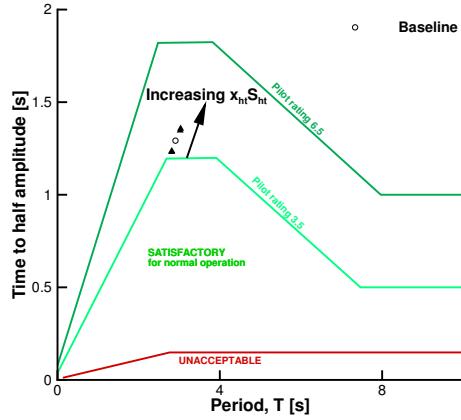
### 4.3.3 New Horizontal Tail Geometry

Other four configurations were created modifying the horizontal tail. The difference is the horizontal tail area ( $S_{ht}$ ) and position on the fuselage ( $x_{ht}$ ). The horizontal tail of configuration 11 was moved forward by 3%, while configuration 12 was moved backward by the same value. The horizontal tail area of configuration 13 was increased by 10%, while configuration 14 decreased by the same value. Table 4.5 shows the position and area of horizontal tail for these four configurations compared to the baseline. The value

Configuration	$x_{ht}/l_{fus}$ [-]	$S_{ht}$ [ $m^2$ ]
Baseline	0.8364	23.78
Configuration 11	0.8664	23.78
Configuration 12	0.8064	23.78
Configuration 13	0.8364	26.16
Configuration 14	0.8364	21.40

**Table 4.5:** Configuration with different horizontal tail

of the horizontal tail position in table was divided by the length of fuselage ( $l_{fus}$ ). Data fusion was also used to get the aerodynamic data and Edge Euler solver was used to compute additional data for these four configurations. The phugoid characteristics do not change much by comparing with the baseline. The period  $T$  is  $89.5 \pm 1.5$  seconds and the damping ratio  $2\zeta\omega_s$  is  $0.049 \pm 0.025$  rad/s. The short period characteristics were improved by increasing the horizontal tail dimensional volume  $x_{ht} S_{ht}$ . The trend of short period characteristics between baseline and 8 configurations are shown in Fig. 4.15. As for the wing sweep angle analysis, the same considerations may be done for the horizontal tail position. Increasing the horizontal tail volume, the phugoid mode increases the time to half amplitude but a longer fuselage or a bigger horizontal tail cause an higher drag. So increasing the baseline horizontal tail surface or position, better handling qualities were obtained but aerodynamic efficiency was reduced.



**Figure 4.15:** Short period characteristics according to ICAO criteria at Mach number of 0.78 and altitude of 11,887 m; arrows indicate increasing values of horizontal tail volume

## 4.4 Conclusions

The performance and stability characteristics of a regional jet design are presented. The baseline configuration is studied with different aerodynamic models. DATCOM results are very close to computational fluid dynamics and wind tunnel experimental data, underlining the effectiveness of the empirical method DATCOM for a traditional configuration. Some differences were found between the vortex lattice method Tornado because of linear assumptions. The CFD method Edge is the only one to capture the nonlinearity observed during the wind tunnel tests. The stall angles of attack predicted by Edge and measured in the wind tunnel are very similar but the CFD computation obtained a smaller effect from the nonlinear aerodynamics. The Kriging and data fusion methods are used to fill the aerodynamic tables with a reduced computational cost. The generated aerodynamic models are then compared and performance and stability characteristics computed. Among the computed models, Tornado is the only one to predict a positive elevator deflection to trim the aircraft during cruise and an unstable behaviour for the spiral mode. The impact of geometry changes on the aircraft dynamic characteristics is then investigated. The wing sweep angle is found to have a larger influence than the wing aspect ratio on the performance and stability characteristics. Increasing the wing sweep leads to a higher angle of attack and a less negative elevator deflection to trim the aircraft. Furthermore a more damped short period mode is obtained for higher sweep angle of the wing. The short period mode is influenced by the horizontal tail size and position. A higher tail volume coefficient causes a more damped short period.



## Chapter 5

# Alternative aerodynamic models for flight dynamics

The traditional aircraft aerodynamic model formulation is based on aerodynamic table or aerodynamic derivatives. The table presents the external aerodynamic forces and moments for every entry of a discrete flight envelope domain. The derivatives instead linearize the external actions close to the steady flight cruise conditions.

A schematic example of the aerodynamic table is given in Table C.1. Static and dynamic states are combined with control variables to generate a model that cover the full aircraft flight envelope. For every point the aerodynamic model is then exploited to compute the external forces and moments. This model has a few limitations since the flight envelope is presented in a non-continuous form, and so any middle point has to be approximated via interpolation. Furthermore, for physics based models, all the entries with non-zero dynamic state variables have to be computed with a time accurate solution, approximating the time history as a pure harmonic. This is a strong hypothesis since the state variables during the flight can change following any generic time history. The intrinsic nonlinear nature of the fluid dynamics governing equations lead to potentially big errors caused by this approximation. For example, different approximated position of eddies, shocks or boundary layer separation may lead to very different pressure distribution over the body surface.

Bryan's method approximation traditionally used to write the motion equation and to compute the static and dynamic aircraft properties, is based on the linearization of the aerodynamic response respect to the states and control variables, starting from a planar steady flight state in cruise conditions. Such approximation is valid only under small perturbations hypothesis, and consider a linear dependence between aerodynamic actions and parameters [38].

Both the formulations are based on very restrictive hypothesis, and so their validity in enclose in a small portion of the flight envelope. In case of time history involving fast and complex manoeuvres, the linear assumptions lead to important errors for the

evaluation of the aerodynamic forces and moments of the aircraft. The existing flight simulators are usually based on a tabular aerodynamic model. The use of inaccurate aerodynamic forces and moments for deriving the motion may cause misleading results that, time integrated, generate very different states. Let suppose, for example, the landing mission phase starting from the exact same condition: the real aircraft may land if a certain (non unique) elevator deflection and throttle time history is executed, but if the same commands time history is imposed in a flight simulator with a too strictly aerodynamic model, the simulation might lead to very different results, in the worst case leading to aircraft stall or crash. In this way an inaccurate flight simulator used for training might require the wrong command sequences.

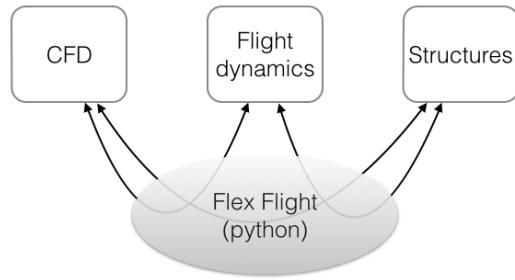
The traditional methods can be accurate and well representative models, only for standard flight conditions, with not too fast and complex manoeuvres. A more precise and not restricted method is presented in this Chapter. FlexFlight is a *Python* programme that implements time accurate models. The programme architecture is disclosed in the next Section, however the concept is based on a time accurate solver of a flexible flying aircraft. The structure deformation, the equation of motion and the CFD solver interact between each other at every time step to obtain a time accurate result. This model is no more based on the assumptions for a priori aerodynamic table generation and the simulation results are only dependent on the type of adopted models (e.g. VLM, Euler, RANS, etc. for the aerodynamic model and beam, half-shell, etc. for the structure).

## 5.1 Architecture

FlexFlight is a *Python* script that aims to intermediate between different blocks. A module represents one of the disciplines for the study of a flexible moving object in a fluid medium: CFD, motion equation integrator and structure. Any programme solving one of the involved disciplines may be inserted in the corresponding block, allowing the user to make it communicate with the others blocks. The modules interfaces are created in a way that each discipline can either be used as standalone for single discipline analysis or be imported as a module by a higher level class or script that performs a multi-disciplinary analyses.

Figure 5.1 shows the Flex Flight modular architecture. In the presented case the aerodynamic model block is filled with the *C* written parallel meshless (PML) CFD solver [39] developed at the University of Liverpool and both the flight dynamics and structures blocks with a *Fortran* written code [40].

This architecture allows the user to add any new required block, e.g. an auto-pilot as controller, making it able to communicate with any other block. Furthermore the open source *python* programming language does not impose any bounds about the



**Figure 5.1:** FlexFlight architecture scheme

programming language of the blocks functions, allowing to exploit any new or already developed scripts.

In order to perform coupled fluid/structure/flight analyses, a mesh deformation tool is needed to transfer information between the structural and fluid solver. The transfer of information between non-matching fluid and structural grids is here implemented with a linear fluid/structure interface. The scheme is exploited to transfer aerodynamic loads from the aerodynamic to the structural model, and map structural deformations from the structural to the aerodynamic model at no additional cost. For time accurate coupled solutions, a fully implicit partitioned approach is used. The solution sequencing between the fluid, flight and structural models is achieved within the pseudo-time stepping iterations. For every real time steps, inner sub-iterations are run calling all the modules, until convergence is reached.

For further information about the used blocks formulations the reader is referred to [39, 40].

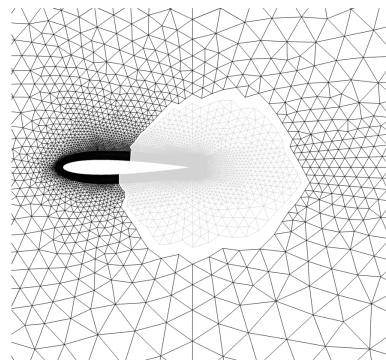
## 5.2 Validation of the CFD module PML

### 5.2.1 Two dimensional steady state

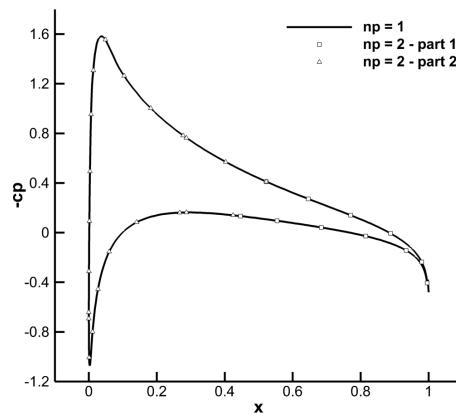
A two dimensional steady flow simulation of a NACA 0012 aerofoil is considered. The flow conditions are  $M = 0.6$  and  $\alpha_0 = 3.16$  deg at sea level in standard atmospheric conditions. The flow is modelled using the Euler equations on a hybrid grid of 12,304 point. The mesh is structured normally to the aerofoil surface for a distance of  $0.1 \cdot c$  and the rest of the domain is filled with an unstructured grid. The mesh convergence analysis is performed in [39]. Results from the CFD solver PML are generated and compared with the solution obtained using FlexFlight in CFD standalone mode.

### 5.2.1.1 Stand alone results comparison for one or two processors computations

In order to be able to use two processors during the computation, the pre-processor has to be run. This programme divides the domain into two parts, each of which will be computed by a single processor. Figure 5.2 shows the mesh division for two processors. The resulting pressure coefficient ( $C_p$ ) over the chord is then shown in Fig. 5.3. As required by the multi processor run, the obtained results do not change.



**Figure 5.2:** 2 dimensional NACA 0012 two processors grid division



**Figure 5.3:**  $C_p$  distribution for the CFD solver with one and two processors

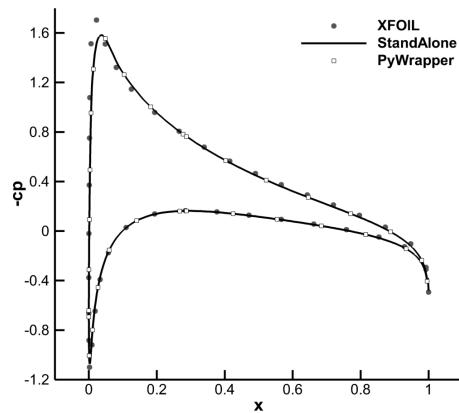
The resulting forces and moments coefficients for the NACA0012 at  $M = 0.6$  and  $\alpha_0 = 3.16$  deg are summarized in Table 5.1.

$\alpha_0$ [deg]	$C_D$ [-]	$C_Y$ [-]	$C_L$ [-]	$C_m$ [-]
3.16	$-2.16 \cdot 10^{-3}$	0.0	$5.07 \cdot 10^{-1}$	$-2.48 \cdot 10^{-3}$

**Table 5.1:** Resulting external aerodynamic forces and moments

### 5.2.1.2 Stand alone and FlexFlight results comparison for one processor computation

A comparison between the resulting pressure coefficient over the chord between stand alone CFD computation and via the Python interface in FlexFlight is shown in Fig. 5.4 using one processor. Furthermore, in order to have an external validation, the non viscous results obtained from the open source code XFOIL [11]<sup>1</sup> are presented. No



**Figure 5.4:**  $C_p$  distribution on the aerofoil for the stand alone CFD solver and via Python

difference appear in the PML results and so the interface does not interfere with the fluid dynamic computation. The XFOIL pressure distribution is very similar, with the biggest difference in the upper part of the aerofoil where the pressure is minimum.

### 5.2.1.3 Computing time

The results were obtained both with the stand alone flow solver PML and with the external Python interface FlexFlight. In Table 5.2 the computing time is shown for one and two processors computations.

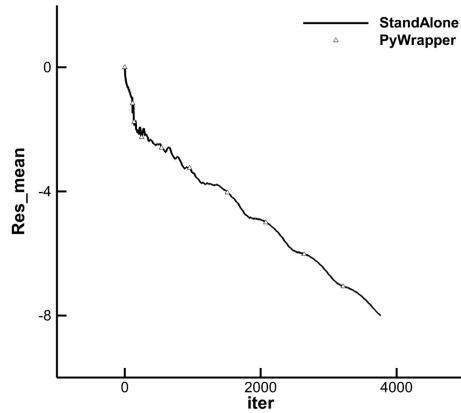
Number of processors	Stand Alone	FlexFlight
1	208 [s]	224 [s]
2	109 [s]	120 [s]

**Table 5.2:** Computation time results for a mean residual of  $10^{-8}$

The resulting computational time for two processors is roughly the half of the one with a single processor. FlexFlight check if the tolerance is reached every  $n$  steps (in this case  $n = 200$ ). The time difference between the stand alone and the FlexFlight interface is given due to the fact that the computation does not stop as soon as the tolerance is reached. Figure 5.5 presents the residual trend for a single processor computation for

<sup>1</sup><http://web.mit.edu/drela/Public/web/xfoil/> [retrieved October 10, 2014]

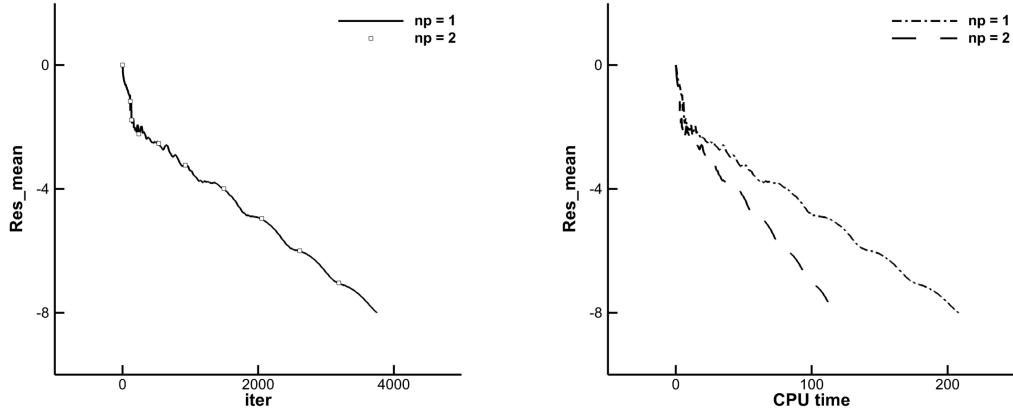
the stand alone and the FlexFlight interface is shown. The resulting trend is the same



**Figure 5.5:** Residual trend for the stand alone solver and with the FlexFlight interface

and so this prove that the wrapper does not influence the computation at all.

The residual behaviour for a single processor computation is compared with a two processors computation for the CFD stand alone solver in Fig. 5.6. The number of iteration to get the required tolerance ( $10^8$ ) is the same but the processor unit (CPU) time is much less if two processors are working simultaneously.



(a) Residual against iteration number for one and two processors computations (b) Residual against CPU time for one and two processors computations

**Figure 5.6:** Residual trends for one and two processors computations

#### 5.2.1.4 Conclusion

The pressure coefficient results are identical for all the considered studies. Neither the multi processor usage nor the FlexFlight interface cause any difference in the results.

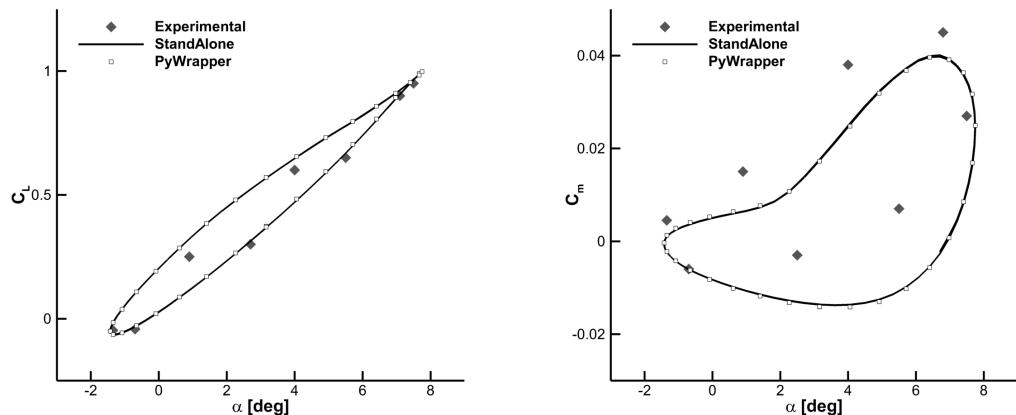
The only notable difference is about the computing time that is close to be halved if two processors are used and is slightly bigger for the FlexFlight interface because the required tolerance is checked only on every external cycle and so the last internal cycle does not interrupt also if the input tolerance is obtained.

### 5.2.2 Two dimensional unsteady

A two dimensional unsteady flow simulation of a NACA 0012 aerofoil is considered. The same grid as for the steady solution is considered, and the same initial steady state is taken with flow conditions of  $M = 0.6$  and  $\alpha_0 = 3.16$  deg at sea level in standard atmospheric conditions. The unsteady solution is then started. A prescribed motion is generated considering an pitch oscillation of amplitude  $\alpha_A = 4.59$  deg and reduced frequency  $k = \omega c/(2 V_\infty) = 0.0811$ . Results from the solver PML are compared with results using the FlexFlight interface.

#### 5.2.2.1 Stand alone and FlexFlight interface results comparison for one processor computation

The comparison between the resulting lift and pitching moment coefficients over the time history using the stand alone CFD solver and via the wrapper Python script are shown in Figure 5.7. Experimental data extracted from the AGARD-R-702 study of an oscillating NACA0012 presented in [41] are included.



(a)  $C_L$  compared with the changing in time angle of attack  $\alpha$  (b)  $C_m$  compared with the changing in time angle of attack  $\alpha$

**Figure 5.7:** Lift and pitching moment for a pitching sinusoidal motion with amplitude  $\alpha_A = 4.59$  deg and reduced frequency  $k = 0.0811$

There is no difference in the results between the stand alone CFD solver and the solution computed through the interface, so the wrapper does not interfere with the

unsteady fluid dynamic computation. The experimental data are in good agreement with the computational results and the comparison is in line with what presented in [18].

### 5.2.2.2 Conclusion

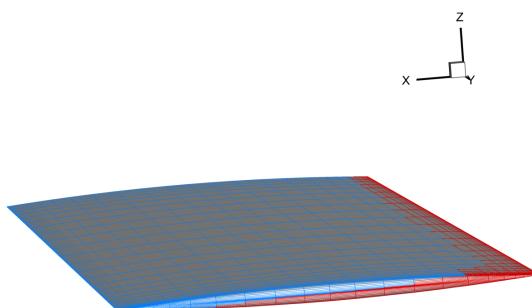
From the presented results the flow hysteresis phenomenon appears clearly. Anyway about the two obtained solutions the python wrapper does not interfere with the unsteady flow solver.

### 5.2.3 Three dimensional steady

A three dimensional steady flow simulation of a Goland wing is considered. The wing is a rectangular shape (wing span of 3.33 m and chord of 1 m) with elliptical aerofoil section. The flow conditions are Mach number of 0.6 and angle of attack of 3.16 deg. The flow is modelled using the Euler equations on a grid of 187,523 points. The values obtained from the solver PML are first compared in case one or two processors were used and then are likened with the results using the FlexFlight interface.

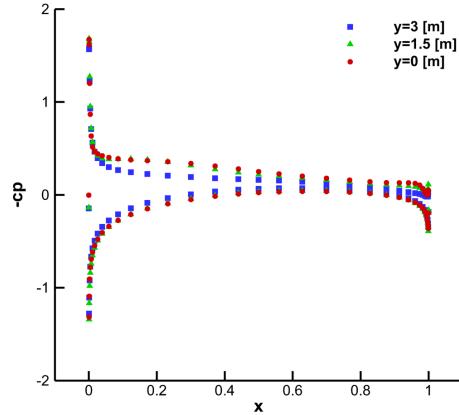
#### 5.2.3.1 Stand alone results comparison for one or two processors computations

In order to be able to use two processors during the computation a pre-processor script have to be run. This executable divides the domain into various parts, each of which will be computed by a single processor. Figure 5.8 shows how the three dimensional domain is divided for two processors.



**Figure 5.8:** Two processors Goland wing surface mesh division

The resulting pressure coefficient ( $C_p$ ) over the chord at the wing root, and 1.5 m and 3 m from it is then shown in Fig. 5.9. As required by the multi processor run,



**Figure 5.9:** Pressure coefficient on the chord at different location along the wing span

the obtained external forces aerodynamic coefficients are identical. The resulting forces and moments coefficients are summarized in Table 5.3.

$\alpha$ [deg]	$C_D$ [-]	$C_Y$ [-]	$C_L$ [-]	$C_\ell$ [-]	$C_m$ [-]	$C_n$ [-]
3.16	$4.86 \cdot 10^{-2}$	$3.96 \cdot 10^{-3}$	$9.45 \cdot 10^{-1}$	1.42	$-4.74 \cdot 10^{-2}$	$6.47 \cdot 10^{-3}$

**Table 5.3:** Half wing resulting force and moment aerodynamic coefficients

### 5.2.3.2 Stand alone and FlexFlight interface results comparison for one processor computations

No difference appears in the results between the solution obtained with the standalone programme and the Python interface code. So the Python script does not interfere with the fluid dynamic computation.

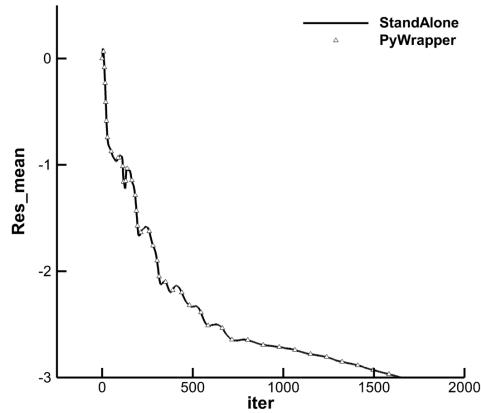
### 5.2.3.3 Computing time

The results were obtained both with the stand alone flow solver PML and with the external FlexFlight interface. Table 5.4 presents the computing time for one, two and four processors computations.

Number of processors	Stand Alone	PyWrapper
1	5,394 [s] (1h29'54")	5,904 [s] (1h38'24")
2	3,427 [s] (57'07")	2,523 [s] (42'03")
4	1,704 [s] (28'24")	2,425 [s] (40'25")

**Table 5.4:** Computation time in seconds for a mean residual of  $10^{-3}$

The resulting computational time with a double number of processors is roughly the half. The time difference between the stand alone and the FlexFlight interface is given due to the fact that the computation does not stop as soon as the tolerance is reached, and more output writing operations are executed during the process. Figure 5.10 presents the residual behaviour for four processors computation for the stand alone and the FlexFlight interface.

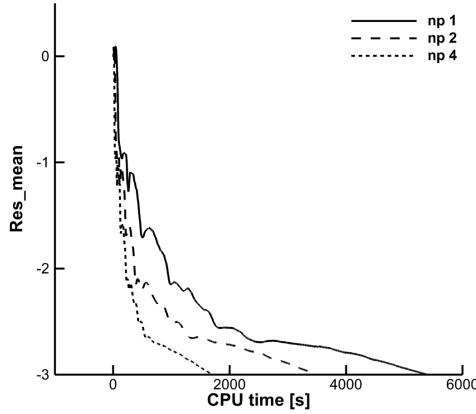


**Figure 5.10:** Residual trend for the stand alone solver and with the FlexFlight interface; computation with 4 processors

The resulting trend is the same and so it proves that the wrapper does not influence the computation. The residual trend respect to the central processor unit (CPU) time for a single processor computation is compared with a two and four processors computations for the CFD stand alone solver is presented in Fig. 5.11. The number of iteration to get the required tolerance ( $10^{-3}$ ) is the same but the overall computational time decreases if more processors are used.

#### 5.2.4 Conclusion

The aerodynamic force and moment coefficients results are identical for all the considered studies. Neither the multi processor usage nor the FlexFlight interface cause any difference in the results. The only notable difference is about the computing time that is close to be halved if the double number of processors are used and is slightly bigger for the FlexFlight interface because the required tolerance is checked only on every external cycle and so the last internal cycle does not interrupt also if the input tolerance is obtained.



**Figure 5.11:** Residual temporal trend for 1,2 and 4 processors computations; the stand alone solver is used

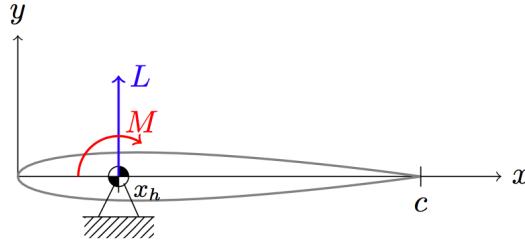
### 5.3 Flight dynamics and CFD coupling

The aim of the creation of FlexFlight is the ability to generate a versatile tool for simulating a flying flexible aircraft. The model includes motion equation integration, flow field CFD solution and structural deformation. The model can compute a time accurate solution of a flying deformable aircraft. The solution is obtained via time accurate solutions of the three disciplines, passing the necessary information between each other. The full model approximations are then only dependent on the single disciplines adopted models.

In this Section the CFD flow field solution is coupled with motion equations integration of the flight dynamics. The structural deformation is here neglected. Time step is validated and geometrical parameters, as initial angle of attack and hinge position, motion influence is investigated. Finally different aerodynamic models are adopted and the insight of the different results is presented.

The rigid body flight dynamic simulation considering the aerodynamic external forces of a 2 dimensional aerofoil NACA0012 is performed. The initial perturbed condition is at an angle of attack of 1 deg ( $\alpha_0$ , angle between flow and the chord). The aerofoil is constrained so that the only degree of freedom is the pitch angle. The gravity is not considered and the only external action is the aerodynamic pitching moment computed referring to the hinge. The considered chord dimension is 1 m and the mass per unitary span is 10 kg/m. The location of the center of gravity (CG) coincides with the hinge position. The torsional inertia moment respect to the CG is 10 kg·m for all the cases. A schematic sketch of the system is drawn in Fig. 5.12. The solution is computed on hybrid grids of 12,304 points for the Euler equations and on 15,017 points (refined close to the aerofoil surface) for the Navier–Stokes equations. The mesh is structured extruding the aerofoil surface for  $0.1 \cdot c$  and then unstructured; the first

layer for the RAN grid is  $0.00001 \cdot c$  thick. The considered time step is 0.0049 s (non-dimensional value  $\Delta\tau = \Delta t \cdot \frac{V}{c} = 0.2454369$  [-]). 320 time steps are computed, so that the simulation lasts 1,568 s. Resulting forces from the flow solver PML are used by the flight dynamics code XBEAM to find the rigid body rotation. The interface between the two codes is managed with the Python wrapper.



**Figure 5.12:** NACA 0012 with the reference frame and the aerodynamic external forces

### 5.3.1 Formulation

A simplified aerodynamic model studied in this section considers the single degree of freedom (DoF) system theory. The governing dynamic equations of the presented two dimensional test case are presented in Eq. (5.1).

$$m h'' = L - W \quad (5.1a)$$

$$m x'' = D - T \quad (5.1b)$$

$$I_{\alpha\alpha} \alpha'' = M_{\alpha} \quad (5.1c)$$

$m$  denotes the mass,  $I_{\alpha\alpha}$  is the inertia moment around the hinge,  $h$  and  $x$  are the vertical and horizontal translations respectively,  $\alpha$  is the rotation around the hinge starting from the chord aligned with the free stream velocity positive clockwise,  $W$  is the weight,  $T$  is the thrust,  $L$ ,  $D$ , and  $M_{\alpha}$  are respectively the aerodynamic lift, drag, and pitching moment around the center of gravity.

Considering the aerofoil vertical and horizontal translations fixed, the only degree of freedom is the pitch and so the angular moment equilibrium in the  $y$ -axis becomes the only governing equation of the 1 DoF system (Eq. (5.1c)). Equation (5.2) shows the rotational equilibrium around the hinge position in conventional aerodynamic forces explicit terms.  $M_{\alpha}^{q.c.}$  denotes the pitching moment at the quarter chord and  $x_h/c$  the position of the hinge starting from the aerofoil leading edge as fraction of the chord length.

$$M_{\alpha} = M_{\alpha}^{q.c.} + \left(x_h - \frac{c}{4}\right) \cdot (L \cos \alpha - D \sin \alpha) \quad (5.2)$$

Small perturbations hypothesis can be made for  $\alpha$  so that the equation can be linearized as presented in Eq. (5.3).

$$M_\alpha \simeq M_\alpha^{q.c.} + (x_h - \frac{c}{4}) \cdot (L - D \cdot \alpha) \quad (5.3)$$

### 5.3.1.1 Two-dimensional thin aerofoil Theodorsen theory

The flow field can be approximated by the analytical aerodynamic model formulated by Theodorsen [42] for an irrotational and incompressible two-dimensional flow. This theory is then more accurate for low Mach numbers. In the considered case the aerodynamic loads arise only from the aerofoil pitch motion as presented in Eq. (5.4). The members are all non-dimensional and the time derivatives are computed with the adimensionalized time  $\tau = t 2V_\infty / c$ . The presented  $C_m(\tau)$  is referred to the hinge location defined by  $a_h$  that is adimensionalized by the semi-chord  $\frac{c}{2}$ , defined as zero in  $\frac{c}{2}$  and positive toward the aerofoil trailing edge.

$$\begin{aligned} C_m(\tau) = & \pi (1/2 + a_h) [\alpha(0) + (1/2 - a_h)\alpha'(0)] \Phi_w(\tau) + \\ & \pi (1/2 + a_h) \int_0^\tau \Phi_w(\tau - \sigma) [\alpha'(\sigma) + (1/2 - a_h)\alpha''(\sigma)] d\sigma + \quad (5.4) \\ & \frac{\pi}{2} a_h (-a_h \alpha'') - (1/2 - a_h) \frac{\pi}{2} \alpha' - \frac{\pi}{16} \alpha'' \end{aligned}$$

Great attention must be given on this formulation since both time and pitching moment are adimensionalized by  $\frac{c}{2}$ , and not  $c$ . In the Eq. (5.4) the Wagner function  $\Phi_w$  accounts for the shed wake time history. The analytical expression is based in terms of Bessel functions, but for a practical evaluation the exponential approximated form of Jones [43] is given in Eq. (5.5), where constant values are  $\Psi_1 = 0.165$ ,  $\epsilon_1 = 0.0455$ ,  $\Psi_2 = 0.335$ , and  $\epsilon_2 = 0.3$ .

$$\Phi_w(\tau) = 1 - \Psi_1 e^{-\epsilon_1 \tau} - \Psi_2 e^{-\epsilon_2 \tau} \quad (5.5)$$

Starting from this analytical model Eq. (5.1c) can be rewritten as presented in Eq. (5.6). By integrating Eq. (5.6) is possible to compute the dynamic motion of the aerofoil and the system states time history.

$$\alpha''_{(\tau)} = \frac{C_m(\tau) \cdot 1/2 \rho v^2 c^2}{I_{\alpha \alpha}} \cdot (c/2V_\infty)^2 \quad (5.6)$$

### 5.3.1.2 Aerodynamic stability derivatives theory

A simplified aerodynamic model studied in this section considers a linear dependency of the pitching moment coefficient respect to the pitch and pitch velocity variation. Equation (5.3) may be further simplified: the drag effect on the pitching moment may be neglected and so is possible to group the lift and pitching moment coefficients respect

to the quarter chord as expressed in Eq. (5.7).

$$C_m \simeq C_m^{q.c.} + \left( \frac{x_h}{c} - 0.25 \right) \cdot C_L \quad (5.7)$$

In agreement to Bryan's theory, the linearized forms of the pitching moment coefficient can be considered. Variations with respect to the angle of attack value and time first derivative are considered. Furthermore the symmetry of the problem allows to neglect the  $C_{m_0}$  term: for low speed, chord aligned with flow direction and small aerofoil thickness, the aerodynamic can be considered linear and the pressure coefficient distribution symmetric, resulting in a zero pitching moment at  $\alpha = 0$  deg. Equation (5.1c) assumes then the classical mass–spring–damper form as presented in Eq. (5.8), representing the total rotational equilibrium for the external aerodynamic pitching moment and the linearized dynamical behaviour of the system.

$$\frac{I_{\alpha\alpha}}{1/2 \rho V_\infty^2 c^2} \alpha'' - C_{m_{\alpha'}} \alpha' - C_{m_\alpha} \alpha = 0 \quad (5.8)$$

In analogy with the mass–spring–damper equation, the approximated dynamic oscillation frequency and damping can be then easily computed as showed in Eq. (5.9).

$$\omega_0 = \sqrt{\frac{-(1/2\rho V_\infty^2 c^2) C_{m_\alpha}}{I_{\alpha\alpha}}} \quad (5.9a)$$

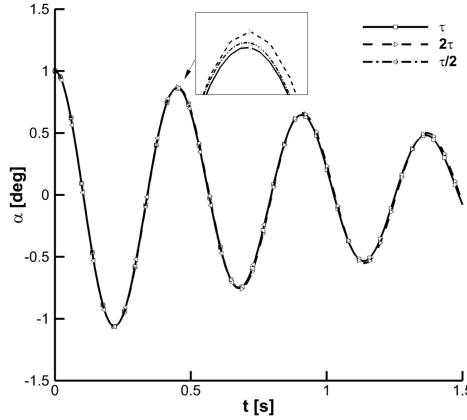
$$\zeta = \frac{-C_{m_{\alpha'}}}{2\sqrt{-C_{m_\alpha} \frac{I_{\alpha\alpha}}{1/2 \rho V_\infty^2 c^2}}} \quad (5.9b)$$

### 5.3.2 Results

The considered flow conditions is  $M = 0.147$  ( $V_\infty = 50$  m/s at sea level for standard conditions). The non-dimensional time can be computed as  $\tau = t \cdot (2V_\infty/c)$ . Baseline initial condition is 1 deg of angle of attack. The used tolerance was of  $1 \cdot 10^{-13}$  for the steady initial computation and  $1 \cdot 10^{-3}$  for the time steps coupling flight dynamic and the CFD code.

#### 5.3.2.1 Time step validation

In this section the exact same problem is analysed, but three different time steps are adopted. The baseline is of 0.0049 s, resulting in an non-dimensional  $\Delta\tau = \Delta t \cdot \frac{V}{c} = 0.2454369$  [–], and the double and the half of this value are adopted. Figure 5.13 shows the resulting free to pitch aerofoil motions with different time steps. No difference can be seen in the overall behaviour, also if some small differences appear when the solution is zoomed. This might be caused by the different integration step, and so a numerical



**Figure 5.13:** Time step effect on the flight-dynamic simulation; free stream velocity at 50 m/s and hinge position at 5% of the chord

error connected issue. Table 5.5 shows the difference between the time steps results. The differences are really small and so the time step can be considered to not influence the overall solution.

$\Delta t$ [s]	$\alpha_{maxII}$ [deg]	Variation %
0.0025	0.8647890	+0.4
0.0049	0.8612986	0
0.0098	0.8726739	+1.3

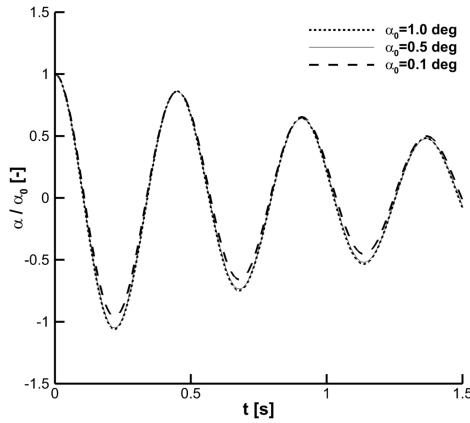
**Table 5.5:** First oscillation extrema for the NACA0012 with different time steps, starting from  $\alpha_0 = 1$  deg

### 5.3.2.2 Initial condition influence

Three simulations were run considering different initial condition:  $\alpha_0 = 1$ ,  $\alpha_0 = 0.5$ , and  $\alpha_0 = 0.1$  deg. The resulting aerodynamic coefficients follow the linearized aerodynamic theory, the drag is the same for all the cases ( $C_D \simeq -0.003$ ) while the lift and pitching moment are linearly dependent to the angle of attack. As previously disclosed, the lift and pitching moment are not present at zero incidence ( $C_{L0} = C_{m0} = 0$ ) and so the linearized dependency is  $C_L = 0.13207 \cdot \alpha$ , and  $C_m = -0.02887 \cdot \alpha$ .

In order to assess the impact of the initial condition on the initial fluid dynamic forces and the consequent motion, the pitch angle time histories were adimensionalized by the initial values and the resulting motion is presented in Fig. 5.14.

Table 5.6 shows the extrema value of amplitude of the first oscillation and the relative differences. A positive value of variation means that the aerofoil motion is non conservative and so it has taken some energy from the flow field around it, increasing the oscillation energy (e.g. for  $\alpha_0$  of 1 and 0.5 deg). For  $\alpha_0 = 0.1$  deg the system is



**Figure 5.14:** Adimensionalized pitch angle time history; free stream velocity of 50 m/s

damped starting from the first oscillation, and so the phenomenon is strictly related with the initial condition.

$\alpha_0$ [deg]	$\alpha_{min}$ [deg]	Variation %
0.1	-0.09594	-4.1
0.5	-0.52448	+4.9
1	-1.06044	+6.0

**Table 5.6:** First oscillation extrema for the NACA0012 with different initial conditions

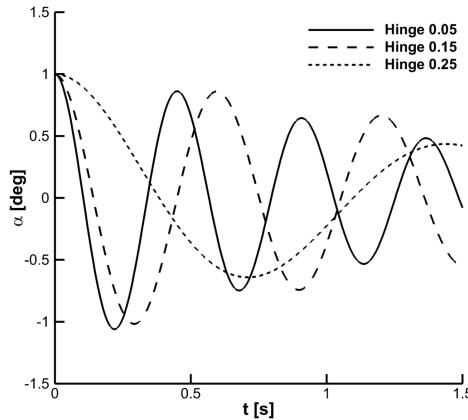
### 5.3.2.3 Hinge position effect

Simulation results for different position of the hinge are shown in Fig. 5.15. Hinge locations ( $x_h$ ) of 5, 15 and 25 % of the chord behind the leading edge are considered.

From thin aerofoil theory the centre of pressure for a low flying speed (50 m/s) aerofoil is at one quarter of the chord behind the leading edge ( $x_{a.c.} \cong c/4$ ). For this reason any position backward the 25% of the chord results in an unstable behaviour while any forward is stable. Equations (5.9a) and (5.9b) give the analytical approximated formulation of the oscillation characteristics. For a symmetric aerofoil there is neither lift nor pitching moment resultant for  $\alpha = 0$  deg ( $C_{m0} = C_{L0} = 0$ ) and so the resulting pitching moment coefficient expression is presented in Eq. (5.10).

$$C_m = C_L \cdot \frac{x_h - x_{a.c.}}{c} \quad (5.10)$$

The  $C_m$  is then expected to increase for hinge positions closer to the leading edge. This causes higher oscillation frequency and damping, both linearly dependent to the hinge position proximity to the leading edge ( $\omega_0 \propto \sqrt{x_h - x_{a.c.}}$  and  $\zeta \propto \sqrt{x_h - x_{a.c.}}$ ). It is important to consider that the rotational inertia around the hinge ( $I_{\alpha\alpha}$ ) is considered



**Figure 5.15:** Hinge position effect over the free to pitch aerofoil motion; free stream velocity at 50 m/s

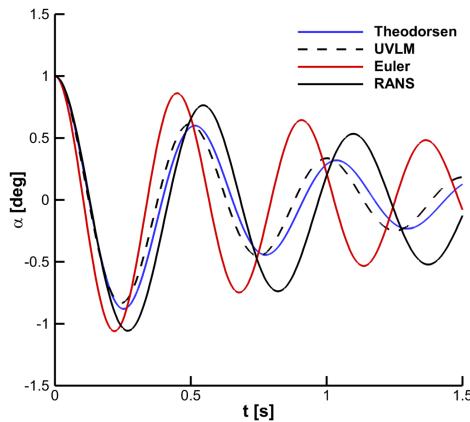
constant although this presumes a different density distribution over the structure (center of gravity location always coincident with the hinge position). A deeper analysis may be conducted using Huygens–Steiner theorem for the rotational inertia computation considering the center of gravity location fixed at the quarter of the chord, so that the total inertia increases for hinge position closer to the leading edge. This may causes a reduction of the previously assessed effect of the hinge position over the oscillation, and the inverse effect may appear for large values of mass so that a hinge position closer to the leading edge would lead to lower frequency and less damped motion oscillations. Furthermore the gravity should be then considered, generating an external steady force, that would make the system unstable unless the center of gravity is located exactly in the initial centre of pressure position ( $x_{c.g.} \equiv x_{a.c.}$  unstable equilibrium point).

About a 3 dimensional wing, the aeroelasticity might be studied approximating the wing via 2-D strip theory and the section connection is usually modelled with the elastic axes, whose chord wise position  $x_{a.e.}$  is represented in this study as the hinge position. However in aeroelastic model the structural connection between different wing sections is represented by a torsional stiffness, that is the presented study was not considered.

#### 5.3.2.4 Fluid–dynamic method comparison

Figure 5.16 shows the resulting free to pitch aerofoil motions, using different aerodynamic models. The Theodorsen thin aerofoil theory presented in Sec. 5.3.1.1 is used to compute instantaneously the external aerodynamic moment acting on the aerofoil. The motion equations are then integrated to compute the speed and rotational displacement. The dynamic solution is started from a steady state solution for  $\tau \rightarrow \infty$ , but for computational limits a value of 1.1 s (equal to  $\tau = 60$ ) was found sufficient to obtain a converged value of pitching moment. The  $C_{L_\alpha}$  value is extracted from the

Euler solution via linear interpolation. The used value is 5.9274 instead of  $2\pi$  from thin aerofoil theory, anyway the effect of this variation does not significantly influence the results. The unsteady vortex lattice method (UVLM) developed in [40] is also taken into account as aerodynamic model for the system dynamic analyses. Finally, both Euler and RANS CFD simulations from the flow solver PML are used coupled with the flight dynamics solver XBEAM. The presented motions are very different for a



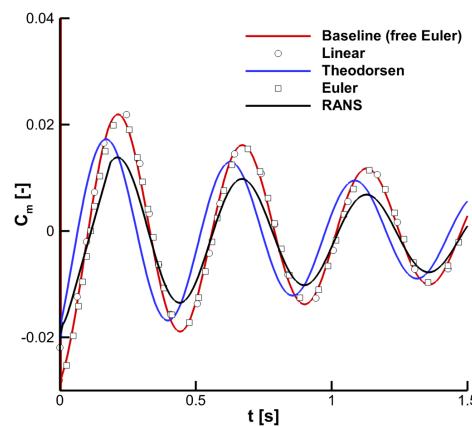
**Figure 5.16:** Fluid-dynamic method effect over the free to pitch aerofoil motion; free stream velocity at 50 m/s and hinge position at 5% of the chord

big time since the resulting dynamic is influenced from every previous state, and the integration enlarge the error. A good comparison can be seen in the very first half oscillation trend. All the methods present the same  $\alpha-\tau$  slope except for the Euler solution that has a bigger aerodynamic moment and so an higher frequency (see Eq. (5.9a)). The methods then split after the first minimum following different time histories. A very good analogy is however showed between the Theodorsen and UVLM models, both maintaining similar dynamical frequency and damping. Euler and RANS CFD models show similar damping values, but the oscillation frequency is much higher for Euler compared to RANS. A viscous or non-viscous flow model causes different resulting aerodynamic forces acting on the aerofoil surface, leading to mismatching results. Theodorsen and UVLM have frequencies in the between the two CFD models, but the damping is higher. The UVLM and Theodorsen models are based on linear aerodynamic assumptions, that is coherent for low speed analyses, but the integration of the motion equation do not consider nonlinear aerodynamic effect that may be caused by the unsteadiness nature in the flow field.

### 5.3.2.5 Free to forced motion comparison

In the following section a new comparison is made between the resultant forces of a free to pitch simulation and forced motion simulations.

The time history of the free to pitch aerofoil with Euler flow model is taken as baseline of this comparison. The free stream velocity is 50 m/s and the hinge located at 5% of the chord from the leading edge. The obtained  $\alpha(t)$  time history is then imposed as forced motion to dynamic systems governed by different aerodynamic models. The linear derivative aerodynamic model, the Theodorsen theory, the viscous and non-viscous flow solved via CFD are considered. Figure 5.17 presents a comparison between the resulting pitching moment coefficient time history obtained from the Euler free to pitch simulation, referred as **Baseline** and the results obtained imposing the motion to the different aerodynamic models.



**Figure 5.17:** Euler free motion applied to forced motion computations

The aim of this study is to analyse the different system response to the same input changing the model. A linear approximation is first considered between pitching moment and pitch angle. Considering the steady aerodynamic derivative theory, the pitching moment can be expressed as  $C_m = C_{m0} + C_{m\alpha} \cdot \alpha$ . Starting from Eq. (5.10) the formulation can be easily rewritten in terms of lift coefficient slope as presented in Eq. (5.11) (positive pitching up).

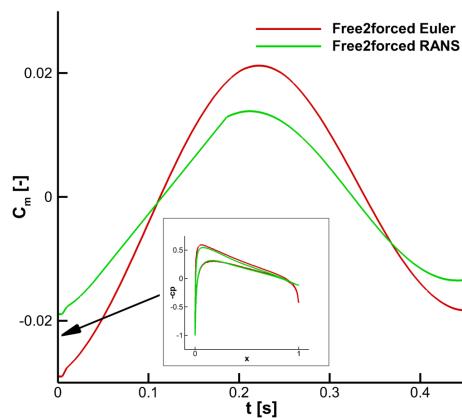
$$C_m = C_{L\alpha} \cdot \alpha \cdot \frac{x_h - x_{a.c.}}{c} \quad (5.11)$$

The Theodorsen model for a pitching aerofoil presented in Eq. (5.4) is then used. For both Theodorsen and the linear aerodynamic models, the lift coefficient slope was extracted via linear interpolation of the available data ( $C_{L\alpha} = 5.8759$ ) instead of using  $2\pi$  according to the ideal aerodynamics of the thin aerofoil theory.

Two unsteady CFD analysis with forced motion were then considered, with the non-viscous equations modelling the flow and the Navier–Stokes Reynolds–averaged. The results indicate a very good approximation of the linear model compared to the Euler forced motion, indicating the absence of nonlinearities in the flow time history.

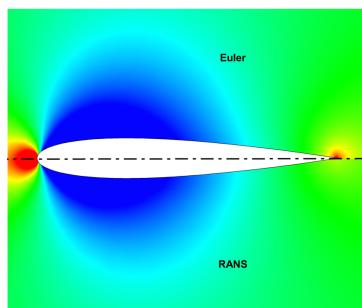
Both of them are very similar to the baseline as predictable, since they all have the same flow model and the same motion. The Theodorsen model is quite close to the other solutions, indicating the good quality of the analytical model. Finally the RANS solution shows a gap of pitching moment magnitude compared to the free to pitch Euler solution, but that may be attributed to the viscous term in the equation, changing the pressure distribution over the aerofoil surface and so the resulting pitching moment.

Figure 5.18 highlights the resulting pitching moment coefficient from both Euler and RANS CFD simulations. The pressure coefficient distributions over the aerofoil surface are presented at the initial time step ( $\alpha = 1$  deg). Furthermore in Fig. 5.19 the



**Figure 5.18:** Euler free motion applied to forced motion Euler and RANS CFD computations, highlighting the pressure coefficient difference at the initial state

upper pressure field (positive y axes) is compared for the not viscous (upper) and viscous (lower) cases. The pitching moment gap is due to different boundary conditions applied



**Figure 5.19:** Pressure field comparison between Euler and RANS steady simulations at  $\alpha = 1$  deg; only the upper part of the flow field is shown with upper part presenting not viscous solution and lower viscous

for the two CFD computations. The non viscous solution is generated considering the Kutta condition: stagnation point at the trailing edge. This boundary condition is due to the impossibility for a non viscous flow to be not continuous. For a viscous flow the

trailing edge the viscosity damps pressure gap. Euler equations are a mathematical simplification and so the Kutta condition is not real but aims to simulate the viscous effect on the aerofoil flow field.

### 5.3.3 Conclusions

FlexFlight is a very versatile and efficient interface to compute time accurate multi-disciplinary solutions. The block based architecture allows the user to freely change the used modules and insert new ones. This tool can be used for very miscellany analyses, skipping the approximations applied for the generation of aerodynamic or structural models given as input to a flight simulator.

In this Section a coupled analyses of aerodynamic and flight dynamic is presented. The traditional procedure is to first generate an aerodynamic model, based on aerodynamic derivatives or tabular results, and then start a flight simulator. This process is expensive and affected to errors during the generation of the aerodynamic model. For the FlexFlight simulation the aerodynamic model generation is not needed and the time accurate coupling between aerodynamic and flight dynamic, is not influenced by any approximation. A direct dependence exists between the modules models adopted and the results.

The aerodynamic module influence over a free to pitch two dimensional aerofoil motion is computed. The result shows the motions difference generated with different flow models. The time integration causes that every small moment difference influences all the following results, so that the total final motion is really different. The flight simulations are however generated with no approximation between flow field and motion equation integration. Full order aeroelastic models can then be created: this may be used to study the influence of a flexible body over the rigid body flight dynamic modes or aeroelastic phenomena (e.g. static divergence, flutter, or buffeting).



## Chapter 6

# Design optimization with CFD

The studies until here presented, deal with the analyses of the aircraft flight dynamics. In Chapter 3 an efficient tool to generate the aerodynamic model with CFD is first presented. The process to find the stability and handling characteristics of a test case aircraft is shown in Chapter 4. Chapter 5 presents then an alternative formulation for the flight simulation, that is more expensive to execute but does not require any aerodynamic table generation. All these analyses aim to obtain higher fidelity results about the aircraft stability and handling characteristics (e.g. manoeuvrability, controllability, static and dynamic stability, etc.) earlier in the aircraft design process. These would be precious information for the design engineers, which may reduce the development cost avoiding costly retrofitting. The aircraft design process is a typical multi-disciplinary optimization (MDO), whose domain is defined by the aircraft design variables and the minimization objects might be the production cost, the fuel consumption, or any other parameter of interest. The parameters are subject to some linear or non linear constraints (e.g. feasibility, certification restrictions, design requirements, etc.). The final target of the studies conducted in the previous Chapters, would be to introduce the aircraft stability and handling characteristics analyses inside a optimization loop. This process should be autonomous and efficient, in order to use a optimizator to find a optimum point. In the whole design process, the stability and handling characteristics can be considered as constraints, for the certification requirements, or as objective functions to optimize the flying aircraft qualities. In this Chapter the CFD will be introduced in a optimization loop. A single input, single output problem is first used to show the optimizator capabilities. A multi input, multi output problem is then presented to analyse the best aircraft trim conditions. Finally the optimization loop to find the optimum geometry to optimize the flying aircraft stability and handling characteristics is formulated. The test case analyses presented in Chapter 4 can be considered as one step of a optimization loop, so, making the process autonomous, the optimum geometry parameters combination may be found.

## 6.1 Optimizer formulation

In the presented study three optimizer tools are used. The Matlab<sup>©</sup> integrated function *fmincon* is used with the active set and interior point algorithms. A response surface reconstruction technique is then used to evaluate the optimum point.

### 6.1.1 Active set algorithm

In mathematical optimization, a problem is defined using an objective function to minimize  $f(x)$ , and a set of constraints  $c_i(x)$  as expressed in Eq. (6.1).

$$\begin{aligned} & \underset{x}{\text{minimize}} && f(x) \\ & \text{subject to} && c_1(x) \geq 0, \dots, c_m(x) \geq 0 \\ & && e_1(x) = 0, \dots, e_n(x) = 0 \end{aligned} \tag{6.1}$$

The feasible region is the set of all  $x$  to search for the optimal solution that satisfy the constraints. Given a point  $x$  in the feasible region, a constraint  $c_i(x) \geq 0$  is called active at  $x$  if  $c_i(x) = 0$  and inactive at  $x$  if  $c_i(x) > 0$ . Equality constraints  $e_i(x)$  are always active. The active set at  $x$  is made up of those constraints  $c_i(x)$  that are active at the current point. The active set is particularly important in optimization theory as it determines which constraints will influence the final result of optimization. In quadratic programming, as the solution is not necessarily on one of the edges of the bounding polygon, an estimation of the active set gives us a subset of inequalities to watch while searching the solution, which reduces the complexity of the search. In general the active set algorithm presents the following structure <sup>1</sup>:

- Find a feasible starting point
- repeat until "optimal enough":
  - solve the equality problem defined by the active set (approximately)
  - compute the Lagrange multipliers of the active set
  - remove a subset of the constraints with negative Lagrange multipliers
  - search for infeasible constraints
- end repeat

### 6.1.2 Interior point algorithm

Interior point methods (also referred to as barrier methods) are a certain class of algorithms to solve linear and nonlinear convex optimization problems. In contrast to

---

<sup>1</sup>For further information about the algorithm implemented in Matlab<sup>©</sup> please visit <http://www.mathworks.com/help/optim/ug/constrained-nonlinear-optimization-algorithms.html> [retrieved October 10, 2014]

the simplex algorithm, which finds an optimal solution by traversing the edges between vertices on a polyhedral set, interior-point methods move through the interior of the feasible region. Many interior-point methods have been proposed and analysed. Early successful implementations were based on affine scaling variants of the method. For both theoretical and practical purposes, barrier function or path-following methods have been the most popular since the 1990s. The class of primal-dual path-following interior point methods is considered the most successful <sup>2</sup>.

### 6.1.3 Response surface reconstruction technique

The response surface reconstruction technique is based on the evaluation of system output over the entire optimization variables domain. The function is computed in some sample points, which are used to generate a Kriging interpolation model. This model is then used to evaluate the function response over the entire domain. The point where the reconstructed response is minimum, is the minimization output. In order to decide which sample points had to be computed, the cognitive sampling function developed in Chapter 3 is used. The *minMAX* option makes the script to first find the local minima and maxima positions, and then advise to sample close to them. In this way more information are obtained close to local maxima and minima, generating a well representative response surface reconstruction close to the output minimum point.

## 6.2 CFD integration in the optimization loop

In the following Sections the optimization problem for flying rigid bodies with CFD is performed. The target of this study is to find the state conditions such that the drag is minimized. Two cases are considered. First a two dimensional Rae2822 aerofoil is taken into consideration. The problem is single input single output system, the input angle of attack is searched so that the output drag is minimized. Second the Piaggio Avanti business jet is analysed. The optimization problem now is still minimizing the drag coefficient, but the constraints are defined as the trim conditions. Such constraints are non linear, and so the complexity of the problem increases. The new considered design variables are the speed, the angle of attack and the elevator deflection. In order to solve these optimization problems various tools were used, and the respective results are compared. The CFD solver that is used in this section is Edge, see Chapter 2. The presented study can be considered as inverse engineering. In fact the direct way of general acting in engineering is to fixed certain design parameters, as angle of attack or cruise speed, and then study the best geometry or configuration for those given values. In the following section the geometry is considered fixed, and the best design

---

<sup>2</sup>For further information about the algorithm implemented in Matlab<sup>©</sup> please visit <http://www.mathworks.com/help/optim/ug/constrained-nonlinear-optimization-algorithms.html> [retrieved October 10, 2014]

parameters are searched in order to obtain the smallest drag value. The project was created in the first place in order to obtain the best geometry for the starting design parameters, but it is not always true that those parameters suits the best for the developed geometry. Aim of this study is to find such values, and see if a different flight condition might lead to smaller drag values and a more efficient flight.

### 6.3 Rae2822 aerofoil test case

In the following section a two dimensional RAE2822 aerofoil is used as test for drag optimization problem with Reynolds averaged Navier–Stokes equations (RANS). The optimization problem has only one design variable that is the angle of attack and only one function minimization object that is the drag coefficient. The angle of attack values constraints were imposed between -5 and 5 deg, and the first guess was considered at 0 deg. The flow is modelled using the viscous fluid dynamic equations on a structured triangular grid of 22,088 points. The considered value of Reynolds number is  $1.7 \cdot 10^7$  and Mach number 0.75 (equivalent to 225 m/s at sea level in standard atmospheric condition). The maximum number of iterations for every CFD computation was selected to 10,000 to get a root mean squared residual between consecutive iterations equal to machine accuracy ( $10^{-14}$ ). In order to have faster CFD computations 4 processors were used.

#### 6.3.1 Formulation

The studied minimization problem for the 2 dimensional aerofoil can be easily expressed in the form presented in Eq. (6.2).

$$\begin{aligned} & \underset{\alpha}{\text{minimize}} && C_D(\alpha) \\ & \text{subject to} && \alpha_{min} \leq \alpha \leq \alpha_{MAX} \end{aligned} \tag{6.2}$$

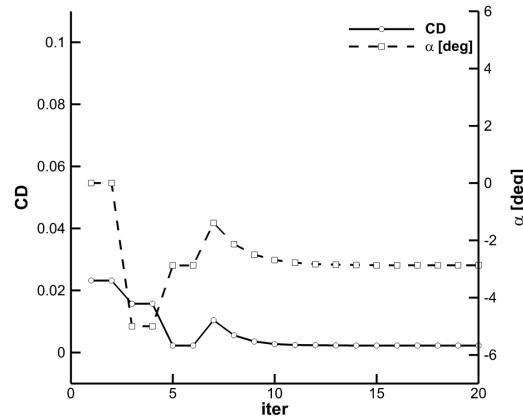
The subject function can be defined as a single input and single output (SISO) system.

#### 6.3.2 Results

##### 6.3.2.1 Active set

First of all the *fmincon* Matlab<sup>©</sup> function was used with the active set algorithm. This algorithm can take large steps, which adds speed to the overall optimization. The algorithm is effective on some problems with non-smooth constraints and it is not a large-scale algorithm. Medium-scale methods internally create full matrices and use dense linear algebra. If a problem is sufficiently large, full matrices take up a significant amount of memory, and the dense linear algebra may require a long time to execute. In this case, however, the computational time of the optimization is irrelevant compared

to the required time to compute every sample with a full CFD computation, and so the optimization cost is negligible. Figure 6.1 shows the behaviour of the optimization

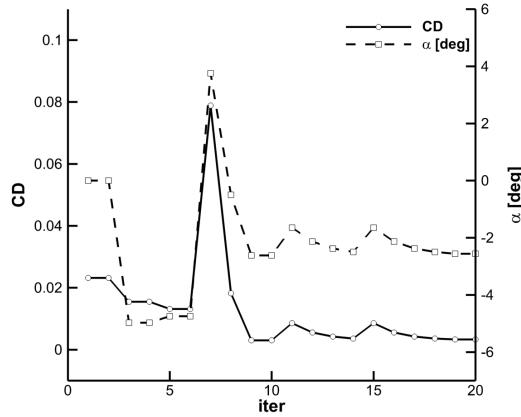


**Figure 6.1:** Active set algorithm iteration history

during the performed steps. Both input (angle of attack) and output (drag coefficient) are shown in the same graph so that the optimization time history can be evaluated. It can be easily seen how the algorithm computes a local derivative on every chosen point to choose the moving direction, by computing two samples very close to each other. This permits the algorithm to individuate a local minimum of the subject function in a very few computations (around 12 optimization iterations), that satisfy the minimization problem. Unfortunately Table 6.1 shows that the found minimum is local and not global, and so the algorithm did not look for other points, but considered itself successful at the first local minimum.

### 6.3.2.2 Interior point

The interior point algorithm is now chosen as default in the new Matlab<sup>©</sup> versions (previously it was the active set algorithm) because of the versatility. It can handle large, sparse problems, as well as small dense problems. The algorithm satisfies bounds at all iterations, and can recover from *NaN* or *Inf* results. It is a large-scale algorithm, which means that it uses linear algebra and it does not need to store, nor operate on, full matrices. This may be done internally by storing sparse matrices, and by using sparse linear algebra for computations whenever possible. Furthermore, the internal algorithms either preserve sparsity, such as a sparse Cholesky decomposition, or do not generate matrices, such as a conjugate gradient method. Similarly to the previously described active set algorithm results, the active set optimization steps are presented in Fig. 6.2. About this algorithm it can be seen how it also evaluate locally the first derivative by computing close points, but, differently the active set, it sample a larger part of the domain, so that the possibility of a global minimum elsewhere decreases.



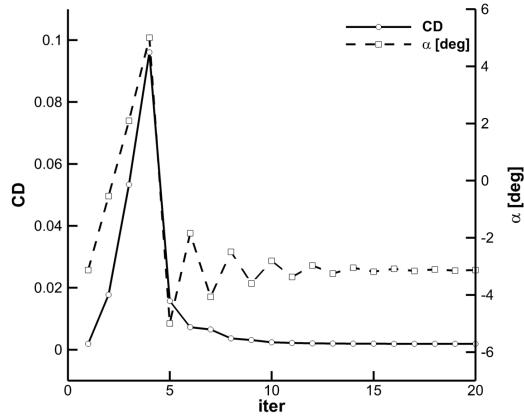
**Figure 6.2:** Interior point algorithm iteration history

However the minimum seems to be found after 18 iterations, but we do not know if such a value would be confirmed if the minimization had not been stopped. The resulting value in fact is higher than the one found with active set algorithm, but the decided computational budget was 20 CFD computations (since each of them is very expensive), and so it can be considerate that the interior point algorithm was stopped to be not efficient enough for the requirements.

### 6.3.2.3 Response surface reconstruction

The last adopted method is response surface reconstruction with a Kriging interpolation model. The sampling choice is empowered to the Matlab<sup>©</sup> function created for the CEASIOM project. In this section the Latin hypercube section was first used for the first samples, and the *MAXmin* method for the following ones. The function is based on the full reconstruction of the function via a Kriging interpolation model and then decides where to sample by comparing the predicted interpolation error and considering the presence of local minima or maxima. For this computation the *error tol* input value was set to 1000 (that is a parameter by which the error close to local minima or maxima is augmented for the comparison with the rest of the domain). The target of such a function is the ability to reconstruct a non linear function with the fewest number of samples, giving particular attention where the non linearity appears.

Figure 6.3 shows the behaviour of the optimization during the steps. Almost the whole domain is investigated, but the advised first samples on the boundary were not computed since the interest was known being not on the border. The drag function is almost a second order polynomial and so the interpolation is accurate with the Kriging model. This permits to the algorithm to individuate a local minimum after the first 3 evaluations of the subject function and then concentrate the attention around it until the requirements number of CFD computations is reached. The script was written so



**Figure 6.3:** Sampling algorithm iteration history

that the predicted minimum value is computed only in the last available computation. We would expect that if the maximum number of computations was set at around 10, it would have still found a very good approximation of the minimum. Table 6.1 shows that this is the best between the three methods, in fact the computation does not rely on local function evaluations but on the whole domain function reconstruction.

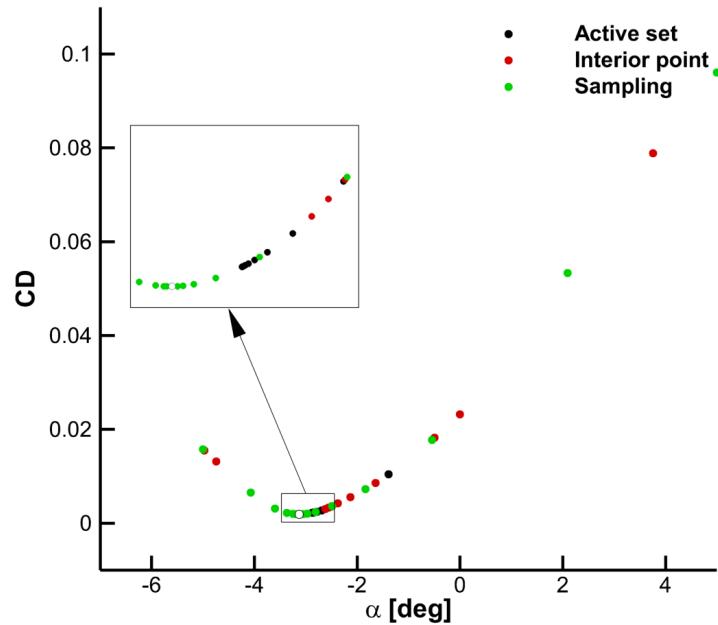
#### 6.3.2.4 Comparison

Figure 6.4 shows the drag compared the angle of attack function with the data coming from the three methods. The part of the domain close to the global minimum is zoomed to give emphasis to the samples that are computed for each method. The surface reconstruction method appears evidently as the most precise in finding the global minimum as soon as possible and then concentrates the attention around it in order to have a more accurate value. The other two methods seem to stop at some degree higher, probably satisfied by the fact that a local minimum is found. This permits to conclude that such methods implemented in *fmincon* are not suitable for a highly non linear problem such as the aerodynamic viscous equations. Probably with more function evaluations they would be able to get similar results, but in this case the cost requirements are very important. Furthermore the results are summarized in Table 6.1. Figure 6.5 presents the flow field around the two dimensional aerofoil at the

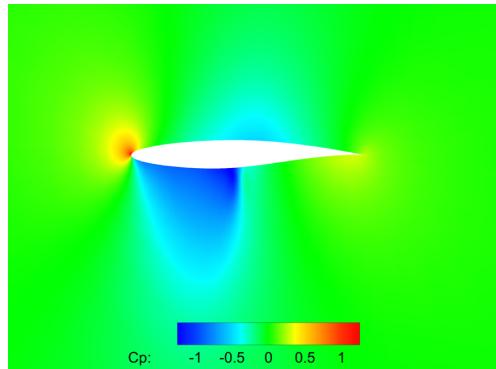
Algorithm	$C_D$	$\alpha$ [deg]
Active set	0.002215061	-2.873574
Interior point	0.003290489	-2.558745
Sampling	0.001904354	-3.130000

**Table 6.1:** RAE2822 drag optimization algorithm results comparison after 20 iterations

minimized drag with the active set algorithm, by showing the pressure coefficient value in the domain around the aerofoil.



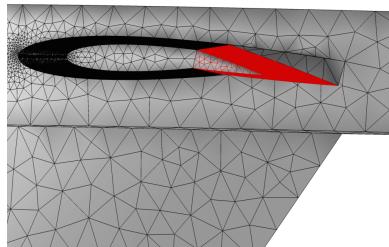
**Figure 6.4:** Drag coefficient samples with the different algorithms



**Figure 6.5:** Pressure coefficient at the active set minimum computed drag ( $C_D = 0.002215$  at  $\alpha = -2.87$  deg)

## 6.4 Piaggio Avanti test case

In the following section a three dimensional model of the Piaggio Avanti P180 II [8]<sup>3</sup> is used for drag optimization problem with Reynolds averaged Navier–Stokes equations (RANS). The optimization problem has three design variables that are the angle of attack, the elevator deflection and the speed. These values are both constrained between upper and lower boundaries and they are expected to satisfy the non linear system identifying the longitudinal vertical and rotational trimming conditions. The function minimization object is the drag coefficient. The flow is modelled using the viscous fluid dynamic equations on a grid of 15,852 points. The used mesh is a unstructured tetrahedron mesh of half aircraft model. The used boundary conditions were of not slippery wall for the aircraft surfaces, symmetry for the plane of symmetry and farfield. The yaw and roll angles are zero, and so no flux is expected through the symmetry plane. For this reason half model grid is more efficient in terms of results generation about the only longitudinal plane. Furthermore the peculiar presence of the canard in the Piaggio Avanti was not considered as a variable into the optimization problem since it has only a flap that is not expected to be used in cruise conditions for trimming, while on the fixed horizontal T-tail stabilizer there are left and right elevators. The maximum number of iterations for every computation was selected to 10,000 to get a root mean squared residual between consecutive iterations of  $10^{-14}$ . In order to have faster CFD computations 16 processors were used on the University of Southampton cluster Lyceum. In order to be able to autonomously act on the elevator deflection, a mesh deformation technique is used at every step of the optimization. This permit to not create a new mesh for any new geometry, but deform the starting grid so that the elevator deflection is generated. This causes a modification of the cells grid around the moving surface, and so a limit of 12 deg was found being the biggest acceptable deflection. An example of mesh deformation result is showed in Fig. 6.6. The mesh



**Figure 6.6:** Mesh deformation detail for a deflection of -12 deg (down)

deformation is more realistic compared to the alternative transpiration boundary conditions method. This can be used for non viscous flow simulations, and it does not physically deform the mesh but imposes the flow velocity vector to be tangent to the

---

<sup>3</sup><http://www.piaggioaero.com/#/en/products/p180-avanti-ii/overview>

deformed geometry. It is faster but the results are not as accurate as a proper mesh deformation and it cannot be applied to RANS simulations.

#### 6.4.1 Formulation

The constraints of the Piaggio test case are given as upper and lower acceptable variables values and a non linear representing the trimming conditions expressed, in wind axis coefficients terms, in Eq. (6.3).

$$\begin{cases} C_L - \frac{W}{1/2\rho V_\infty^2 S} = 0 \\ C_D - \frac{T}{1/2\rho V_\infty^2 S} = 0 \\ C_m = 0 \end{cases} \quad (6.3)$$

$C_L$  represents the lift coefficient,  $W$  the aircraft weight,  $T$  the thrust and  $C_m$  the pitching moment coefficient. The second equation is supposed always satisfied, since the pilot can freely act of the throttle with no other relevant consequences on the other two equations. Both the lift and pitching moment coefficients are functions of the aircraft state and control vectors, that in the studied case is reduced to the longitudinal plane as  $\mathbf{x} = (V_\infty, \alpha, \delta_{ele})$  with  $q = 0$  (pitching rotational speed). Such nature of the equations makes the constraint non linear, since the values of the  $C_L$  and  $C_m$  is known only after that the CFD computation is finished. This fact does not permit the optimization to compute only  $\mathbf{x}$  values such that the constraint are satisfied, but it is possible to know if the chosen design variables were right to satisfy the constraints only after that the computation is concluded. The minimization problem can be then presented in the form presented in Eq. (6.4).

$$\begin{aligned} & \underset{\mathbf{x}}{\text{minimize}} && f(\mathbf{x}) \\ & \text{subject to} && \mathbf{u}\mathbf{b} \leq \mathbf{x} \leq \mathbf{l}\mathbf{b} \\ & && \mathbf{C}_{nl}(\mathbf{x}) = \mathbf{0} \end{aligned} \quad (6.4)$$

In the presented study:

$$\begin{cases} \mathbf{x} = (V_\infty, \alpha, \delta_{ele}) \\ f(\mathbf{x}) = C_D(\mathbf{x}) \\ \mathbf{l}\mathbf{b} = (v_{\infty min}, \alpha_{min}, \delta_{ele min}) \\ \mathbf{u}\mathbf{b} = (v_{\infty MAX}, \alpha_{MAX}, \delta_{ele MAX}) \end{cases}$$

In this case the subject function can be defined as a multi-input and single-output (MISO) system. The non linear nature of the problem is contained in the  $C_{nl}(x)$  that is given by the first and third equations of 6.3. Bryan's method would permit to express

analytically the non linear constraint, making it linear as presented in Eq. (6.5).

$$\mathbf{C}_{nl}(\mathbf{x}) = \mathbf{0} \rightarrow \begin{bmatrix} C_{L_0} \\ C_{m_0} \end{bmatrix} + \begin{bmatrix} C_{L_{V_\infty}} & C_{L_\alpha} & C_{L_{\delta_{ele}}} \\ C_{m_{V_\infty}} & C_{m_\alpha} & C_{m_{\delta_{ele}}} \end{bmatrix} \cdot \begin{bmatrix} V_\infty \\ \alpha \\ \delta_{ele} \end{bmatrix} = \begin{bmatrix} W \\ 1/2\rho V_\infty^2 S \\ 0 \end{bmatrix} \quad (6.5)$$

This approximation however can be used as starting point of optimization problem, or as mid-step between the found minimum and the exact trimming variable values. However these choices require the computation of the 6 aerodynamic derivative whose result is only approximated since they should be recomputed in the starting state and not in a general one. In order to obtain a totally autonomous and precise process it is possible to insert the non linear constraint in the objective function, so that the optimization should not consider the trimming equations as constraints but as optimization objective. The problem may be then re-elaborated as shown in Eq. (6.6).

$$\begin{aligned} & \underset{x}{\text{minimize}} \quad f_i(\mathbf{x}), \quad i = 1, \dots, n. \\ & \text{subject to} \quad \mathbf{ub} \leq \mathbf{x} \leq \mathbf{lb} \end{aligned} \quad (6.6)$$

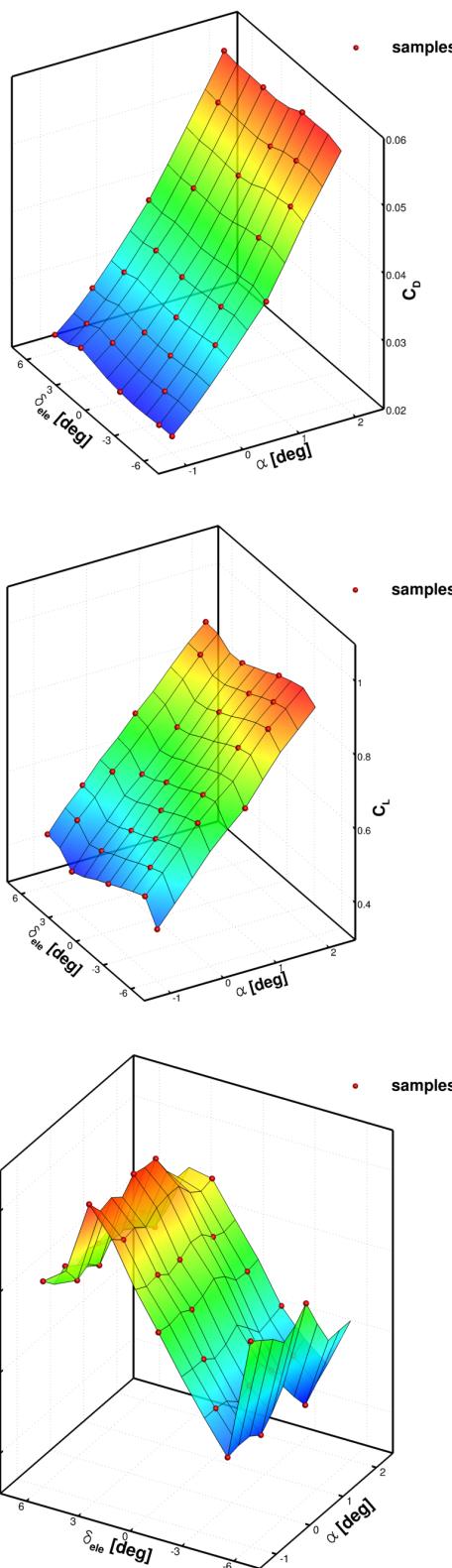
Where:

$$\begin{cases} n = 3 \\ f_1(\mathbf{x}) = C_D(\mathbf{x}) \\ f_2(\mathbf{x}) = \|C_L(\mathbf{x}) - \frac{W}{1/2\rho V_\infty^2 S}\| \\ f_3(\mathbf{x}) = \|C_m(\mathbf{x})\| \end{cases}$$

Where  $\|\cdot\|$  represents the absolute value. This new formulation lead to a increased problem complexity and the subject function can be defined as a multi input and multi output (MIMO) system.

#### 6.4.2 Results at fixed speed

In order to validate the function, the aerodynamic table is first generated at the fixed speed of 250 m/s (Mach number of 0.83 at 39,000 ft). Kriging interpolation model is generated starting from 30 CFD simulations. The resulting aerodynamic response surface about the longitudinal plane are then presented if Fig. 6.7. The drag, lift and pitching moment coefficient are plotted changing angle of attack and elevator deflection. The semi-empirical Howe's method, implemented in CEASIOM, was used to evaluate the aircraft mass and the center of gravity location, and so the pitching moment is computed with reference to the estimated CG. The pitching moment results clearly shows the non linear behaviour changing the elevator deflection beyond  $\pm 3$  deg. A linear fit around the point  $\mathbf{x} = (V_\infty, \alpha, \delta_{ele}) = (250, 0, 0)$  is then executed to estimate the aerodynamic derivatives. As previously stated, the  $C_{m_{\delta_{ele}}}$  linear approximation



**Figure 6.7:** Drag, lift and pitching moment with reference to the estimated CG coefficients; Mach 0.83 at 39000ft ( $v_\infty=250$  m/s)

validity is confined in a narrow domain part ( $\delta_{ele} \in [-3, 3]$  deg). The elevator deflection is defined positive upward. Table 6.2 shows the resulting linear approximation values. The aerodynamic derivatives with reference to the velocity are approximated to 0. The resulting derivatives allow to compute the aircraft longitudinal trimming conditions.

$C_\chi$	$C_{\chi_O} [-]$	$C_{\chi_{V_\infty}} [s/m]$	$C_{\chi_\alpha} [1/deg]$	$C_{\chi_{\delta_{ele}}} [1/deg]$
$C_L$	0.6001	$\approx 0$	0.1447	-0.0104
$C_m$	-0.2151	$\approx 0$	0.0190	0.0529
$C_D$	0.0310	$\approx 0$	0.0115	0.000065

**Table 6.2:** Piaggio aerodynamic derivatives

Equation 6.5 presented the linearized optimization constraints that represents the trimming conditions, that can be reduced to the linear system presented in Eq. (6.7) if the velocity derivatives are neglected.

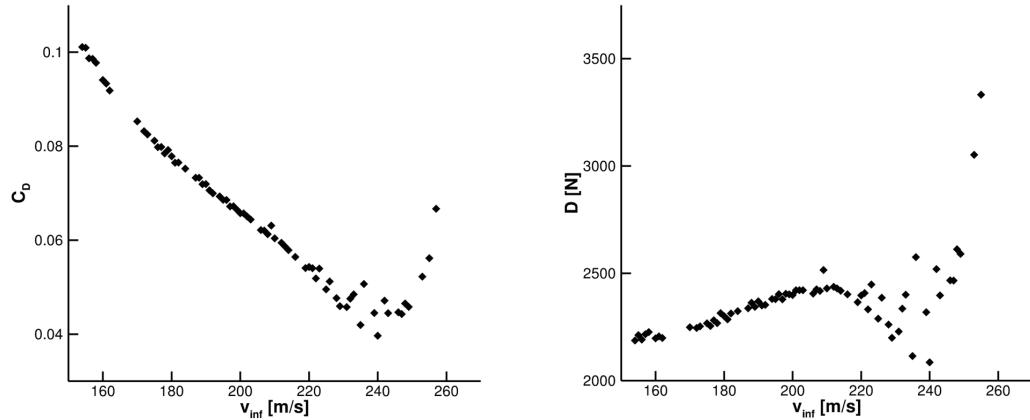
$$\begin{bmatrix} C_{L_0} \\ C_{m_0} \end{bmatrix} + \begin{bmatrix} C_{L_\alpha} & C_{L_{\delta_{ele}}} \\ C_{m_\alpha} & C_{m_{\delta_{ele}}} \end{bmatrix} \cdot \begin{bmatrix} \alpha \\ \delta_{ele} \end{bmatrix} = \begin{bmatrix} \frac{W}{1/2\rho V_\infty^2 S} \\ 0 \end{bmatrix} \quad (6.7)$$

Considering the Howe's method predicted mass of 4590 kg, the weight is  $W = M \cdot g = 45028 N$ . At the cruise altitude of 39,000 ft (11,887 m) the air density in standard conditions is  $\rho = 0.3164 \text{ kg/m}^3$ . The aircraft model wing reference surface is  $S = 5.7674 \text{ m}^2$ . The weight coefficient is then easily computed  $\mu = \frac{W}{1/2\rho V_\infty^2 S} = 0.78957$ . The location of the center of gravity is estimated at 7.31 m behind the aircraft nose. The linear system solution indicates the longitudinal trimming values  $\alpha = 1.56 \text{ deg}$  and  $\delta_{ele} = 3.49 \text{ deg}$ . The incidence is mainly due to the first equation, vertical equilibrium, on which the elevator does not influence much. This value can be then considered quite accurate. The elevator deflection value is then applied in order to obtain the rotational equilibrium. The resulting value is out the linear interval and so the result cannot be considered accurate. From the  $C_m$  plot, it can be seen how the rotational equilibrium seems impossible, since no intersection appears between the surface and the  $C_m = 0$  plane. This issue is strongly influenced by the center of gravity estimation. The pitching moment coefficient is always resulting in a pitch down effect ( $C_m \leq 0$ ) that makes the longitudinal trimming impossible. This result is caused by a too forward predicted location of the CG along the aircraft fuselage. Considering the obtained trimming values and the computed aerodynamic derivatives, the resulting drag coefficient is 0.0492.

#### 6.4.3 Results without rotational equilibrium

Since the center of gravity location is bad estimated but the drag aerodynamic derivative respect to the elevator deflection is very small, the optimization analyses considering only the vertical equilibrium (not rotational) is performed. The surface recon-

struction technique is adopted. Right Figure 6.8 presents the resulting drag coefficient changing the velocity. The angle of attack is computed in order to satisfy the vertical equilibrium (lift equal to the weight).



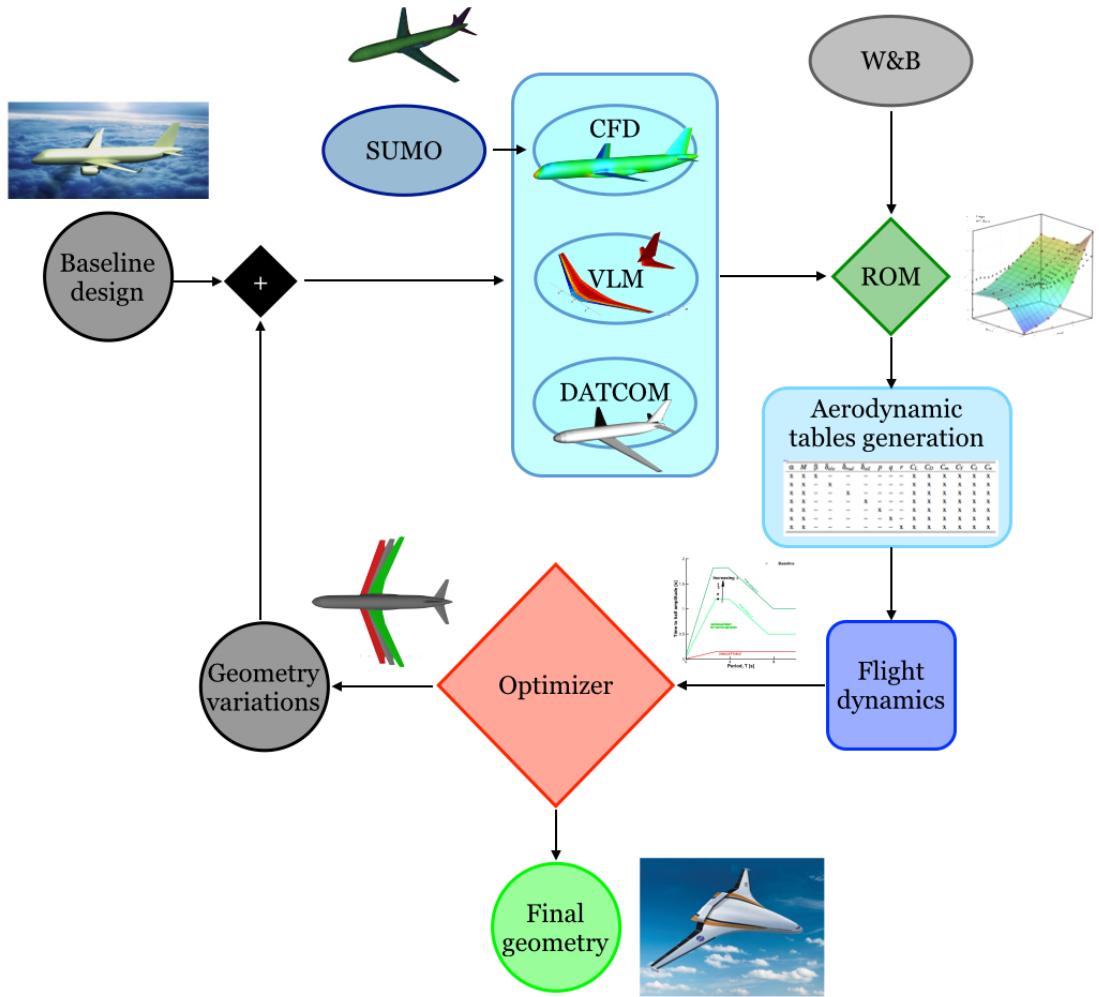
**Figure 6.8:** Drag coefficient and drag force trends changing the velocity; with vertical equilibrium imposed

The final optimization output is  $V_\infty = 237.4$  m/s (Mach number 0.8 at 39,000 ft). The vertical trim is obtained with  $C_L = 0.876$  resulting from 2.03 deg of angle of attack and the resulting minimum drag coefficient value is  $C_D = 0.039153$ . Although the drag force is directly proportional to the squared of the speed, the same output is reached if the drag is considered as optimization objective. Right figure 6.8 shows the drag force trend changing the velocity, with the lift compensating the weight. The sparse nature of some points may be caused by functions oscillations generated by the Kriging interpolation model.

## 6.5 Geometry optimization problem definition

The final aim of introducing autonomous CFD computations in an optimization loop is to emulate the aircraft design process. The design iteration would become automatic, and so a mathematical optimizer tool can be exploited to reach a better design point.

Figure 6.9 gives a graphical representation of the aircraft stability characteristics optimization changing the geometry. The loop start from the definition of the baseline geometry, the optimization objectives and some design constraints (e.g. maximum wing span). First the aerodynamic mesh can be generated with an external tool (e.g. the surface modeller SUMO). After that some results are obtained from physical-based (e.g. CFD) or statistical (e.g. DATCOM) methods, a full aerodynamic table can be computed with the surrogate models presented in Chapter 3. In the mean time, weight and balance evaluations can be obtained with semi-empirical methods (e.g. Howe's



**Figure 6.9:** Optimization loop graphical representation; geometry analysis for aircraft stability characteristics optimization

method). The statical and dynamical stability characteristics (e.g. trim and rigid aircraft flight dynamics modes) can be finally evaluated.

The optimization objectives (e.g. drag coefficient to trim the aircraft at defined flight conditions) are then given as input to the optimizer tool, which proposes some geometry variations (e.g. larger wing span). This new geometry can be then used to restart the loop until the optimizer convergence is reached.

Some iterations of the presented loop are walked through in Chapter 4, but that study is not automatic and no optimizer was involved in the process. The optimization cycle for the analyses of the external wing geometry minimizing drag and pitching moment, is presented by Liu et al. in [44].

## 6.6 Conclusions and future work

The aerofoil Rae2822 test case was a good starting point for the integration of CFD inside the optimization loop. The familiarity of the problem allowed to have a full understanding of the process and of the obtained results. The Piaggio Avanti full optimization was not run because a good quality mesh for RANS simulations was difficult to generate. Furthermore the center of gravity estimated location was demonstrated to be unlikely and so a different method should be used to generate a more accurate results. The results obtained at fixed speed, showed the limitations of considering linear aerodynamic derivatives, compared to a full aerodynamic model. Although the assessment cost for aerodynamic derivatives is lower compared to the generation of a full model with CFD, using interpolation can efficiently reduce this. The approximated models should still be able to capture flow non linearities. The studies conducted considering the only vertical equilibrium gave good results. The minimization of drag and drag coefficient gave the same optimum point. Unfortunately the accuracy of these results is strongly dependent on the quality of the RANS simulations and so of the input mesh. The aim of the optimization problem for this test case was to compute the flight condition to obtain the lowest drag coefficient, satisfying the longitudinal trim conditions. However, such problem is not an aircraft design step, but a inverse engineering problem: given the designed geometry computes the best flight conditions. The final aim of integrating the CFD in a optimization loop, is to create an autonomous cycle for the aircraft design process. The test case in Chapter 4 represents some manual steps of the geometry optimization.

## Chapter 7

# Conclusions and Outlook

The models are means for the men gleaning of the reality. Every discipline tries to figure out different aspects of it, and to make them understandable with the intellect. Human ambition is to create enough accurate models to be able to predict their characteristics. Science deals with the understanding of the cause-consequence relation. Art can be then defined as science when the beauty research is rational.

The most known physical model evolution is the step from the Galilean to the Einstein special relativity. Both the models are attempts of capturing the reality, translating them in a rational language. The validity of the models is validated with the scientific process, and they are overtaken only when more general models becomes available.

About the computational fluid dynamics different models exist. The influence of flows around flying objects can be analysed considering only the external object surfaces, or a whole part of the atmosphere around it. Today scientific progress sees the overtaking of viscous over Euler flow models for computational fluid dynamic.

About the flight dynamics simulation, the state of art is based on the exploitation of aerodynamic models with strong underlining assumptions. Tabular and linear derivatives aerodynamic models validity for flight simulation is today questioning. New models are overtaking the traditional approaches, capitalising the better computational facilities. The direct interaction between aerodynamic, structural, and etc. models with flight dynamic equations allows an efficient and exact analysis of the disciplines interaction.

Faster processors facilities allow the development, implementation and building of new flight simulators, for which time-accurate computational fluid dynamics simulations can be run in real-time. This new approach will void the exploitation of assumptions for the aerodynamic model generation, and the errors would rise only from the different adopted models approximations.

Furthermore they would authorise the exploitation of computational fluid dynamics earlier in the aircraft design process. The availability of higher fidelity models start-

ing from the conceptual phase decrease the aircraft project performances uncertainty, avoiding costly retro-fitting and reducing the necessary design iterations number. An overall better solution would be then reached by the design process convergence.

In the presented work development steps were done in these directions.

The computerized environment for aircraft synthesis and integrated optimization methods is used as platform to achieve higher fidelity aerodynamic models during the conceptual aircraft design. Flight simulation is usually obtained with aerodynamic tabular model and the application of higher fidelity methods would increase the resulting stability and handling characteristics accuracy. A more efficient method to generate reduced order models of tabular computational fluid dynamics based aerodynamic databases is developed. A real aircraft design case is then presented: the accuracy of reduced order model for aerodynamic tables is assessed and the availability of obtaining higher fidelity flight simulation results early in the design process is showed. A new programme blocks architecture is then presented. The aim is the development of an open interface, able to communicate with any software. The final object is the generation a full flying aircraft model, in which all the related disciplines communicate freely and real time between each others. The computational fluid dynamics block is validated and some aerodynamic and flight dynamics coupling results are presented for a two dimensional, one degree of freedom rigid model.

The aircraft design process is a multi-disciplinary optimization problem. Some steps toward the integration of computational fluid dynamics inside an optimization loop are developed. Some examples are presented and the feasibility of using an autonomous process is assessed.

Future work will include the development of an automatic optimization process that aims to simulate the aircraft design. The optimization loop can aim to obtain better aircraft stability characteristics by geometry modifications. A single iteration would then include mesh generation, some new computational fluid dynamics simulations, a new full aerodynamic table creation, and the flight simulation. The exploitation of reduced order models for the generation of higher fidelity aerodynamic tables will be included.

The flight simulation results obtained with the aerodynamic tables should then be placed side by side with the real-time simulation and flight dynamics coupled solution. The static stability results, as well as the flight dynamics modes characteristics, can be compared. The rigid aircraft dynamical modes can be obtained in the aerodynamic and flight dynamic coupling programme via similar procedures to the mode excitation during flight tests.

The tools used and developed in the presented thesis are freely available on request, for further information please visit the websites:

<http://www.southampton.ac.uk/engineering/about/staff/adr1d12.page> and  
<http://www.ceasiom.com>.

To briefly summarize, the main points of the work presented in this thesis are

1. exploitation of CFD for the generation of flight simulation models; physics-based simulations can now be used early in the aircraft design process; a range of applications demonstrated the potential of CFD to reduce overall design cycle cost.
2. development of higher efficiency sampling method for highly nonlinear functions; the application of reduced order model reduces the number of required CFD computation for high-fidelity aerodynamic tables generation.
3. alternative model presentation for flight simulation; the traditional used approximations for inter-disciplinary data exchanging are avoided; an open blocks architecture is defined for direct coupling of multi-disciplinary real-time computations.
4. optimization loop development including automatic CFD computations; steps towards an automatic loop for aircraft design process are accomplished.



# Bibliography

- [1] Scharl, J., Mavris, D. N., and Burdun, I. Y., "Use of Flight Simulation in Early Design: Formulation and Application of the Virtual Testing and Evaluation Methodology," *World Aviation Conference, San Diego, California, USA*, 2000.
- [2] Slotnick, J., Khodadoust, A., Alonso, J., Darmofal, D., Gropp, W., Lurie, E., and Mavriplis, D., "CFD Vision 2030 Study: A Path to Revolutionary Computational Aerosciences," NASA Langley Research Center, NASA/CR-2014-218178, 2014.
- [3] Rizzi, A., "Modeling and simulating aircraft stability and control, The SimSAC project," *Progress in Aerospace Sciences*, Vol. 47, No. 8, 2011, pp. 573–588.
- [4] Cavagna, L., Ricci, S., and Travaglini, L., "NeoCASS: An integrated tool for structural sizing, aeroelastic analysis and MDO at conceptual design level," *Progress in Aerospace Sciences*, Vol. 47, No. 8, 2011, pp. 621–635.
- [5] Williams, J. E. and Vukelich, S. R., "The USAF Stability and Control Digital DATCOM," McDonnell Douglas Astronautics Company - St Louis Division, St Louis, Missouri 63166, 1979, AFFDL-TR-79-3032.
- [6] Vallespin, D., Da Ronch, A., Badcock, K. J., and Boelens, O., "Vortical Flow Prediction Validation for an Unmanned Combat Air Vehicle Model," *Journal of Aircraft*, Vol. 48, No. 6, 2011, pp. 1948–1959.
- [7] Morari, M. and Ricker, N. L., "Model Predictive Control Toolbox," User's Guide for use with MATLAB, 1995, Version 1.
- [8] Williams, J. E. and Vukelich, S. R., "Avanti P180 II, specification and Description," Piaggio Aero Industries SpA, Revision 5.0, 2005.
- [9] Quagliotti, F., "Meccanica del volo," Lecture notes, 2013.
- [10] Milne-Thomson, L. M., *Theoretical Aerodynamics*, ISBN 0-486-61980-X, Dover Publications, 1973.
- [11] Drela, M., "XFOIL: An Analysis and Design System for Low Reynolds Number Airfoils," Tech. Rep. 54, MIT, 1989.
- [12] Tomac, M. and Eller, D., "From geometry to CFD grids—An automated approach for conceptual design," *Progress in Aerospace Sciences*, Vol. 47, No. 8, 2011, pp. 589–596.

- [13] 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, No. 8, 2011, pp. 597–620.
- [14] Ghoreyshi, M., Badcock, K. J., Da Ronch, A., Vallespin, D., and Rizzi, A., "Automated CFD analysis for the investigation of flight handling qualities," *Mathematical Modelling of Natural Phenomena*, Vol. 6, No. 3, 2011, pp. 166–188, doi: 10.1051/mmnp/20116307.
- [15] Goetzendorf-Grabowski, T., Mieszalski, T., and Marcinkiewicz, E., "Stability analysis using SDSA tool," *Progress in Aerospace Sciences*, Vol. 47, No. 8, 2011, pp. 636–646.
- [16] 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, doi: 10.2514/1.J051304.
- [17] Vallespin, D., Badcock, K. J., Da Ronch, A., White, M. D., Perfect, P., and Ghoreyshi, M., "Computational fluid dynamics framework for aerodynamic model assessment," *Progress in Aerospace Sciences*, Vol. 52, 2012, pp. 2–18, doi: 10.1016/j.paerosci.2011.12.004.
- [18] Da Ronch, A., McCracken, A. J., 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, 2013, pp. 694–707, doi: 10.2514/1.C031674.
- [19] Da Ronch, A., Badcock, K. J., McFarlane, C., Beaverstock, C., Oppelstrup, J., Zhang, M., and Rizzi, A., "Benchmarking CEASIOM Software to Predict Flight Control and Flying Qualities of the B-747," *27th Congress of the International Council of the Aeronautical Sciences, ICAS, Nice, France*, 2010.
- [20] Rizzi, A., Eliasson, P., Goetzendorf-Grabowski, T., Vos, J. B., Mengmeng, Z., and Richardson, T. S., "Design of a canard configured TransCruiser using CEASIOM," *Progress in Aerospace Sciences*, Vol. 47, No. 8, 2011, pp. 695–705.
- [21] Lophaven, S. N., Nielsen, H. B., and Søndergaard, J., "DACE, a MATLAB Kriging Toolbox," Tech. rep., Technical University of Denmark, 2002.
- [22] 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.
- [23] Santini, D., *Adaptive Fidelity Aero Data Generation*, Master's thesis, Politecnico di Milano and Royal Institute of Technology, Stockholm, 2009.
- [24] Zhang, M., Tomac, M., Wang, C., and Rizzi, A., "Variable Fidelity Methods and Surrogate Modeling of Critical Loads on X-31 Aircraft," *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition, Grapevine, Texas, USA*, Vol. 1081, 7–10 January 2013.

- [25] Ghoreyshi, M., Post, M., Cummings, R., Da Ronch, A., and K.J., B., "Transonic Aerodynamic Loads Modeling of X-31 Aircraft," *AIAA JOURNAL*, Vol. 51, No. 10, October 2013, pp. 2447–2464.
- [26] Rizzi, A., Meng, P., Nagel, B., Boehnke, D., Anemaat, W. A. J., and Carroll, J., "Collaborative Aircraft Design using AAA and CEASIOM linked by CPACS Namespace," *4th CEAS Air & Space Conference, Linkoping, Sweden*, September 2013.
- [27] Anemaat, W. A. and Kaushik, B., "Geometry Design Assistant for Airplane Preliminary Design," *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition, Orlando, Florida, USA*, Vol. 162, 2011.
- [28] Si, H., "TetGen: a quality tetrahedral mesh generator and 3D delaunay triangulator," User's Manual, WIAS technical Report No. 13, 2013.
- [29] Richardson, T. S., McFarlane, C., Isikveren, A., Badcock, K., and Da Ronch, A., "Analysis of conventional and asymmetric aircraft configurations using CEASIOM," *Progress in Aerospace Sciences*, Vol. 47, No. 8, 2011, pp. 647–659.
- [30] Jungo, A., *Development of the CEASIOM aircraft design environment for novel aircraft configurations*, Master's thesis, Ecole polytechnique federale de Lausanne, 2014.
- [31] Rizzi, A., Zhang, M., Nagel, B., Boehnke, D., and Saquet, P., "Towards a Unified Framework using CPACS for Geometry Management in Aircraft Design," *50th AIAA Aerospace Sciences Meeting, Nashville, Tennessee, USA*, January 2012.
- [32] Roberto, M. M., *Design and Analysis of the Control and Stability of a Blended Wing Body Aircraft*, Master's thesis, Royal Institute of Technology, Stockholm, 2014.
- [33] Cristofaro, M., Da Ronch, A., Vos, J., and Rizzi, A., "Recent developments in the CEASIOM framework using the common language CPACS," *4th EASN Association International Workshop on Flight Physics and Aircraft Design, Aachen, Germany*, October 2014.
- [34] "Edge User Guide," FOI, Swedish Defence Research Agency, dnr 03–2870, 5.2, 2011.
- [35] "Edge Theoretical Formulation," FOI, Swedish Defence Research Agency, dnr 03–2870, 5.2, 2011.
- [36] Da Ronch, A., "CFD Star Documentation," SimSAC Report, Deliverable Number D 3.2–6, 2009.
- [37] Cristofaro, M., Wang, Y., and Da Ronch, A., "Towards computational flight dynamics of a passenger jet aircraft," *ICAS International council of aeronautical sciences Conference, St Petersburg, Russia*, Vol. 0462, September 2014.
- [38] Pamadi, B. N., *Performance, stability, dynamics, and control of airplanes*, AIAA education series, 2004, ISBN: 1-56347-583-9.

- [39] Kennett, D. J., *On the Development of a Meshless Method to Study Multibody Systems Using Computational Fluid Dynamics*, Ph.D. thesis, University of Liverpool, 2013.
- [40] Da Ronch, A., McCracken, A. J., Tantaroudas, N. D., Badcock, K. J., Hesse, H., and Palacios, R., “Assessing the Impact of Aerodynamic Modelling on Manoeuvring Aircraft,” *52nd Aerospace Sciences Meeting, National Harbor, Maryland, USA*, January 2014.
- [41] Landon, R. H., “NACA 0012. Oscillating and Transient Pitching,” Compendium of Unsteady Aerodynamic Measurements AGARD-R-702, NATO Science and Technology Organization, 1982.
- [42] Theodorsen, T., “General theory of aerodynamic instability and the mechanism of flutter,” National Advisory Committee for Aeronautics, 1935, NACA-496.
- [43] Jones, R. T., “The Unsteady Lift of a Wing of Finite Aspect Ratio,” National Advisory Committee for Aeronautics, 1940, NACA-681.
- [44] Lyu, Z., Kenway, G. K. W., and Martins, J. R. R. A., “RANS-based Aerodynamic Shape Optimization Investigation of the Common Research Model Wing,” *52nd Aerospace Sciences Meeting, National Harbor, Maryland, USA*, Vol. 0567, January 2014, doi: 10.2514/6.2014-0567.
- [45] Watkins, C. J. C. H. and Dayan, P., “Q–Learning,” *Machine Learning*, Vol. 8, No. 3-4, May 1992, pp. 279–292.
- [46] Zhang, M., Cristofaro, M., Wang, Y., Da Ronch, A., and Rizzi, A., “Investigating the Piaggio Avanti design using CEASIOM,” *ICAS International council of aeronautical sciences Conference, St Petersburg, Russia*, Vol. 0464, September 2014.
- [47] Ciampa, P. D., Nagel, B., Meng, P., Zhang, M., and A., R., “Modeling for Physics Based Aircraft Pre–design in a Collaborative Environment,” *4th CEAS Air & Space Conference, Linkoping, Sweden*, September 2013.
- [48] Lötstedt, P., “Accuracy of a Propeller Model in Inviscid Flow,” *Journal of Aircraft*, Vol. 32, No. 6, 1995, doi: 10.2514/3.46880.

## Appendix A

# The Kriging interpolation toolbox DACE

DACE (Design and Analysis of Computer Experiments) is a Matlab<sup>©</sup> toolbox to use Kriging approximations on computer models developed by Soren N. Lophaven, Hans Bruun Nielsen and Jacob Sondergaard at the Technical University of Denmark [21]. In this Appendix some theoretical basis are given and the influence of the inputs over the generated interpolation model is investigated.

### A.1 Theoretical basis

Considering a set of data:

$$Y = \begin{bmatrix} y_{11} & y_{12} & \dots & y_{1q} \\ y_{21} & y_{22} & \dots & y_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ y_{m1} & y_{m2} & \dots & y_{mq} \end{bmatrix} = \begin{bmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_m \end{bmatrix} \quad \text{with } \mathbf{y}_i \in \mathfrak{R}^q \quad (\text{A.1})$$

in which  $q$  is the number of different variables considered and  $m$  is the number of sites where they have been taken and the set of sites:

$$S = \begin{bmatrix} s_{11} & s_{12} & \dots & s_{1n} \\ s_{21} & s_{22} & \dots & s_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ s_{m1} & s_{m2} & \dots & s_{mn} \end{bmatrix} = \begin{bmatrix} \mathbf{s}_1 \\ \vdots \\ \mathbf{s}_m \end{bmatrix} \quad \text{with } \mathbf{s}_i \in \mathfrak{R}^n \quad (\text{A.2})$$

in which  $m$  is still the number of sites analysed and  $n$  is the dimension of the vector space. The target is to find a model  $y'$  that expresses the  $q$  responses  $y(x) \in \mathfrak{R}^q$  for every site of the domain of  $n$  dimension (input  $x \in \mathfrak{R}^n$ ). In order to do so it used a

regression model which is a linear combination of  $p$  chosen functions  $f_j : \Re^n \longrightarrow \Re$  and  $R(\theta, w, x)$  that is the correlation model with parameters . For further details see [21].

## A.2 Practical application

For examples it is possible to consider the classical aeronautic problem about aero table generation. Consider e.g. that we want to calculate  $[C_L, C_{L_\alpha}, C_m, C_{m_0}]$  for every different combination of  $[\alpha, M, \beta]$  inside the flight envelope. In reference of what said in the previous section we have a problem of 4 variable ( $q = 4$ ) in a 3 dimensions vector space ( $n = 3$ ) generated by all the different combinations of the flight conditions  $[\alpha, M, \beta]$ . Kriging approximations aims to fulfil this table by calculating with CFD only some of the conditions (only  $m$  sites) and then build a model to predict all of them. In order to do so it is needed to calculate the real values of the unknowns at the border of the flight envelope (our domain) and at some points in the domain and then make Kriging approximation work. In case of multi-dimensional domain, to avoid extrapolation, it is preferred to place initial samples only at the edges of the flight envelope. In fact in order to start the Kriging computation only  $\sum_{i=1}^n i = 6$  are needed (otherwise if a *poly2* regression model is used it would find a error equal to zero anywhere at the first step) but there are  $2^n = 8$  edges. For a one or two dimensional domain, beyond the boundary samples, in order to decide where to sample initially is possible to use Latin hypercube sampling that is a strategy for generating random sample points ensuring that every portions of the domain is represented. These are the main steps:

1. Divide the interval of each dimension into  $m$  non-overlapping intervals having equal probability and same size
2. Sample randomly from a uniform distribution a point in each interval in each dimension
3. Pair randomly the point from each dimension.

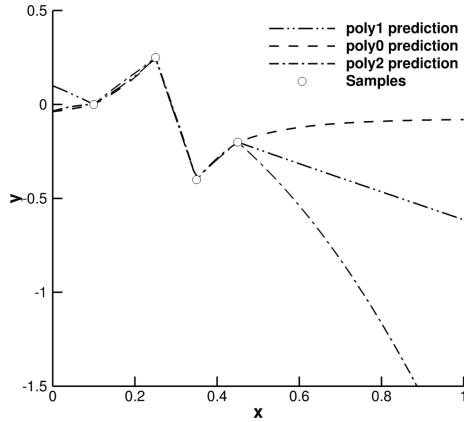
The function takes as input:  $m$  (Number of sample points to generate) and  $n$  (Number of dimensions). It gives as output:  $S$  ( $m \times n$  matrix with the generated sample points chosen from uniform distributions on  $m$  subdivisions of the interval  $]0, 1[$ ). Obviously the result cannot be considered good at the first step but Kriging approximation gives us information about the predicted error and so it is possible to sample in those points and then a new Kriging approximation. This step can be performed until a certain stop criterion is satisfied. It might be a condition on the maximum or averaged predicted error, on the number of iterations or on the computational time. At the end of this process we obtain a data table in 3 dimensions  $[\alpha, M, \beta]$ , in which every position contain

the value of every unknown  $[C_L, C_{L_\alpha}, C_m, C_{m_0}]$ . The values are exact in the position of where the sampling had been done and approximated in all the others.

### A.3 Numerical examples over the influence of the input parameters

In this section the different parameters required as input for the Kriging approximation will be analysed trying to understand their actual influence on the behaviour. The following graphs contain the Kriging approximation, the samples positions and the estimated error (Mean Squared Error). In order to have an easy visualization a one-dimension ( $n=1$ ) and single unknown ( $q=1$ ) computation has been computed considering to have only 4 sampling ( $m=4$ ). First of all the results using different regression models are compared. In DACE toolbox are provided 3 of them:

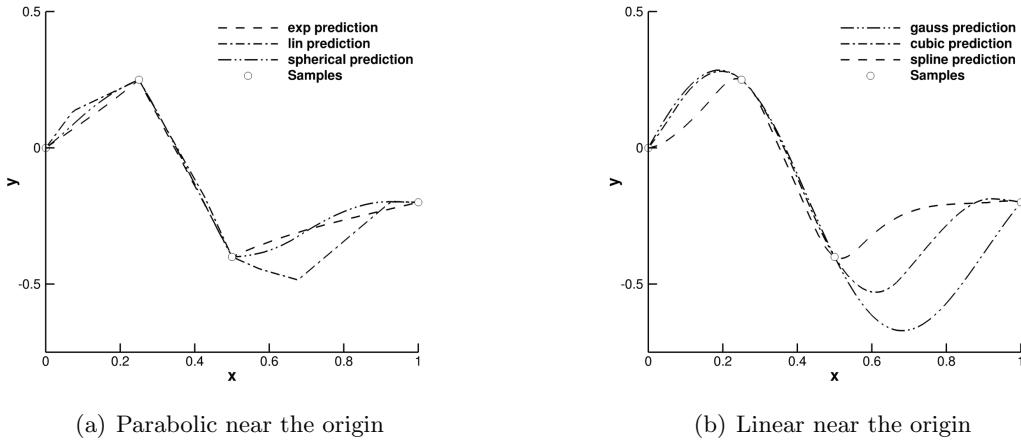
1. Constant (**poly0**)
2. Linear (**poly1**)
3. Quadratic (**poly2**)



**Figure A.1:** Comparing the three regression models in DACE

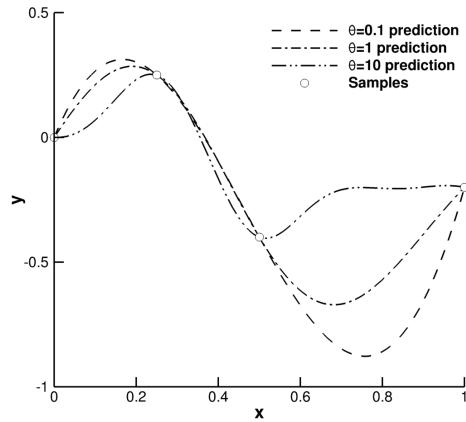
The importance of the choice of the regression model over the Kriging approximation can be seen from Fig. A.1. It can be easily seen that, far from the samples, the trend follows the least squares fit approximation of the only samples. So for the constant model that is the average of the values, for the linear one, the straight line closer to samples and for the quadratic the parabola. This is true for a one-dimension domain but we aspect the same behaviour for more dimensions as well because for multi-dimensional domain polynomial functions maintain the same behaviour (e.g. for a 2-dimensional domain the polynomial of order 0, 1 and 2 become a horizontal plane, an sloping plane and a paraboloid respectively). The choice of the regression model is very important in the

case of sampling points in the inner domain, if the borders are sampled the importance of the regression model decrease. In DACE toolbox are available various correlation models: exponential (**exp**), generalized exponential (**expg**), Gaussian (**gauss**), linear (**lin**), spherical (**spherical**), cubic (**cubic**) and spline (**spline**).



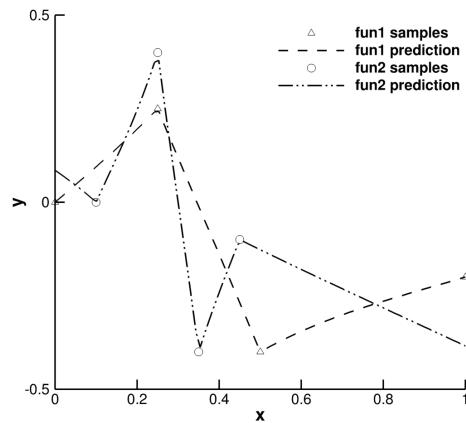
**Figure A.2:** Comparing the correlation models in DACE

The correlation functions can be separated into two groups, one containing functions that have a parabolic behaviour near the origin (**gauss**, **cubic** and **spline**), and the other containing functions with a linear behaviour near the origin (**exponential**, **linear** and **spherical**). The choice of correlation function should be motivated by the underlying phenomenon, e.g. a function we want to optimize or a physical process we want to model. If the underlying phenomenon is continuously differentiable, the correlation function will likely show a parabolic behaviour near the origin (as for **gauss**, **cubic** and **spline**), but conversely, physical phenomena usually show a linear behaviour near the origin and **exp**, **expg**, **lin** or **spherical** would usually perform better. Furthermore for large distances the correlation is 0 according to the linear, cubic, spherical and spline functions, while it is asymptotically 0 when applying the other functions. [21] The correlation models are always dependent on a parameter,  $\theta$ , that changes the correlation shape. In Fig. A.3 some results of Gaussian correlation using different  $\theta$  are presented.



**Figure A.3:**  $\theta$  parameter effect on Gaussian correlation models

The correlation modifies the least squares fit interpolation approximation so that it would pass through the samples. A larger value of  $\theta$  leads to a faster approach to the least squares fit and so the domain of influence of the samples over the approximation is more narrow. The result for a 1d dimension ( $n=1$ ) and double unknown ( $q=2$ ) with 3 sampling both ( $m=3$ ) is showed in Fig. A.4. It can be easily seen that the two resulting approximation functions are completely independents, as if was made a separate computation for both of them.



**Figure A.4:** Problem with two unknowns



## Appendix B

# Controller design

### B.1 Model predictive control

The MPC control law can be most easily derived by referring to Fig. B.1. For any assumed set of present and future control moves  $\Delta u(k), \Delta u(k+1), \dots, \Delta u(k+m-1)$  the future behaviour of the process outputs  $y(k+1|k), y(k+2|k), \dots, y(k+p|k)$  can be predicted over a horizon  $p$ .

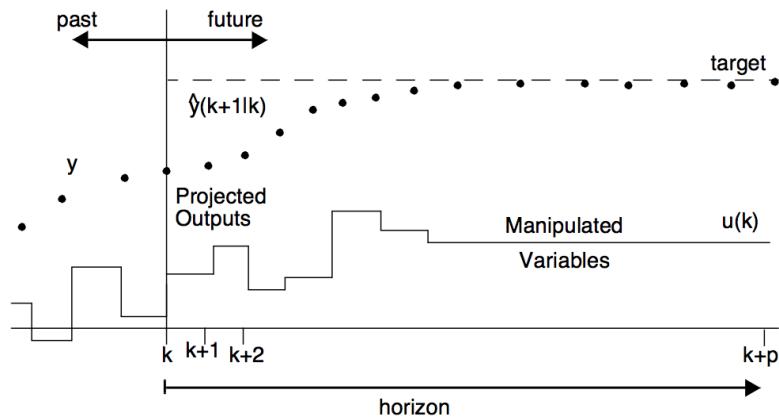


Figure B.1: MPC working scheme [7]

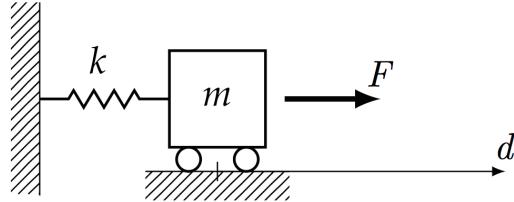
The  $m$  present and future control moves ( $m \leq p$ ) are computed to minimize a quadratic objective. Though  $m$  control moves are calculated, only the first one ( $\Delta u(k)$ ) is implemented. At the next sampling interval, new values of the measured output are obtained, the control horizon is shifted forward by one step, and the same computations are repeated. The resulting control law is referred to as “moving horizon” [7]. This method is usually used in industry to make a variable follow a required trend (e.g. control the temperature with the power in a dryer). The general state space problem

analysed is expressed by the system presented in Eq. (B.1).

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases} \quad (B.1)$$

In Eq. (B.1)  $u$  represents the manipulated variables that can be adjusted to obtain the required output trend for  $y$  and  $x$  is the state of the system.

### B.1.1 Application to a mass spring system



The physical mass-spring model is easily described with the force equilibrium Eq. (B.2).

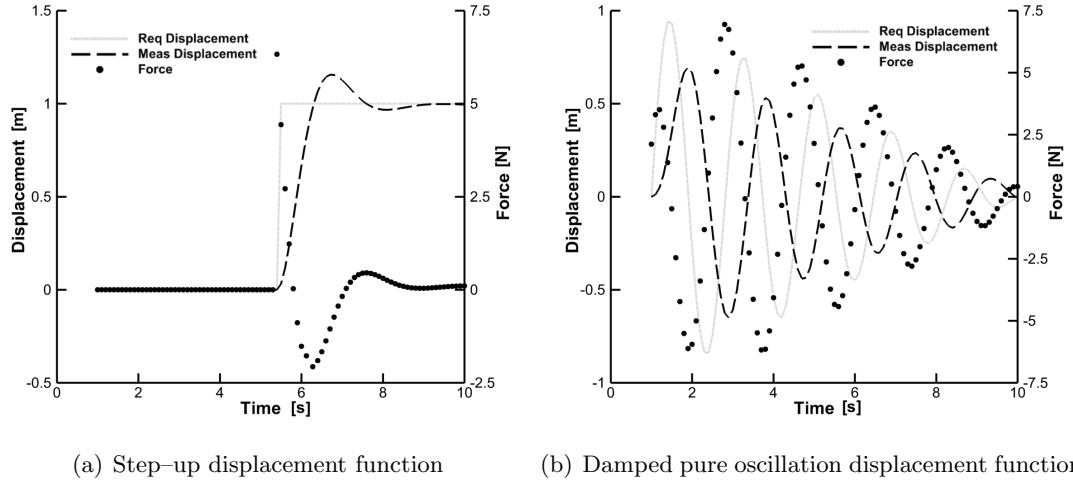
$$F = m\ddot{d} + kd \Rightarrow \begin{Bmatrix} \ddot{d} \\ \dot{d} \end{Bmatrix} = \begin{bmatrix} 0 & 1 \\ -k/m & 0 \end{bmatrix} \begin{Bmatrix} d \\ \dot{d} \end{Bmatrix} + \begin{Bmatrix} 0 \\ 1/m \end{Bmatrix} F \quad (B.2)$$

This may be easily converted in the previously described state space problem of Eq. (B.1) considering:

$$\begin{cases} \mathbf{A} = \begin{bmatrix} 0 & 1 \\ -k/m & 0 \end{bmatrix} \\ \mathbf{B} = \begin{Bmatrix} 0 \\ 1/m \end{Bmatrix} \end{cases} \quad \text{and} \quad \begin{cases} \mathbf{C} = \{1 \ 0\} \\ \mathbf{D} = 0 \end{cases} \quad (B.3)$$

so that the state vector  $x$  is composed of both displacement and speed of the body, the output  $y$  is the displacement and the manipulated variable  $u$  is the external force that acts to make the output follow a required trend. Furthermore a nonlinear system was considered. A very meaningful example is a nonlinear spring for which  $k = k_1 + k_3 x^2$  where a  $k_3 \leq 0$  means that the spring applies less force if it is stretched and vice versa. In this case the  $\mathbf{A}$  matrix has to be computed at every step and this modification may bring to some imprecision in the MPC prediction. After that the physical parameters are defined, it is sufficient to chose the required displacement trend to make the controller work.

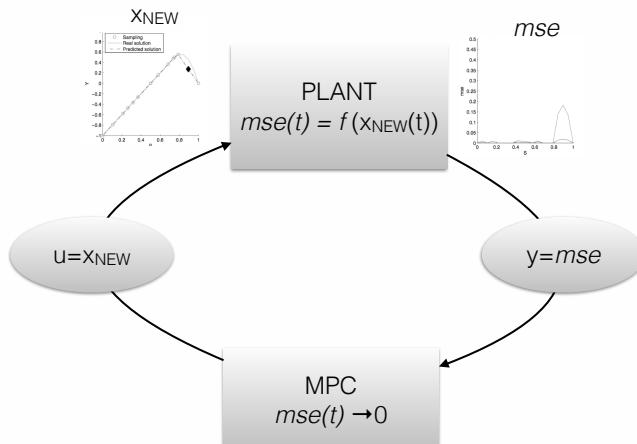
A step-up function and a damped pure oscillation are the displacement trends required to the controller in Fig. B.2. The adopted mass value is  $m = 1\text{kg}$ , while the stiffness are:  $k_1 = 0.1\text{ N/m}$  and  $k_3 = 0\text{ N/m}^3$ .



**Figure B.2:** Model predictive controller applied to a mass spring system

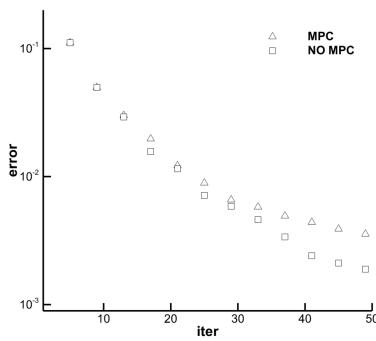
### B.1.2 Application to sampling problem

In this case an explicit law in the form of Eq. (B.1) is not available, but an approximated form may be generated after some iterations considering the produced data. In the problem analysed the variables are:  $u$  represents the coordinates of the new sample, the time confused with the iteration number  $t$ , and  $y$  the mean squared error  $mse$ . The target of the model is to minimize the error and so  $y = 0$ . Figure B.3 presents a schematic representation of the problem.



**Figure B.3:** MPC definition scheme for sampling problem

First of all a single dimension domain problem is studied, using the  $C_{L-\alpha}$  similar function developed in the Chapter 3. The plant model, which is identified from the trend of errors until that point acquired, is generally accurate. To have a better accuracy the first points were removed, because really different from the general trend. Then the MPC can be used simulating a prediction of the output trend (the error) for the required horizon and only the first of the required manipulated variables (the new sample coordinates).



**Figure B.4:** MPC test on  $C_{L-\alpha}$  function sampling problem presented in Chapter 3

The resulting samples are confined to the domain and if a point is already sampled another method is used for that iteration. However the result in Fig. B.4 is worse than the previous developed methods. The firsts 10 iterations bring to the same errors values because MPC is not able to compute a good approximation of the error trend over the iterations. Starting from iteration number 10 MPC start working but it produces a linear increasing or decreasing of the new sample coordinates, until it arrives at the borders of the domain. Then one iteration is made with the other methods and then MPC start again from that new starting point. The overall linear dependency between state, input and output causes the method to consider a decreasing sample coordinate being good to decrease the overall error. So this method does not consider the nature of the problem, and try to find a good evaluation of the next samples considering the values of the coordinates as a quantity to be modelled to obtain the target output value.

The mass-spring problem in the example, is evidently a dynamic problem. About table generation, instead, it is a "sort of static" process, since no dynamic occurs, that is not usually solved with MPC.

As final attempt new poles were inserted in the extracted transfer function. In order to have a stable model this must have a negative real part, and two conjugate imaginary parts. However the addition of a real pole or a couple of complex conjugate poles did not improve the behaviour of the model and the process did not reach better results than the sampling decisions methods developed in Chapter 3.

## B.2 Game theory: Q-learning

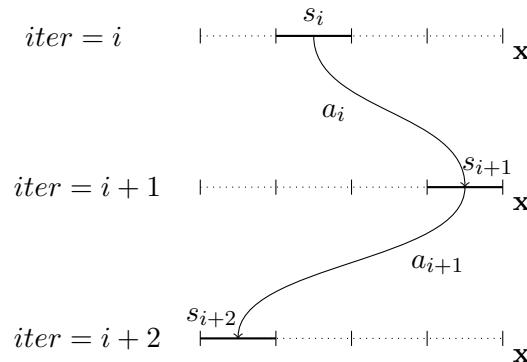
Q-learning is a model-free reinforcement learning technique. Specifically, Q-learning can be used to find an optimal action-selection policy for any given (finite) Markov decision process (MDP). It works by learning an action–value function that ultimately gives the expected utility of taking a given action in a given state and following the optimal policy thereafter. When such an action–value function is learned, the optimal policy can be constructed by simply selecting the action with the highest value in each state. One of the strengths of Q-learning is that it is able to compare the expected utility of the available actions without requiring a model of the environment [45]. The problem model consists of an agent, states  $S$  and a set of actions per state  $A$ . By performing an action  $a \in A$ , the agent can move from state to state. Executing an action in a specific state provides the agent with a reward  $R$  (a real or natural number). The goal of the agent is to maximize its total reward. It does this by learning which action is optimal for each state. Before learning has started,  $Q$  returns a fixed value, chosen by the designer. Then, each time the agent selects an action, and observes a reward and a new state that both may depend on both the previous state and the selected action. The core of the algorithm is a simple value iteration update. It assumes the old value and makes a correction based on the new information.

$$Q_{t+1}(s_t, a_t) = \underbrace{Q_t(s_t, a_t)}_{\text{old value}} + \underbrace{\alpha_t(s_t, a_t)}_{\text{learning rate}} \left[ \underbrace{R_{t+1}}_{\text{reward}} + \underbrace{\gamma}_{\text{discount}} \underbrace{\max_a Q_t(s_{t+1}, a)}_{\text{future value}} - \underbrace{Q_t(s_t, a_t)}_{\text{old value}} \right]$$

where  $R_{t+1}$  is the reward observed after performing  $a_t$  in  $s_t$  and  $\alpha_t(s, a)$  is the learning rate with  $0 < \alpha \leq 1$ . The discount factor  $\gamma$  trades off the importance of sooner versus later rewards with  $0 < \gamma \leq 1$ . The learning rate determines to what extent the newly acquired information will override the old information. A factor of 0 will make the agent not learn anything, while a factor of 1 would make the agent consider only the most recent information. In fully deterministic environments, a learning rate of  $\alpha_t(s, a) = 1$  is optimal. The discount factor determines the importance of future rewards. A factor of 0 will make the agent short-sighted by only considering current rewards, while a factor approaching 1 will make it strive for a long-term high reward. Since Q-learning is an iterative algorithm, it implicitly assumes an initial condition before the first update occurs. A high initial value can encourage exploration (update rule lower values). Furthermore the first reward  $r$  may be used to reset the initial conditions. According to this idea, the first time an action is taken the reward is used to set the value of  $Q$ .

### B.2.1 Application to sampling decision

In the case of the sampling decision, the problem is represented in Fig. B.5. The Q-learning method is adapted in order to be able to apply it. The sampling sequence can be thought as a path that the method follow in order to obtain the smallest error. In order to have a finite and not too big number of possible choices at every step the domain was divided to  $n$  parts ( $n=4$  in Fig. B.5). So at every iteration the state is the position of the last chosen sample and the  $n$  domain parts in which the next sample might be performed compose the set of available actions  $A$ . The reward  $R_{t+1}$  was defined as the difference between the function value given by the interpolation in the new sample point and the real computed value, so that a good approximation of the interpolation would lead to a low reward and vice versa. About the two parameter, they were taken as:  $\gamma = 0$ , because there is no influence of the action on the future available actions (in any part it can go wherever it wants) and  $\alpha = 0.9$  so that the learning is fast and easily forget about the past actions. At every iteration the Q-learning method decided in which part position the next sample and, in that sub-domain, the maximum of the predicted  $mse$  was searched and the newsample position was placed there. The



**Figure B.5:** Q-learning theory application applied to the sampling decision method

state does not really influence the choice because at every iteration all the actions are available, so reducing the state to one, it reduced the problem complexity obtaining better and faster results of the method. However, as for the model predictive control, the Q-learning method did not bring any improvement to the methods discussed in Chapter 3. This is imputable to the fact that the method make always a choice for which domain part analyse, that the other methods have not and so they can carry their research over the whole domain at every iteration.

## Appendix C

# Investigating the Piaggio Avanti design using CEASIOM

The paper about the Piaggio Avanti P180 II [46] is presented in this Appendix. The results were created within an international team effort, during which the author actively collaborated.

### C.1 Abstract

In early steps of aircraft design the unification of configuration definition is important to avoid user–input errors. Also coupling with each other can strengthen different tools with different specifications simultaneously provided that the geometry definition is transferred with minimum data loss. This is vitally useful especially when geometry data is transferred in order to perform high–fidelity analysis. This paper reports the analysis for the pitch control of a three–lifting–surface aircraft Piaggio Avanti using CEASIOM, a tool–chain software for aircraft preliminary design, with the baseline configuration coming from the conceptual design code AAA, linked by a common name–space CPACS for the means of data collaboration.

### C.2 Introduction

Figure C.1 spells out the details in the early steps of aircraft design for the definition of the configuration. The figure illustrates two design loops in the conceptual design phase that follow the first–guess sizing (usually done by a spread–sheet) to obtain the initial layout of the configuration. The first one, the pre–design loop, is aimed at establishing a very quick (time–scale can be from one to a few weeks) yet technically consistent sized configuration with a predicted performance. The second one, the concept–design loop, is a protracted and requires intensive effort involving more advanced first–order trade studies to produce a refinement in defining the minimum goals of a candidate

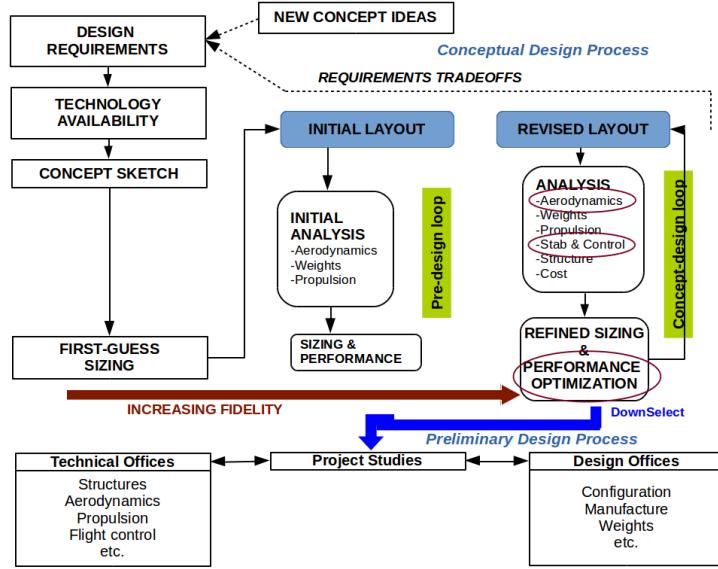
project. At the end of the conceptual design phase all the design layouts will have been analysed, and the best one, or possibly two, designs will be down-selected to the preliminary design phase. During the preliminary definition, project design is still undergoing a somewhat fluid process and indeed warrants some element of generalist-type thinking, but the minimum goals of the project have already been established during the conceptual definition phase and the aim is to meet these targets using methods with higher order than those used during the conceptual definition phase. Furthermore, the participants in this working group are mostly genuine specialists in each respective discipline. Figure C.1 indicates the way in which data, or information, is passed between specialist groups during the design process. The specialist groups must consider the level of advanced technology to be adopted together with all of the other active constraints on the design. The data flow lines indicate how the technology areas influence the aircraft configuration through its performance. The specialist departments/offices provide the input data to the project designers who then coordinate a systematic search to find the optimum configuration and settle disputes between conflicting specialist opinions. There exists today a good deal of inefficiencies in interactions between all these various groups.

This paper shows the application of the high fidelity aircraft design code CEASIOM [3], the Computerised Environment for Aircraft Synthesis and Integrated Optimization Methods for Piaggio Avanti configuration which comes from Advanced Aircraft Analysis AAA [27] by investigating its longitudinal stability and control. The goal is to model the known three-channel control surfaces and to show how the three-lifting-surface for pitch control gives lower trim drag than conventional two-lifting-surface configurations.

### C.3 Conceptual Design Tool Aaa

Advanced Aircraft Analysis (AAA) provides a powerful framework to support the iterative and non-unique process of aircraft preliminary design. The AAA program allows design engineers and preliminary design engineers to take an aircraft configuration from early weight sizing through open loop and closed loop dynamic stability and sensitivity analysis, while working within regulatory and cost constraints.

The current version of AAA is based on the methods of Airplane Design Parts I–VIII by Jan Roskam, Airplane Flight Dynamics Parts I–II by Jan Roskam, Airplane Aerodynamics and Performance by Jan Roskam and Eddie Lan and methods developed for airplane design by DARcorporation engineers. Since 1991, when DARcorporation acquired the rights for AAA and continued the development as a commercial venture, AAA has been improved and upgraded several times.



**Figure C.1:** The two design loops in the *conceptual design phase* process and the *down-select* to project study in *preliminary design*

AAA enables a fully functioning three-dimensional aircraft drafting tool Shark/AP [26]. More information about AAA geometry format and description can be found in [27] or on the website <sup>1</sup>.

## C.4 Preliminary Design Toolset Ceasiom

The Computerized Environment for Aircraft Synthesis and Integrated Optimization Methods, CEASIOM, developed within the European 6th Framework Programme Sim-SAC (Simulating Aircraft Stability And Control Characteristics for Use in Conceptual Design), is a framework tool for conceptual aircraft design that integrates discipline-specific tools like: CAD and mesh generation, computational fluid dynamics (CFD), structures, stability and control analysis, etc., all for the purpose of early preliminary design [3]. It is an *ad hoc* framework that offers possible ways to increase the concurrency and agility of the classical conceptual–preliminary process outlined in Fig. C.1. CEASIOM software has four core functions: geometry and mesh generation, CFD, aeroelastic analysis, and stability and control (flight dynamics). Significant features developed and integrated in CEASIOM as modules are:

- Geometry module CPACScreator-sumo [12,47]. A customized geometry construction system coupled to *automated* surface and volume grid generators, resulting model exported to Computer Aided Design (CAD) via Initial Graphics Exchange Specification (IGES) standard.

<sup>1</sup><http://www.darcorp.com/Software/AAA/> [retrieved October 10, 2014]

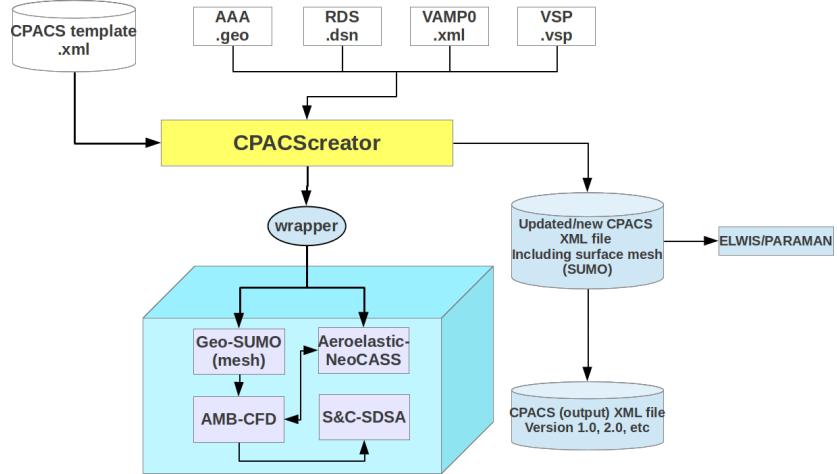
- Aerodynamic Model Builder AMB–CFD [13]. A complete toolbox of aerodynamic analysis methods ranging from the empirically based DATCOM to physics-based linear and non–linear CFD (Euler & RANS) offering broad choice in fidelity:
  - Digital DATCOM.
  - Steady/unsteady vortex–lattice code (VLM) TORNADO for low–speed (linear) aerodynamics and aeroelasticity.
  - CFD solvers in EDGE code. Euler solver (EDGE code in Euler mode) for inviscid flow cases where total pressure and vorticity fields are too complex to model with isentropic equations e.g. at high speed or swirling flow. Examples of these are shock waves and propeller slipstreams. RANS (Reynolds–Averaged Navier–Stokes) flow simulator (e.g. EDGE CFD code) for high fidelity viscous flow analysis at extreme flight conditions.
- Stability and Control module S & C (e.g. SDSA [15]). A simulation and dynamic stability and control analyser and flying–quality assessor. Includes:
  - Performance prediction.
  - Test flights by six Degrees of Freedom flight simulation.
  - Stability Augmentation System (SAS).
- Aero–elastic module NeoCASS [4]. Quasi–analytical structural analysis methods that support aero–elastic problem formulation and solution.

CEASIOM is intended to support engineers in the conceptual/preliminary design process of the aircraft, with emphasis on the improved prediction of stability and control properties of elastic aircraft achieved by higher–fidelity methods than found in contemporary aircraft design tools. Moreover CEASIOM integrates into one application the main design disciplines, e.g. aerodynamics, structures, and flight dynamics, impacting on the aircraft performance. It is thus a multi–disciplinary analysis toolbox brought to bear on the design of the aero–servo–elastic aircraft [20, 29]. CEASIOM however does not carry out the initial sizing of a baseline configuration, and thus needs to collaborate with a tool like AAA, which was described in Section C.3.

## C.5 Interfaces and Wrappers

If an analysis module is not developed to explicitly serve a central data model it is unlikely that the module and the central model share the same parameterization. Hence conversions need to be made. The first step in such a conversion is the filtering of data. By applying mapping rules only the data relevant for the analysis module is transferred. In a second step the tool wrappers do the conversion of the data.

Figure C.2 shows that all the related software tools for aircraft concept–design are linked to the central model approach CPACS [31] (visualized via CPACSCreator), then the data are sent to the higher order physics-based analysis tools CEASIOM. The baseline geometry studied in this paper is obtained from AAA–CPACS interface [26].



**Figure C.2:** Different conceptual design tools linked to preliminary design tool–chain CEASIOM by CPACS

## C.6 Ceasiom Down–Select Configuration & Pitch Control Study

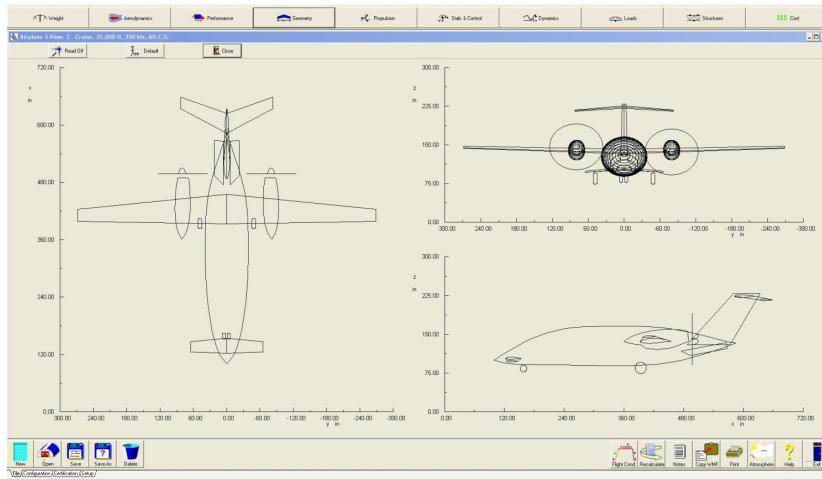
**Table C.1:** Structure of the aerodynamic database constructed in CEASIOM for use in the flight simulation SDSA module.

$\alpha$	$M$	$\beta$	$\delta_{ele}$	$\delta_{rud}$	$\delta_{ail}$	$p$	$q$	$r$	$C_L$	$C_D$	$C_m$	$C_Y$	$C_\ell$	$C_n$
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	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

Figure C.3 shows the AAA 3–view drawing of the final conceptual design for the Piaggio Avanti. This is the configuration that is down–selected from conceptual design and is now ready for preliminary design, as illustrated in Fig. C.1.

A primary goal of preliminary design is to obtain the final *wing design with optimized performance*: e.g. maximized aerodynamic efficiency or minimized drag coefficient ( $C_D$ ), usually starting with the cruise point.

The wing aerofoils are chosen by a skilful engineer as the initial design, which may not be optimum, but can be used as a good starting point. Then it should be put in an optimization loop to determine the optimized aerofoil shapes (thickness and cambers), twist distributions according to the limits of lift coefficient ( $C_L$ ), pitching moment coefficient ( $C_m$ ), bending moment, span loading, fuel tank volumes, etc. for corresponding flight conditions. Take the Piaggio Avanti for example, the aerofoils of the wing are well-designed with cambers and twists to give sufficient lift in order to balance the weight during cruise. Additionally to get a well designed wing a multi-level optimization [44] should be considered, namely, optimizations for cruise, take-off and landing.



**Figure C.3:** 3-view drawing of Piaggio–Avanti in AAA

Another goal is to determine a *database of aerodynamic forces and moments* that cover sufficiently the flight envelope so that it is appropriate input to a flight simulator for the study of the vehicle performances and handling qualities. Table C.1 indicate how the aerodynamic database computed in CEASIOM is organized. It shows the static and quasi-static stability coefficients and the control coefficients. In Table C.1 are presented the lift, drag and lateral force coefficients ( $C_L$ ,  $C_D$ ,  $C_Y$  respectively), pitching, rolling and yawing moment coefficients ( $C_m$ ,  $C_\ell$ ,  $C_n$  respectively). The angle of attack is presented as  $\alpha$ ,  $M$  is the Mach number and  $\beta$  the side slip angle while  $q$ ,  $p$  and  $r$  are the three rotations in pitch, roll and yaw. The three control surfaces that can be deflected are the elevator ( $\delta_e$ ), the rudder ( $\delta_r$ ) and the aileron ( $\delta_a$ ). The dynamic derivatives ( $C_{m_\alpha}$ ,  $C_{Z_\alpha}$ ,  $C_{X_\alpha}$ ,  $C_{Y_\beta}$ ,  $C_{\ell_\beta}$ ,  $C_{n_\beta}$ ) are instead computed only for different Mach numbers.

The coefficients must be obtained for each of these parameters throughout the flight envelope, hence the data is voluminous. In this paper we only focus on longitudinal control analysis in order to validate the advantage of this three-lifting-surface configuration. Thus only the second and sixth rows in Table C.1 are filled with CFD Euler computations.

## C.7 Piaggio Avanti

Piaggio Avanti is a three-lifting-surface twin-engine turboprop aircraft that has a small forward wing (canard) to produce extra lift and a T-tail for longitudinal and lateral control. It is claimed that it can save up to around 30% fuel compared with similar aircraft due to the non-traditional configuration. The main wing is designed to have laminar flow over a very high percentage of the wing chord, and the fixed canard is designed to stall before the main wing, resulting in a nose-down effect improving the airplane good performance at high angles of attack. The Piaggio Avanti is designed to cruise at Mach number of 0.62 at an altitude of 39,000 ft (economy cruise). More about the cruise speed and altitude can be referred in [8] and summarized in Table C.2 according to Instrument Flight Rules IFR Range & Payload graph.

**Table C.2:** Cruise speed at maximum cruise power [8]

Description	Speed [kts]	Altitude [ft]
Service ceiling at OEI	–	24,000
Maximum speed <sup>2</sup>	395	30,000
Cruise	370	37,000
*Economy cruise	356	39,000
Service ceiling	320	41,000

The typical mid-cruise weight is estimated by:

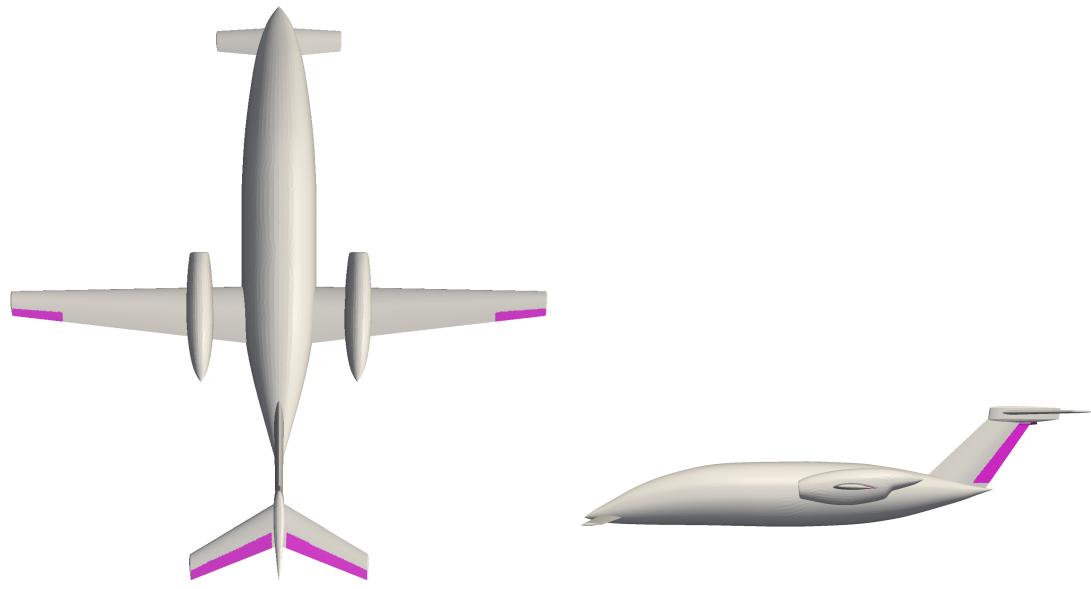
$$W = \text{Operating weight} + 4\text{PAX} + 1/2\text{Full fuel} \quad (\text{C.1})$$

at ISA condition and IFR reserves. The lift is produced to balance the weight, which is around 450 kN estimated by taking maximum payload and half full fuel [8]. The CFD solvers operate inside CEASIOM and then all the data are sent to Stability and Control Analysis module SDSA in CEASIOM to model/simulate the pitch controls.

Figure C.4 shows its 3-channel standard control surfaces illustrated on the 3-view drawing. The canard is a fixed lifting surface, and elevators on the horizontal tail control the pitch.

Tornado VLM and Edge Euler computations are carried out at Mach number 0.62 (economy cruise 356 kts) in order to build a complete aero-database to verify the advertised the superior flight qualities. The VLM method is fast but with lower fidelity. It is used to quickly generate a complete data-set as Table C.1 is shown that is sent into SDSA for stability and control analysis.

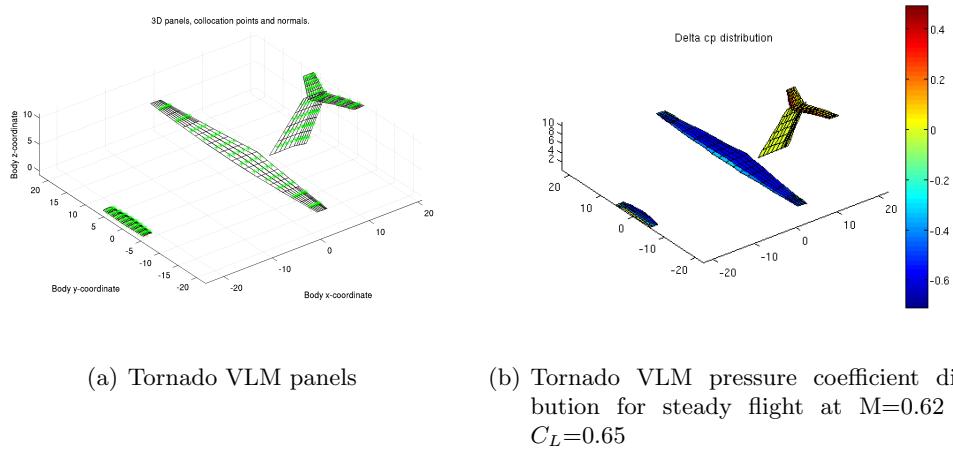
At the next higher level of detail, the graphical surface modelling tool **sumo** can be used to define a more detailed geometry based on a moderate number (often less than 30) spline surfaces. This description is used to generate input for CFD solutions based on the Euler equations. Horizontal trim is studied for both full and canard off configurations.



**Figure C.4:** Control surface illustrated on 3-view drawing [8]

### C.7.1 Aerodynamics using Vortex Lattice Method

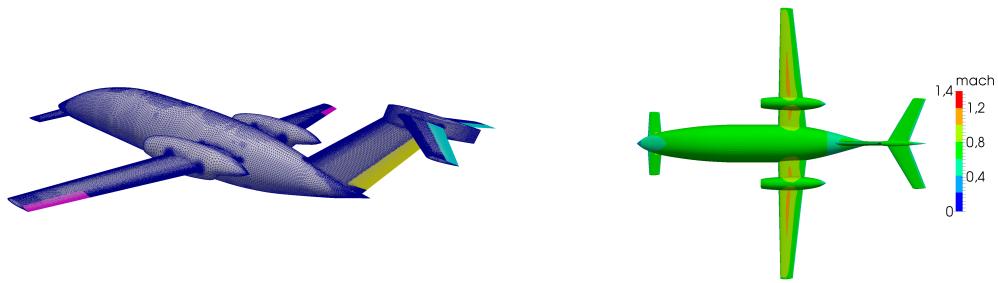
In VLM method the compressibility Prandtl–Glauert correction was applied for high subsonic Mach number (0.62). Figure C.5 are the VLM mesh and solution for economy cruise at Mach number of 0.62.



**Figure C.5:** Tornado VLM solutions from CEASIOM

### C.7.2 Euler solutions

Figure C.6(a) shows an Euler mesh used in this paper generated in `sumo` with 9.4 million nodes. The results for movable control surfaces are computed not by physically deflected, but by transpiration boundary condition in `Edge`. Figure C.6(b) shows the Mach contour on the same condition predicted by CEASIOM Euler solver. We see that a weak shock is formed at the mid-chord of the wing due to high lift. Note that the VLM model only includes lifting surfaces such as wing, vertical tail and stabilizer, while the `sumo` geometrical representation also includes aerofoil thickness and non-lifting surfaces such as the fuselage. Again the VLM only treats the flow fields around the lifting surfaces linearly, which has poor fidelity on non-linear aerodynamics such as the wing in transonic flow. All of these above motivate us to turn to Euler solver to get more accurate solutions.

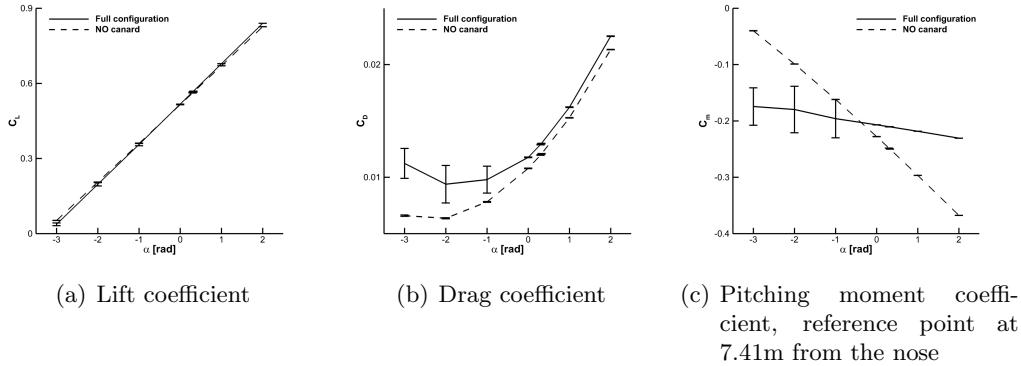


(a) Euler mesh generated by `sumo` and `Tet-Gen` [28] (b) Euler computation for cruise at  $C_L = 0.565$  and Mach=0.62 at 39,000 ft

**Figure C.6:** Euler solutions from CEASIOM

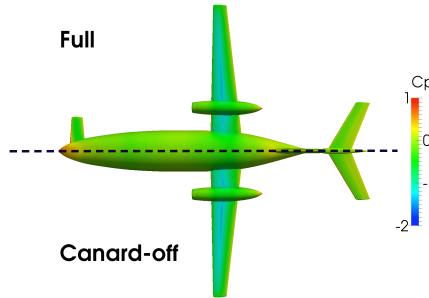
Figure C.7 shows the forces and moments predicted for both the full and the canard-off configurations from Euler solver in CEASIOM. The canard is very small that only produces slight lift, so the total lift from both configurations are quite close. We see significant differences on  $C_D$  and  $C_m$  for both configurations. Note that the error bars showing maximum deviations of the last 500 iterations indicated very poor convergence for steady flow computed for drag coefficient and pitching moment for negative angles of attack. Hence, the simulations below zero degree angle of attack is computed in unsteady model. This means that the deviation bars for the results of negative angle of attack cases indicates the actual unsteadiness predicted by simulation.

Figure C.9 shows the pressure distributions  $C_p$  from Euler solutions at trimmed flight conditions for both full & canard-off configurations. For the full configuration, the horizontal trim is achieved by deflecting the elevator up (negative) 3.33 deg at  $\alpha=0.56$  deg to maintain the desired lift. For the alternative geometry without canard, the horizontal trim can be achieved at  $\alpha=0.62$  deg with elevator deflection at  $\delta_e=-4.25$  deg. The elevator deflection angles are small for both configurations at trimmed flight, however the canard-off aircraft is too stable compared with its fully configured



**Figure C.7:** Euler solutions at Mach number of 0.62, canard ON & OFF cases; error bars represent maximum deviations during the last 500 CFD iterations

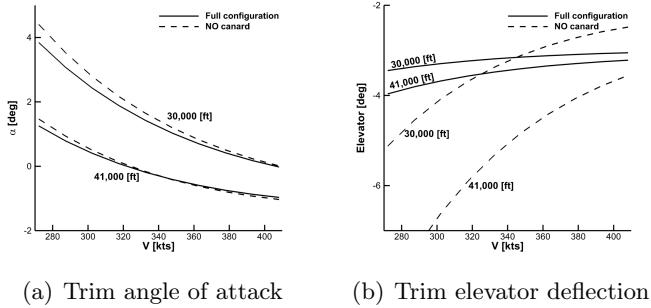
counterpart. During the presented study the center of gravity position was assumed unchanged between canard ON & OFF cases at 23.41 ft from the nose. As figure C.7(c) shows, the static margin for the full configuration is around 7.5% MAC, while for the canard-off one is around 42% MAC. Adding the canard moves the aerodynamic center forward that reduces the static margin accordingly.



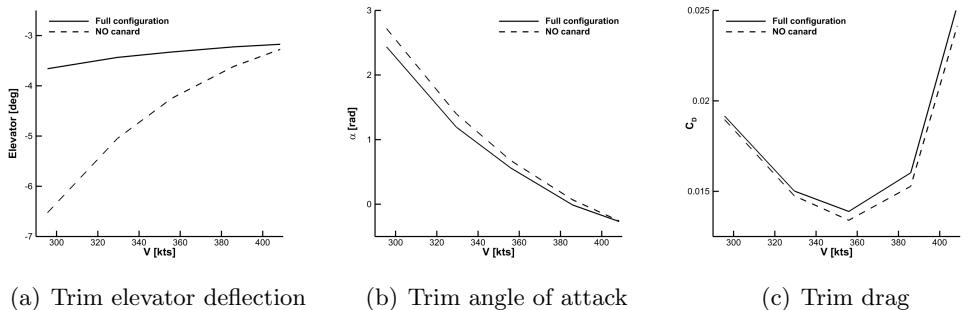
**Figure C.8:**  $C_p$  from Euler solutions for trimmed flight at Mach number of 0.62 ( $U = 356$  kts at 39,000 ft), full & canard-off configurations

Then we would like to ask why both configurations have very close trim angles. If we go back to figure C.7(c), we could see that, round  $\alpha = 0$  (computed cruise lift coefficient  $C_L \approx 0.56$ ), the pitching moments  $C_m$  for both configurations are quite close. This means to get the pitching moments balanced for both configurations require similar efforts, namely, similar nose-up moments provided from the horizontal stabilizer. However, if we expand the speed and altitude from the economy cruise point according to Table C.2, we found that although the differences of the trimmed angles of attack between the full and canard-off configurations are within 0.5 deg for concerned speeds and altitudes, the elevator deflections vary significantly for both configurations. From figures C.9 and C.10 we can see the full canard configuration superiors in the horizontal

trim flight: the elevator deflected angles are almost kept constantly at small negative values; while for the canard-off one the behaviour is on a common level, the elevator deflects less (in absolute value) as the speed increases.



**Figure C.9:** Longitudinal trim calculated and interpolated/extrapolated from Euler solutions for full (solid line) & canard-off (dashed line) configurations



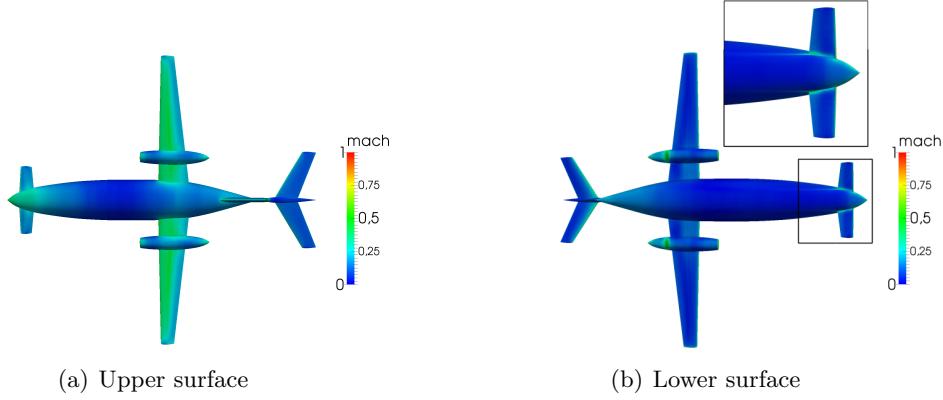
**Figure C.10:** Economy cruise trimmed conditions at altitude of 39,000 ft, calculated from CEASIOM-Euler

### C.7.2.1 Innovative design

The forward canard contributes to lift, since it is a fixed surface, and the pitch angle of the forward wing is configured so that it always stalls before the main wing. The resulting automatic nose-down effect assures excellent in flight behaviour at high angles of attack. These aerodynamic advantages resulting from the aircraft innovative design and construction, cause the airflow to be laminar over a very high percentage of the aircraft wing chord.

### C.7.2.2 Steady pull-up

To judge the manoeuvrability of the Piaggio Avanti, the *elevator per g* is calculated using Euler solver, by maintaining a steady pull-up with a constant angular velocity



**Figure C.11:** Mach contour from CEASIOM–Euler solutions for Piaggio Avanti 3g steady pull-up manoeuvre

as if the aircraft were attached to the end of a whirling arm provided very far away. Figure C.11 shows the Mach plot for the full configuration aircraft with 3g pull-up manoeuvre predicted from CEASIOM–Euler, at economy cruise condition. The free stream velocity is theoretically zero (but set to very small value, e.g. 1 m/s to assure numerical convergence in CFD computation) and the aircraft is evoked by the angular velocity. Canard provides additional lift, also notice that the flow starts to accelerate at the lower side of root of the canard. This manoeuvre can be achieved with incremental control force and elevator deflection, in order to climb to a higher altitude, for instance, to the service ceiling altitude.

Euler simulations for both full and canard–off configurations for 3g steady pull-up manoeuvre are obtained, for flight at 39,000 ft. The *elevator per g* for the full configuration is -1.58 deg, whereas the value for the canard–off configuration is around -4.8 deg, indicating that the full configuration has better manoeuvrability.

## C.8 Conclusions

The aircraft design stages, conceptual and preliminary, are necessarily collaborative by their very nature. An example design exercise was carried out to illustrate the collaborative aspects of design using the tools AAA and CEASIOM, working respectively on conceptual and preliminary design. The chosen example is the Piaggio Avanti that has three lifting surfaces and has an advantage in horizontal flight performance. The exercise brought out some of the details involved when exporting the configuration geometry from conceptual design, where the model is usually not water-tight and meshable, to preliminary design where a meshed model is the necessary starting point for further design work. In the example a small computer routine was written to convert the configuration data from AAA to input data for CEASIOM–sumo so that

the configuration was water-tight and meshable [26]. The common-language used to minimize the re-work is a standard for the data describing the aircraft, i.e., the CPACS standard proposed by DLR (see Ref [31]).

Euler simulations for both full and canard-off configurations at steady and level flight conditions were computed. The trim analysis is carried out a number of different cruise conditions, showing that the full configuration with a canard has advantages with minimized elevators angle changes while no (significant) drag was added. The *elevator per g* for the full configuration is much less than the configuration without a canard, when the aircraft is under a steady pull-up manoeuvre, provided that the neutral point is fixed. All of these above validates that the three-lifting-surface configuration has some aerodynamic advantages than conventional configurations.

In the future a more realistic model with propellers can be made in CEASIOM for Euler simulations. The effects are well modelled for inviscid flow and these should be considered in the design process, one good example is Lötstedt's work [48].