

CRANFIELD UNIVERSITY

VIS SRIPAWADKUL

GEOMETRICAL REPRESENTATIONS FOR EFFICIENT
AIRCRAFT CONCEPTUAL DESIGN AND OPTIMISATION

School of Engineering

PhD
Academic Year: 2008–2012

Supervisor: Prof Marin D Guenov
June 2012

CRANFIELD UNIVERSITY

School of Engineering

PhD

Academic Year 2008 -2012

VIS SRIPAWADKUL

GEOMETRICAL REPRESENTATIONS FOR EFFICIENT
AIRCRAFT CONCEPTUAL DESIGN AND OPTIMISATION

Supervisor: Prof Marin D Guenov

June 2012

© Cranfield University 2012. All rights reserved. No part of this publication may be reproduced without the written permission of the copyright owner.

ABSTRACT

Geometrical parameterisation has an important role in the aircraft design process due to its impact on the computational efficiency and accuracy in evaluating different configurations. In the early design stages, an aircraft geometrical model is normally described parametrically with a small number of design parameters which allows fast computation. However, this provides only a coarse approximation which is generally limited to conventional configurations, where the models have already been validated. An efficient parameterisation method is therefore required to allow rapid synthesis and analysis of novel configurations. Within this context, the main objectives of this research are: 1) Develop an economical geometrical parameterisation method which captures sufficient detail suitable for aerodynamic analysis and optimisation in early design stage, and 2) Close the gap between conceptual and preliminary design stages by bringing more detailed information earlier in the design process.

Research efforts were initially focused on the parameterisation of two-dimensional curves by evaluating five widely-cited methods for airfoil against five desirable properties. Several metrics have been proposed to measure these properties, based on airfoil fitting tests. The comparison suggested that the Class-Shape Functions Transformation (CST) method is most suitable and therefore was chosen as the two-dimensional curve generation method. A set of blending functions have been introduced and combined with the two-dimensional curves to generate a three-dimensional surface. These surfaces form wing or body sections which are assembled together through a proposed joining algorithm. An object-oriented structure for aircraft components has also been proposed. This allows modelling of the main aircraft surfaces which contain sufficient level of accuracy while utilising a parsimonious number of intuitive design parameters.

Three aircraft configurations: a twin-jet airliner, an unmanned aerial vehicle and a blended wing body, have been chosen for evaluation, covering both conventional and unconventional configurations. Aerodynamic analyses have been performed with a

potential flow solver and the results have been validated with high fidelity models and wind-tunnel data according to availability. The validation shows marked improvements in accuracy for an unconventional design where the empirical models are no longer valid. Finally, for evaluation purpose, the proposed parameterisation method and the selected potential flow solver have been integrated with relevant test cases for aircraft conceptual design. It is demonstrated that, by introducing higher fidelity into the conceptual design stage, it extends the scope of the design to more unconventional concepts with reliable aerodynamic analysis at affordable computational cost. Currently, the proposed parameterisation method is able to generate only the surfaces of the main aircraft components such as wing, tail, fuselage, and nacelle. Future work can be extended in a few directions, including the definition of control surfaces and a more detailed mission analysis, e.g., at take-off and landing which is required for noise prediction.

Keywords:

Surface Parameterisation, Aircraft Conceptual Design, Class-Shape Function Transformation, Aerodynamic Analysis, Multi-objective Optimisation

Publication by the author related to this thesis

Sripawadkul, V., Padulo, M., Guenov, M.D.,2010, A Comparison of Airfoil Shape Parameterization Techniques for Early Design Optimization, 13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference, 13-15 September 2010, Fort Worth, Texas, AIAA 2010-9050.

Guenov, M.D., Nunez, M., Molina-Cristobal, A., Sripawadkul, V, Datta, V, and Riaz, A.,2013, Composition, Management and Exploration of Computational Studies at Early Design Stage,CIAERO Computational Intelligence book chapter.

ACKNOWLEDGEMENTS

First, I would like to express my sincere gratitude to Prof. Marin Guenov, my supervisor, for his guidance, encouragement, patience and tremendous support through the course of my study. His determination and dedication to work truly inspires me.

My sincere thanks also go to all my colleagues in Advanced Engineering Design Groups: Marco, Mattia, Jeremy, Libish, Aditya, Arturo, Varun and Atif, for their kind supports, helpful comments and suggestions. Being part of this highly active research group has been a valuable experience.

I would like to extend my gratitude to Ministry of Science, the Royal Thai Government and Department of Aerospace Engineering, Kasetsart University, for the generous financial supports throughout my study.

Thank you my Thai friend in Cranfield for all the help, supports, and many good moments we shared together.

I dedicate this thesis to my mother and my sister, who are always giving me strength to continue pursuing my dreams.

Vis Sripawadkul

TABLE OF CONTENTS

ABSTRACT	i
ACKNOWLEDGEMENTS	iii
LIST OF FIGURES	vi
LIST OF TABLES	ix
NOMENCLATURE	x
1 Introduction	15
1.1 Conceptual Design	15
1.2 Preliminary Design	17
1.3 Multidisciplinary nature of design	17
1.4 Fidelity Level	21
1.5 Shape Parameterisation	23
1.6 Aim and Objectives	24
1.7 Thesis structure	24
2 Literature Survey	25
2.1 Curve Parameterisation methods	25
2.1.1 NACA airfoil	26
2.1.2 Ferguson's Curve	27
2.1.3 Splines	29
2.1.4 Hicks-Henne Shape Functions	30
2.1.5 Parameter Section (PARSEC)	31
2.1.6 Class-Shape function Transformation (CST)	34
2.2 Curve Parameterisation Methods Comparison	43
2.3 Surface Parameterisation Methods	38
2.3.1 Partial Differential Equations	38
2.3.2 NURBS	41
2.3.3 Free Form Deformations	42
2.4 Geometry Generation Tools	43
2.4.1 Characteristic Curves	46
2.4.2 ICAD Multi-Model Generator	47
2.4.3 Rapid Aerospace Geometry Engine	48
2.4.4 Vehicle Sketch Pad	49
2.5 Summary and Conclusions	51
3 Curve Parameterisation	52
3.1 Assessment Criteria	53
3.2 Airfoil Parameterisation Assessment	54
A. Parsimony	54
B. Completeness	56
C. Orthogonality	58
D. Flawlessness	60
E. Intuitiveness	61
3.3 Results and analysis	61
A. Parsimony	61
B. Completeness	63
C. Orthogonality	64
D. Flawlessness	65

E.	Intuitiveness	66
3.4	Summary	67
3.5	Cross-section Parameterisation Comparison	68
3.6	Conclusions.....	69
4	Aircraft Parameterisation Method	70
4.1	Axes Conventions	71
4.2	Geometry Classes	72
4.2.1	Cross-section	73
4.2.2	Distribution Class	75
4.2.3	Surfaces Class.....	78
4.2.4	Component Class.....	80
4.2.5	Aircraft.....	81
4.3	Surface Lofting	82
4.3.1	Wing	82
4.3.2	Body Section	84
4.3.3	Grid discretisation	86
4.4	Intersection of surfaces	87
4.5	Surface Area Calculation	89
4.6	Examples of the Parameterised Aircraft Models	90
4.6.1	Conventional transport	90
4.6.2	Blended Wing Body	94
4.6.3	Unmanned Aerial Vehicle	97
4.7	Summary.....	100
5	Aerodynamic Analysis and Validation.....	101
5.1	Aerodynamic Coefficients	102
5.1.1	Lift	102
5.1.2	Drag	102
5.2	Potential Flow	104
5.3	Friction/ Form Drag Model.....	107
5.3.1	Composite formula	108
5.3.2	Form factor	109
5.3.3	Wetted area calculation	109
5.4	Convergence Study	111
5.5	Aerodynamic Analysis Models from the Selected Test Cases	114
5.5.1	USMAC	114
5.5.2	FLOPS	115
5.6	Results and Analysis	116
5.6.1	Conventional.....	116
5.6.2	Blended Wing Body	118
5.6.3	Unmanned Aerial Vehicle: UAV-KU4	119
5.7	Conclusions.....	121
6	Applications in Optimisation and Evaluations	122
6.1	Integration into Model-Based Design Tool	122
6.1.1	Optimisation Study #1	125
6.1.2	Optimisation study #2.....	129
6.1.3	Optimisation study #3.....	131
6.2	Evaluations	134
7	Summary and Conclusions	136

7.1	Literature Survey	136
7.2	Airfoil Parameterisation Method Comparison.....	136
7.3	Surface Parameterisation Method	137
7.4	Aerodynamics analysis	137
7.5	Evaluation	138
7.6	Novelty and Contribution to knowledge.....	138
7.7	Current Limitations.....	139
7.8	Future work.....	139
REFERENCES		141
APPENDICES		148
Appendix A	PARSEC-CST Equivalent form	148
Appendix B	Empirical Model from Industrial Cases	149
Appendix C	MATLAB code for Geometry Generation	153

LIST OF FIGURES

Figure 1-1	Aircraft Design Process (Raymer, 1992).....	16
Figure 1-2	Increasing Numbers of Design Parameters (Price <i>et al.</i> , 2006).....	17
Figure 1-3	Design Spiral (Keane and Nair, 2005).....	18
Figure 1-4	Schematic of convention structural design process (Sensmeier and Samareh, 2005).....	19
Figure 1-5	Aerodynamics analysis (Vandenbrande <i>et al.</i> , 2006)	19
Figure 1-6	Fidelity Levels in Various Disciplines (Price <i>et al.</i> , 2006).....	21
Figure 1-7	Design Fidelity in Conceptual and Preliminary Design Stages Aircraft Models (Kroo <i>et al.</i> , 2005)	22
Figure 2-1	Changes in an airfoil described by a series of points (Keane and Nair, 2005)	26
Figure 2-2	Airfoil Described by 2 Ferguson Curves (Sobester and Barrett, 2008).....	28
Figure 2-3	B-Spline curve representing airfoil with control points	30
Figure 2-4	A set of 10 Hicks-Henne shape function $t=4$	31
Figure 2-5	Design Parameters for PARSEC method (Padulo <i>et al.</i> , 2009).....	32
Figure 2-6	RAE 2822 profile (top) with corresponding shape functions (bottom).....	36
Figure 2-7	Airfoil Profiles [From top left, NACA 0012, top right: RAE2822, bottom left: NACA 63(2) -015 and the bottom right, NASA SC(2)-0714].....	37
Figure 2-8	Blend circular cylinder to a plane (Bloor and Wilson, 1989a)	38
Figure 2-9	Aircraft Representing by 5 PDE surface patch (Ugail, 2003)	40
Figure 2-10	PDE surfaces joining fuselage and wing section, with $a=1$ (middle) and $a=100$ (right) (Pasadas and Rodriguez, 2008)	40
Figure 2-11	NURBS approximation of BWB (Mastin <i>et al.</i> , 1996)	42
Figure 2-12	Free-form deformation grids on the wing-body intersection (Samareh, 2004).....	42
Figure 2-13	Characteristic functions (Trapp and Sobieczky, 1999).....	46
Figure 2-14	Example of curve represented by base functions (Trapp and Sobieczky, 1999).....	47
Figure 2-15	Wing trunk and connection element (La Rocca <i>et al.</i> , 2002)	48
Figure 2-16	Example of RAGE fuselage and wing built by lofting functions (Rodriguez and Sturdza, 2006).....	48

Figure 3-1 Overview of the Research Methodology	52
Figure 3-2 Parsimony Test	56
Figure 3-3 Airfoil Fitting Test	57
Figure 3-4 Non-orthogonal set of CST design variables (Ceze et al., 2009)	59
Figure 3-5 Inconsistent Maximum Crest Locations (Padulo <i>et al.</i> , 2009)	60
Figure 3-6 Intersections between upper and lower surface (Padulo <i>et al.</i> , 2009)	60
Figure 3-7 Number of Hicks-Henne bumps on representation of RAE2822 airfoil	62
Figure 3-8 Parameterised RAE2822 airfoil by Ferguson's curve and PARSEC	63
Figure 3-9 Examples of irregular shapes produced with PARSEC method.....	65
Figure 4-1 Aircraft components	70
Figure 4-2 Aircraft geometric axis conventions	71
Figure 4-3 Classes and attributes for full aircraft geometry generation.....	72
Figure 4-4 Bodycross Description	75
Figure 4-5 Cross-section shapes corresponding to NCI Coefficients.....	75
Figure 4-6 Bernstein polynomial (left) with its corresponding engine cowling line (right).....	77
Figure 4-7 Closed surface form functions	78
Figure 4-8 Wing section definitions	79
Figure 4-9 Body section descriptions	79
Figure 4-10 Conventional wing (left) and Blended Wing with multiple wing sections (right).....	80
Figure 4-11 Fuselage (left) and nacelle (right).....	80
Figure 4-12 Wing section with linear interpolation (left) and Bernstein distribution (right).....	83
Figure 4-13 Nose surface generation.....	84
Figure 4-14 Cockpit surface generation	85
Figure 4-15 Fuselage surface generation.....	85
Figure 4-16 Wing surface: linear distribution (left) and cosine distribution (right).....	86
Figure 4-17 Fuselage nose surface: linear distribution (left) and cosine distribution (right).....	87
Figure 4-18 Intersecting of the horizontal tail to the rear fuselage; pre-intersection (left) and post-intersection (right)	88
Figure 4-19 Arbitrary quadrilateral and the midpoint parallelogram (Softsurfer, 2012)	89
Figure 4-20 Airbus A320 3-View Configuration (Airbus, 2010).....	91
Figure 4-21 Parameterised A320 in 3 View	94
Figure 4-22 Parameterised MOB.....	96
Figure 4-23 Unmanned Aerial Vehicle (In-noi <i>et al.</i> , 2004).....	97
Figure 4-24 Parameterised UAV-KU4	99
Figure 5-1 CFD fidelity with corresponding level of Geometry descriptions (Adapted from Rizzi <i>et al.</i> , 2010, CAD model from WIPD-Pro CAD 2010).....	101
Figure 5-2 Drag Component (Gur <i>et al.</i> , 2010).....	103
Figure 5-3 The vortex horseshoe position and the collocation point (Melin, 2010)	106
Figure 5-4 TN D-5971 in Vortex Lattice analysis with Mach numbers from 0.55 to 0.85	107
Figure 5-5 A320 in Vortex Lattice analysis with Mach numbers from 0.55 to 0.85 ...	107
Figure 5-6 TN D-5971 Wind Tunnel Model (left) (Capone, 1970) and Parameterised Model (right)	109
Figure 5-7 Lift and Drag coefficients with respect to number of panels (chord wise).	112

Figure 5-8 Computation time with respect to number of panels (chord wise)	112
Figure 5-9 Lift and Drag coefficients with respect to the number of panel (span wise)	113
Figure 5-10 Computation time with respect to the number of panel (span wise)	114
Figure 5-11 A320 lift and induced drag coefficients with VLM and FLOPS	116
Figure 5-12 A320 Drag Polar at various Mach numbers with VLM (left) and FLOPS (right).....	118
Figure 5-13 BWB Drag Polars with RANS, FLOPS, and VLM.....	119
Figure 5-14 Lift coefficient versus angle of attack.....	120
Figure 5-15 UAV-KU4 Drag Polar	120
Figure 6-1 Subset of USMAC Workflow (geometry and aerodynamic modules)	123
Figure 6-2 Integration of USMAC with VS Geo and VLM.....	124
Figure 6-3 Optimisation study: FLOPS	127
Figure 6-4 Optimisation study: FLOPS + VLM.....	127
Figure 6-5 Objectives Space [Original FLOPS (white) and FLOPS+VLM (yellow)].	128
Figure 6-6 FLOPS + VLM with airfoil design parameters (A7-A12).....	130
Figure 6-7 Airfoil shapes at the Pareto front	131
Figure 6-8 Prototype 63 ₂ -415 airfoil (black-dashed), optimal section at root (blue), and optimal section at tip (red) with the corresponding CST parameters	133

LIST OF TABLES

Table 2.1 Airfoil Parameterisation Comparison Studies	44
Table 3.1 List of desirable properties for airfoil parameterisation methods	53
Table 3.2 Design parameters for each parameterisation methods	54
Table 3.3 Parsimonious numbers of design variables	63
Table 3.4 Percentage of airfoils with satisfactory fitting approximation	64
Table 3.6 Comparison of selected airfoil parameterisation methods	67
Table 5.1 Calculated Wetted Area [m ²].....	110
Table 5.2 Friction Drag Model Validation	111
Table 5.3 Number of span wise panels on each section	113
Table 5.6 Aerodynamic Coefficients Comparison	117
Table 5.7 Lift and drag coefficients for BWB.....	119
Table 6.1 Induced Drag Models Merging	125
Table 6.2 Evaluations of the integration of VSGeo and VLM into FLOPS.....	134

NOMENCLATURE

ACRONYMS

BWB	Blended Wing Body
CAD	Computer Aided Design
CFD	Computational Fluid Dynamic
CST	Class-Shape function Transformation
EDET	Empirical Drag Estimation Technique
FEM	Finite Element Analysis
FFD	Free Form Deformation
FLOPS	Flight Optimisation System
LE	Leading Edge
MDO	Multidisciplinary Design and Optimisation
NACA	National Advisory Committee for Aeronautics
PARSEC	Parameter Section
PDE	Partial Differential Equation
TE	Trailing Edge
UAV	Unmanned Aerial Vehicle
USMAC	Ultra Simplified Model of Aircraft
VLM	Vortex Lattice Method
VSGeo	Visualised Surface Geometry

SYMBOLS

b	Wing span
c	Chord length
CL	Lift coefficient
CDi	Induced drag coefficient
CDf	Friction drag coefficient
CDc	Compressible drag coefficient
CD	Total drag coefficient
F_i	Force in i axis
H	Fuselage cross-section height

M_i	Moment in i axis
NC	Cross-section coefficient
ND	Distribution coefficient
\mathbf{n}_i	Normal unit vector
p	Pressure
q	Dynamic reference pressure
R_{le}	Leading edge radius
Re	Reynolds number
S_{ref}	Reference surface area
S_{wet}	Wetted surface area
W	Fuselage cross-section width
X_{up}, X_{lo}	Upper and lower crest location
Z_{up}	Camber line
Z_{up}, Z_{lo}	Upper and lower crest value
$Z_{xx,up}, Z_{xx,lo}$	Upper and lower curvature at crest location
Z_{te}	Trailing edge vertical coordinate
ΔZ_{te}	Trailing edge thickness
Λ_{le}	Leading edge sweep angle
α_{te}	Trailing edge direction
β_{te}	Trailing edge wedge angle
ψ	Normalised chord wise position (x/c)
η	Normalised span wise position (y/b)
ζ	Normalised chord wise vertical coordinate (z/c)
ρ	Fluid density
v	Speed
ϕ	Scalar flow potential
USMAC	
span	span
Awing	wing planform area
dfus,	fuselage diameter
lfus	fuselage length

Aht	horizontal tail area
Avt	vertical tail area
dnac	nacelle diameter
ar	aspect ratio
ne	number of engine
wAwing	wing wetted area
Aref	wing reference area
Lref	reference length
wAfus	fuselage wetted area
wAht	horizontal tail area
wAvt	vertical tail area
wAnac	nacelle wetted area
mass_crz	total mass at cruise
g_crz	gravitational acceleration at cruise
Mach_crz	Mach number at cruise
Pamb_crz	Ambient pressure at cruise
Tamb_crz	Ambient temperature at cruise
cx0_crz	friction drag coefficient
cx _i _crz	induced drag coefficient
cx _c _crz	pressure drag coefficient
Kind	induced drag factor
Kcx0	friction drag factor
Kcx _p	pressure drag factor

FLOPS

aircraftBaseArea	aircraft base area
apprSpeed	maximum allowable landing approach velocity
canFormFac	canard form factor
canWetArea	canard wetted area
canCharacLen	canard characteristic length
CDITAB[]	vector of induced drag coefficients
crzMachNum	cruise Mach number
designRange	design range

desLiftCoeff	design lift coefficient
desMachNum	design Mach number
dihedAng	dihedral angle
fuseCrossSecArea	fuselage cross-section area
fuseFormFac	fuselage form factor
fuseDia	fuselage diameter
fuseDia_wingSpan_R	fuselage diameter/wingspan ratio
fuseLenDia_R	fuselage length /diameter ratio
horTailCharacLen	horizontal tail characteristic length
horTailFormFac	horizontal tail form factor
horTailWetArea	horizontal tail wetted area
horTailSwp	horizontal tail sweep
horTailTapR	horizontal taper ratio
ldnFldLen	maximum allowable landing field length
liftCoeffs []	vector of lift coefficients
machNums[]	vector of Mach numbers
maxCrzAlt	maximum cruise altitude
nacFormFac	nacelle form factor
nacWetArea	nacelle wetted area
nacCharacLen	nacelle characteristic length
noOfLiftCoeff	number of lift coefficients
noOfMachNum	number of Mach numbers
noOfVerTail	number of vertical tail
noOfFuse	number of fuselage
noOfEngine	number of engine
rampW	ramp weight
takoffFldLen	maximum allowable take-off field length
verTailCharacLen	vertical tail characteristic length
verTailFormFac	vertical tail form factor
verTailSwp	vertical tail sweep
verTailTapR	vertical tail taper ratio
verTailWetArea	vertical tail wetted area

wingAsp_R	wing aspect ratio
wingCamber	wing camber
wingCharacLen	wing characteristic length
wingFine_R	wing finess ratio
wingFormFac	wing form factor
wingRefArea	wing reference area
wingSkinFricCoeff	wing skin friction coefficient
wingSweep	wing sweep
wingTap_R	wing taper ratio
wingThkChd_R	wing thickness to chord ratio
wingWetArea	wing wetted area

1 Introduction

Aircraft life cycle is an iterative process which can be divided into eight stages, starting from customer requirements, conceptual design, preliminary design, detailed design, manufacturing and assembly, flight test, operation, and finished in a disposal stage, as demonstrated in Figure 1-1. Successful review of solutions from an earlier stage is passed to the next one where more details are added in each subsequent stage as the design progresses.

Engineering especially aerospace put special attention to the conceptual and preliminary design stages where the design concepts are explored and chosen. Different design options are considered in conceptual stage and continue to work out in more detail in preliminary design stage.

1.1 Conceptual Design

The conceptual design phase addresses the highest level questions about the proposed aircraft. In particular the main requirements and desired functions are considered. Normally a number of potential configurations are outlined which will undergo a trade-off study. The solution which best matches the requirements will be chosen. Like any design process this phase is highly iterative, but it tends to be the most open and unconstrained phase of aircraft design, so the largest number of design solutions will be explored here. (Price *et al.*, 2006)

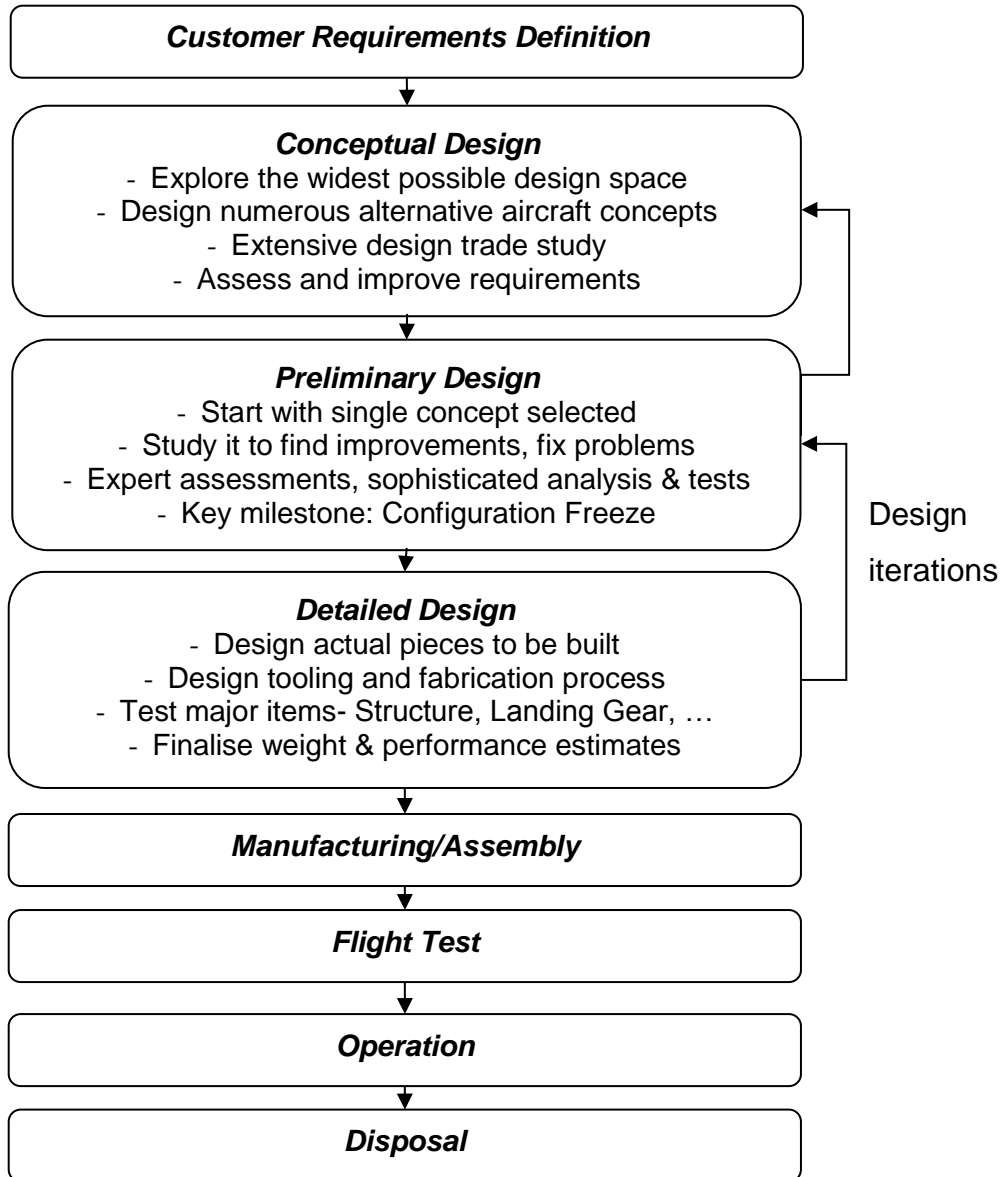


Figure 1-1 Aircraft Design Process (Raymer, 1992)

The result of the conceptual phase is a configuration with the basic size and arrangement of its main aspects for instance the wing, empennage, engines, fuselage, control surfaces, etc. These data are built from few initial equations and empirical data which provide a starting point for the design. As the design proceeds, more values are being fixed and the design space is being gradually reduced. However, at the same time the number of design parameters is increasing as shown in Figure 1-2.

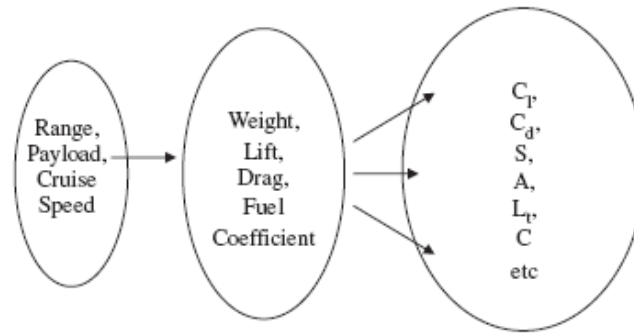


Figure 1-2 Increasing Numbers of Design Parameters (Price *et al.*, 2006)

1.2 Preliminary Design

Once the concept has been accepted and the design space is sufficiently well defined or constrained, preliminary design may proceed. Each major system of the aircraft is now considered again and more detailed estimates of size, thickness, material are made. The number of design parameters is now elevated.

Design and analysis continues through the detailed stage including activities such as stressing of the final shape and form of the components. Manufacturing details are added, specifying exactly how the aircraft will be built.

1.3 Multidisciplinary nature of design

As discussed earlier, aircraft design process is highly iterative in nature. Each discipline is reviewed in turn in order to keep the detailed definitions within acceptable bounds as shown in Figure 1-3. It is highly unusual for any detail change to affect the overall concept since the design of such local features is highly constrained. However, such changes can be difficult to handle if they occur. This is the motivation to investigate how to facilitate bridging between the conceptual and preliminary design stages efficiently.

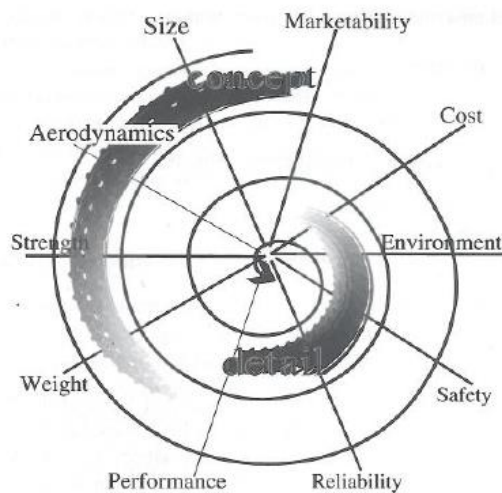


Figure 1-3 Design Spiral (Keane and Nair, 2005)

Different levels of model fidelity are used in each design stage. Figure 1-4 shows a schematic representation of the conventional conceptual and preliminary design phases. Vehicle weights are estimated as a function of the configuration parameters using historical or generic analysis-based data. The actual structural design is typically not selected until the preliminary design phase. Thus, if the aircraft being designed is not much different from those on which the weights database is constructed, the designed weights will be fairly close to the estimated ones obtained during the conceptual design. However, if the new configuration is different from original, there will be a large uncertainty in weight estimation. At this stage of the design process, making significant changes to the aircraft configuration can be extremely difficult since these changes will affect many systems, not just structures.

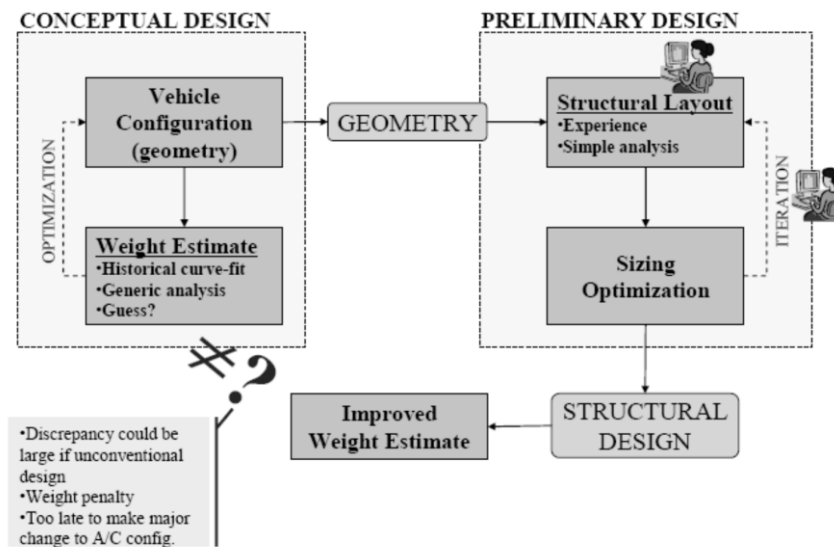


Figure 1-4 Schematic of convention structural design process (Sensmeier and Samareh, 2005)

Similar scenarios occur in aerodynamic analysis and optimisation. In a conceptual study, the aerodynamics coefficients are calculated with simple empirical models. Once the outline configuration is fixed, aerodynamic analysis commences. The geometrical model is usually described in high fidelity with thousands of grid points. In this stage, computational tools have no direct impact on the geometry of the design and the designer has to make a decision on how to change the design as a result of studying the outputs, as illustrated in Figure 1-5.

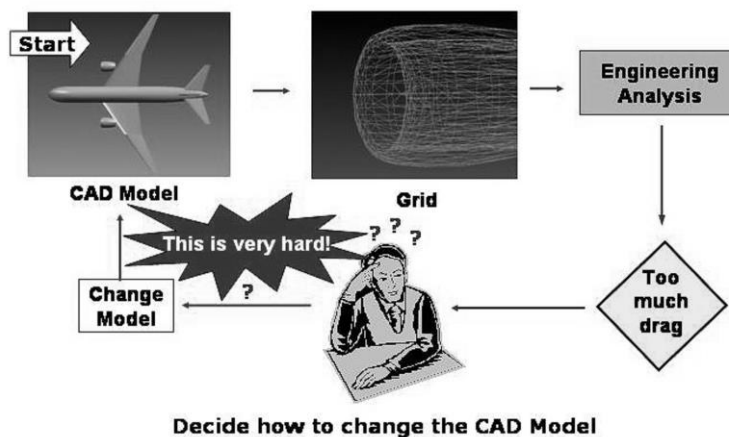


Figure 1-5 Aerodynamics analysis (Vandenbrande *et al.*, 2006)

Even though the structural and aerodynamic analyses are coupled in the design process, shape parameterisation is more concerned with aerodynamic analysis than structural analysis since flow analysis are often the most subtle and difficult to deal with (Keane and Nair, 2005). An efficient parameterisation method which contains sufficient detail descriptions and intuitive design parameters is therefore required early in the design process and this is the second motivation behind this research.

1.4 Fidelity Level

The *fidelity* of a mathematical or computational model is the degree to which the model accuracy reflects reality. Higher fidelity models correspond closer to the reality (Robinson, 2007). Low fidelity models tend to use simple equations and look up tables, and often will not have any geometric models associated (Price *et al.*, 2006). Medium fidelity models will mostly have some form of linear analysis and high fidelity models contain a lot of detail and often model non-linear behaviour. Thus, in order to make trade-off study in the early design stages, low fidelity models are often used. These models are generally fast and many different disciplines may be considered at the same time. High fidelity models are to be used later in the design phase in local areas since they are expensive in time and resources (Armstrong *et al.*, 2002).

It becomes obvious that the appropriate mix of fidelity level is needed to be optimised in order to get reliable answers for as many configurations as possible (Knight *et al.*, 2002, and Blair *et al.*, 2006).

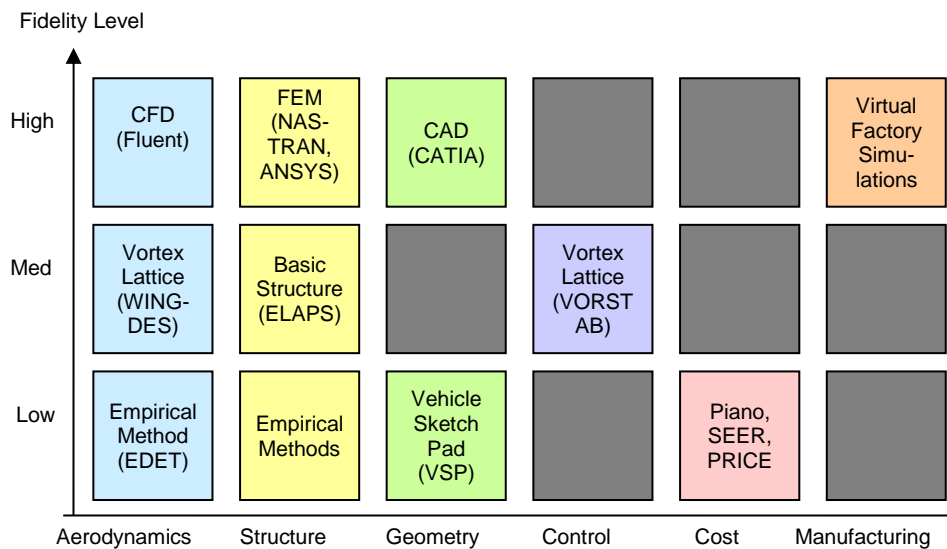
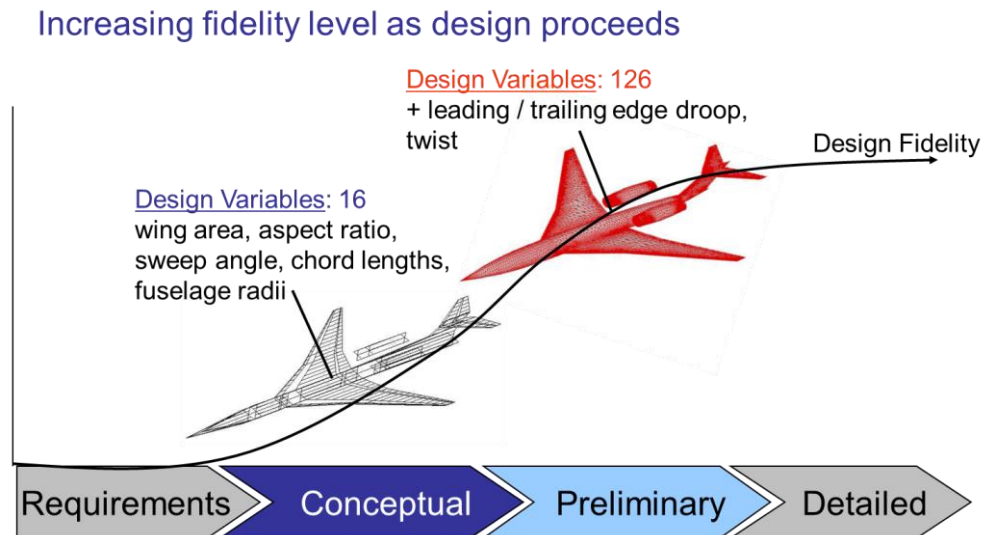


Figure 1-6 Fidelity Levels in Various Disciplines (Price *et al.*, 2006)

Figure 1-6 summarises tools with different fidelity levels used in various analysis disciplines. In aerodynamic analysis, the model is first analysed with simple empirical model or Vortex Lattice method in low-medium fidelity level, before further analysed with high fidelity tools such as Fluent in Computational Fluid Dynamic (CFD) in later

stage. In structural analysis, the high fidelity model used for Finite Element Analysis might contain thousands of parts, much more detailed than the one in the low fidelity.

At any point of the design, trade-offs may be required focused on one given aspect. The data needed for that may come from several levels of fidelity. For instance, a Finite Element Analysis (FEA) model may take data from low fidelity aerodynamic model.



**Figure 1-7 Design Fidelity in Conceptual and Preliminary Design Stages
Aircraft Models (Kroo *et al.*, 2005)**

Figure 1-7 shows examples of supersonic aircraft models in two levels of fidelity (Choi *et al.*, 2005). The low fidelity model on the left uses 16 design variables: the wing area, aspect ratio, sweep angle, the location of the wing root leading edge, the thickness of chord length at three locations on the wing, the minimum cockpit diameter, the minimum cabin diameter, and the fuselage radii at six locations. While the medium fidelity model (Figure 1-7 right) uses 126 design variables: leading edge and trailing edge droop, twist, and 15 Hicks-Henne bumps at each of 7 locations on the wing. These two models present the use of the parameterisation of relevant level of descriptions.

1.5 Shape Parameterisation

Nowadays, computer aided design (CAD) is widely used as drawing and drafting tool. Modern CAD has become more sophisticated and can capture complex geometry using Non-uniform Rational B-Spline (NURBS) curves and surfaces. CAD allows geometrical data transferred between analysis tools. However, using control points as design variables in CAD is considered to be computationally demanding, and does not provide physical meaning to designers to understand the change they are trying to make over the design. Other analytical techniques were then developed to overcome this challenge. The two main needs for efficiency in design is the compromise between:

- Achieving sufficient level of detail and local control
- Minimising complexity of design task

1.6 Aim and Objectives

The overall aim is to investigate how to provide designers with more information earlier in the design process through an efficient geometrical parameterisation method.

The objectives supporting this aim have been formulated as follows:

1. Develop an economical geometrical parameterisation method which captures sufficient detail focusing on aerodynamic analysis and optimisation in early design stage
2. Close the gap between conceptual and preliminary design stages by bringing more detailed information earlier in the design process.

1.7 Thesis structure

This thesis contains 7 chapters. The next chapter (Chapter 2) gives an insight of shape parameterisation methods. It focuses first on the parameterisation methods for airfoils and then extended to surface generation methods. In Chapter 3, five chosen methods for airfoils were evaluated against a set of desirable criteria and summarised in comparison matrices. Chapter 4 presents the proposed parameterisation method for three dimensional aircraft components and then assembly into full aircraft configuration including the joining algorithm for intersections. Chapter 5 presents the aerodynamic analysis results and validation for representative aircraft configurations. The proposed parameterisation method and the selected aerodynamic analysis tool has been integrated with an industrial test case and evaluated in Chapter 6. The conclusions, novelty, current limitation and future work are finally summarised in Chapter 7.

2 Literature Survey

In order to develop an economical geometrical parameterisation method, the widely used shape parameterisation methods must be studied and evaluated. This chapter gives a review of shape parameterisation methods. First it focuses on the parameterisation methods for curves and then extends to surface generation methods. The list of desirable properties for application in optimisation has also been summarised at the end of this chapter.

2.1 Curve Parameterisation methods

Since the basic process of design involves the making of decisions that change the product definition, the purpose of geometric parameterisation is to aid in this design process by providing increasingly powerful manipulation schemes for changing design definitions. (Keane and Nair, 2005)

~~The starting~~One of the simplest form of parameterisation is indiscrete form, which uses sequence of coordinates of sample points with linear interpolation between these points to generate curves and surfaces. Since it uses each point as a design variable, this method provides flexible parameterisation and allows designers to manipulate the shape easily. The higher level of accuracy can be achieved by adding more grid points. However, this leads to high computational cost of the analysis. It is also difficult to maintain a smooth geometry since each point is independent from the others, as shown in Figure 2-1. Moreover, this method cannot provide consistent representation for analysis in various disciplines since the grid formulations are different. Analytical methods were then developed to improve the control of curves.



Figure 2-1 Changes in an airfoil described by a series of points (Keane and Nair, 2005)

An attempt to link multiple points together with control over local curvature and smoothness leads to the use of spline curves and polynomials. Detail formulations of each parameterisation method are described as below:

2.1.1 NACA airfoil

National Advisory Committee for Aeronautics (NACA) has developed one of the earliest geometric parameterisation for airfoils through experimental and form a famous NACA-4 digit series. These airfoils have been used extensively in the aircraft industry but have been gradually replaced by more advanced methods developed on CFD basis. The NACA Report 460 (Jacobs *et al*, 1933) presented the definition of NACA-4 digit which can be summarised as follows:

An airfoil is described with an expression for camber line with a thickness distribution on either side of this line. This forms the upper and lower surfaces in two-dimensional (x,z) coordinates. The camber line, z_c , consists of one parabola from leading edge to the point of maximum camber, and another parabola extending from this point to the trailing edge as:

$$z_c = z_{c,max} \left(\frac{1}{x_m^2} \right) \left(2x_m \frac{x}{c} - \left(\frac{x}{c} \right)^2 \right) \text{ for } 0 \leq \frac{x}{c} < x_m$$

$$z_c = z_{c,max} \left(\frac{1}{(1-x_m)^2} \right) \left(1 - 2x_m + 2x_m \frac{x}{c} - \left(\frac{x}{c} \right)^2 \right) \text{ for } x_m \leq \frac{x}{c} < 1 \quad [2.1]$$

where $z_{c,max}$ is maximum camber, x_m is chord wise position of maximum camber, c is chord length.

The thickness distribution is described with polynomial function where the coefficients are determined by fitting Clark Y and Gottingen-398 airfoils.

$$z_t = 5z_{tmax}(0.2969\sqrt{x/c} - 0.1260 x/c - 0.3537(x/c)^2 + 0.2843(x/c)^3 - 0.1015(x/c)^4)[2.2]$$

where z_{tmax} is airfoil maximum thickness.

The airfoil coordinates are given by

$$\begin{aligned} x_U &= x - z_t \sin(\theta) & z_U &= z_c - z_t \cos(\theta) \\ x_L &= x + z_t \sin(\theta) & z_L &= z_c - z_t \cos(\theta) \end{aligned} \quad [2.3]$$

where $\theta = \arctan\left(\frac{dz_c}{dx}\right)$

The NACA-4 digit system uses three design variables: the first digit refers to the airfoil maximum camber, the second digit is the position of maximum camber in tenth of the chord, and the last two digits are maximum thickness as a per cent of chord. This results in smooth shape and high flexibility while guarantees airfoil shapes. Although these airfoils are easy to produce, they generate high lift compare to the new airfoils. (Sadrey, 2012)

2.1.2 Ferguson's Curve

Ferguson's curve was introduced in Computer Aided Design (CAD) in 1964. A curve $Z(t)$ is defined as the polynomial on parametric points, t , as follows:

$$Z(t) = \sum_{i=0}^3 a_i t^i, t \in [0,1] \quad [2.4]$$

The curve starts from point $S(0) = A$, until the end point, $S(1) = B$, with corresponding tangent vectors $\left.\frac{dZ}{dt}\right|_{t=0} = T_A$ and $\left.\frac{dZ}{dt}\right|_{t=1} = T_B$

From the endpoint conditions:

$$\begin{aligned}
 \mathbf{A} &= \mathbf{a}_0 \\
 \mathbf{B} &= \mathbf{a}_0 + \mathbf{a}_1 + \mathbf{a}_2 + \mathbf{a}_3 \\
 \mathbf{T}_A &= \mathbf{a}_1 \\
 \mathbf{T}_B &= \mathbf{a}_1 + 2\mathbf{a}_2 + 3\mathbf{a}_3
 \end{aligned}
 \tag{2.5}$$

Rearranging these equations in terms of coefficients, equation 2.5 becomes

$$\mathbf{Z}(t) = \mathbf{A}(1 - 3t^2 + 2t^3) + \mathbf{B}(3t^2 - 2t^3) + \mathbf{T}_A(t - 2t^2 + 3t^3) + \mathbf{T}_B(-t^2 + t^3)
 \tag{2.6}$$

This can be presented in a matrix form:

$$\mathbf{Z}(t) = \begin{bmatrix} 1 & t & t^2 & t^3 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -3 & 3 & -2 & -1 \\ 2 & -2 & 1 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \\ \mathbf{T}_A \\ \mathbf{T}_B \end{bmatrix}
 \tag{2.7}$$

An arbitrary airfoil can be described with two Ferguson curves, one for the upper and one for the lower curve, as shown in Figure 2-2. Six parameters are required to construct a complete airfoil. The curve starts from leading edge (A) to trailing edge (B) with their corresponding tangent vectors (\mathbf{T}_A and \mathbf{T}_B) with camber angle α_c and boat tail angle α_b defining the orientation of the tangent vectors, $\mathbf{T}_B^{\text{lower}}$ and $\mathbf{T}_B^{\text{upper}}$, respectively.

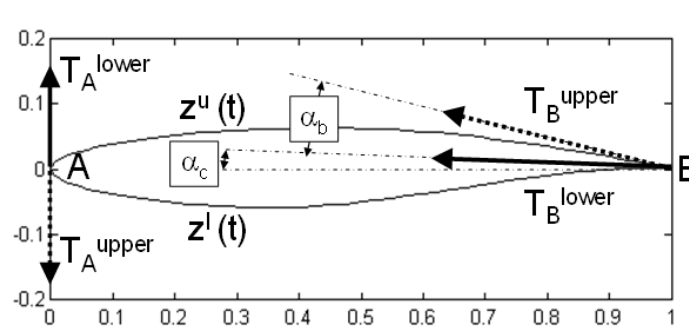


Figure 2-2 Airfoil Described by 2 Ferguson Curves (Sobester and Barrett, 2008)

2.1.3 Splines

Splines roots in ship-hull design where the drafting was done by passing thin strips of wood through points laid out on the floor. The concept has been adopted to form a mathematical representation of curve passing through a set of defined points. Formulations of Ferguson's curve and B-Splines representation are summarised as follows:

A Bezier curve is based on the similar logic as Ferguson's curve but defined with a slightly different algorithm(Sobester and Barrett, 2008). A Bezier curve of order n is described as:

$$P(t) = \sum_{i=0}^n B_i^n(t) \cdot P_i \quad [2.8]$$

$$\text{where } B_i^n(t) = \binom{n}{i} \cdot (1-t)^{n-i} \cdot t^i, i = 0, 1, \dots, n [2.9]$$

and P_i represents the set of $n+1$ control points, t is parametric point varying from 0 to 1.

A Bezier curve is accurate when representing simple curves. As the curve complexity increases, higher degree of polynomials must be increased, which results in a larger error. In order to represent the complex curve, it is more efficient to separate the curve into segments and use a set of low-order Bezier curves instead.

The B-spline is the generalisation of Bezier curve, which is defined as

$$C(t) = \sum_{i=0}^m P_i \cdot N_{i,p}(t) \quad [2.10]$$

where p is order, P_i are control points, $N_{i,p}$ is B-spline basis function.

The example of airfoil generated by B-Spline curve is shown in Figure 2-3.

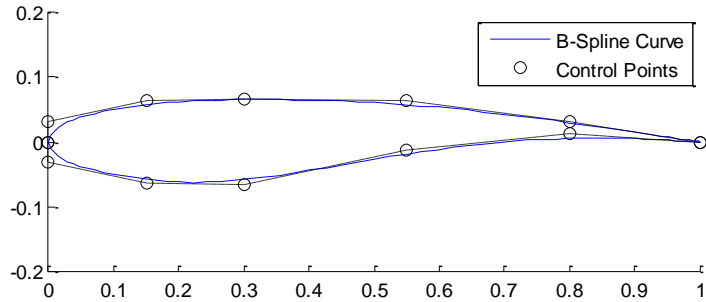


Figure 2-3 B-Spline curve representing airfoil with control points

One advantage of the B-spline method is that the degree of the polynomial is not limited. The designer can increase the polynomial degree without changing the number of control points. It also allows local control by limiting the number of control points as design variables.

2.1.4 Hicks-Henne Shape Functions

Hicks and Henne (1978) introduced a compact formulation for parameterisation of airfoil sections. The perturbation is a linear superposition of “bumps” or analytical shape function on the baseline airfoil geometry. The shape functions are defined as

$$b_i(x) = \sin^t(\pi x^{m_i}) \quad [2.11]$$

$$\text{where } m_i = \frac{\ln(0.5)}{\ln(x_{Mi})} \quad [2.12]$$

x_{Mi} is the position of the maximum point of the bump ranges from 0 to 1, t controls the width of the bump. Figure 2-4 shows sets of 10 Hicks-Henne bump functions with parameter $t=4$. The contribution of each parameter is determined by the value of the participating coefficients, α_i associated with shape function.

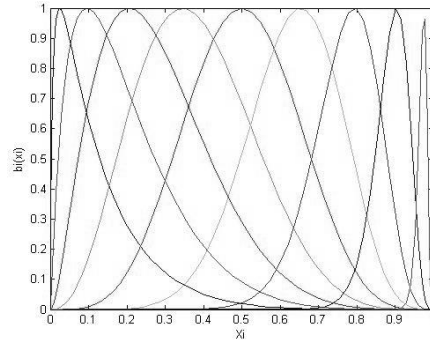


Figure 2-4 A set of 10 Hicks-Henne shape function $t=4$.

The airfoil is finally described by:

$$z = z_{basis} + \sum_{i=1}^N \alpha_i b_i(x) \quad [2.13]$$

All participating coefficients are initially set to zero, so the first computation gives the baseline geometry.

Hicks-Henne bump functions are used in airfoil design and optimisation process. Wide range of airfoil shapes can be generated and the computed gradients are always smooth. However, since a baseline airfoil shape is required, its selection is crucial to guarantee whether an optimal and realistic shape can be obtained.

2.1.5 Parameter Section (PARSEC)

The Parameter Section (PARSEC) method (Sobieczky, 1998) has been developed on the basis of airfoil geometry with the aim to keep the number of required design parameters as low as possible. The author focuses on strong control over the curvature of an airfoil by using design parameters such as leading edge radius, upper and lower crest curvature. Similar to 4-digit NACA series, the author chooses polynomial as a function to generate each upper and lower surface of the airfoil, but with a higher order. This results in eleven design parameters used to describe an airfoil, as follows:

- r_{LE} : leading edge radius,
- X_{up}, X_{lo} : upper and lower crest location,
- Z_{up}, Z_{lo} : upper and lower crest value,
- $Z_{xx,up}, Z_{xx,lo}$: upper and lower curvature at crest location,
- ΔZ_{tc} : trailing edge thickness,

Z_{te} : trailing edge vertical coordinate,
 β_{te} : trailing edge wedge angle, and
 α_{te} : trailing edge direction

These design parameters are illustrated in Figure 2-5.

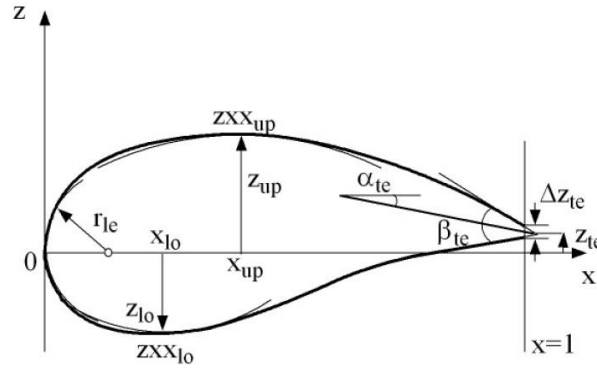


Figure 2-5 Design Parameters for PARSEC method (Sobieczky, 1998)

The airfoil shape for each upper and lower curve is described by the linear combination of polynomial shape functions:

$$Z_i = \sum_{i=1}^6 a_{n,i} X_i^{\frac{n-1}{2}} \quad [2.14]$$

The coefficients a_n are determined by solving system of equations [2.15] with the defined geometric parameters

1. z coordinate at leading edge is described in terms of leading edge radius
2. z coordinate at trailing edge
3. z coordinate at upper crest location
4. slope of trailing edge angle
5. zero slope at upper crest location
6. curvature at upper crest location

$$\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 \\
x_{te}^{\frac{1}{2}} & x_{te}^{\frac{3}{2}} & x_{te}^{\frac{5}{2}} & x_{te}^{\frac{7}{2}} & x_{te}^{\frac{9}{2}} & x_{te}^{\frac{11}{2}} \\
x_{up}^{\frac{1}{2}} & x_{up}^{\frac{3}{2}} & x_{up}^{\frac{5}{2}} & x_{up}^{\frac{7}{2}} & x_{up}^{\frac{9}{2}} & x_{up}^{\frac{11}{2}} \\
\frac{1}{2}x_{te}^{-\frac{1}{2}} & \frac{3}{2}x_{te}^{\frac{1}{2}} & \frac{5}{2}x_{te}^{\frac{3}{2}} & \frac{7}{2}x_{te}^{\frac{5}{2}} & \frac{9}{2}x_{te}^{\frac{7}{2}} & \frac{11}{2}x_{te}^{\frac{9}{2}} \\
\frac{1}{2}x_{up}^{-\frac{1}{2}} & \frac{3}{2}x_{up}^{\frac{1}{2}} & \frac{5}{2}x_{up}^{\frac{3}{2}} & \frac{7}{2}x_{up}^{\frac{5}{2}} & \frac{9}{2}x_{up}^{\frac{7}{2}} & \frac{11}{2}x_{up}^{\frac{9}{2}} \\
-\frac{1}{4}x_{up}^{-\frac{3}{2}} & \frac{3}{4}x_{up}^{-\frac{1}{2}} & \frac{15}{4}x_{up}^{\frac{1}{2}} & \frac{35}{4}x_{up}^{\frac{3}{2}} & \frac{53}{4}x_{up}^{\frac{5}{2}} & \frac{99}{4}x_{up}^{\frac{7}{2}}
\end{bmatrix}
\begin{bmatrix}
a_1 \\
a_2 \\
a_3 \\
a_4 \\
a_5 \\
a_6
\end{bmatrix}
=
\begin{bmatrix}
\sqrt{\frac{2}{r_{le}}} \\
z_{te} \\
z_{up} \\
\tan(\theta_{te,up}) \\
0 \\
\left[\frac{d^2z}{dx^2}\right]_{x=x_{up}}
\end{bmatrix}
\tag{2.15}$$

Six coefficients are required for each upper and lower curve. This results in the total of 12 coefficients. However, since the first coefficients for each upper and lower curve, $a_{1,up}$ and $a_{1,lo}$, are set to be equal, in order to force continuity at the leading edge, the total number of coefficients is reduced to 11.

The minimum and maximum range of PARSEC design parameters are studied (Padulo *et al.*, 2009) as presented in Table 2.1.

Table 2.1 Minimum and maximum values of PARSEC design variables

Variables	r_{LE}	x_{up}	x_{lo}	z_{up}	z_{lo}	$z_{xx,up}$	$z_{xx,lo}$	z_{te}	Δz_{te}	β_{te}	α_{te}
min	0.003	0.25	0.191	0.040	-0.061	-0.726	0.14	0.0037	0.0001	-0.28	0.032
max	0.018	0.46	0.521	0.067	-0.020	-0.197	1.00	0.0052	0.0008	-0.05	0.131

The advantages of the PARSEC method are:

1. No baseline airfoil is needed to generate airfoil profile
2. Wide range of airfoil shapes can be generated
3. Intuitive design variables
4. Airfoil thickness can be expressed by simple bound or linear constraints

However, since the number of design parameters is fixed, the PARSEC airfoil may reach only a certain geometrical accuracy. Padulo *et al.* (2009) also found that the certain sets of PARSEC design parameters leads to erroneous shapes such as additional bumps or overlapping. These errors will be demonstrated in Chapter 3.

2.1.6 Class-Shape function Transformation (CST)

Class-Shape function Transformation (CST) (Kulfan, 2006) is one of the recent geometry parameterisation methods developed with application for aircraft component shapes. The author introduces “shape function” which is a simple analytical function which provides direct control key geometry parameters e.g. leading edge radius, trailing edge boattail angle. A “class function” is also combined to generate wide ranges of geometries. A two-dimensional curve is represented as the product of the class function $C(x/c)$, and a shape function $S(x/c)$:

$$\zeta(\psi) = C_{N_2}^{N_1}(\psi) \cdot S(\psi) \quad [2.16]$$

where $\psi = x/c$ and $\zeta = z/c$

The Class function is given in a generic form by:

$$C_{N_2}^{N_1}(\psi) = (\psi)^{N_1} [1 - \psi]^{N_2}, 0 < \psi < 1 \quad [2.17]$$

The exponents N_1 and N_2 range from 0 to 1 which yields arbitrary shape.

In order to generate a general symmetric airfoil, the exponents N_1 and N_2 are equal to 0.5 and 1.0 respectively. The first term, $\sqrt{\psi}$, produces round leading edge, while the second term, $[1 - \psi]$, makes sharp trailing edge. This selection of exponents N_1 and N_2 makes the basis shape of the “airfoil” class. Other airfoils can be derived from this class function.

Equation of airfoil upper and lower profile respectively then becomes:

$$\begin{aligned} \zeta_U(\psi) &= C_{1.0}^{0.5}(\psi) \cdot S_U(\psi) + \psi \cdot \Delta\zeta_U \\ \zeta_L(\psi) &= C_{1.0}^{0.5}(\psi) \cdot S_L(\psi) + \psi \cdot \Delta\zeta_L \end{aligned} \quad [2.18]$$

The additional last terms define the upper and lower edge thicknesses, respectively.

To generate arbitrary airfoil shape, Bernstein polynomial was chosen as the shape function which is added to the class function. This polynomial has a mathematical property of “Partition of Unity”, stating that for any order of the polynomials, the

summation of the polynomial values at any point always equals to 1. The Bernstein polynomial always keeps the shape of the airfoil when superimposed on the class function.

The definition of a Bernstein polynomial of order n is:

$$S(\psi, r) = K_r^n \cdot \psi^r (1 - \psi)^{n-r} \quad [2.19]$$

where n is the order of Bernstein polynomials, K is Binomial coefficient, which directly related to the order of the Bernstein polynomials used and it is defined as:

$$K_r^n \equiv \binom{n}{r} \equiv \frac{n!}{r!(n-r)!} \quad [2.20]$$

The complete equations to represent the upper and lower surface of CST airfoils become

$$\zeta_U(\psi) = \sqrt{\psi} \cdot (1 - \psi) \cdot \sum_{i=0}^N \left[A_{U,i} \cdot \frac{N!}{i!(N-i)!} \psi^i (1 - \psi)^{N-i} \right] + \psi \cdot \Delta\zeta_U \quad [2.21a]$$

$$\zeta_L(\psi) = \sqrt{\psi} \cdot (1 - \psi) \cdot \sum_{i=0}^N \left[A_{L,i} \cdot \frac{N!}{i!(N-i)!} \psi^i (1 - \psi)^{N-i} \right] + \psi \cdot \Delta\zeta_L \quad [2.22b]$$

From equations 2.21, the coefficients A_i of the shape function will be used as the design variables. The number of these coefficients corresponds to the order of the Bernstein polynomial. For instance, for polynomials of order 3, which has four terms, the number of coefficients is 4.

Equation 2.21(a) can be rearranged to present the shape function of the upper profile, S_u as follows:

$$S_U(\psi) = \sum_{i=0}^N \left[A_{u,i} \cdot \frac{N!}{i!(N-i)!} \psi^i (1 - \psi)^{N-i} \right] = \frac{\zeta_U - \psi \cdot \Delta\zeta_U}{\sqrt{\psi} \cdot (1 - \psi)} \quad [2.23a]$$

and similarly for the lower profile, S_l

The polynomial coefficients A_i of the shape functions can be obtained by performing least square fit on the shape functions generated from the original profile.

Figure 2-6 provides an example of RAE2822 airfoil with its transformed shape functions reconstructed from CST formulation.

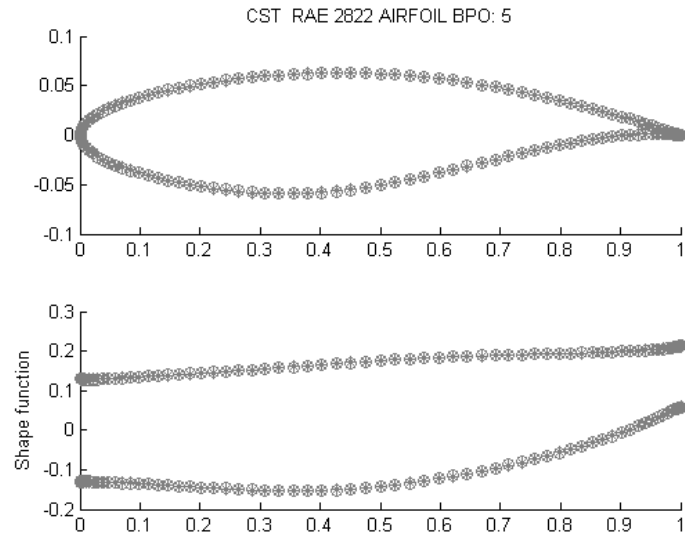


Figure 2-6 RAE 2822 profile (top) with corresponding shape functions (bottom)

Examples of airfoils representing four different shapes are presented in Figure 2-7: top left symmetrical NACA 0012, top right: RAE2822 camber airfoil, bottom left: NACA 63(2) -015 and the bottom right, supercritical NASA SC(2)-0714. The set of CST coefficients representing these airfoils with Bernstein Polynomial order 5 are presented in Table 2.1.

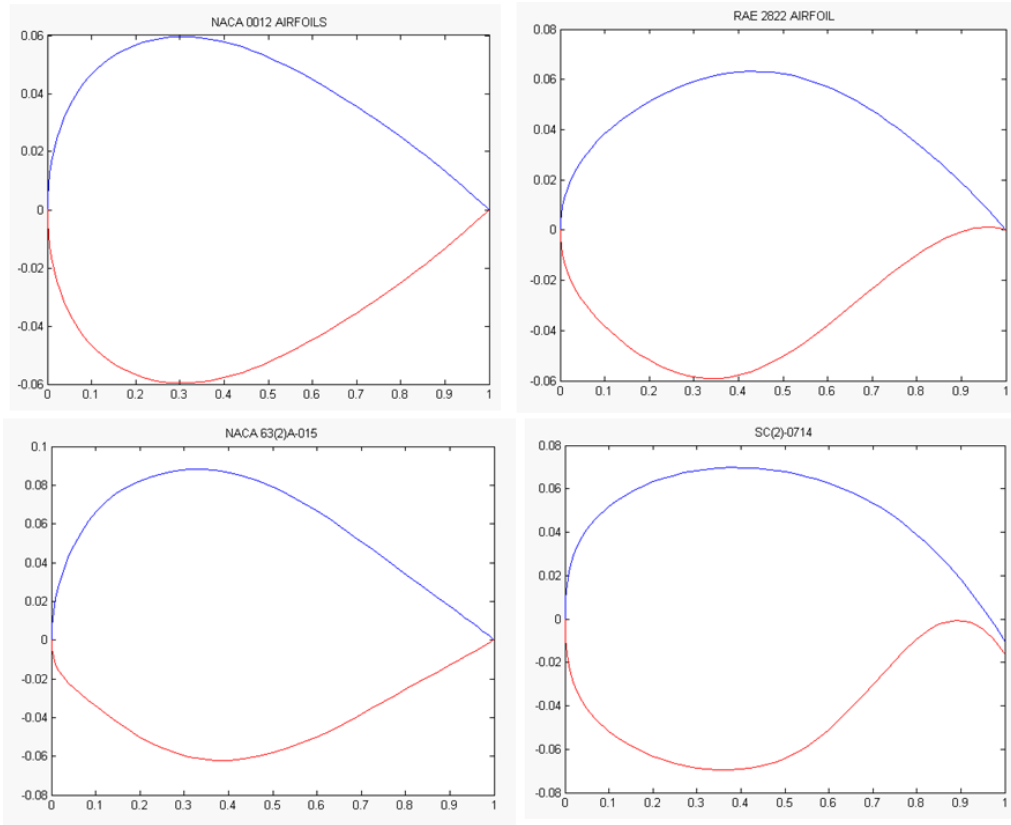


Figure 2-7 Airfoil Profiles [From top left, NACA 0012, top right: RAE2822, bottom left: NACA 63(2) -015 and the bottom right, NASA SC(2)-0714]

Table 2.1 CST Airfoil Polynomial Coefficients

Airfoils	Au1	Au2	Au3	Au4	Au5	Au6
NACA 0012	0.175228	0.142394	0.173494	0.127051	0.143224	0.141154
RAE 2822	0.130016	0.134272	0.164348	0.209591	0.175838	0.21112
NACA 63(2)A-105	0.225653	0.241502	0.214564	0.25777	0.157689	0.186596
NASA SC(2)-714	0.230561	0.086264	0.285186	0.145279	0.277266	0.319361

Airfoils	AI1	AI2	AI3	AI4	AI5	AI6
NACA 0012	-0.17526	-0.14201	-0.17466	-0.12541	-0.14440	-0.14076
RAE 2822	-0.13234	-0.11908	-0.22646	-0.11391	-0.09583	0.05915
NACA 63(2)A-105	-0.13410	-0.05420	-0.30965	-0.09450	-0.15613	-0.13046
NASA SC(2)-714	-0.23146	-0.07965	-0.18201	-0.30954	0.01190	0.27918

By using the polynomials for parameterisation such as PARSEC and CST method, the polynomials may end up with coefficients with opposing signs which results in oscillating behaviour or wavy curves due to round-off errors (Keane and Nair, 2005).

2.2 Surface Parameterisation Methods

Presented in this section are two famous surface parameterisation methods used in conceptual and preliminary design: the Partial Differential Equation (PDE) and Free-form deformation surface (FFD).

2.2.1 Partial Differential Equations

Partial Differential Equations (PDE) were first introduced as a surface generation method by Bloor and Wilson (1989a and 1989b). By solving elliptic partial differential equation with boundary value, the solution of the PDE describes the blending surface between those boundary conditions.

From elliptic partial differential equation:

$$\left(\frac{\partial^2}{\partial u^2} + a^2 \frac{\partial^2}{\partial v^2}\right) \underline{X}(u, v) = 0 \quad [2.24]$$

where $\underline{X}(u, v) = (x(u, v), y(u, v), z(u, v))$ is parametric surface of u and v

and $0 \leq u \leq 1, 0 \leq v \leq 1$

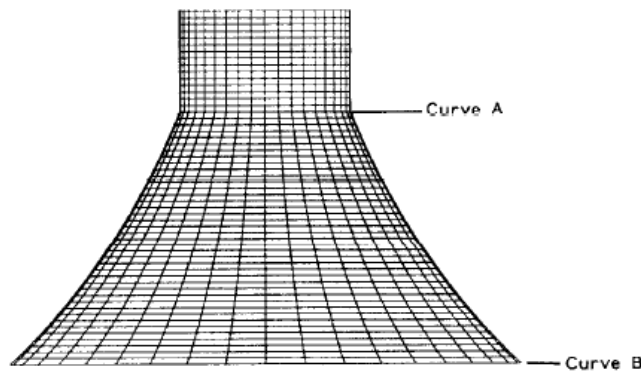


Figure 2-8 Blend circular cylinder to a plane (Bloor and Wilson, 1989a)

Figure 2-8 illustrates the PDE surface which blends a circular cylinder to an orthogonal plane. The PDE is subjected to the boundary conditions:

$$\text{(Curve A)} \quad x(0, v) = \cos v, \quad y(0, v) = \sin v, \quad z(0, v) = H \quad [2.25]$$

$$\text{(Curve B)} \quad x(1, v) = R \cos v, \quad y(1, v) = R \sin v, \quad z(1, v) = 0 \quad [2.26]$$

By solving Eq.2.24 under boundary conditions 2.25 and 2.26 by the method of separation of variables, an analytical solution is obtained:

$$x = \left(\cosh au + \frac{(R - \cosh a)}{\sinh a} \sinh au \right) \cos v$$

$$y = \left(\cosh au + \frac{(R - \cosh a)}{\sinh a} \sinh au \right) \sin v$$

$$z = H \cdot (1 - u) \quad [2.27]$$

To generate the surface patch of PDE, uses equation 2.24 with

$$\text{boundary conditions:} \quad \underline{X}(0, v) = P_1(v), \quad \underline{X}(1, v) = P_2(v) \quad [2.28]$$

$$\text{and their corresponding derivatives: } \underline{X}_U(0, v) = D_1(v), \quad \underline{X}_U(1, v) = D_2(v) \quad [2.29]$$

The surface patch is then described as;

$$X(u, v) = A_0(u) + \sum_{n=1}^{\infty} [A_n(u) \cos(nv) + B_n(u) \sin(nv)] \quad [2.30]$$

$$\text{with coefficients} \quad A_0 = a_{00} + a_{01}u + a_{01}u^2 + a_{03}u^3$$

$$A_n = a_{n1}e^{anu} + a_{n2}ue^{anu} + a_{n3}e^{-anu} + a_{n4}e^{-anu}$$

$$B_n = b_{n1}e^{bnu} + b_{n2}ue^{bnu} + b_{n3}e^{-bnu} + b_{n4}e^{-bnu} \quad [2.31]$$

The coefficients in equation 2.29 are determined by imposing the boundary conditions. Ugail (2003) demonstrated that it was possible to represent aircraft geometry with small set of design variables. The aircraft-like surface can be generated using only 5 PDE surface patches, as shown in Figure 2-9.

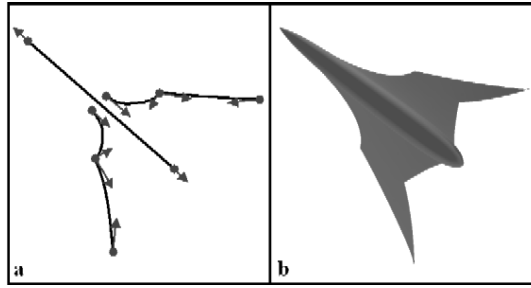


Figure 2-9 Aircraft Representing by 5 PDE surface patch (Ugail, 2003)

Pasadas and Rodriguez (2008) generated blending surface connecting fuselage to a wing geometrical model. The wing is described by polynomials of order 8, while the fuselage is described in terms of trigonometric function as follows:.

$$f(x, y) = (y, -4x^2 + 159.8x^3 - 747.7x^4 + 1620.5x^5 - 1920x^6 + 1195.3x^7 - 303.7x^8, 0.4(-0.7x - 41.5x^2 + 429.3x^3 - 1706.4x^4 + 3464.6x^5 - 3800x^6 + 2140x^7 - 485.5x^8)), (x, y) \in \Omega \quad [2.32]$$

$$g(x, y) = (0.5 \cos(2\pi x), y, 0.5 \sin(2\pi x)), (x, y) \in \Omega \quad [2.33]$$

A numerical solver is required to solve the partial differential equation (Equation 2.24) of this problem.

Figure 2-10 shows the blending surfaces generated with the control coefficients ‘a’ equal to 1 (middle) and 100 (right). This coefficient described the influence of the tangent at boundary conditions to the surface.

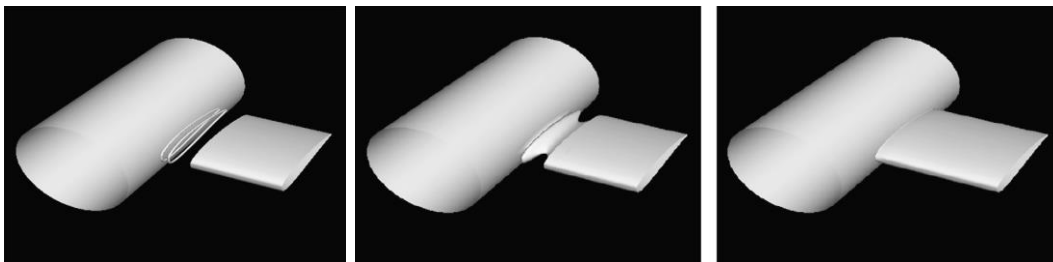


Figure 2-10 PDE surfaces joining fuselage and wing section, with a =1 (middle) and a=100 (right) (Pasadas and Rodriguez, 2008)

PDE surface has very high level of smoothness and complete accuracy at their intersections. It is obvious that, in order to find a solution of PDE, every curve or section has to be described by proper mathematical function, which is not always

possible. Samareh (1999) pointed out that using PDE approach to parameterise an existing complex model can be a time-consuming process. Keane and Nair (2006) also stated that:

The main deficiency of the PDE is the lack of local control. To gain local control of PDE generated surfaces for instance, modern airfoil sections which include various bulges and curves to control highly non-linear flow such as boundary layer separation, it is usually necessary to convert them to piecewise collection of NURBS representations and then manipulate these local functions.

PDE is suitable for surface presentation which is useful for Computational Fluid Dynamics rather than modelling the internal parts such as spar or stringers for Structural analysis.

2.2.2 NURBS

Non-Uniform Rational B-Splines (NURBS) has become standards in CAD industry. Similar to splines representation in 2D, the NURBS surface is defined by a set of control points ($\bar{\mathbf{P}}_{i,j}$) which can be manipulated to change the shape of the surface.

A NURBS surface of degree p in the u -direction and degree q in the v -direction is a piecewise rational function in a form:

$$P(u, v) = \frac{\sum_{i=0}^I \sum_{j=0}^J N_{i,p}(u) N_{j,q}(v) w_{i,j} \bar{\mathbf{P}}_{i,j}}{\sum_{i=0}^I \sum_{j=0}^J N_{i,p}(u) N_{j,q}(v) w_{i,j}} \quad [2.34]$$

where $\bar{\mathbf{P}}_{i,j}$ are the points on control net, $w_{i,j}$ are the weight, $N_{i,p}(u)$ and $N_{j,q}(v)$ are B-Spline basis functions defined on the knot vector u and v .

Mastin *et al.* uses NURBS to construct approximation model of blended wing body for CFD analysis. The surface is controlled by sets of control point or a control polygons shown in Figure 2-11. The model can be used by CAD model for further refinement or modification of the original geometry.

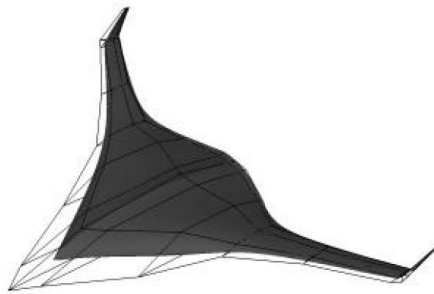


Figure 2-11 NURBS approximation of BWB (Mastin *et al.*, 1996)

2.2.3 Free Form Deformations

The Free Form Deformation (FFD) method uses a three-dimensional mesh in which each node is considered to be a control point of NURBS volume. Therefore the positions of the control points are then used as the design variables. This number of design variables can be large in three dimensional representations since the number of nodes increases as the cube of number of points placed along the edges of the mesh.

Figure 2-12 shows FFD surface blending between wing and fuselage of a general aircraft.

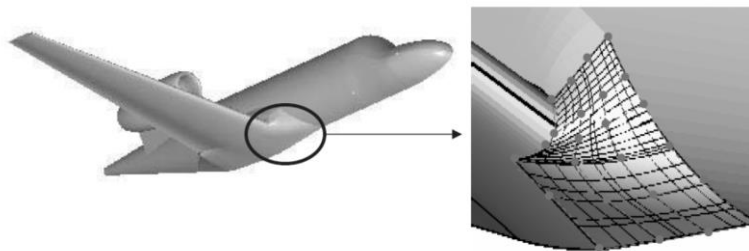


Figure 2-12 Free-form deformation grids on the wing-body intersection (Samareh, 2004)

Anderson *et al.* (2009) commented that this is a convenient method for multidisciplinary design and optimisation where several unrelated meshes need to be modified simultaneously.

2.3 Parameterisation Methods Comparison

There are a number of research studies on parameterisation comparison. Samareh (1999) has proposed the metrics for comparison for parameterisation methods used in high fidelity analysis which are basis vector, domain element, partial differential equation (PDE), discrete, polynomial and spline, CAD, analytical approach, free-form deformation (FFD), and multidisciplinary aero/structural shape optimisation using deformation (MASSOUD) approach. The rating was given in three levels: good, fair, and poor based on the author’s judgement, as shown in Figure 2-13. For the first assessment criterion, “airplane shape design variables”, the analytical and MASSOUD methods are rated as “good”, PDE and CAD received “fair”, while other methods received “poor”. Next assessment on the “compact set of design variables”, discrete method received “poor” rating, due to each point is design variable itself, the domain element, polynomial and spline, and FFD received “fair” rating, while the remaining methods received “good” rating. The assessment criteria continue in smooth geometry, local control, etc. For CAD, one obvious deficiency comparing to other methods is that the sensitivity can not be calculated analytically.

Approaches \ Criteria	Basis vector	Domain element	PDE	Discrete	Polynomial and spline	CAD	Analytical	FFD	MASSOUD
Consistent parameterization									
Airplane shape design variables									
Compact set of design variables									
Smooth geometry									
Local control									
Analytical sensitivity									
Grid deformation									
Setup time									
Existing grids									
CAD connection									

Figure 2-13 Comparison of parameterisation approaches [Samareh, 1999]

This comparison metrics is extended to compare with an assessment numerically for each assessment criteria. Literatures on comparison studies are compiled in Table 2.2.

Table 2.2 Airfoil Parameterisation Comparison Studies

References:	Mesh Point	Ferguson Curve	Hicks-Henne	B-splines/ NURBS	PARSEC	CST
Wu <i>et al.</i> (2003)	*		*		*	
Kumano <i>et al.</i> (2006)				*	*	
Sobester and Keane (2007)		*		*		
Castonguay and Nadarajah (2007)	*		*	*	*	
Azamatov <i>et al.</i> (2008)				*		*

Wu *et al.* (2003) performed a comparison on three parametric methods: mesh-point, Hicks-Henne shape function, and PARSEC, for the design and optimisation of turbo machinery cascades by an adjoint equation method. The results show that the PARSEC method is not suitable for representing a blade shape. Comparing between Hicks-Henne shape functions and mesh-point method, the authors found that the Hicks-Henne shape functions converge to the optimum faster, but the mesh-point method can reach higher accuracy.

Sobester and Keane (2007) used Ferguson Curve for rapid airfoil generation. This yields less accurate approximation than B-Spline representations due to its limited number of design variables in the formulation yet provides wide range of design alternatives. Kumano *et al.*(2006) also compared B-Spline with PARSEC and found that B-spline with 13 control points (26 design variables) yields better approximation than PARSEC.

Castonguay and Nadarajah (2007) studied the effect of shape parameterisation on aerodynamic shape optimisation based on accuracy and evaluation costs. Four methods: mesh points, Hicks-Henne bump functions, B-Spline curves and PARSEC method, have been studied. The authors found that the Hicks-Henne bump functions are able to provide the design space where the target pressure is obtainable, however can obtain lower accuracy than mesh point and B-Spline curves. Both Hicks-Henne bump

functions and B-Spline curve can be used efficiently in drag minimisation studies however the Hicks-Henne bump functions cannot provide necessary shape modifications for the inverse design problem. Moreover, the PARSEC method cannot capture the leading edge shape of the ONERA M6 to reproduce the target pressure.

Azamatov *et al* (2008) proposed a geometry representation algorithm which is a combination of the Class/Shape function Transformation (CST) and B-spline. The results show that CST achieves good accuracy in most analytical representations. The CST method with 4 control variables can fit the existing airfoil better than NURBS with 10 control variables. However, it has difficulties to use in complex aircraft geometry such as fuselage fairing. The proposed Hybrid method introduces breakpoint to divide complex curves into smaller sections. This yields better accuracy, with more control parameters for each subdivision required as a trade-off.

Most of these researches performed only pair-wise comparisons. This leads to a systematic review of the parameterisation methods numerically based on the selected criteria, which will be presented in Chapter 3.

2.4 Geometry Generation Tools

This section presents geometry generation tools which have been developed for multidisciplinary design application and used at conceptual design stage.

2.4.1 Characteristic Curves

Trapp and Sobieczky (1999) developed a geometrical representation method for aircraft design in an object-oriented manner. Fifteen basic curves were provided in the library. Each of these curves is described by algebraic or other explicit functions e.g. polynomials, trigonometric, or conic functions. They are defined within a normalised unit square with maximum four parameters. The minimum two parameters are the slopes at the start and the end of the curve, a and b , respectively. Examples of these basic curves are demonstrated in Figure 2-14.

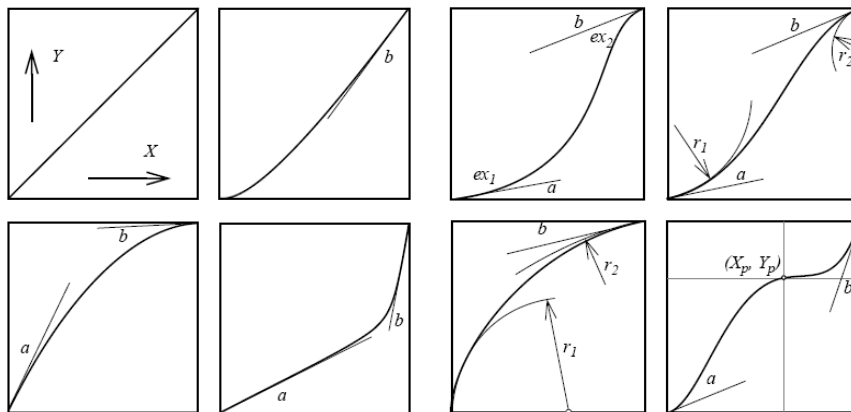


Figure 2-14 Characteristic functions (Trapp and Sobieczky, 1999)

These basic functions are then mapped on the intervals of characteristic curve. The unit square can be stretched to fit the intervals as shown in Figure 2-15. In order to ensure smoothness of curve, the slopes and curvatures at the connecting borders have to correspond.

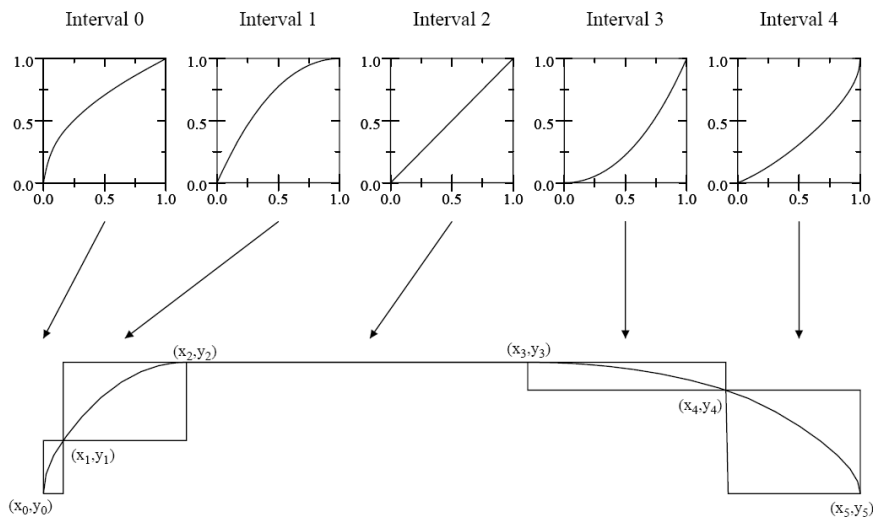


Figure 2-15 Example of curve represented by base functions (Trapp and Sobieczky, 1999)

2.4.2 ICAD Multi-Model Generator

La Rocca *et al.* (2002) developed the ICAD Multi-Model Generator for the Blended Wing Body (BWB) aircraft. It holds the information about the BWB aircraft and generates the consistent models at the different levels of fidelity. A full parametric definition of the aircraft has been implemented in the KTI ICAS environment. The tool generates the different models for aerodynamics, structure, mass distribution. These models are feed to different analysis boxes, mainly finite element analysis and computational fluid dynamic.

For surface modelling, the whole BWB is modelled in object oriented structure. It consists of parts such as a fuselage, wings, winglet, etc. The fundamental element is called “wing trunk” which is defined by geometrical parameters such as span, sweep, twist, and dihedral angle, with reference to leading edge line or quarter chord line. The wing trunks are then assembled together via connection element as shown in Figure 2-16 to form a blended body aircraft.

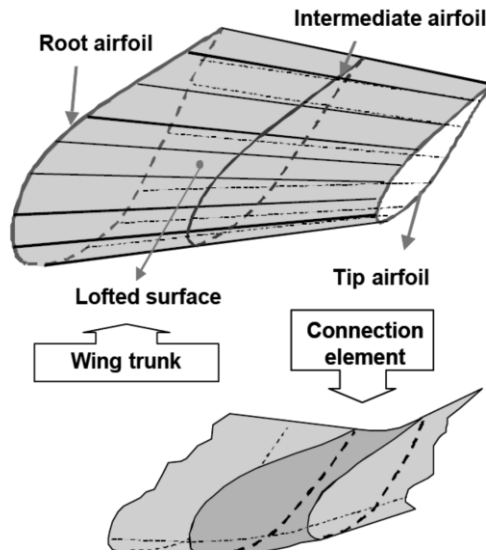


Figure 2-16 Wing trunk and connection element (La Rocca *et al.*, 2002)

This tool has been developed for blended wing body aircraft only therefore there is just one component type: wing trunk. More diverse shapes for the conventional aircraft are used in two following examples.

2.4.3 Rapid Aerospace Geometry Engine

There are other tools under development for CAD-free support such as Rapid Aerospace Geometry Engine: RAGE (Rodriguez and Sturdza, 2006). This tool contains collection of parametrically defined geometry components. These components are created by mathematically stacking the lofting crosssections such as airfoils or fuselage cross-sections. All these subcomponents are mathematically defined by user-provided parameters and then lofted together to form a component as shown in Figure 2-17.

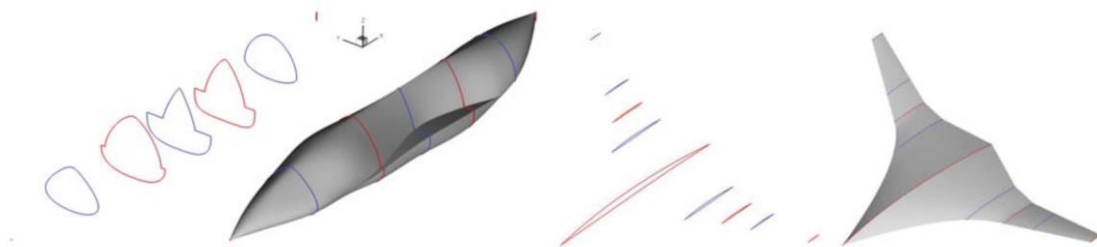


Figure 2-17 Example of RAGE fuselage and wing built by lofting functions (Rodriquez and Sturdza, 2006)

2.4.4 Vehicle Sketch Pad

The geometry tool Vehicle Sketch Pad, VSP, was developed by NASA (Hahn, 2010) and is openware freely available worldwide. In the US, it has been used in over 100 universities, aerospace companies, and other government agencies (Fredericks *et al.*, 2010). It contains library of 11 aircraft parts e.g. wing, fuselage, jet engine, etc. Each component is defined with geometric parameters; for instance the wing component is modelled with span, area, sweep, taper ratio, thickness to chord ratio as design parameters. The library of NACA 4-digit and 6-series, biconvex and wedge airfoil are included, with option for user input airfoil as well. This structure is used for all 11 parts, and the wide range of components can be modelled using these parts, for example, a landing gear can be modelled using fuselage library.

The early version of VSP has been developed to work with Flight Optimisation System: FLOPS (McCullers, 2011) which is an aircraft designtool used in conceptual design stage that requires only text inputs. This text listing has been replaced with an interactive graphical user interface which leads to the development of Rapid Aircraft Modeller (RAM), which is more user friendly environment.

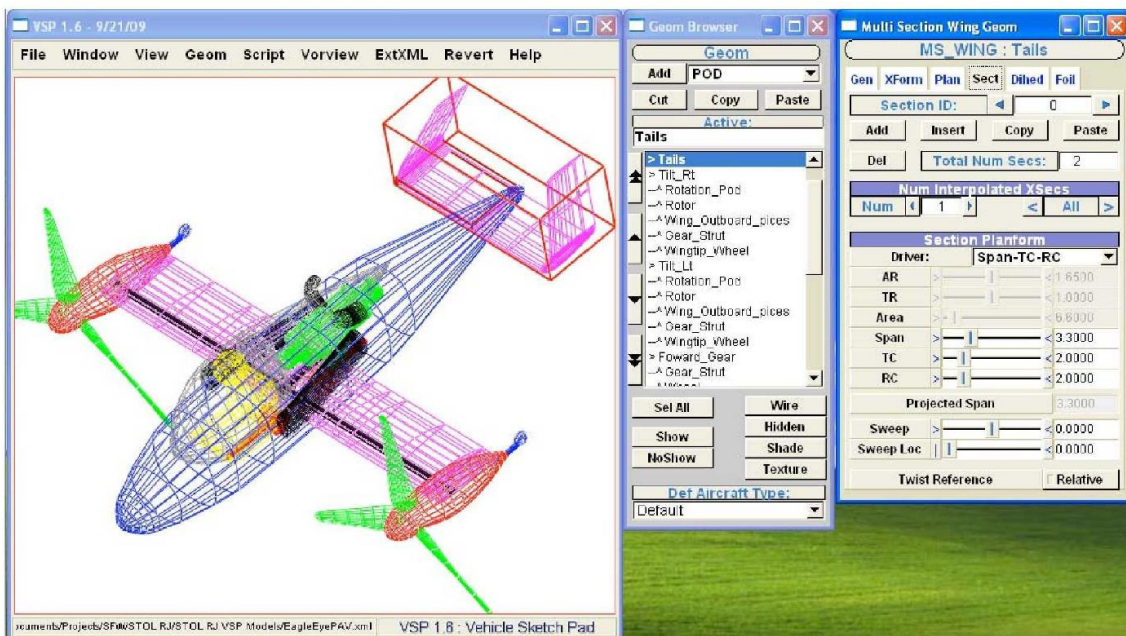


Figure 2-18 Vehicle Sketch Pad parameterised graphical user interface (Hahn, 2010)

One of the main benefits of VSP comparing to CAD software is the amount of time used in generating aircraft model. This is primary due to VSP has library of aircraft components and user is able to select certain parameters to generate or modify the shapes while traditional CAD requires draw-extrude-edit process of parts. The author demonstrated that the benefit of using parametrically defined parts are: it is more intuitive and use less time for designers to model an aircraft comparing to traditional CAD software.

The program has been linked with a vortex lattice solver: VorLax (Miranda *et al.*, 1977) via VorView tool which creates a meta-model compatible with this solver. Hahn (2010) conclude that *“it is possible to manually define certain characteristics on an initial model, and then have VorView automatically create similar meta-models after that. While not perfect, certain tasks such as sweep optimization may be performed effectively”*. The example of high lift aerodynamic assessment of a STOL regional jet design is shown in Figure 2-19. Eventhough the Vehicle Sketch Pad is freeware, its associated aerodynamic solver i.e. VorLax is limited to users due to licence agreement of NASA.

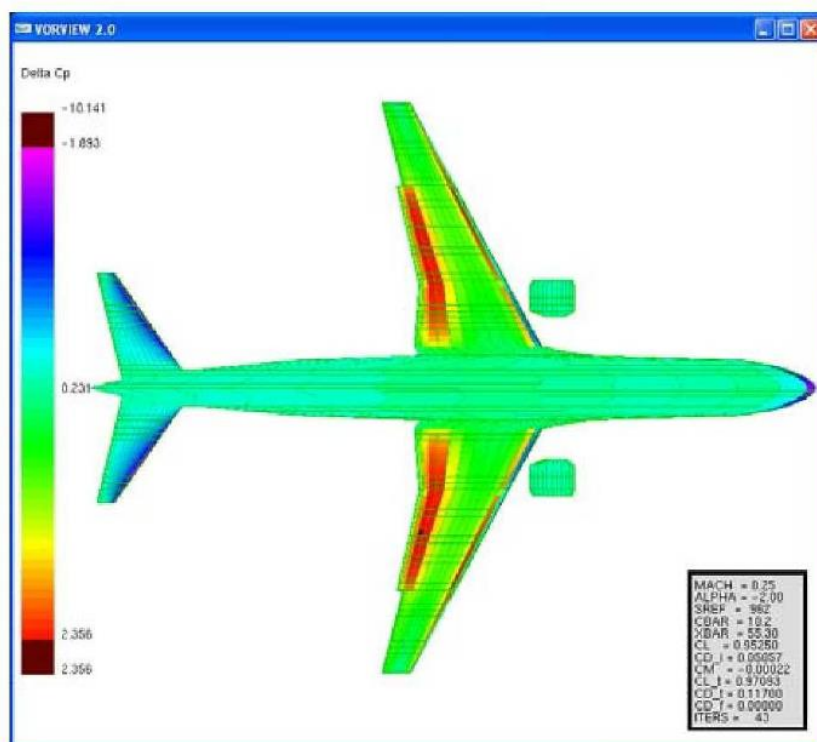


Figure 2-19 VorLax/Vorview analysis (Hahn, 2010)

2.5 Summary and Conclusions

This chapter presents the survey on shape parameterisation methods with an initial focus on airfoil profiles. The previous studies on parameterisation method comparison have also been discussed but are inconclusive. This necessitates amore complete comparison of the parameterisation methods in order to choose the proper method to be extended to aircraft surface generation. Five widely-cited methods are chosen: Ferguson's curve, Hicks-Henne bump functions, B-Splines representation, PARSEC, and CST method. These methods are based on different formulations and design parameters, and will be compared against the proposed metrics containing the relevant assessment criteria in Chapter 3. The surface generation method will then be discussed in Chapter 4.

3 Curve Parameterisation

Aircraft parameterisation process begins with the parameterisation of 2D curves, which is the most detailed level of descriptions required for the aerodynamics analysis. This is then followed by the development of methods for 3D surface generation and subsequently methods for component modelling and assembly. Validation forms an integral part of the process followed by the evaluation of the whole approach with regard to aircraft conceptual design. The overview of the research methodology is presented in Figure 3-1.

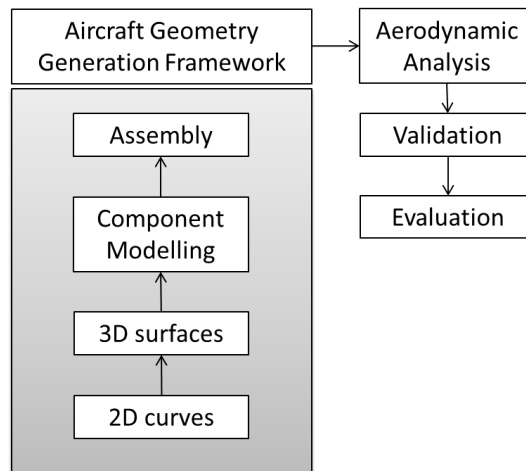


Figure 3-1 Overview of the Research Methodology

This chapter presents the evaluation of the two dimensional curve parameterisation methods. Since the airfoil profile has a direct effect on the aerodynamic efficiency, which determines the aircraft performance, the assessment in this chapter first focuses on airfoil parameterisation. Five widely cited methods: Ferguson's curve, Hicks-Henne bump functions, B-Splines representation, PARSEC, and CST method, were chosen for comparison. Amongst these five methods, Hicks-Henne bump functions and PARSEC method were developed specially for airfoil parameterisation while Ferguson's curve, B-

Splines representation, and CST method are applicable to wider configurations. The method which is the most efficient will be chosen and used as a base function for the parameterisation of the three-dimensional surfaces.

3.1 Assessment Criteria

In order to assess the selected methods for parameterisation, a list of desirable criteria from three authors: Samareh (1999), Kulfan (2006) and Padulo *et al.*(2009), are compiled in Table 3.1. Similar criteria are listed in the corresponding rows. The terminology compiled by Padulo *et al.* (2009) has been chosen as the assessment criteria which covered the properties compiled by the two previous literatures.

Table 3.1 List of desirable properties for airfoil parameterisation methods

Samareh, (1999)	Kulfan, (2006)	Padulo <i>et al.</i> (2009)
<ul style="list-style-type: none"> - Compact set of design variables - Fast and easy to use 	<ul style="list-style-type: none"> - Requires relatively few variables to represent a large enough design space to contain optimum aerodynamic shape for a variety of design conditions and constraints. - Provides easy control for designing and editing the shape of a curve. 	<ul style="list-style-type: none"> - <i>Parsimony</i>: least possible number of design parameters which can represent arbitrary shape up to a specific level of accuracy
<ul style="list-style-type: none"> - Extendable to new situation 	<ul style="list-style-type: none"> - Mathematically efficient and numerically stable process that is fast, accurate and consistent 	<ul style="list-style-type: none"> - <i>Completeness</i>: can describe any shape, up to a specific degree of accuracy
<ul style="list-style-type: none"> - Shape perturbation maintains smooth geometry 	<ul style="list-style-type: none"> - Well-behaved, produces smooth and realistic shapes 	<ul style="list-style-type: none"> - <i>Flawlessness</i>: does not generate ill-behaved shapes
<ul style="list-style-type: none"> - Design variables directly related to airplane shape design variables 	<ul style="list-style-type: none"> - Allows specification of key design parameters. - Intuitive: geometry algorithm should have intuitive and geometric interpretation 	<ul style="list-style-type: none"> - <i>Intuitiveness</i>: designers get the physical meaning of the design parameters
<ul style="list-style-type: none"> - Sensitivity can be calculated analytically 	<ul style="list-style-type: none"> - Robust: the represented curve will not change its geometry under geometric transformations such as translation, rotation and affine transformations 	<ul style="list-style-type: none"> - <i>Orthogonality</i>: each airfoil shape corresponds to a unique set of input parameters

3.2 Airfoil Parameterisation Assessment

From the five assessment criteria, the methodology for assessment of each respective property had been formulated as follows:

A. Parsimony

The first criterion to be considered is parsimony. It is the most crucial factor since the number of design parameters has a direct influence on the computational speed. The evaluation seeks for the method which can capture main airfoil geometrical features with the least possible number of design variable.

Amongst the five methods considered, the Ferguson's curves and the PARSEC method was formulated on the basis of a fixed number of design parameters, while the other three methods have varying design parameters according to the formulation of each method, as summarised in Table 3.2.

Table 3.2 Design parameters for each parameterisation methods

Parameterisation Methods	Degrees of Freedom	Design Parameters
Ferguson's Curve	[fixed] 8	Bu(y), Bl(y), TAU, TAl, TBu, TBl, alpha, beta
PARSEC	[fixed] 11	r_{LE} , X_{up} , X_{lo} , Z_{up} , Z_{lo} , $Z_{xx,up}$, $Z_{xx,lo}$, ΔZ_{te} , Z_{te} , β_{te} , α_{te}
Hicks-Henne bumps function	[free] Number of bumps	- Bumps maximum positions - Bumps value
B-Splines	[free] Number of Control Points	(x,z) coordinate of each control point
CST	[free] Order of Bernstein Polynomials	Bernstein polynomials coefficients a_{ui} , a_{li}

The algorithm for determining the least number of design variables for the target airfoil is summarised in Figure 3-2 and operates as follows:

1. From the target airfoil, the initial fitting was conducted to achieve the set of design parameters which yields the first airfoil approximation. For each parameterisation method, the initial set of design parameters can be derived from the target airfoil

2. coordinates as follows:

2.1. The design variables of Hicks-Henne bump functions were determined by the maximum height of each bump at its corresponding chord wise position.

2.2. B-spline curve uses the position of each control points as the design variables. In this fitting study, the positions of each point on the chord were fixed, leaving all control points free to move in the z-direction, i.e. perpendicular to the chord only.

2.3. For the CST method, the MATLAB curve fitting tool *cfit* was used to determine the Bernstein polynomials coefficients. The fitting was performed on the graph of shape function rather than the airfoil geometry since it has less curvature than the airfoil geometry itself. This yields better results for least square fit function used in the fitting process. By utilising the benefit of shape function concept, the fitting process was performed on shape function first (see Figure 3-2 below), the Bernstein polynomial coefficients results from fitting was then used to regenerate airfoil geometry (Figure 3-2 above).

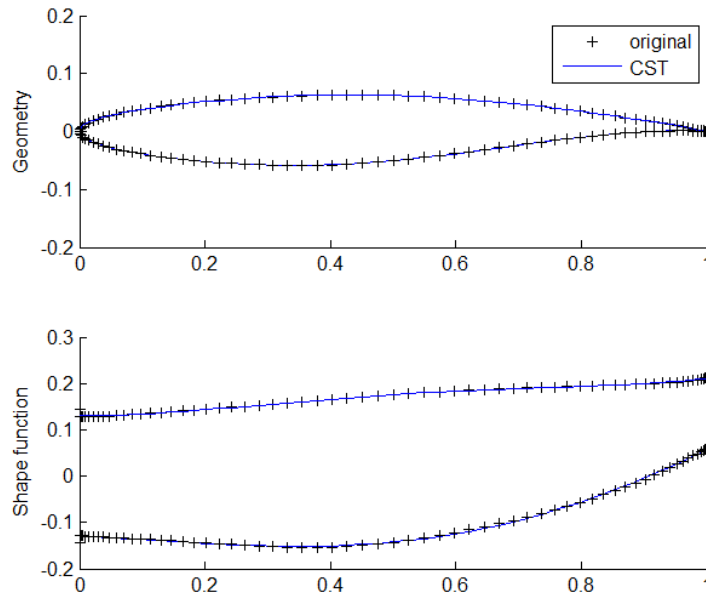


Figure 3-2 Airfoil Geometry and Shape function of RAE2822 Airfoil

3. The next step is to improve the quality of the fitting. The initial set of design variables (dv0) is used as a starting point. The regenerated airfoil profile was generated and the residuals between the regenerated and the target airfoil were calculated. The root mean

square of the residuals is then used as an objective to be minimised with the MATLAB optimiser, `fminunc`.

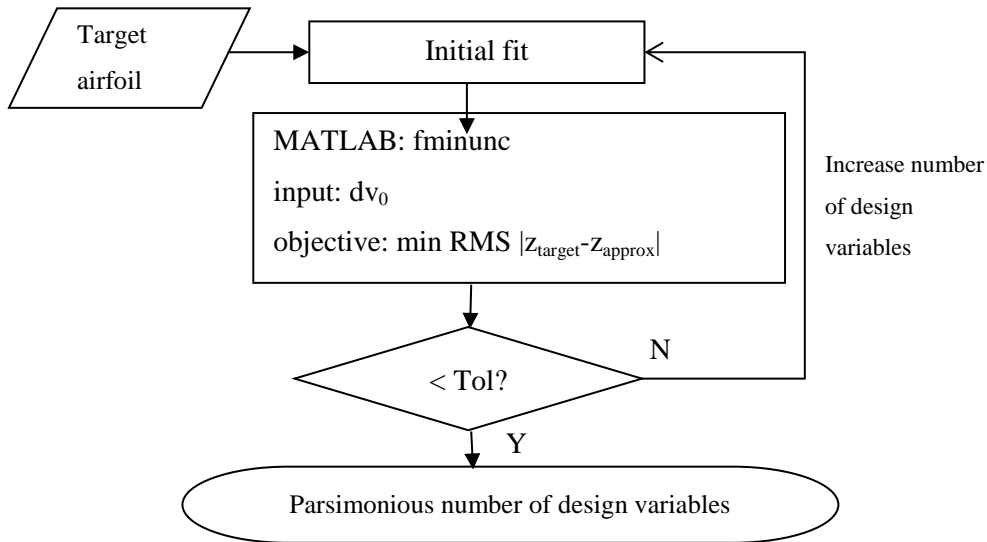


Figure 3-3 Parsimony Test

4. The typical wind tunnel tolerances (Kulfan, 2006) were used as the criteria to determine whether the fitting is acceptable or not. The approximated and the target airfoil are considered to be “geometrical exact” if the residuals are within the wind tunnel tolerance of $\pm 3.5 \times 10^{-4}$ from the leading edge to 20% units of chord, and $\pm 7 \times 10^{-4}$ elsewhere.
5. If the wind tunnel tolerance cannot be reached, the number of degrees of freedom for each method which summarised in Table 3.2 is increased by one for the next iteration. The process repeats until the geometry exactness is attained.

B. Completeness

Completeness assesses whether the parameterisation can describe any airfoil up to specific level of accuracy. The purpose of this study is to determine how many airfoils currently used in industry each parameterisation method is able to capture.

Most of the procedures repeat the parsimony number determination in the previous study but applied to the airfoil database. The methodology for fitting test is summarised in Figure 3-4.

1. From the airfoil database (UIUC, 2009) containing over 1,500 airfoils used in industry, only the airfoils described with more than 60 x-z coordinates were chosen for the study. This was to ensure that they can be directly used in the test without any pre-smoothing. The number of airfoils chosen for this study was therefore reduced to 250 airfoils.
2. For each selected airfoil, an initial set of design variables was generated through the first approximation. The initial fitting followed the same procedure used for three parameterisation methods assessed in section 3.1 with the additional two remaining parameterisation methods:
 - 2.1. For the Ferguson's curves, the trailing edge positions, the tangent vectors at the leading and trailing edges were calculated from the airfoil coordinates.
 - 2.2. For the PARSEC method, all parameters can be derived from the coordinates. The maximum crest position, the maximum thickness value on each surface, and trailing edge thickness can be determined directly from the coordinates. The remaining parameters such as leading edge radius, wedge angles, and trailing edge thickness were calculated from the coordinates.

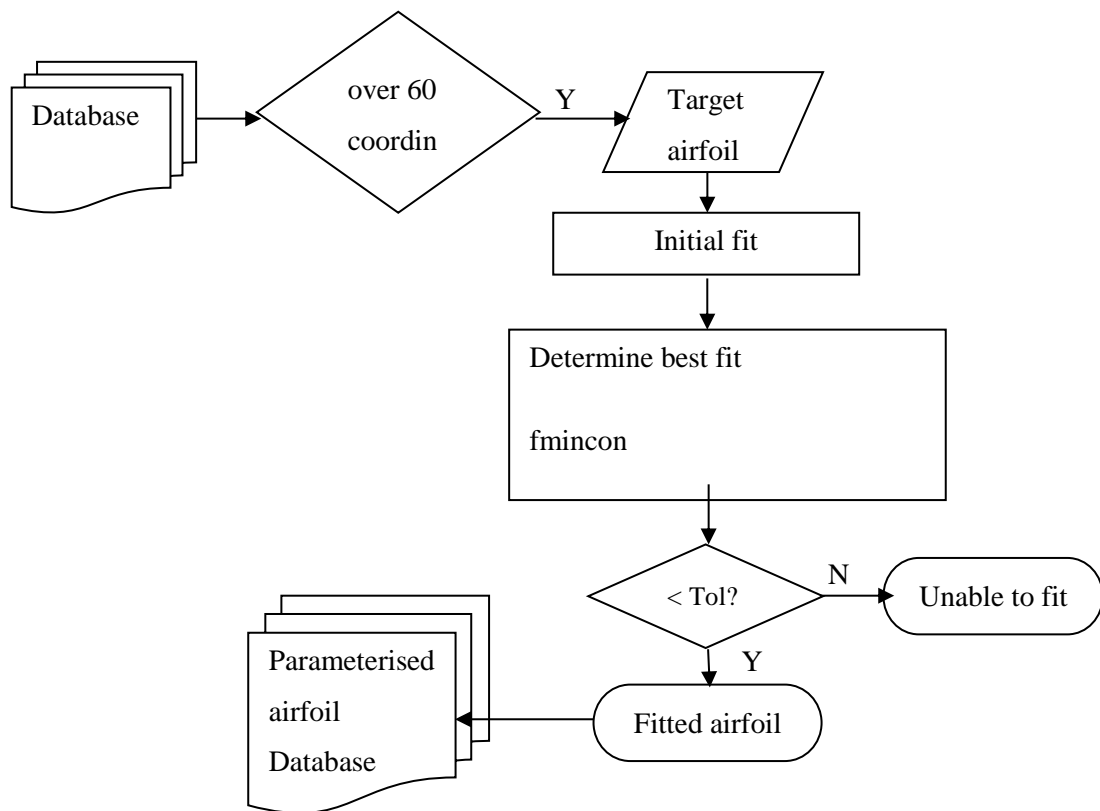


Figure 3-4 Airfoil Fitting Test

3. The selected airfoil coordinates are imported into the MATLAB environment where the design parameters and parameterised airfoils are generated. The root mean square of the residuals of each approximation is used as the objective function for the MATLAB optimiser, `fmincon`. The process continues until the residuals fall within the standard wind tunnel tolerances or until the maximum number of iterations is reached.
4. The design variables which correspond to the fitted airfoil were stored in the new database for the future use. If the standard wind tunnel tolerance cannot be reached, the method is considered unable to fit that particular airfoil and discarded.

The result of this fitting test should provide more realistic range of design parameters and will be used as a design range for prove of orthogonality described in the next subsection.

C. Orthogonality

Orthogonality guarantees that each airfoil shape corresponds to a unique set of input parameters. This property is particularly relevant to the parameterisation methods on which an airfoil is constructed by combining existing sets of airfoils with other analytical functions.

With this regard, Keane and Nair (2005) stressed that

“...the design functions should be as geometrically orthogonal as possible. Lack of orthogonality implies a non-unique mapping of the parameter values to the geometry. The resulting spurious multimodality of the objective function can significantly degrade the search process”

Keane and Nair (2005) also pointed out that for the Hicks-Henne formulation, the sine bumps which are used as base functions are non-orthogonal, and thus the solutions such as the inverse problem of a certain pressure distribution are not guaranteed to be attained.

Ceze et al (2009) have tested the orthogonality of the CST and found that at high degree of the polynomials, there exists another set of coefficients which yields the same profile

as a set of unitary coefficients, as shown in Figure 3-5. The circles represent an airfoil for which all the coefficients are equal to 1, while the stars represent another set of polynomial coefficients which produces the same shape function and airfoil profile. However, this behaviour starts occurring at the polynomial of order 30 which is far beyond the parsimonious polynomials order from 5 to 8 considered in this research.

The procedure to check the orthogonality property of the PARSEC method was conducted as follows:

1. The design spaces were defined from the range of each variable found in section 3.2 B.
2. One thousand samples from this design space were generated through Latin Hypercube.
3. The condition number of each sample was calculated. The condition number indicates the accuracy of the results from matrix inversion. Large condition number means the matrix is ill-conditioned, yields error in solving system of linear equations and causes non-uniqueness in the solution.

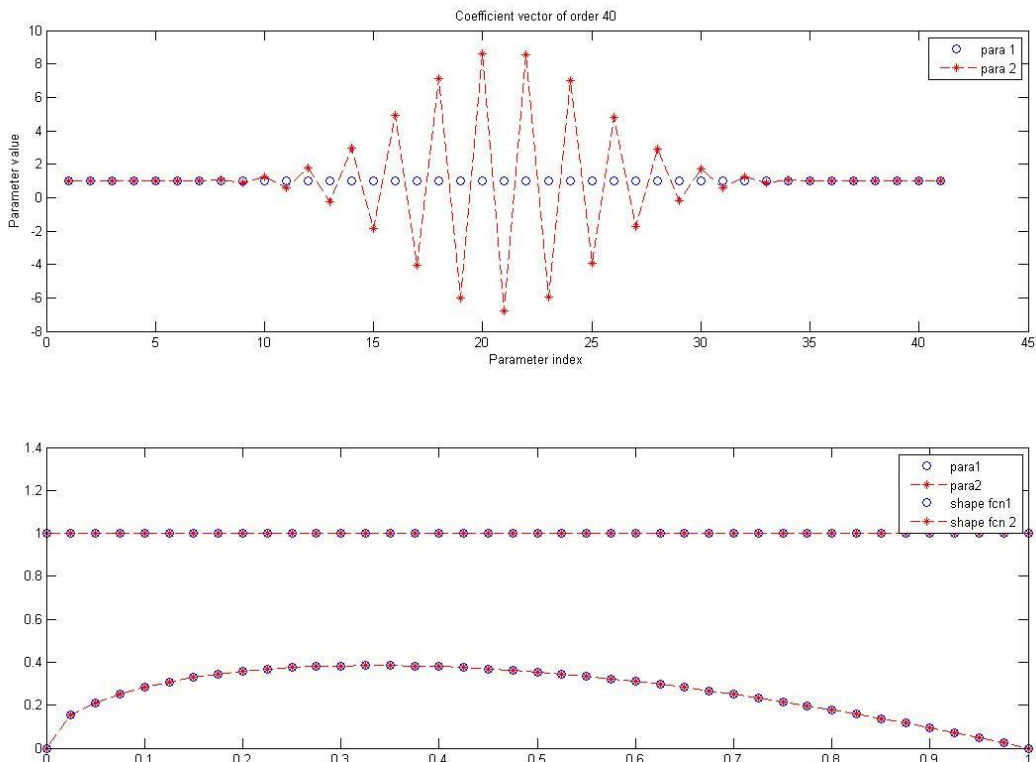


Figure 3-5 Non-orthogonal set of CST design variables (Ceze et al., 2009)

D. Flawlessness

In some cases, the optimisation might arrive at the solution with non-airfoil-like shapes. A previous research on PARSEC formulation (Padulo *et al*, 2009) found an undesirable bump which occurs at another position of the profile in addition to the defined maximum crest location (Figure 3-6) or intersections of the upper and the lower surfaces (Figure 3-7).

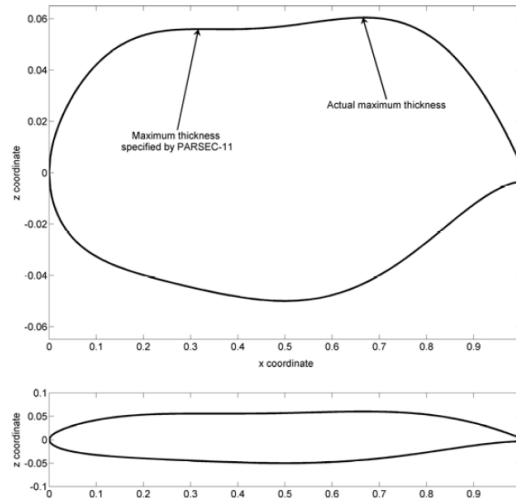


Figure 3-6 Inconsistent Maximum Crest Locations (Padulo *et al.*, 2009)

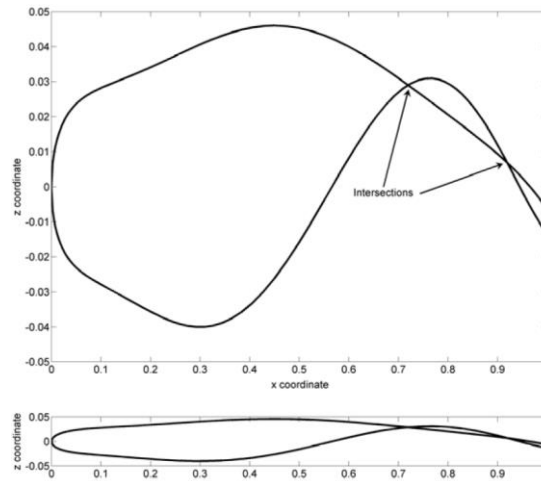


Figure 3-7 Intersections between upper and lower surface (Padulo *et al.*, 2009)

In order to assess flawlessness, the samples generated to prove orthogonality were examined for the irregular shapes mentioned above.

E. Intuitiveness

Intuitiveness assesses whether the method relates the design parameters to the physical design meaning which in turn simplifies the choice of input bounds or design judgement. The design parameters are assessed according to the following criteria:

1. The design parameters are directly related to the geometrical parameters
2. Designers can understand the effect on geometry when manipulating one or more of these design parameters

3.3 Results and analysis

The results from the analysis of the five parameterisation methods: Ferguson Curves, Hicks-Henne bump functions, B-Splines, PARSEC, and CST, against five desirable criteria, are summarised in this section:

A. Parsimony

The minimum number of design variables required for Ferguson's Curve and PARSEC are fixed due to their formulation. The parsimonious numbers of B-Splines and CST found agree with literature (Nadaraja, 2005 and Kulfan, 2006). For Hicks-Henne bumps, the number of required design variables is determined from fitting airfoils with typical wind tunnel tolerance. The effect of number of bumps on representation of the RAE2822 airfoil is presented in Figure 3-8. The maximum residuals from the leading edge to 20% chord (lower line) are within the tolerance of 3.5×10^{-4} , while the residuals on the rest of the airfoil fall within the tolerance of 7×10^{-4} when using 16 bumps or more. This is the number of bumps required for each upper and lower profile, and therefore resulting in total of 32.

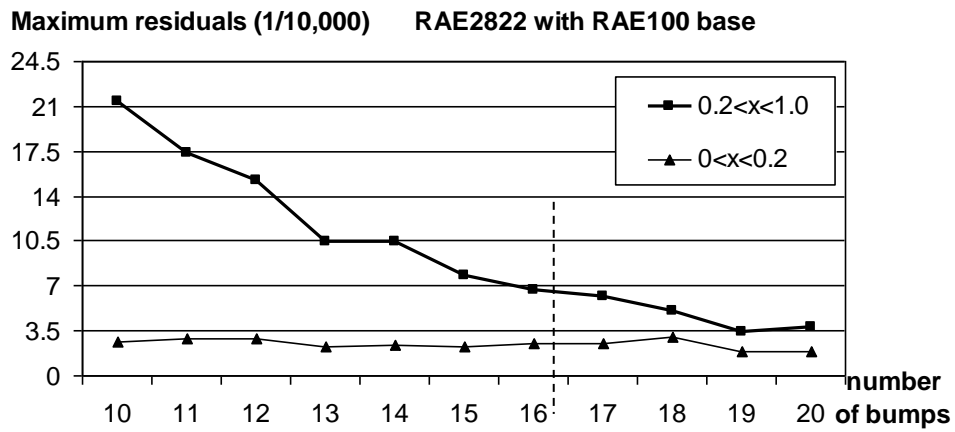


Figure 3-8 Number of Hicks-Henne bumps on representation of RAE2822 airfoil

Amongst the five methods, the Ferguson’s curve is the most parsimonious with 8 variables. The parsimonious numbers of the remaining methods are presented in Table 3.3.

Table 3.3 Parsimonious numbers of design variables

<i>Method</i>	<i>Number of variables</i>	<i>Remarks</i>
Ferguson's Curves	8	Fixed variables
Hicks-Henne	32	Depends on number of bumps
B-Splines	13	Depends on number of control points
PARSEC	11	Fixed variables
CST	11	Depends on the order of polynomials

B. Completeness

Fitting performances were evaluated against the wind tunnel tolerance. Examples of a parameterised RAE2822 airfoil with Ferguson's curve and PARSEC are presented in Figure 3-9. As can be seen, Ferguson's curve shows limitations in capturing the sharp curvature on the lower profile, whereas PARSEC, as well as the other three methods which perform in the same level of accuracy, achieve better approximation of the profile.

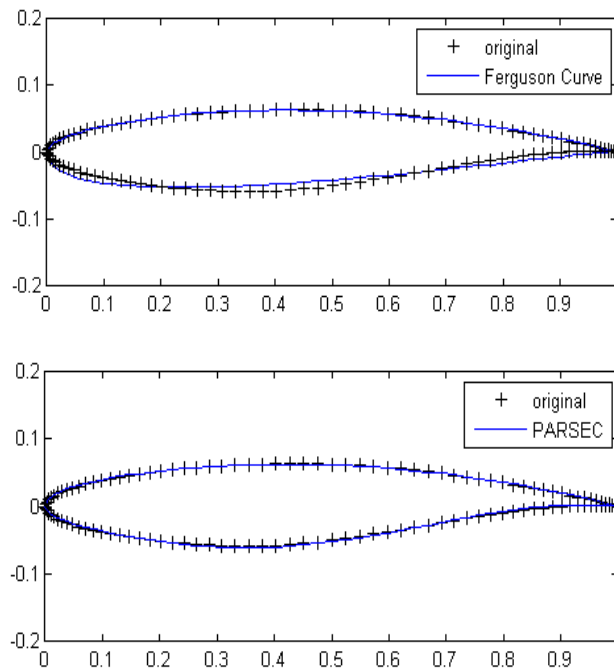


Figure 3-9 Parameterised RAE2822 airfoil by Ferguson's curve and PARSEC

The parameterised airfoil which falls within acceptable tolerance after performing a fitting test is considered a good approximation. The number of good approximation

airfoils, out of the selected 250 airfoils is summarised in Table 3.4. The results show that B-splines method performed best by capturing 98.4 per cent of the total airfoils followed by Hicks-Henne bump functions which captured 95.2 per cent. The PARSEC and CST perform almost equally with 94 per cent and 93.6 per cent, respectively. The Ferguson’s curve which has limited number of design variables did not perform as well as the other methods, with only 60 per cent of airfoils captured.

Table 3.4 Percentage of airfoils with satisfactory fitting approximation

Methods	Number of Good Approximation [/ 250]	Percentage
Ferguson’s Curve	150	60.0
Hicks-Henne	238	95.2
B-Splines	246	98.4
PARSEC	235	94.0
CST	234	93.6

C. Orthogonality

The Ferguson’s curves are generated based on 4 design parameters. At fixed boattail and tail closure angles, each a unique set of design parameters representing in a certain airfoil, therefore it is orthogonal. For B-splines, Mason *et al.* (1993) demonstrated that it is possible to construct orthogonal splines based on B-Splines for smoothing applications.

A conclusion has been drawn from the literature (see section 3.2 C) that Hicks-Henne bump functions are non-orthogonal, while CST method is orthogonal when the order of polynomials below 30, which is beyond the parsimonious order of 6.

For PARSEC method, the orthogonality was assessed by examination of the matrix condition numbers of 1,000 samples generated from the design range obtained in the previous section. The calculated condition numbers of these samples ranged from 3,945 to 22,137, which were higher than the single digit values. Moreover, since PARSEC formulation can be rearranged to polynomials degree 5.5 (see appendix A) which is lower than 30, this confirms that PARSEC formulation is orthogonal.

D. Flawlessness

The flaws considered in this paper are additional bump at another location apart from at maximum crest position, and compenetration. These behaviours can be observed with an interactive tool “PARSEC-Airfoil Generator” which is provided on the webpage of DLR (2010) in order to understand the influence of each design parameter to the design.

By adjusting only a single parameter may result in erroneous shapes as shown in Figure 3-10.

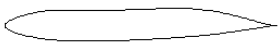

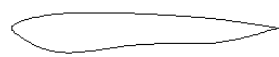

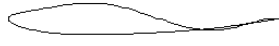
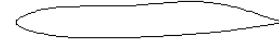
Change x_u from 0.3 to 0.25  An additional bump starts to form on the upper profile close to the trailing edge side	Change y_u from 0.06 to 0.11  An additional bump starts to form on the upper profile closer to the trailing edge side	Change x_l from 0.3 to 0.65  An additional bump starts to form on the lower profile close to the leading edge side.
Change y_{xxl} from 0.45 to 2.00  The lower profile intersects the upper profile near the trailing edge	Change y_{xxu} from -0.45 to -2.0  The upper profile intersects the lower profile near the trailing edge	Change y_{xxu} from -0.45 to 2.0  This produces an addition bump on the upper profile close to the trailing edge side.

Figure 3-10 Examples of irregular shapes produced with PARSEC method

Since CST method is based on the polynomials which proved to be mathematically equivalent (see appendix A), this same scenario is also found in CST airfoil formulation. The design space could be rendered robust through a procedure based on the self-organizing maps, which can be found in the work by Padulo *et al.*(2009).

E. Intuitiveness

The PARSEC method appears to be the most intuitive amongst all methods compared in this research since it directly describes the main geometrical features of the airfoil. All the design parameters e.g. the leading edge radius, the trailing edge thickness, boat-tail angle, etc., carries geometrical meaning. Designers can understand the change on the geometry when manipulating one or more of these design parameters, for instance, increase the leading edge radius, reduce the maximum thickness airfoils, or change the location of maximum crest position. This is useful for optimization processes, especially for non-experienced designer trying to improve the aerodynamic performance of the airfoils.

~~In CST, a set of geometric parameters are related to the shape functions, in the same way as in PARSEC. This allows the designers to directly manipulate the shape of the airfoil using these geometric parameters instead of polynomials coefficients which is very practical. Therefore, these two methods are considered the most intuitive amongst all five methods.~~ CST may or may not be considered intuitive. If the Bernstein polynomial coefficients are used, the method is not intuitive for the designer. However, if the design parameters which can be determined on shape function curve such as leading edge radius, closure thickness, or maximum thickness, as shown in Figure 3-11, is used, the method is considered intuitive which designers have direct control over airfoil geometry.

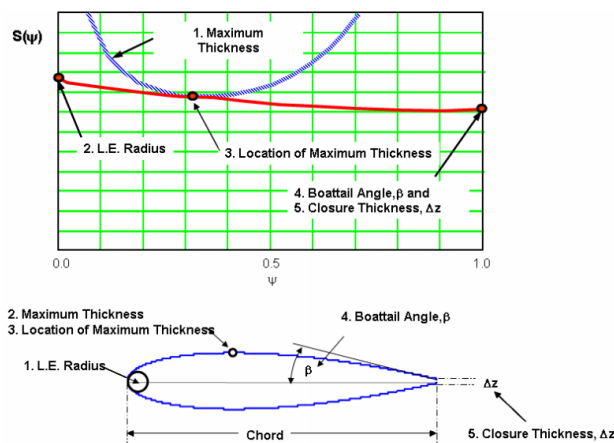


Figure 3-11 Shape function and the corresponding design parameters (Kulfan, 2007)

Ferguson’s curve uses tangent vectors, boat tail angle, and camber angle as design parameters. Designers can change the shape of airfoils intuitively based on these sets of parameters, therefore the method is very intuitive but not as flexible as other method due to limited number of design parameters.

The Hicks-Henne bump functions provide information about the influence of parameter change on the profile by manipulating contributions of each base function, while B-Splines enables designers to manipulate the shape by moving control points. These design variables are not directly related to geometry, but still provide the designer with a physical meaning of the design change. Therefore, these two methods are considered partially intuitive.

~~Last, Ferguson’s curve provides only information about the positions and tangent vector directions at the starting and ending positions of each profile. Therefore it is considered the least intuitive amongst the five methods.~~

3.4 Summary

The performances of the selected airfoil parameterisation methods with respect to the five desirable criteria are summarised in Table 3.5.

Table 3.5 Comparison of selected airfoil parameterisation methods

Methods	Parsimony	Completeness	Orthogonality	Flawlessness	Intuitiveness
Ferguson’s Curve	8	60.0 %	Y	0	Direct
Hicks-Henne	32	95.2 %	N	0	Partial
B-Splines	13	98.4 %	Y	0	Partial
PARSEC	11	94.0 %	Y	error	Direct
CST	11	93.6 %	Y	error	Direct*/Not

*In case of geometric related design parameters used

Considering all five properties, the B-Spline, PARSEC, and CST methods performs well in general. The parsimonious number of design variables for PARSEC and CST are lower than B-Splines with some loss in completeness as a trade-off. Considering Orthogonality, all methods except Hicks-Henne bumps function is orthogonal. Ferguson’s curve, PARSEC and CST uses intuitive design parameters. In case of CST, it can be considered intuitive when the geometric related design parameter is used, if the polynomial coefficient is used, the method is not intuitive at all.

Even though PARSEC and CST produce some irregular shapes, this can be eliminated by setting a proper design bound. In general, the PARSEC and CST are equally performs but CST provides more flexibility considering that the degree of polynomials can be increased for more accurate results, and therefore chosen as the parameterisation method for airfoil.

3.5 Extension of CST application to Cross-section Parameterisation

There are number of parameterisation methods commonly used for cross-section shapes such as conic curves, spline curves. CST method has also wider application from airfoil shape to cross-section shapes. The method is selected due to the following reasons:

1. The method required only 5 design parameters to generate common shape used in fuselage of aircraft.
 - W cross-section maximum width
 - H_u cross-section upper lobe height, measured along the symmetrical line from the maximum width line (see Figure 3-12)
 - H_l cross-section lower lobe depth, measured along the symmetrical line from the maximum width
 - NCu, NCl cross-section shape function coefficients

The cross-section curve is generated through the following equations (Kulfan, 2006):

$$z_u(\eta) = [\eta^{NCu}(1 - \eta)^{NCu}] \cdot H_u \quad [3-1]$$

$$z_l(\eta) = [\eta^{NCl}(1 - \eta)^{NCl}] \cdot H_l \quad [3-2]$$

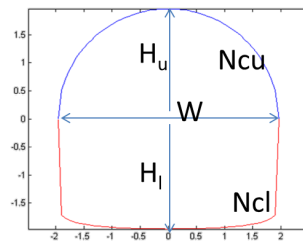


Figure 3-12 CST Cross section description

Comparing to spline-based methods, the bodycross section in Figure 3-12 will require minimum 8 control points: 4 to control over total height and total width, and 4 to control shape of each quadrant.

2. The class-shape function has already been selected for airfoil generation. By keeping the same concept, the extrusion of cross-section shape to generate surfaces of aircraft component would be more convenient. Therefore, CST is chosen for cross-section shape, as well.

3.6 Conclusions

Presented in this chapter are the evaluations of the selected curve parameterisation method. For airfoil profile, PARSEC and CST methods perform best overall. They require small number of design parameters when compared to the other methods studied in this chapter. The design parameters are intuitive, and can be adjusted in order to modify the airfoil profile in wide ranges. CST is chosen as two-dimensional base functions: both airfoil-like and cross-section, and will be combined with proper lofting functions for the generation of three-dimensional surfaces in the next chapter.

4 Aircraft Parameterisation Method

The parameterisation methods for curves were evaluated in Chapter 3. In this chapter, curve parameterisation is extended to surfaces and to major component modelling. A methodology for component parameterisation and assembly is presented, followed by examples of parameterised aircrafts in three different configurations.

The aircraft parameterisation follows a bottom-up process, starting from local to global descriptions as presented in Figure 3-1.

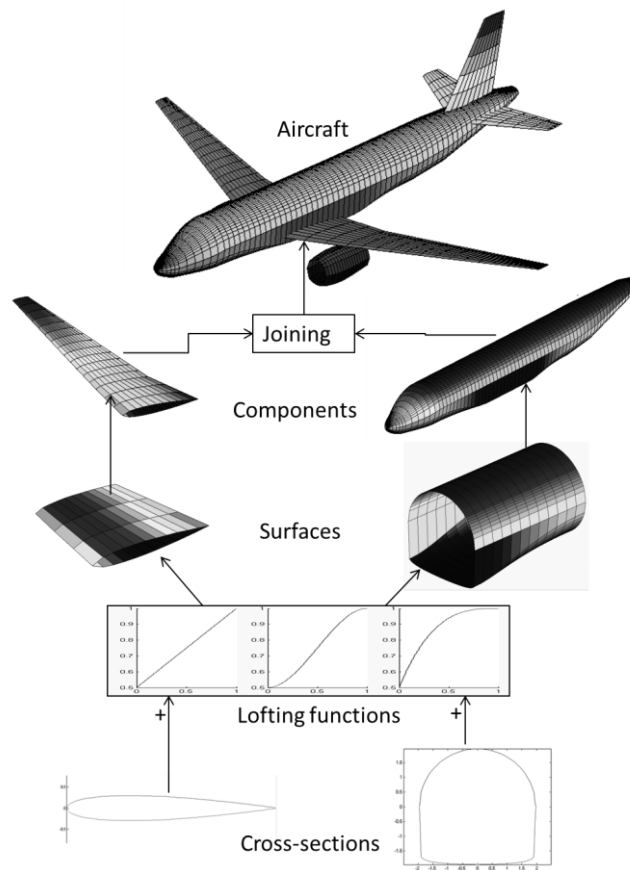


Figure 4-1 Aircraft components

Each surface is generated from a combination of two-dimensional bounded curves with a set of selected distribution functions. The surface sections are used to build up the components, which are then assembled to a full aircraft configuration.

4.1 Axes Conventions

The geometry definition follows the axes conventions as shown in Figure 4-2. The Cartesian coordinates corresponding to each axis are denoted as follows:

axes	directions	normalised mesh	local coordinates	global coordinates
X	longitudinal: parallel to the fuselage reference line positive from nose to tail	ψ	x	X
Y	lateral: perpendicular to the plane of symmetry from root chord to wingtip	η	y	Y
Z	vertical, positive upwards, perpendicular to the xy plane	ζ	z	Z

The normalised meshes are generated in the Cartesian coordinates with two independent variables ψ and η , with their corresponding surface function $\zeta = f(\psi, \eta)$. These normalised meshes are mapped to the local coordinates, xyz, to produce the actual dimensions. These surfaces are translated to the global reference position XYZ for the full configuration assembly.

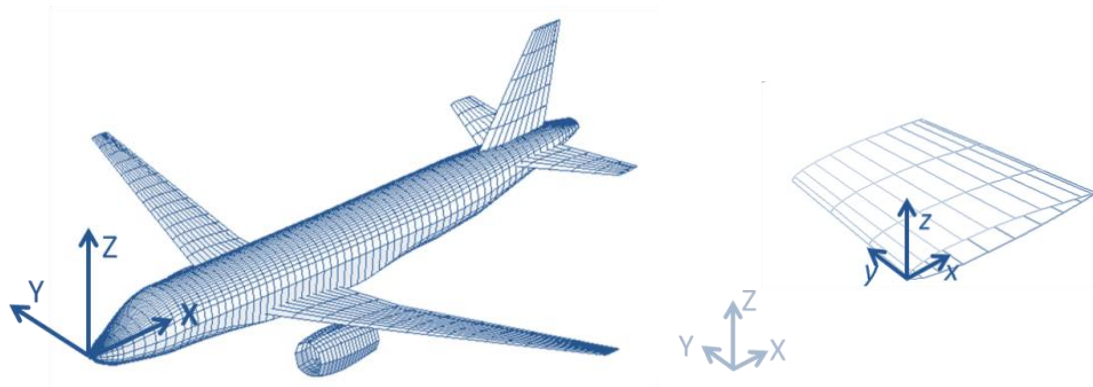


Figure 4-2 Aircraft geometric axis conventions

4.2 Geometry Classes

Corresponding to the component built-up scheme, the geometry classes were introduced as shown in Figure 4-3. There are five levels of classes: Aircraft, Component, Section, Distribution, and Cross-section.

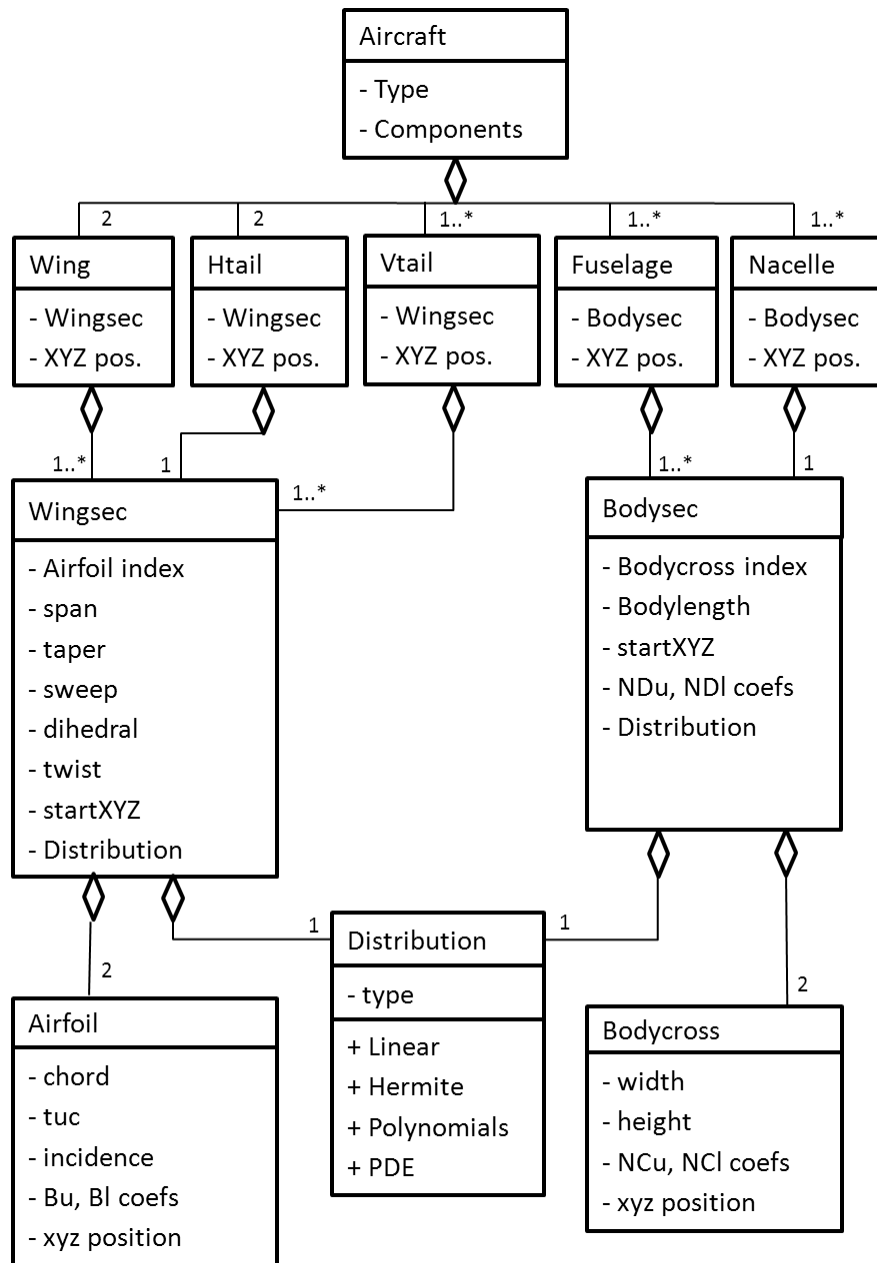


Figure 4-3 Classes and attributes for full aircraft geometry generation.

The “Aircraft” class is the highest class in the hierarchy. Each aircraft contains the following components: “Wing”, “Tail”, “Fuselage”, and “Nacelle”. Each of these components may contain a single or multiple sections. The lifting surface components

contain the “Wingsec” class, while the free-from body components contain the “Bodysec” class.

Each section class is generated from two cross-sections with a distribution function. The cross-section class contains information with regard to the cross-section shapes e.g. “Airfoil” or “Bodycross”. This is the most detailed description considered in this research. The surface generation process starts from this level.

4.2.1 Cross-section

The cross-section class contains the detailed description of each curve in two dimensions. The class can be separated in two main categories as follows:

a) Airfoil Class

The “Airfoil” class constructs with the CST functions through the following design parameters:

- chord airfoil chord length. The chord length of each airfoil can be defined by the user or calculated through the taper ratio (TR) of that wing section with the wing root chord as follows:

$$chord_{i+1} = chord_i \cdot TR_i \quad [4-1]$$

- tuc thickness to chord ratio. By default, this variable is set to be constant for every section along the wing span, or can be specified by user for each particular section.

- inc incidence angle. The incidence angle of each section is calculated from wing setting angle at wing root and the wing section twist as:

$$inc_{i+1} = inc_i \cdot twist_i \quad [4-2]$$

- xyzle the leading edge position is defined at the starting point of each airfoil section, each variable is calculated through sweep and dihedral angle with respect to the wing starting position following the relationships:

$$x_{i+1} = x_i + span_i \cdot \tan(sweep_i) \quad [4-3]$$

$$y_{i+1} = y_i + span_i \quad [4-4]$$

$$z_{i+1} = chord_{i+1} \cdot \tan(inc_i + twist_i) + span_i \cdot \tan(dih_i) \quad [4-5]$$

In case of the quarter chord sweep angle, which is defined at quarter chord line, the leading edge position in x axes is calculated from:

$$x_{i+1} = x_i + 0.25 \cdot chord_i + span_i \cdot \tan(sweep_i) - 0.25 \cdot chord_{i+1} \quad [4-6]$$

- Au, Al CST Bernstein polynomial coefficients. A CST airfoil profiles are generated with the Class-shape Transformation function method (Kulfan, 2006):

$$\zeta_U = \sqrt{x}(1-x) \sum_{i=0}^N \left[A_{u,i} \cdot \frac{N!}{i!(N-i)!} \cdot \psi^i (1-\psi)^{N-i} \right] + \psi \cdot \Delta\zeta_U \quad [4-7]$$

$$\zeta_L = \sqrt{x}(1-x) \sum_{i=0}^N \left[A_{L,i} \cdot \frac{N!}{i!(N-i)!} \cdot \psi^i (1-\psi)^{N-i} \right] + \psi \cdot \Delta\zeta_L \quad [4-8]$$

b) Bodycross Class

The “Bodycross” class generates a cross-section shape of the free-form body surfaces such as fuselage or nacelle through following parameters:

- W cross-section maximum width
- H_u cross-section upper lobe height, measured along the symmetrical line from the maximum width line (see Figure 4-4)
- H_l cross-section lower lobe depth, measured along the symmetrical line from the maximum width
- NC_u, NC_l cross-section shape function coefficients
- XYZ_c centreline position, where the symmetrical line intersects the maximum width line

The cross-section curve is generated through the following equations (Kulfan, 2006):

$$y(\eta) = (1 - 2\eta) \cdot W \quad [4-9]$$

$$z_u(\eta) = [\eta^{N_{Cu}}(1 - \eta)^{N_{Cu}}] \cdot H_u \quad [4-10]$$

$$z_l(\eta) = [\eta^{N_{Cl}}(1 - \eta)^{N_{Cl}}] \cdot H_l \quad [4-11]$$

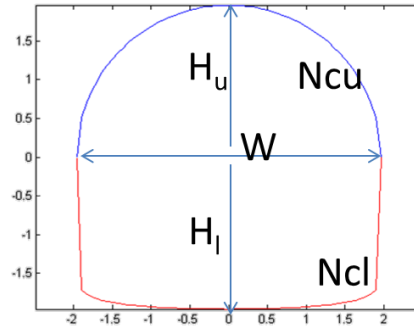


Figure 4-4 Bodycross Description

Figure 4-5 presents three cross-section shapes corresponding to three cross-section coefficients. The upper lobe is kept constant with $N_{Cu}=0.5$, resulting in circular arch. The lower lobe varies from 0.5 (left), 0.1 (middle), and 0.05 (right). As the coefficient value gets lower, the lobe profile becomes close to rectangular. This type of cross-section will be used to generate the aircraft cross-sections where the wing is connected to the fuselage.

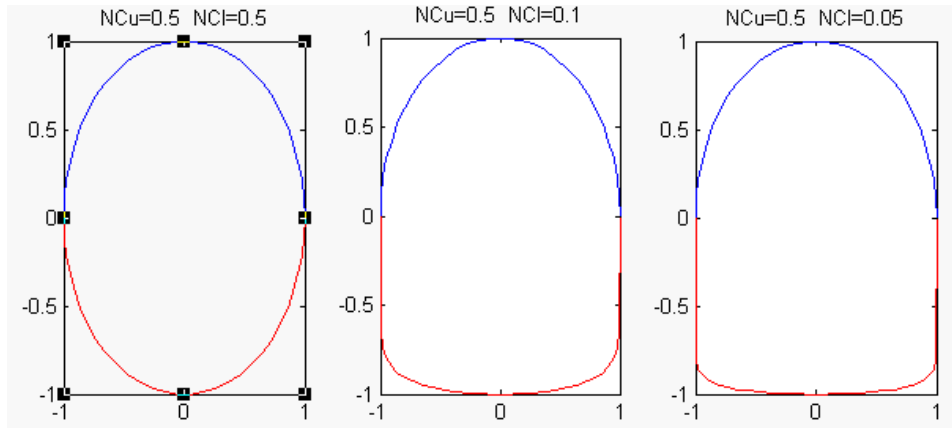


Figure 4-5 Cross-section shapes corresponding to NCl Coefficients

4.2.2 Distribution Class

The surface is generated with the distribution function along the surface. A set of distribution functions was introduced as follows:

a) Linear distribution

The linear distribution is used when the change along the lofting direction is linear. The formulation is as follows:

$$NC(\psi) = (1 - \psi) \cdot NC_i + \psi \cdot NC_{i+1} \quad [4-12]$$

b) Hermite interpolation

Hermite interpolation (Ferguson, 1964) was chosen for lofting due to its simplicity and sufficient control over the surface lofting. The cross-section shape function coefficients are interpolated between NC_i and NC_{i+1} as follows:

$$NC(\psi) = (2\psi^3 - 3\psi^2 + 1) \cdot NC_i + (-2\psi^3 + 3\psi^2) \cdot NC_{i+1} + \dots \\ + (\psi^3 - 2\psi^2 + \psi) \cdot \tan NC_i + (\psi^3 - \psi^2) \cdot \tan NC_{i+1} \quad [4-13]$$

From Equation 4-13, the first two terms produce the blending of the surface between the starting and ending positions. The last two terms provide the control of the curvature through the tangent vectors at each end. In order to ensure the continuity at each joint section, the tangent vectors of the two connecting surfaces are set to be equal. This function is mainly used for the fuselage surface lofting.

c) Bernstein Polynomials

The cubic Bernstein polynomial is also used as a lofting function to provide more control over the surfaces in three-dimension.

$$NC(\psi) = \sum_{i=0}^N \left[B_i \cdot \frac{N!}{i!(N-i)!} \cdot \psi^i (1 - \psi)^{N-i} \right] NC \quad [4-14]$$

Similar to the airfoil parameterisation, the Bernstein coefficients must be determined through the fitting process to the engine cowling line. Figure 4-6 shows an example of engine cowling shape which corresponds to the Bernstein polynomial coefficients, $B = [1.00 \quad 1.24 \quad 1.30 \quad 1.24 \quad 1.12 \quad 1.00]$.

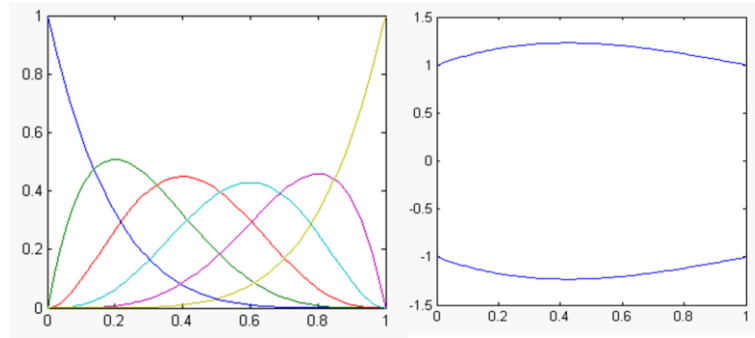


Figure 4-6 Bernstein polynomial (left) with its corresponding engine cowling line (right)

d) Distribution coefficients

For the outermost section which required closing of the surface, the distribution function of the CST method is imposed on top of the whole surface descriptions after the lofting function has been selected. The function is simply in the form of:

$$FD = [\psi^{ND1}(1 - \psi)^{ND2}] \quad [4-15]$$

The coefficients, ND, produce the surface edge as follows:

- ND = 0 Open edge, the form functions becomes one
- ND = 0.05 Closed Cylindrical edge
- ND = 0.5 Closed Spherical edge
- ND = 1.0 Closed Conical edge

Figure 4-7 presents the corresponding shapes for each distribution coefficient, ND1, with ND2 = 0.

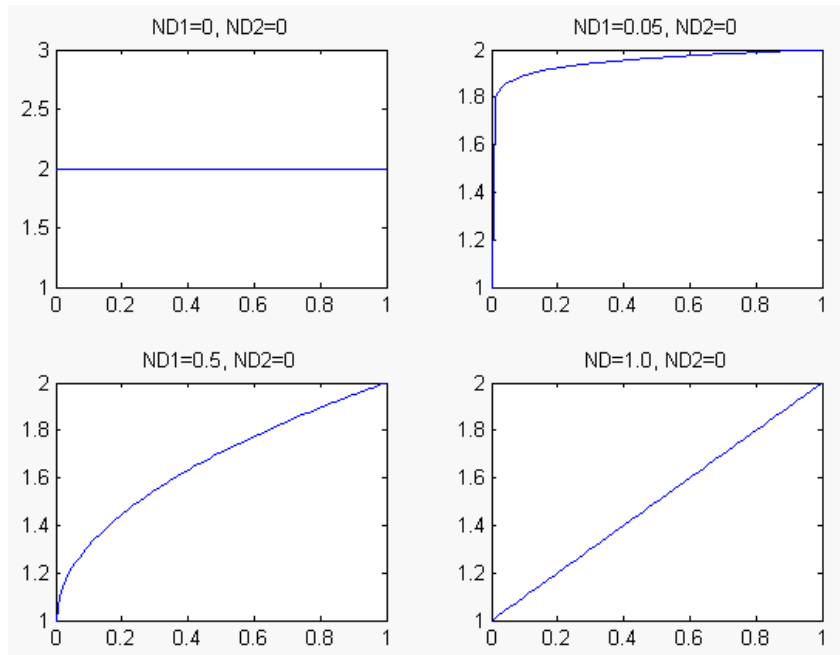


Figure 4-7 Closed surface form functions

4.2.3 Surfaces Class

The “Surface” class produces three-dimensional surface coordinates. Each surface is generated with the two bounding curves defined in the cross-section class with a distribution function. The surface is separated into two categories: “Wingsec” or the lifting surfaces and the body surfaces, denoted by “Bodysec”.

a) Wingsec

Each wing section consists of two bounded airfoils which carry the entities from the “Airfoil” class, one for the inner airfoil (closer to the wing root) and one for the outer airfoil. Both airfoils are set parallel to the streamline. Each wing section is defined with a constant sweep, dihedral and twist angle. The Wingsec class contains the following attributes:

- Airfoil (i) the inner airfoil
- Airfoil (i+1) the outer airfoil
- span span of the section
- taper taper ratio
- sweep sweep angle
- dihedral dihedral angle

- twist twist angle
- startXYZ section start position
- Distribution distribution

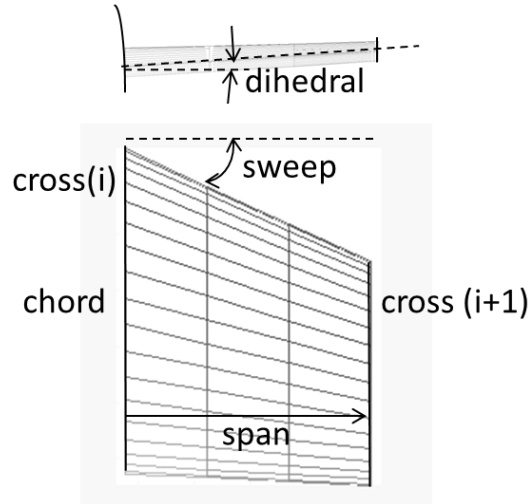


Figure 4-8 Wing section definitions

b) Bodysec

Similar to the Wingsec Class, Bodysec stores a body surface bounded by two cross-section curves. The bodysec class contains the following attributes:

- Bodycross (i) pointer to the forward cross-section
- Bodycross (i+1) pointer to the aft cross-section
- seclength longitudinal length
- startXYZ section start position
- NDU, NDI distribution function coefficient
- distribution type of the blending function describing the transition

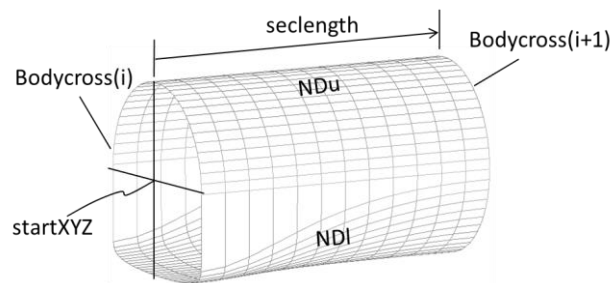


Figure 4-9 Body section descriptions

4.2.4 Component Class

The “Component” class describes each aircraft component, which can be separated in two categories corresponding to the type of section.

a) Lifting surfaces

Lifting surfaces are the components which produce lift e.g. main wing, horizontal and vertical tails. Each surface may contain one or more wing sections. The standard horizontal and vertical tails contain a single wing section, while the main wing consists of two sections: from wing root to crank and crank to wing tip. The blended wing body usually contains more than two wing sections, as shown in Figure 4-10.

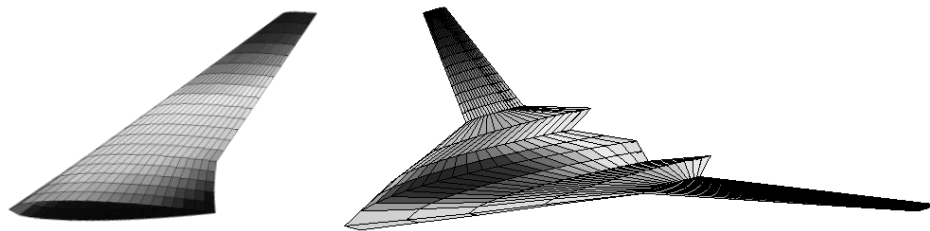


Figure 4-10 Conventional wing (left) and Blended Wing with multiple wing sections (right)

b) Body

The Body class describes a fuselage or nacelle. The fuselage usually contains multiple sections depending on the number of cross-sections defined by the designer. The nacelle is constructed with a single section as shown in Figure 4-11.

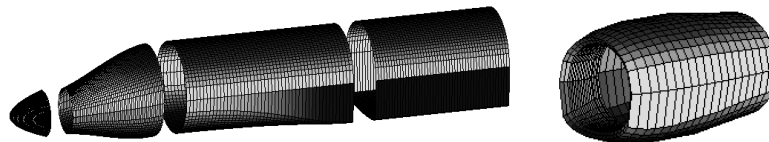


Figure 4-11 Fuselage (left) and nacelle (right)

Each component is defined with its XYZ starting position with respect to the global reference frame. For fuselage is the nose position, while for wing and tails is the apex.

4.2.5 Aircraft

The Aircraft class describes a full configuration with all selected components. This class contains two attributes:

a) Type

This attribute defines type of an aircraft based on its configuration. The current research focuses on three types of aircraft: a generic airliner, blended wing-body, and an unmanned aerial vehicle which demonstrates a more unconventional configuration.

b) Component

There are five main components in a full aircraft configuration: wing, horizontal tail, vertical tail, fuselage and nacelle. The components are chosen to be included in the design corresponding to each type of aircraft. For instance, a generic airliner contains fuselage, wings, the horizontal and vertical tails and the nacelles, while the blended wing body contains two main wings joined in the plane of symmetry instead of a fuselage.

4.3 Surface Lofting

Once the geometry parameters have been defined by the user, the surface generation may proceed. In this research, the parameterisation is performed in MATLAB® (2007) where the surface is generated on the basis of grid meshes.

The lofting combines two cross-section curves with the selected distribution function. The formulations for each type of surface are as follows:

4.3.1 Wing

The wing surface class generates all components, and contains airfoil shape cross-sections such as the main wing, the horizontal tail, the vertical tail. For each wing surface, each airfoil section (station) is set parallel to the stream line. The surface for each wing section is lofted between the two boundary sections in the span wise direction.

The local coordinates x, y, z are functions of the chord-wise distribution (ψ), and the span wise distribution (η).

$$x(\psi, \eta) = \psi \cdot chord \cdot (1 - \eta \cdot (1 - taper)) + \eta \cdot span \cdot \tan(sweep) \quad [4-16]$$

$$y(\psi, \eta) = \eta \cdot span \quad [4-17]$$

$$z(\psi, \eta) = zle(\eta) \cdot (1 - \psi) + \zeta(\psi, \eta) + \eta \cdot zt \quad [4-18]$$

From equation 4-17, the local coordinates $z(\psi, \eta)$ consist of three components:

- the leading edge position,

$$zle(\eta) = \tan(inc + \eta \cdot twist) \cdot chord \quad [4-19]$$

- the CST airfoil profile, for the upper and lower parts:

$$\zeta_u(\psi, \eta) = \sqrt{\psi}(1 - \psi) \cdot \sum_{i=0}^N \left[A_{u,i}(\eta) \cdot \frac{N!}{i!(N-i)!} \cdot \psi^i \cdot (1 - \psi)^{N-i} \right] + \psi \cdot \Delta\zeta_u \quad [4-20]$$

$$\zeta_l(\psi, \eta) = \sqrt{\psi(1 - \psi)} \cdot \sum_{i=0}^N \left[A_{l,i}(\eta) \cdot \frac{N!}{i!(N-i)!} \cdot \psi^i \cdot (1 - \psi)^{N-i} \right] + \psi \cdot \Delta\zeta_l \quad [4-21]$$

- and the wingtip position,

$$zt = span \cdot \tan(dih) \quad [4-22]$$

The Bernstein coefficient $A_{u,i}(\eta)$ is the function of the span-wise position which is lofted with a linear distribution (Figure 4-12 left). Equation 4-12 becomes:

$$A_u(\eta) = (1 - \eta) \cdot A_{u,init} + \eta \cdot A_{u,final} \quad [4-23]$$

In order to gain control over the wing distribution, the wing surface can be lofted with Bernstein polynomial distribution (Equation 4-13), Equation 4-20 therefore becomes:

$$\zeta_u(\psi, \eta) = \sqrt{\psi(1 - \psi)} \cdot \sum_{j=0}^{Ny} \sum_{i=0}^{Nx} \left[B_{u,ij}(\psi, \eta) \cdot \frac{Nx!}{i!(Nx-i)!} \cdot \frac{Ny!}{j!(Ny-j)!} \cdot \eta^j (1 - \eta)^{N-j} \cdot \dots \psi^i (1 - \psi)^{N-i} \right] + \psi \cdot \Delta\zeta_u \quad [4-24]$$

The coefficients $B_{u,ij}$ are to be determined by surface fitting or used as design variables in optimisation process. Figure 4-12 (right) presents wing section lofted with Bernstein distribution with the span wise coefficients $B_{u,j}=[1.0 \quad 1.5 \quad 2.75 \quad 1.5 \quad 0.75 \quad 0.25]$.

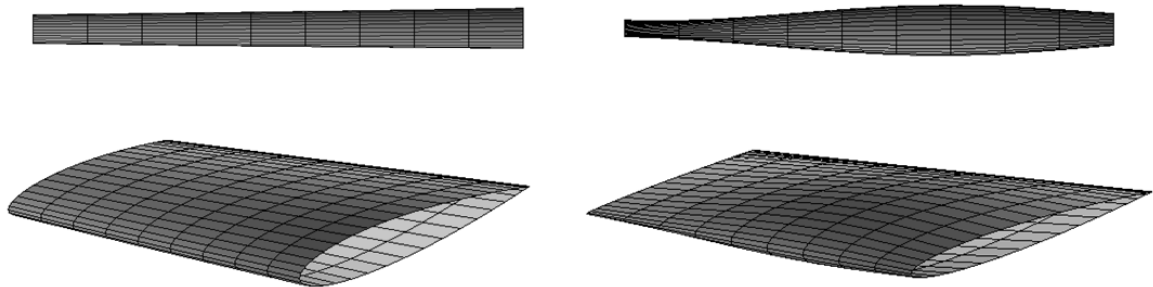


Figure 4-12 Wing section with linear interpolation (left) and Bernstein distribution (right)

4.3.2 Body Section

The body section is generated with the combination

$$x(\psi, \eta) = \psi \cdot length \quad [4-25]$$

$$y(\psi, \eta) = [\psi^{ND1}(1 - \psi)^{ND2}] \cdot (1 - 2\eta) \cdot W(\psi) \quad [4-26]$$

$$z_u(\psi, \eta) = [\psi^{ND1}(1 - \psi)^{ND2}] \cdot [\eta^{NCu(\psi)}(1 - \eta)^{NCu(\psi)}] \cdot H_u(\psi) \quad [4-27]$$

From Equation 4-26, the lower lobe is constructed by replacing $NCu(\psi)$ and $H_u(\psi)$ with $NCl(\psi)$ and $H_l(\psi)$ respectively.

The width ($W(\psi)$), the height ($H(\psi)$), and the cross-section shape coefficient ($NC(\psi)$) are varying in the longitudinal direction. These parameters are lofted with Hermite interpolation (Equation 4-13) with the following set up for different surface shapes:

The nose section (Figure 4-13), which has a close front, is generated from circular cross-section $NCu = NCl = 0.5$ with distribution function $ND1 = 0.5$ and $ND2 = 1.0$.

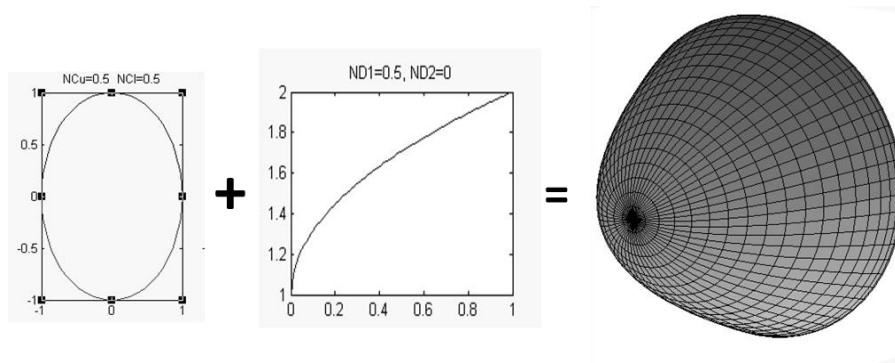


Figure 4-13 Nose surface generation

The cockpit (Figure 4-14) is also defined with circular cross-section ($NCu = NCl = 0.5$). Both sides are open, therefore distribution coefficients: $ND1 = ND2 = 0$. The Hermite interpolation is used with the tangent vector at front section of 1.30, and the tangent vector at the aft is set to 0. This is for smooth blending of the next section.

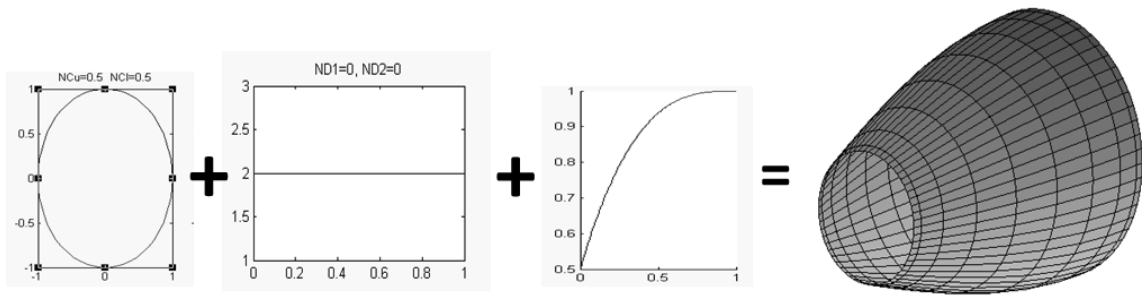


Figure 4-14 Cockpit surface generation

The fuselage section can be defined with varying cross section shapes. For instance, the section connecting the front fuselage to the wing-joining section (Figure 4-15), the front section is defined with circular cross-section ($NCu=NCI=0.5$) while the aft section has circular upper lobe and rectangular lower lobe ($NCu=0.5, NCI=0.05$). Similar to other sections, both sides are open, therefore distribution coefficients: $ND1=ND2=0$. The surface is lofted with Hermite interpolation. For the main fuselage section which has the maximum height and width, the tangent vectors at connections are zero, therefore only the blending terms the Hermite interpolation is used.

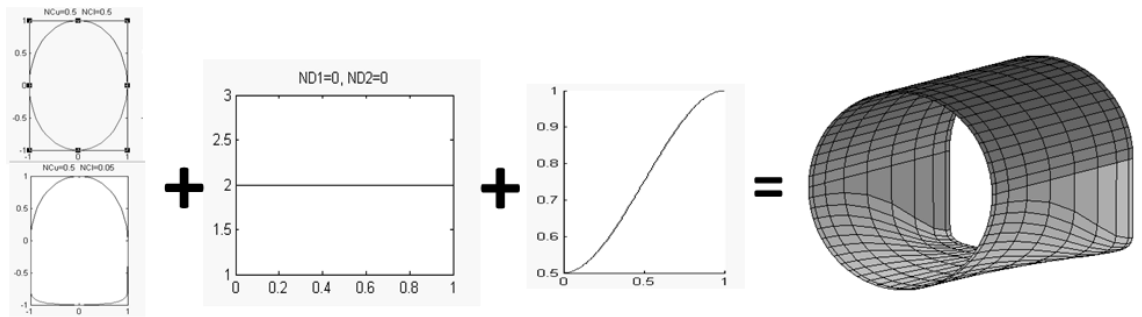


Figure 4-15 Fuselage surface generation

4.3.3 Grid discretisation

The parameterisation is performed in MATLAB (2007) where the surface is generated and plotted on the basis of the grid mesh.

Mesh size determines the accuracy of the surface parameterisation and the aerodynamic analysis. The standard mesh is distributed linearly throughout the xy plane. However, this distribution does not capture the curvature at the leading edge and trailing edge where important flow phenomena take place.

The cosine distribution is set as follow: $\psi = \frac{1}{2} \left(1 - \cos \left(\frac{\pi i}{n} \right) \right), i = 1 \dots n,$

where n = total number of points

Figure 4-16 shows a wing sections and the airfoil cross section generated with the same number of panels, with linear distribution (left) and cosine distribution (right). As clearly shown in the figure, the airfoil with a cosine distribution has a larger number of points describing the leading and trailing edge region.

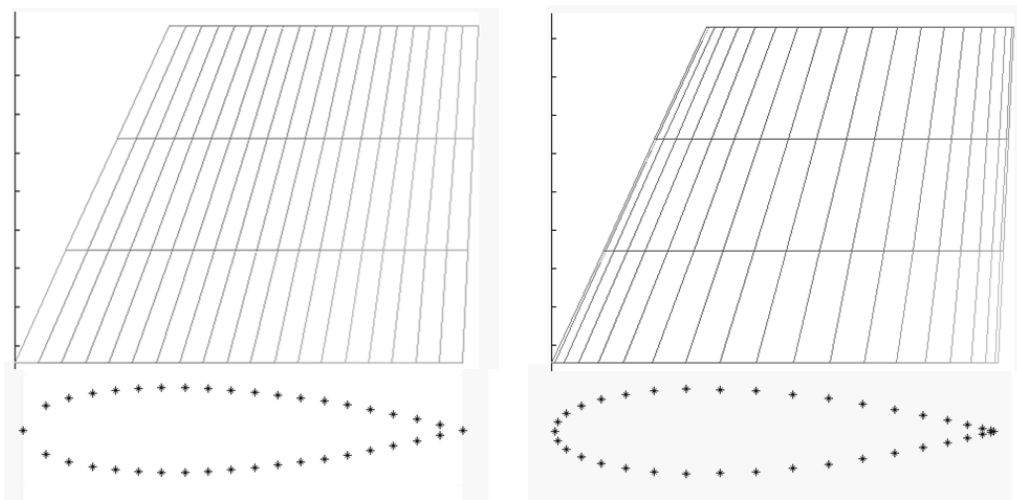


Figure 4-16 Wing surface: linear distribution (left) and cosine distribution (right)

Similar distribution has also been applied to fuselage distribution in the lateral direction (y) which puts more number of the panels in the maximum width region, as shown in Figure 4-17.

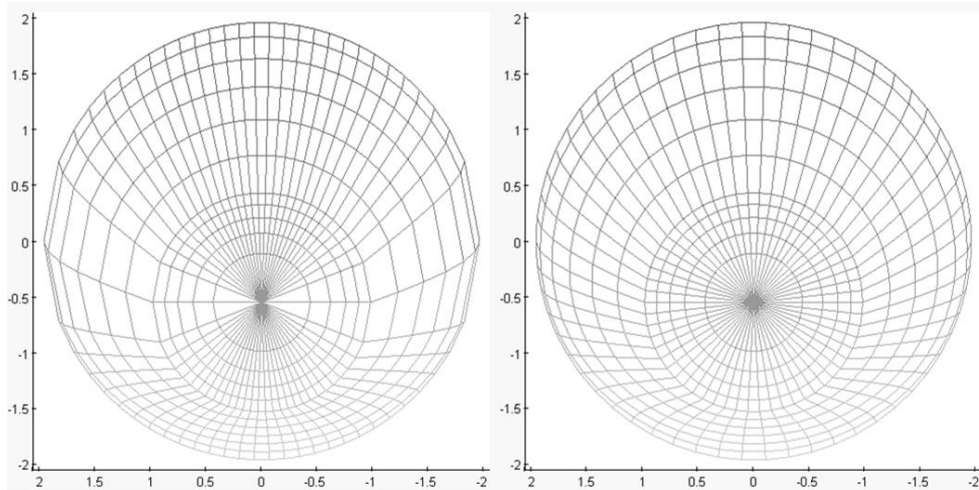


Figure 4-17 Fuselage nose surface: linear distribution (left) and cosine distribution (right)

The optimal number of panels in each dimension is determined by the convergence study during aerodynamic analysis and optimisation process described in Chapter 5.

4.4 Intersection of surfaces

The process of intersecting the two main components i.e. wing or tail components to the fuselage starts at determining the intersecting lines. The algorithm for connecting the wing surface to the fuselage surface has been proposed as follows:

1. Determine the fuselage section to be joined. Set the boundary of the fuselage section corresponding to the wing leading edge and trailing edge (Figure 4-18 left).
2. For all the panel edges of wing surface protruding into the corresponding fuselage surface panels, determine the XY coordinates on the wing panel where it intersects the fuselage surface plane. This is implemented by the *curveintersect* function (Holzt, 2006) written in the MATLAB.
3. For each intersecting point in XY coordinates, interpolate for the Z position on the protruded surface panel. This generates the intersecting line on the fuselage surface.

4. Update the wing root edge and delete the protruded surface inside the fuselage. The wingsec xyz surfaces now contain only actual panels outside the fuselage as shown in Figure 4-18 right. The wetted area of the wing can then be computed from this updated geometry.
5. Update the panel distribution on the fuselage to be coincided with the panel on the intersection line.

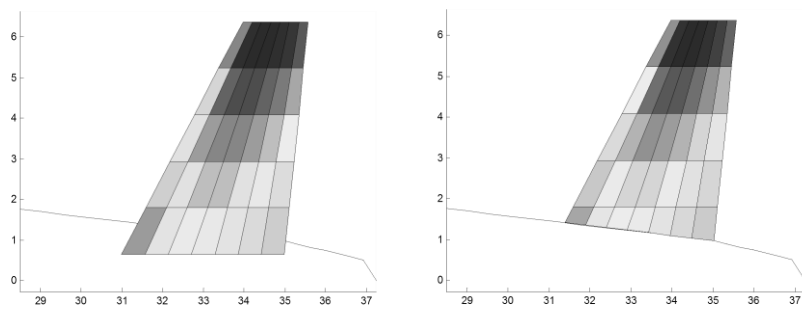


Figure 4-18 Intersecting of the horizontal tail to therear fuselage; pre-intersection(left) and post-intersection (right)

4.5 Surface Area Calculation

Each panel on the surface is a quadrilateral and its surface area is calculated from Varignon's parallelogram theorem (Coxeter and Greitzer, 1967).

The figure formed when the midpoints of the sides of quadrangle area joined in order is a parallelogram, and its area is half that of the quadrangle.

Consider Figure 4-19, the quadrilateral $V_0V_1V_2V_3$ has four midpoints: $M_0, M_1, M_2,$ and M_3 , one on each side. The quadrilateral $M_0M_1M_2M_3$ formed by these four midpoints as its vertices is always a parallelogram.

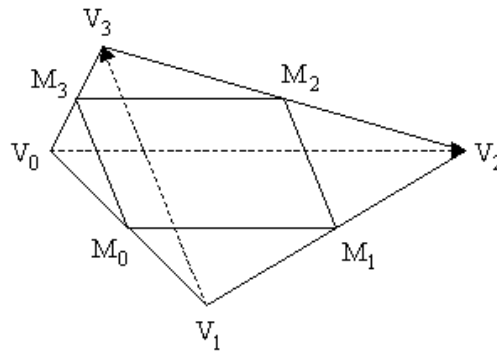


Figure 4-19 Arbitrary quadrilateral and the midpoint parallelogram (Softsurfer, 2012)

The area of the quadrilateral $V_0V_1V_2V_3$ can be computed from the area of the parallelogram $M_0M_1M_2M_3$ as:

$$\begin{aligned}
 A(V_0V_1V_2V_3) &= 2A(M_0M_1M_2M_3) \\
 &= 2|(M_1 - M_0) \times (M_3 - M_0)| \\
 &= \frac{1}{2}|(V_2 - V_0) \times (V_3 - V_1)| \quad [4-28]
 \end{aligned}$$

The total surface area of each component is computed and used for skin friction drag calculation in aerodynamic analysis in Chapter 5.

4.6 Examples of the Parameterised Aircraft Models

The full aircraft configurations are constructed from the main five components: fuselage, wing, horizontal tail, vertical tail, and nacelle. Three types of aircraft have been chosen to demonstrate the capability of the parameterisation to represent wide range of configurations. These are:

1. The generic transport are presented by a twin-engine generic airliners: A320 (Airbus, 2010)
2. Blended Wing-Body: MOB (Smith, 1999).
3. A small unmanned aerial vehicle: UAV-KU4 (In-noi *et al.*, 2004)

4.6.1 Conventional transport

The Airbus A320 was chosen as the example for the conventional transport aircraft. This aircraft features single aisle fuselage with twin engines as shown in Figure 4-20.

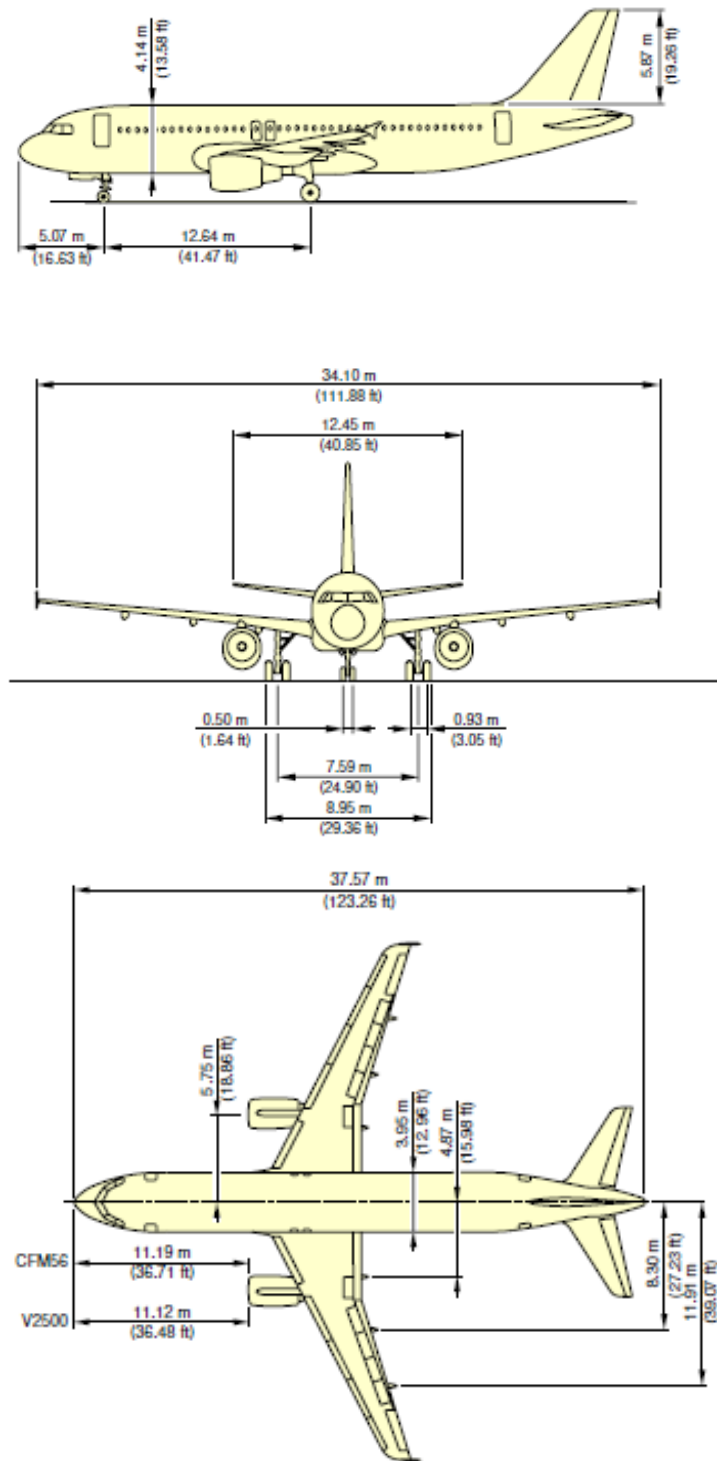


Figure 4-20 Airbus A320 3-View Configuration (Airbus, 2010)

The design parameters for each component are summarised below. The main geometry parameters such as span, chord are taken from Jane's all world aircraft database and the information from the manufacturer (Airbus, 2010). However, since the information about the airfoil section is not available for this aircraft, the airfoil is modelled with unitary set of CST coefficients for demonstration purpose here. A refined example will be presented in Chapter 6 where the airfoil section is used for optimization. Please note that all the dimensions listed in this example are in metric system.

Wing section parameters

section	taper	sweep	twist	dih	XYZ_pos
1	0.6	24.7	- 1.0	3.3	[11.5, 0.0, -1.00]
2	0.3	24.7	- 1.0	3.3	

Wing partition parameters

position	remarks	chord	inc	tuc	span
R	Root	6.5	3.0	0.1	0.0
C	Crank	4.0	2.0	0.1	4.4
T	Tip	1.3	1.0	0.1	17.1

Horizontal tail section parameters

section	taper	sweep	twist	dih	XYZ_pos
1	0.4	27.5	0.0	3.0	[31, 0.6583, 0.9875]

Horizontal tail partition parameters

position	Remarks	chord	inc	tuc	span
R	Root	4.0	0.0	0.1	0.0
T	Tip	0.8	0.0	0.1	10.44

Vertical tail section parameters

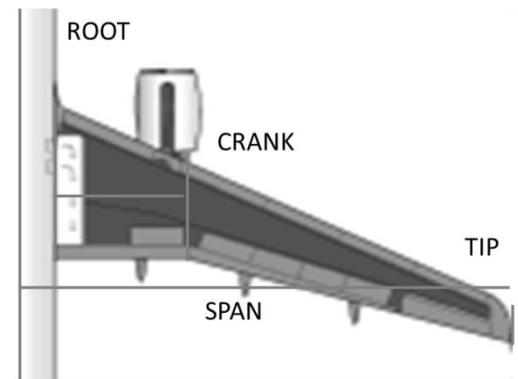
section	taper	sweep	twist	dih	XYZ_pos
1	0.6	34.0	0.0	90	[29.67, 0, 1.90]

Vertical tail partition parameters

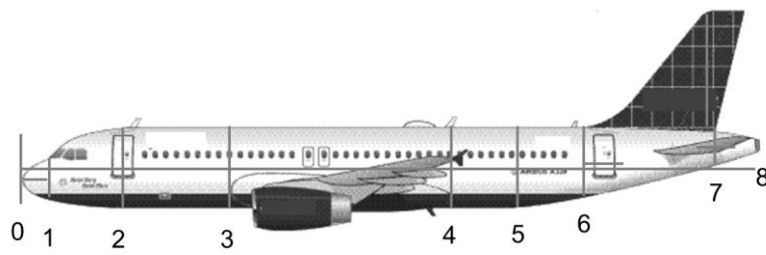
position	remarks	chord	inc	tuc	span
R	Root	5.0	0.0	0.1	0.0
T	Tip	2.1	0.0	0.1	6.26

Nacelle parameters

diameter	1.7
length	4.4



XYZpos	[0 5.4 -2.2]
--------	--------------



Fuselage cross-section parameters

section	remarks	ncu	ncl	x_pos	z_pos	width	height
1	cockpit	0.50	0.50	1.49	-0.55	1.96	1.96
2	fuse front	0.50	0.50	5.22	0.00	3.92	3.92
3	wing box front	0.50	0.05	12.67	0.00	3.92	3.92
4	wing box aft	0.50	0.05	18.63	0.00	3.92	3.92
5	fuse rear	0.50	0.50	25.14	0.00	3.92	3.92
6	tail front	0.40	0.50	28.50	0.20	3.53	3.53
7	tail aft	0.10	0.10	35.02	1.10	1.96	1.76
8	fuse aft	0.50	0.50	37.25	1.33	0.98	0.59

The parameterised aircraft generated with the design parameters defined above is shown in Figure 4-21.

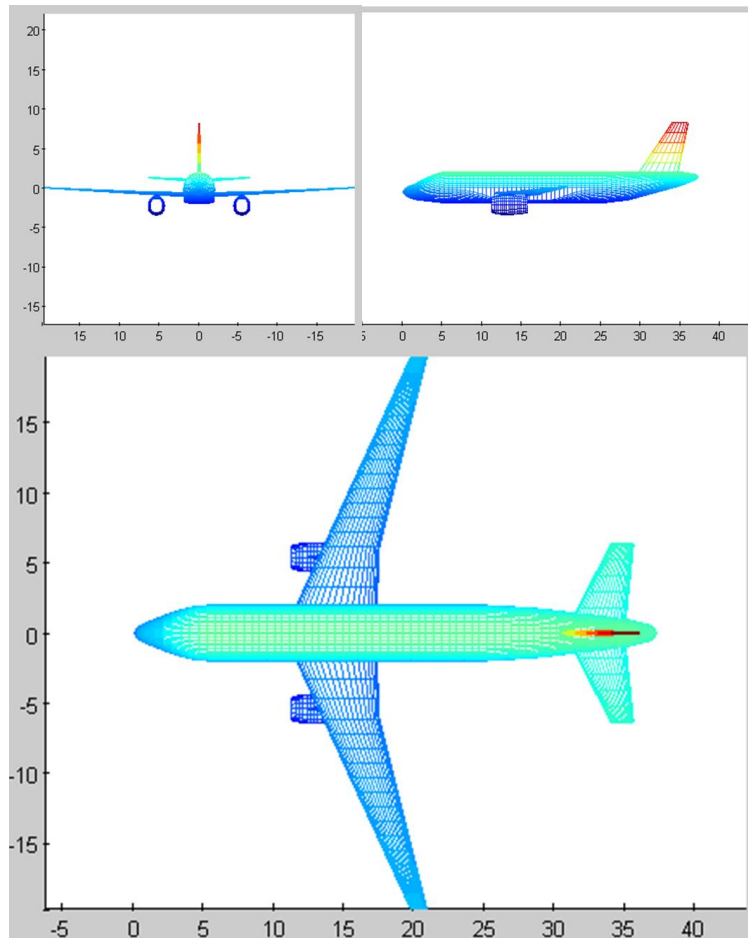


Figure 4-21 Parameterised A320 in 3 View

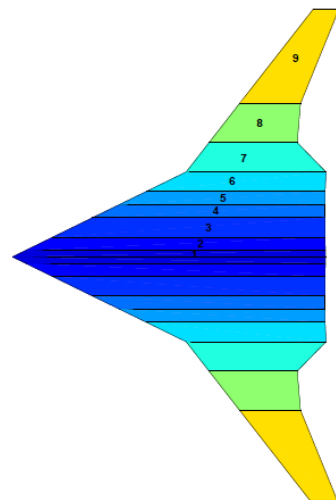
4.6.2 Blended Wing Body

The Blended wing body consists of two symmetrical wings, each defined with multiple sections, joined at the middle to form the fuselage. This design was originally created for the Aerospace Vehicle Design project at Cranfield College of Aeronautics (Smith, 1999). A modified version has later been used as a basic configuration for the MOB (Multi-Objective Blended wing body) project (Morris *et al.*, 2004).

The MOB contains 9 sections. Sections 1 to 6 form the main fuselage, while sections 7 to 9 form the control surface. Each section is described with the following parameters:

Wing section parameters

section	taper	sweep	inc	twist	dih
---------	-------	-------	-----	-------	-----



1	0.957	64	-1.6	0	4.4
2	0.91	64	-1.6	0	4.4
3	0.853	64	-1.6	0.2	4.4
4	0.885	64	-1.4	0.2	4.4
5	0.87	64	-1.2	0.2	4.4
6	0.78	64	-1	0.5	4.4
7	0.63	38	-0.5	0.5	4.4
8	0.69	38	0	1	5.9
9	0.42	38	1	-3.5	7.4

Wing cross-section parameters: Upper surface

section	chord	span	Bu1	Bu2	Bu3	Bu4	Bu5	Bu6
1	48	0	0.2307	0.2916	0.1678	0.255	0.1871	0.208
2	45.9	1	0.2421	0.2829	0.1874	0.2435	0.1918	0.209
3	41.8	3	0.2582	0.2621	0.2254	0.2143	0.203	0.2058
4	35.7	6	0.2611	0.2363	0.2557	0.1778	0.2086	0.2004
5	31.6	8	0.2526	0.2193	0.2591	0.1579	0.2059	0.1937
6	27.5	10	0.2355	0.1981	0.2468	0.1415	0.1949	0.1848
7	21.4	13	0.1714	0.164	0.1602	0.1342	0.1483	0.1574
8	13.5	17.5	0.1364	0.0781	0.1412	0.1133	0.1437	0.1469
9	9.3	23.5	0.121	0.0699	0.1246	0.1018	0.127	0.1309
10	3.906	38	0.121	0.0699	0.1246	0.1018	0.127	0.1309

Lower surface

section	B11	B12	B13	B14	B15	B16
1	0.2143	0.1791	0.1637	0.2386	0.2126	0.2952
2	0.2219	0.1853	0.1951	0.202	0.256	0.2845
3	0.2362	0.2062	0.2141	0.2101	0.2506	0.2803
4	0.2553	0.214	0.265	0.1683	0.268	0.2531
5	0.2521	0.2141	0.2774	0.1442	0.2614	0.2411
6	0.2295	0.2104	0.2534	0.1349	0.2359	0.2136
7	0.1612	0.174	0.1548	0.1179	0.1564	0.1512
8	0.1312	0.0972	0.0898	0.1972	0.0305	-0.0964
9	0.1166	0.0869	0.0789	0.1763	0.0263	-0.0854
10	0.1166	0.0869	0.0789	0.1763	0.0263	-0.0854

The parameterised MOB is presented in Figure 4-22

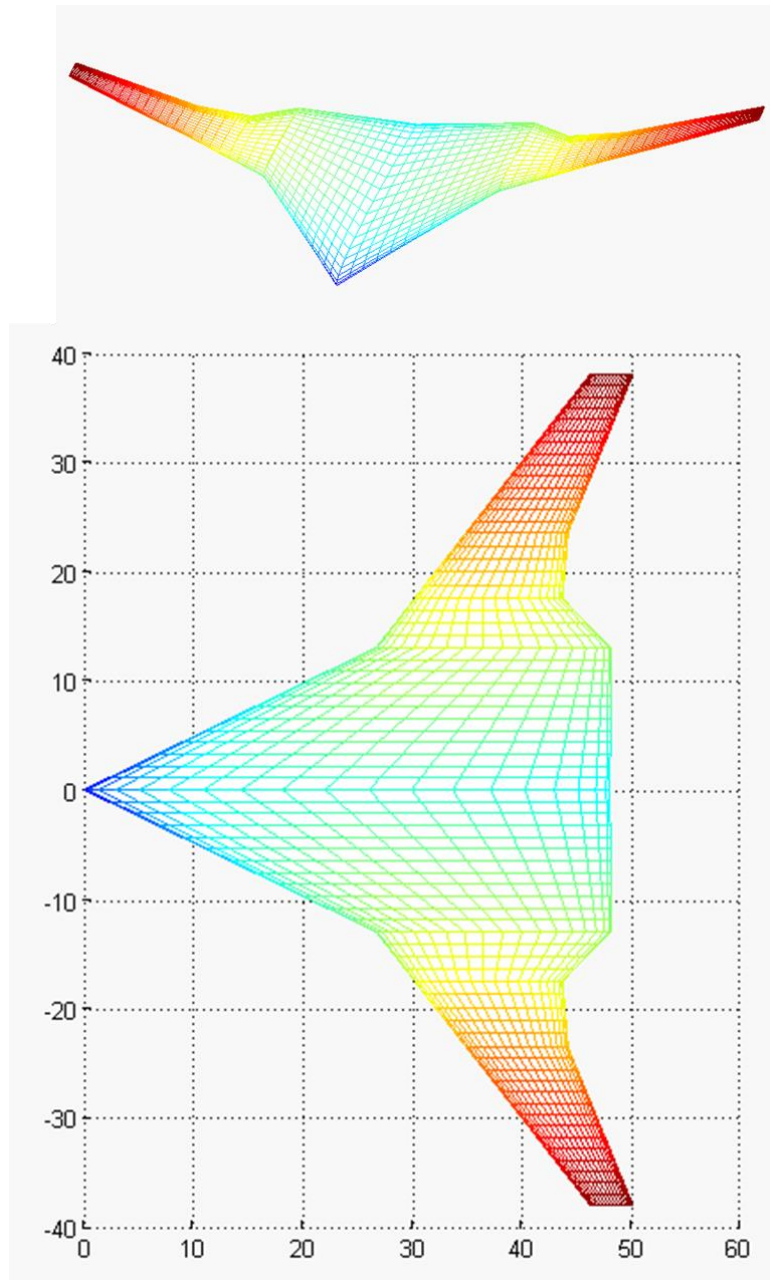


Figure 4-22 Parameterised MOB

4.6.3 Unmanned Aerial Vehicle

The Unmanned Aerial Vehicle: UAV-KU 4 (In-noi *et al.*, 2004), has been developed by the Department of Aerospace Engineering, Kasetsart University, Bangkok, Thailand. This H-tail pusher UAV has a 3.95-meter span with an empty weight of 25 kg plus a 10-kg payload. The half-scale aircraft model has been built for pilot training mission and completed its test flight in 2004. The dimensions of this UAV are presented in Figure 4-23.

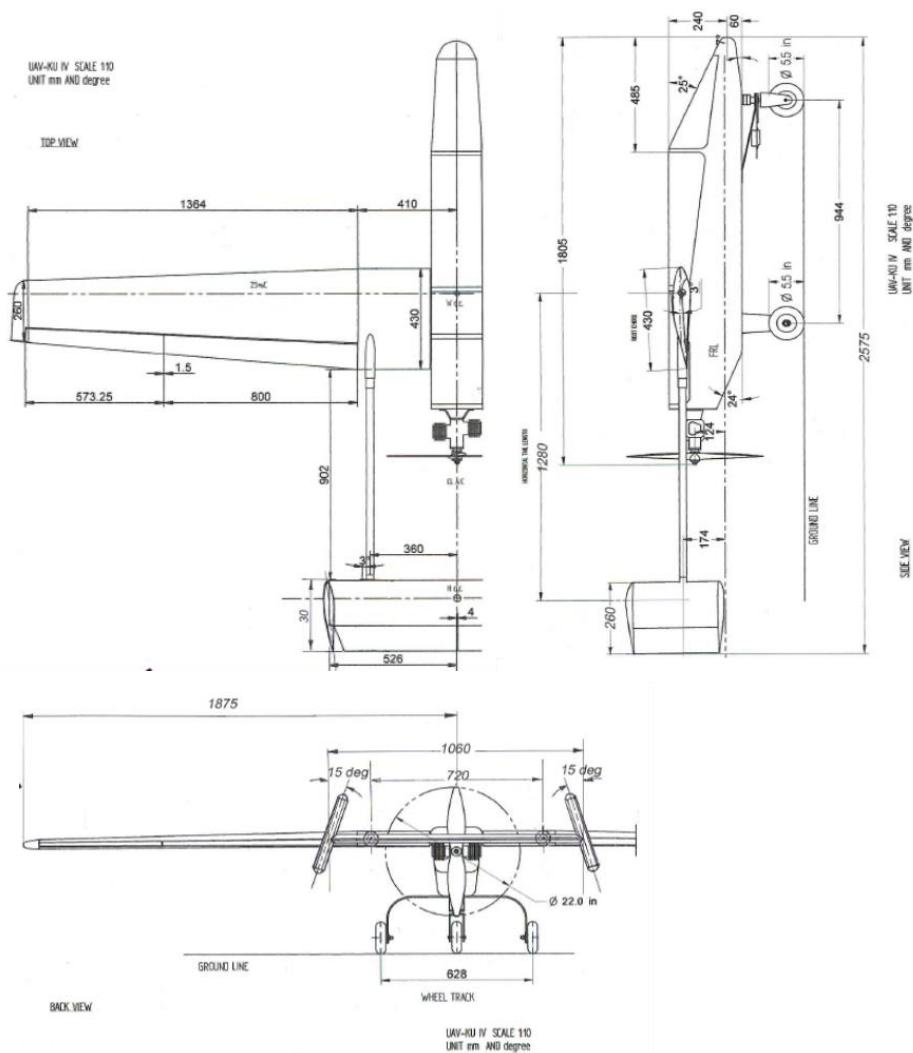


Figure 4-23 Unmanned Aerial Vehicle (In-noi *et al.*, 2004)

The design parameters are summarised below. The airfoil section is modelled with airfoil NACA 63₂-415.

Wing section parameters

section	taper	sweep	twist	dih	XYZ_pos
1	1.0	0	-1.2	1	[0.93, 0.0, 0.174]
2	0.6	0	-1.8	1	-

Wing partition parameters

position	remarks	chord	inc	tuc	span
R	Root	0.43	3.0	0.085	0.0
C	Crank	0.43	1.8	0.085	0.41
T	Tip	0.26	0.0	0.085	1.465

Horizontal tail section parameters

section	taper	sweep	twist	dih	XYZ_pos
1	1.0	0.0	0.0	0.0	[2.264, 0.0, 0.174]

Horizontal tail partition parameters

position	remarks	chord	inc	tuc	span
R	Root	0.3	0.0	0.06	0.0
T	Tip	0.3	0.0	0.06	0.526

Vertical tail section parameters

section	taper	sweep	twist	dih	XYZ_pos
1	1.0	0.0	0.0	-15	[2.264, 0.526, 0.174]

Vertical tail partition parameters

position	remarks	chord	inc	tuc	span
R	Root	0.3	0.0	0.06	0.0
T	Tip	0.3	0.0	0.06	0.36

Fuselage cross-section parameters

section	Remarks	ncu	ncl	x_pos	z_pos	width	height
1	Nose	0.05	0.05	0.07	0.00	0.23	0.18
2	fuse front	0.05	0.05	0.48	0.09	0.23	0.30
3	wingbox front	0.05	0.05	1.04	0.09	0.23	0.30
4	wingbox aft	0.05	0.05	1.24	0.09	0.23	0.30
5	fuse aft	0.05	0.05	1.54	0.15	0.23	0.18

Boom (connecting the empennage to the main wing)

radius	0.017
length	1.047
XYZ-pos	[1.268, 0.360, 0.174]

Airfoil 63₂-415 CST parameters:

Au1 = 0.2126	Al1 = 0.2126
Au2 = 0.1678	Al2 = -0.1565
Au3 = 0.3527	Al3 = -0.0511
Au4 = 0.2240	Al4 = -0.2536
Au5 = 0.2404	Al5 = -0.1048
Au6 = 0.1940	Al6 = -0.0744

The parameterised UAV-KU4 is presented in Figure 4-24.

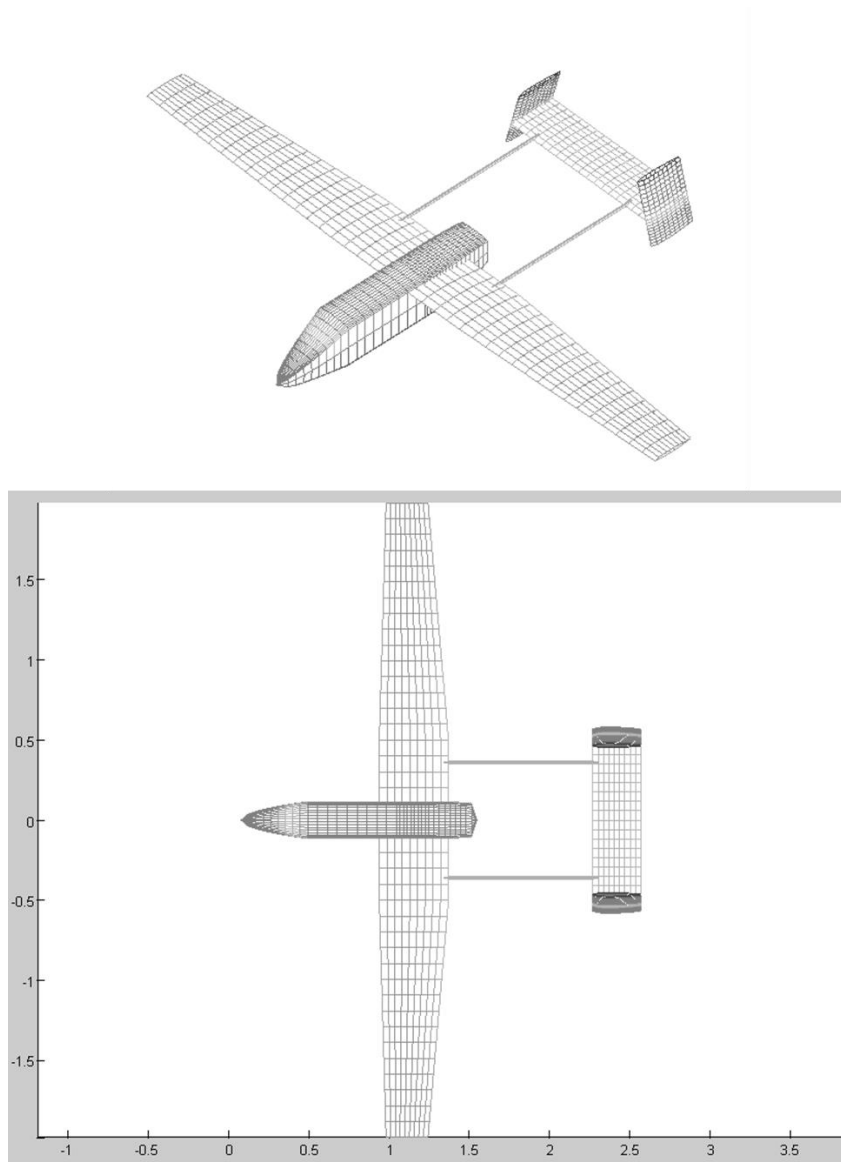


Figure 4-24 Parameterised UAV-KU4

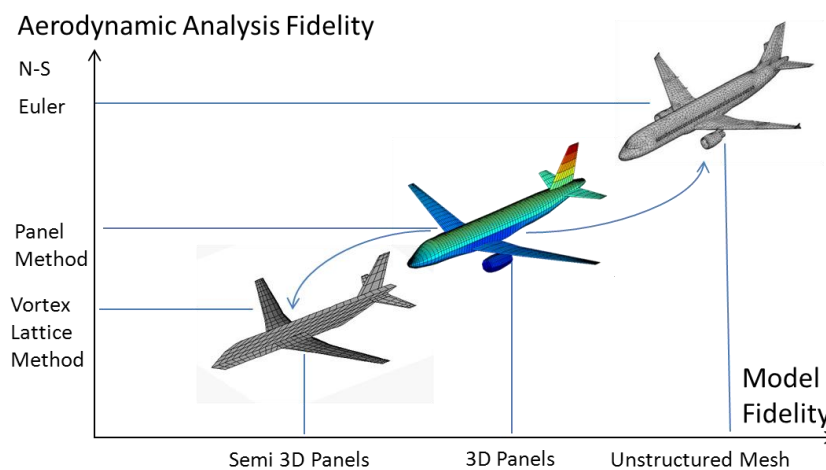
4.7 Summary

Presented in this chapter is a surface generation framework for aircraft design and optimisation. The geometrical model is generated with design parameters describing the geometry and configurations of a level of detailed considered in the conceptual design stage. Kulfan's CST method has been chosen for curve generation due to its simplicity and small number of design parameters required (11 parameters for each airfoil and 6 parameters for each fuselage cross-section). These functions have been combined with a set of selected lofting functions by the authors, resulting in the analytic surfaces describing each component of the aircraft. An algorithm to determine intersections between two intersecting surfaces will be described.

Three configurations of aircraft have been regenerated with the proposed method through the defined set of parameters. The baseline configurations can be directly manipulated through this set of parameters. The parameterised models are now ready for the aerodynamics analysis and optimisation, which will be presented in chapter 5 and 6, respectively.

5 Aerodynamic Analysis and Validation

This chapter presents the aerodynamics analysis of the proposed three-dimensional aircraft parameterisation. Two levels of fidelity are considered: the level of detail of the geometrical model descriptions and the fidelity of the aerodynamic analysis tools. Figure 5-1 presents the mapping of CFD tools to the different levels of model descriptions. In low to medium fidelity analysis, the solvers are based on potential flow i.e. Vortex Lattice Method (VLM) and Panel Method. These solvers calculate the flow forces by putting vortices on the discretised geometrical surfaces (panels). Both methods provide satisfactory induced drag results, but are limited to the inviscid incompressible flow region. The main difference between VLM and Panel Method is that the VLM does not take geometrical thickness into calculation, and the model used in VLM is therefore referred to as a “semi-3D” model.



**Figure 5-1 CFD fidelity with corresponding level of Geometry descriptions
(Adapted from Rizzi *et al*, 2010, CAD model from WIPD-Pro CAD 2010)**

In high fidelity analysis, the solvers are based on Euler and Navier-Stokes equations which use the detailed geometrical surfaces with unstructured mesh. This requires both time and expertise in preparing the model. The proposed geometrical model in this research is positioned in the medium fidelity level which is expected to contain sufficient geometrical details for the medium fidelity analysis (Panel Method). It also provides flexibility to (1) flatten the model to semi-3D model for a Vortex Lattice Method or (2) export model in the standard geometry format for a high fidelity analysis.

The work is structured in four parts:

1. Convergence study on the geometry to determine an optimal number of grids required for aerodynamics analysis
2. Validation of the wetted surface area calculation of the proposed model
3. Solving for the lift and drag coefficients
4. Validation of these coefficients with CFD results or wind-tunnel data

5.1 Aerodynamic Coefficients

The aerodynamic coefficients determine the performance of the aircraft. The two main coefficients are the total lift and drag coefficients which together are combined as lift to drag ratio. This parameter is used to determine the optimal lift coefficients in the conceptual design stage.

5.1.1 Lift

In an empirical calculation, the lift coefficient is a function of the aircraft total weight at a certain position in flight e.g. take-off, climb, landing. Whereas in a potential based flow solver, the lift is calculated through flight condition specified by the flight altitude and speed with the angle of attack at each position, where the induced drag is also calculated at the same time. The formulation of potential flow is discussed in section 5.2.

5.1.2 Drag

The drag of a blunt body in flow contains two main components: induced drag and parasite drag, as presented in Figure 5-2. The induced drag or drag due to lift is

generated from vorticity shed into wake. The parasite drag composed of three components: friction/form drag, interference drag, and wave drag.

Total Drag							
Induced	Parasite						
Due to lift generated vorticity into wake	Friction/Form			Interference (due to intersection)		Wave (due to shock waves)	
	Skin friction	Form/Pressure	Additional profile drag due to lift	Due to intersection	Due to lift	Due to volume	Due to lift

Figure 5-2 Drag Component (Gur *et al.*, 2010)

The friction/form drag is generated due to viscosity. The pressure drag or form drag is generated due to compressibility effect. The airfoil at lift also generates the additional profile drag but is relatively small comparing to the first two components. The interference drag is generated due to the intersection between main components such as wing and fuselage. The wave drag is generated due to shock waves at critical Mach number.

The induced drag is an important component in total drag since it contributes about half of the entire vehicle drag (Gur *et al.*, 2010). There are several methods to calculate the induced drag coefficient, for instance: Trefftz plane analysis, Prandtl’s lifting line, Vortex Lattice Method, Panel Method. All methods mentioned are based on linear equations which limit their validity to the flow in the low speed region. The common approach is to use Weissinger non-linear lifting line method with Prandtl-Glauert correction for 2D lift-slope.

Since a potential flow solver does not take viscosity effect into calculation, the friction drag is modelled with flat-plate skin friction model coupled with form-factor models.

The actual wetted area is calculated through the generated aircraft geometry. The other two components, wave drag and interference drag, cannot be modelled through the low-medium fidelity analysis. In order to achieve more accurate total drag value, these two components are calculated through empirical formulae (Gur *et al.*, 2010). The wave

drag is interpolated between Lock's fourth order law and Korn equation. The interference drag is interpolated between the CFD response surface for thin wing ($t/c < 0.075$) and the Hoerner model for thick wing ($t/c > 0.4$).

5.2 Potential Flow

The potential flow follows the fundamentals of inviscid, incompressible flow. The formulation in this section follows the work of Filkovic (2009).

From the law of conservation of mass:

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho v_i)}{\partial x_i} = 0 \quad [5.1]$$

For inviscid flow, the density (ρ) is constant, equation 5.1 becomes:

$$\frac{\partial v_i}{\partial x_i} = 0 \quad [5.2]$$

where v_i is scalar speed potential, given as a derivative of scalar potential ϕ with respect to coordinate x_i : $v_i = \frac{\partial \phi}{\partial x_i}$

Equations 5.2 now becomes

$$\frac{\partial^2 \phi}{\partial x_i^2} = 0 \quad [5.3]$$

which is called the Laplace equation.

From the law of conservation of momentum:

$$\frac{\partial(\rho v_i)}{\partial t} + \frac{\partial(\rho v_j v_i)}{\partial x_j} = \rho f_i + \frac{\partial \sigma_{ji}}{\partial x_j} \quad [5.4]$$

For irrotational and inviscid flow for steady flow:

$$\frac{v^2}{2} + \frac{p}{\rho} + gz = const \quad [5.5]$$

which is called the Euler-Bernoulli equation.

By solving equation 5.3 and 5.5, the velocity field and the pressure field can be determined. The total force and moment on the body is calculated from:

$$F_i = - \int_S p n_i dS \quad [5.6]$$

where p is the pressure,

n_k is the normal unit vector,

S is the surface area

subscript i denotes the direction X (longitudinal), Y (lateral), or Z (normal to XY plane)

The force and moment coefficients are calculated as a function of Force (F_i) in each direction as follow:

$$C_X = \frac{F_X}{q S_{ref}} \quad [5.7]$$

$$C_Y = \frac{F_Y}{q S_{ref}} \quad [5.8]$$

$$C_Z = \frac{F_Z}{q S_{ref}} \quad [5.9]$$

where q is dynamic reference pressure, S_{ref} is wing reference area

The lift and drag coefficients (C_L , C_D) are calculated through transformation of the coordinate system with respect to the angle of attack (α):

$$\begin{bmatrix} C_L \\ C_D \end{bmatrix} = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} C_Z \\ C_X \end{bmatrix} \quad [5.10]$$

In this research, a non-commercial potential flow solver TORNADO (Melin, 2001) based on the vortex lattice method (VLM) is chosen for aerodynamic analysis. It calculates aerodynamics variables such as lift, drag, bending moment and shear forces along with their coefficients. The vortex horseshoes and collocation point are placed on each panel of the parameterized model, as shown in Figure 5-3.

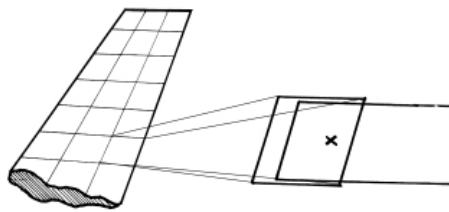


Figure 5-3 The vortex horseshoe position and the collocation point (Melin, 2010)

The VLM method is limited to thin airfoil theory which means that the thickness is not taken into calculation but the method uses the camber line of the airfoil profile to calculate the normal vectors of each panel. The wing geometry is flattened through the in to form semi 3d panels. The CST airfoil was used to generate the camber line. The wake used by the solver can be either fixed wake (standard vortex lattice) or free stream following wake (Tornado method) of which the wake is influenced by the angles of attack and side slip.

For validation, the results are compared with NASA TN D-5971 report (Capone, 1970). Figure 5-4 and Figure 5-5 present the drag polar calculated by the VLM method for the NASA (TN D-5971) and the A320, respectively. The calculation was performed from low Mach number to critical Mach number and shows that for both configurations the drag polars are shifted with respect to the Mach number, but all the induced drag polars follow the same curve. Therefore the induced drag can be calculated at a single value of Mach number, which is the design Mach number in this research.

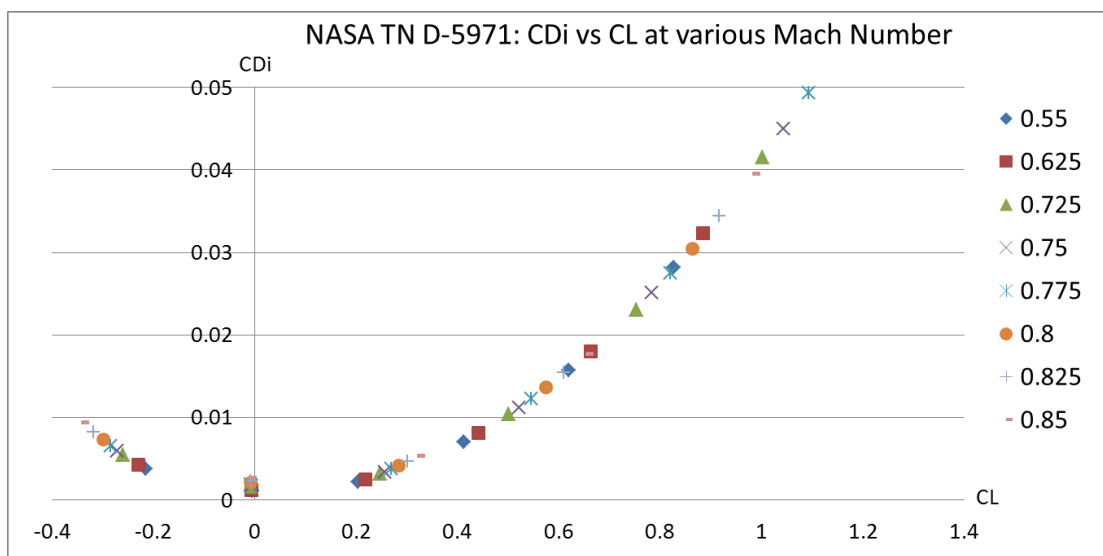


Figure 5-4 TN D-5971 in Vortex Lattice analysis with Mach numbers from 0.55 to 0.85

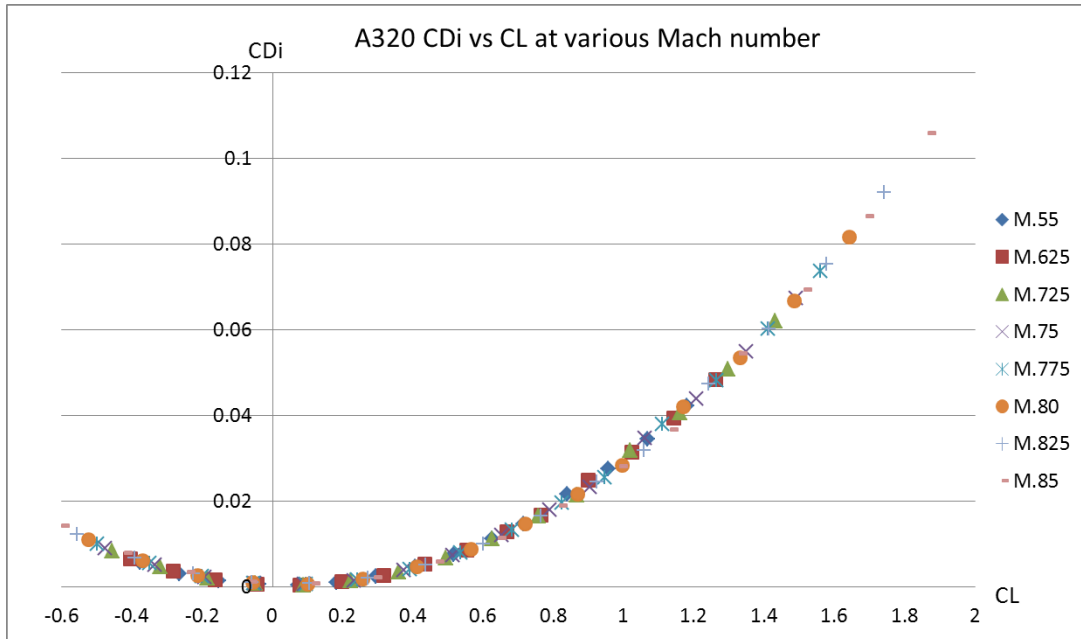


Figure 5-5 A320 in Vortex Lattice analysis with Mach numbers from 0.55 to 0.85

5.3 Friction/ Form Drag Model

Even though the vortex lattice method and the panel method yield acceptable induced drag, the main limitation is the lack of viscosity effect consideration. Models for skin friction drag prediction are therefore included to improve the drag prediction. These methods are based on the theory of Friction/Form drag which is composed of two components:

1. Semi-empirical flat plate skin friction models
2. Form-factor model

The form drag (sometimes referred to as pressure drag) is influenced by the frontal area of each section. The friction drag is due to the movement of the air on the surface which generates boundary layers.

The skin friction drag is a product of skin friction coefficient and the ratio between component wetted area and the reference area. The pressure drag is the product of the form factor and the skin friction coefficient.

The complete formula for friction and form drag is:

$$C_{D,F} = C_F FF \frac{S_{wet}}{S_{ref}} \quad [5.11]$$

where C_F is a flat-plate skin-friction coefficient

FF is a form factor of the component

S_{wet} and S_{ref} are the wetted area and the reference area, respectively

The calculation uses the standard flat plate skin friction theory and compressibility effect on skin friction. The coefficients are calculated at two flow conditions: laminar and turbulent. The detailed formulation of each coefficient can be found in (Mason, 2011). The composite formula is then used to combine the two coefficients to compute the total skin friction drag coefficients.

5.3.1 Composite formula

The transition flow between the laminar and turbulent flow can be modelled using the Schlichting's formula (Schlichting, 1979). For the given transition position, $\frac{x_c}{L}$, the composite Reynolds number is calculated through

$$Re_c = \left(\frac{x_c}{L}\right) Re_L \quad [5.12]$$

The laminar flow skin friction coefficient is then computed at Re_c and the turbulent skin friction coefficients are computed at both Re_c and Re_L . The total skin friction coefficient is calculated through:

$$C_F = C_{F,Turb}(Re_L) - \left(\frac{x_c}{L}\right) [C_{F,Turb}(Re_c) - C_{F,Lam}(Re_c)] \quad [5.13]$$

The determination of the transition position is difficult at low-medium fidelity analysis. For the cases with fixed transition, such as the wind tunnel models, the transition position $\frac{x_c}{L}$ should be set to the appropriate location; however, for commercial aircraft, the value $\frac{x_c}{L}$ can be approximated quite accurately as zero (March, 2008).

5.3.2 Form factor

In order to model the effect of thickness, the form factor is included. There are two types of the form factor: of the wing and of the body of revolution. Gue *et al.* (2010) has compared the formulae to calculate form factor from various authors. These are in the similar equations with different coefficients. For wing or planar surface, the form factor depends on the thickness to chord ratio, $\frac{t}{c}$ and follows the equation:

$$FF_{wing} = 1 + 2.7 \frac{t}{c} + 100 \left(\frac{t}{c}\right)^4 \quad [5.14]$$

For the body shape, the form factor is a function of the ratio between diameter and length, $\frac{d}{l}$ which is the inverse of the fineness ratio, $\frac{l}{d}$

$$FF_{body} = 1 + 1.5 \left(\frac{d}{l}\right)^{1.5} + 7 \left(\frac{d}{l}\right)^3 \quad [5.15]$$

5.3.3 Wetted area calculation

In the sizing process, a wetted area is usually calculated through empirical models. This limits the accuracy as the design moves away from the conventional configuration.

In order to demonstrate this, the calculated wetted areas from an industrial test case (Guenov *et al.*, 2010) and the actual surface areas from the parameterised model are compared against the data provided by NASA TN D-5971 wind tunnel test (Capone, 1970). The two models are shown in Figure 5-6.

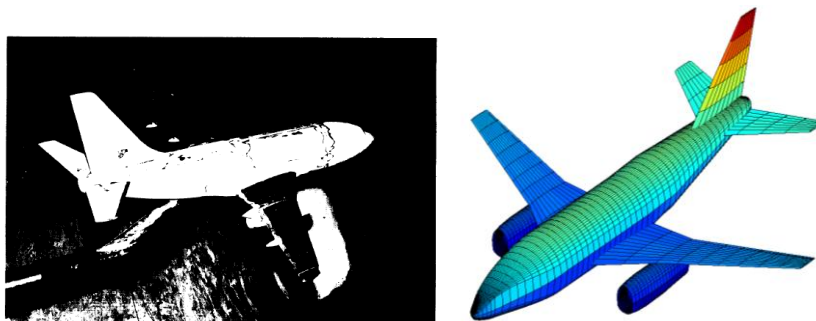


Figure 5-6 TN D-5971 Wind Tunnel Model (left) (Capone, 1970) and Parameterised Model (right)

The wetted area of each component and the corresponding errors are presented in Table 5.1. Note that all surface areas are calculated based on the full scale model.

Table 5.1 Calculated Wetted Area [m²]

Components	TN D-5971	Empirical	% Error	Parameterised	% Error
wing	149.16	147.04	-1.4%	149.67	0.3%
fuselage	266.85	280.30	5.0%	269.39	0.9%
horizontal tail	47.36	51.00	7.7%	48.20	1.8%
vertical tail	42.96	48.11	12.0%	42.22	-1.7%
nacelle	38.48	30.60	-20.5%	37.76	-1.9%

As seen from the table, the wetted area for each component calculated through the parameterised model has an error within 2 per cent which is more accurate than the empirical model. The errors in the empirical model are due to the following reasons:

- The fuselage wetted area is calculated from the defined length and diameter. Since nose and tail section is not actually modelled, the approximation of the wetted area from the empirical method yields higher error than from the parameterised one.
- The wetted area of the wing is calculated from the reference wing area without any information concerning wing thicknesses or airfoil curvatures. This yields a difference between the calculated and the actual surface area. Similar to the wing, the wetted areas of the horizontal and vertical tail are also calculated from the reference area. However, since the reference area of each component is approximated based on the wing reference area and fuselage length rather than the actual geometrical design parameters. This yields a higher error comparing with the wing wetted area.
- The nacelle area is the function of its diameter only. For this chosen design, the nacelle is longer than the usual conventional design therefore the error is more significant.

By modelling the actual surface yields more accurate surface wetted area for each component which is an important parameter in friction/form drag calculation. The friction model (Mason, 2011) is used to calculate the friction drag coefficient. The results are validated with the wind-tunnel results, as presented in Table 5.2. The friction

model demonstrates good agreement with the wind tunnel results at low Mach number (0.1 per cent error at Mach 0.55). The error grows as the flow moves to the transonic region. This is mainly due to an error in transition positions which were determined under the assumption of transition position rather than the exact location which has to be determined through CFD analysis.

Table 5.2 Friction Drag Model Validation

Mach Number	NASA TN D-5971	FRICTION model	Nominal Error	% Error
0.550	0.02093	0.02096	0.00003	0.1%
0.625	0.02055	0.02062	0.00007	0.3%
0.725	0.02016	0.02026	0.00010	0.5%
0.775	0.01996	0.02007	0.00011	0.6%
0.850	0.01964	0.01977	0.00013	0.7%

For an unconventional design such as Blended Wing Body, the transition position can be estimated by the interpolation between the results from the F-14 variable sweep transition flight experiment and the wind-tunnel test data from NASA TN D-338. The detailed work can be found in Leifsson, *et al.* (2005).

5.4 Convergence Study

The convergence test was performed in order to determine the optimal number of panels on each direction for the configuration being studied. This section follows the mesh density study by Filkovic (2009), which was performed on a wing with a single sweep and constant taper ratio. The analysis is performed on a series of number of panels chord wise and span wise. The selected three sets of grid density are: “coarse” (10 panels span wise, 24 panels chord wise), “medium” (20 panels span wise, 40 panels chord wise), and “fine” (40 panels span wise, 60 panels chord wise). The corresponding computation time increases exponentially from 1 second to 1.8 seconds and 15 seconds for the course, optimal and fine mesh, respectively.

In this section, the Multi-Objective Blended wing body (Morris *et al.*, 2004) has been used to perform grid convergence study. The numbers of panels chord wise and span wise are set as follows:

The number of panels along chord wise is constant throughout the wing and varies from 4, 8, 12, 16, 20, 24, 28, and 32. When the number of panel reaches 36, the solver is out of memory and can no longer solve for the aerodynamics coefficients.

From Figure 5-7, the lift and drag coefficients are presented according to the number of panel chord wise. Taking the wind-tunnel exactness criteria of 0.0001 (Kulfan, 2007), convergence is achieved when the number of panels per chord reaches 16, where the changes fall within the required tolerance of 0.0001.

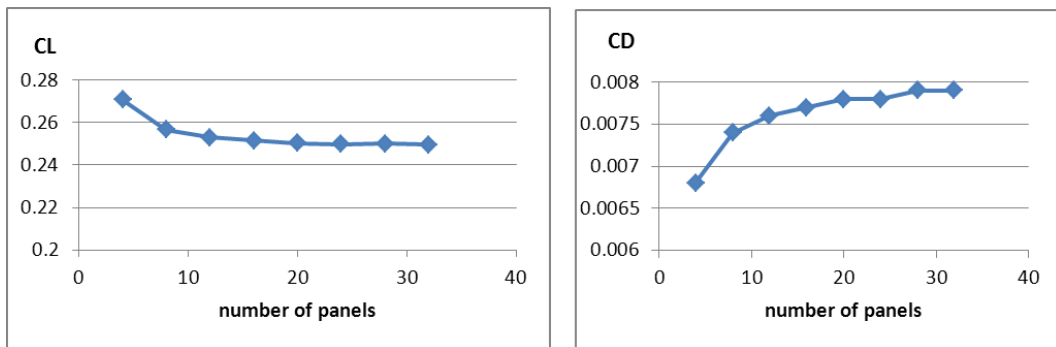


Figure 5-7 Lift and Drag coefficients with respect to number of panels (chord wise)

The computational times corresponding to the number of panels are presented in Figure 5-8. The computational time significantly increases as the number of panel increases. The selected value of the panels is 16, which takes 13.3 seconds per analysis.

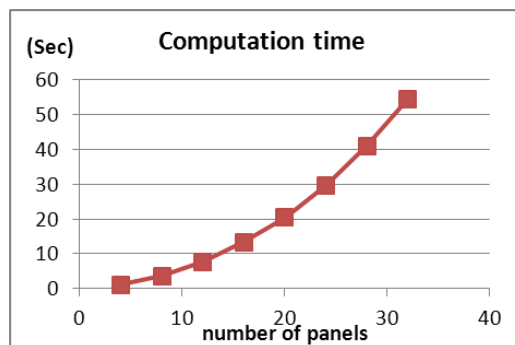


Figure 5-8 Computation time with respect to number of panels (chord wise)

The number of the panels span wise is normalised according to the span of each section. For this configuration, the spans of section are 13.0, 4.5, 6.0, and 14.5 (starting from inboard to outboard). The number of panels on each section is summarised in Table 5.3.

Table 5.3 Number of span wise panels on each section

Case	ny ₁	ny ₂	ny ₃	ny ₄
1	3	1	1	3
2	5	2	2	6
3	8	3	4	9
4	10	4	5	12
5	13	5	6	15
section span (m)	13.0	4.5	6.0	14.5

The number of span wise panels is proportional to the span of each section. First, the panel numbers are defined by rounding the span length, resulting in case 5 in Table 5.3. Once the number of panel is increased the solver is out of memory, therefore case 5 is set as the maximum number of panels. The other sets are derived from this set by proportionally reducing number of panels by 20% at each step until the minimum number, 1 is reached (case 1).

The lift and drag coefficients are presented in Figure 5-9. Both coefficients start to converge at case 3, where the change in induced drag coefficient between the two cases is reduced to 0.0002.

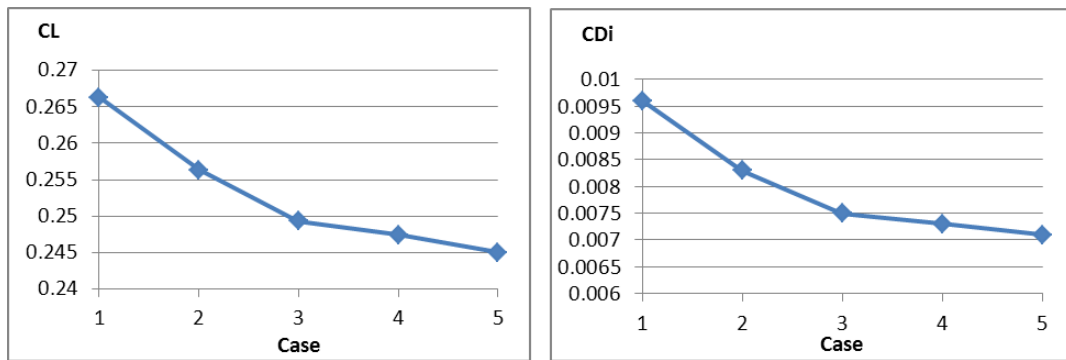


Figure 5-9 Lift and Drag coefficients with respect to the number of panel (span wise)

The computation time for each case is presented in Figure 5-10. For the chosen case (case 3), the computation time is 17.20 seconds per run.

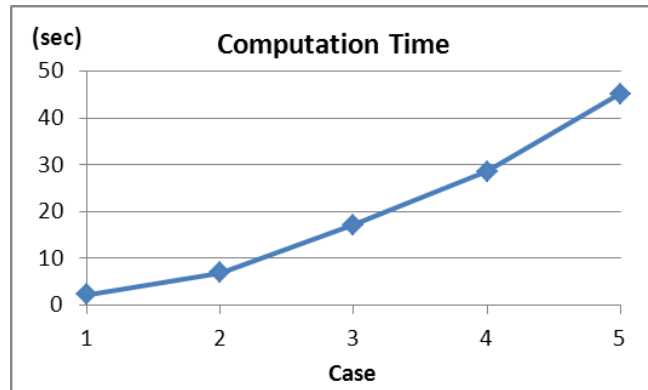


Figure 5-10 Computation time with respect to the number of panel (span wise)

5.5 Aerodynamic Analysis Models from the Selected Test Cases

Two test cases developed for aircraft conceptual design have been studied in this research. Each test case contains a set of models in various disciplines, sufficient for aircraft performance and sizing study. In this section, only the models which are related to aerodynamic analysis are discussed as follows:

5.5.1 USMAC

The Ultra Simplified Model of Aircraft (USMAC) (see appendix B) is an aircraft sizing test case provided by an industrial partner and has been extensively used by the Engineering Design Group at Cranfield University (Guenov *et al.*, 2010). It contains 97 models and 125 variables. Most of the models are based on empirical or statistical data.

The USMAC models which are involved in aerodynamic analysis are presented in and can be separated into four sub-modules: Lift, Friction drag, Induced drag, and Pressure drag. In the test case, all the models are calculated at three flight conditions: climb, cruise, and static thrust. For simplification, only the models for cruise condition are presented here.

The lift coefficient is calculated from mass, speed, gravitational acceleration, Mach number and pressure at each flight condition. The induced drag models are based on the empirical relationship between the lift coefficient and the wing aspect ratio. The friction drag is calculated based on the considered flight condition and the wetted areas, while the pressure drag is calculated through the flight Mach number and the characteristic Mach number.

5.5.2 FLOPS

Flight Optimisation System: FLOPS (McCullers, 2011) is a computational program for multidisciplinary aircraft design in conceptual and preliminary design stage. The analysis can be separated into nine primary modules: 1) weights, 2) aerodynamics, 3) engine cycle analysis, 4) propulsion data scaling and interpolation, 5) mission performance, 6) take-off and landing, 7) noise footprint, 8) cost analysis, and 9) programme control.

The aerodynamic module uses Empirical Drag Estimation Technique: EDET (Feagin and Morrison, 1978) as summarised in Table B.2 in appendix B. These models can be divided into 6 sub-groups: Lift, Mach number, induced drag, friction drag, pressure drag, and compressible drag.

The friction drag model follows the standard skin friction calculation method discussed in section 5.3. The inputs are: skin friction coefficients, wetted area of each component, form factors, finess ratio, and wing reference area. The friction drag coefficients are calculated at a vector of predefined Mach numbers. This vector is defined by FLOPS, usually in the range between 0.2 - 0.85, with a refined distribution in the transonic region at Mach number between 0.75 - 0.80.

The pressure drag coefficients are calculated at the defined set of lift coefficients and Mach numbers which together form a matrix of drag coefficients. Each row of these drag polars varies with the lift coefficients and each column varies with the Mach numbers.

5.6 Results and Analysis

Three representative aircraft configurations, conventional transport, blended wing body, unmanned aerial vehicle, have been used for aerodynamic analysis.

5.6.1 Conventional

The A320 geometrical model is analysed with USMAC and FLOPS aerodynamic models and VLM. The lift and induced drag polar shows agreement between FLOPS and VLM as shown in Figure 5-11. Both methods yield close results in the low lift coefficient region. However, the drag polar calculated with VLM shows small drop at lift coefficient around 0.65 due to the compressibility effect.

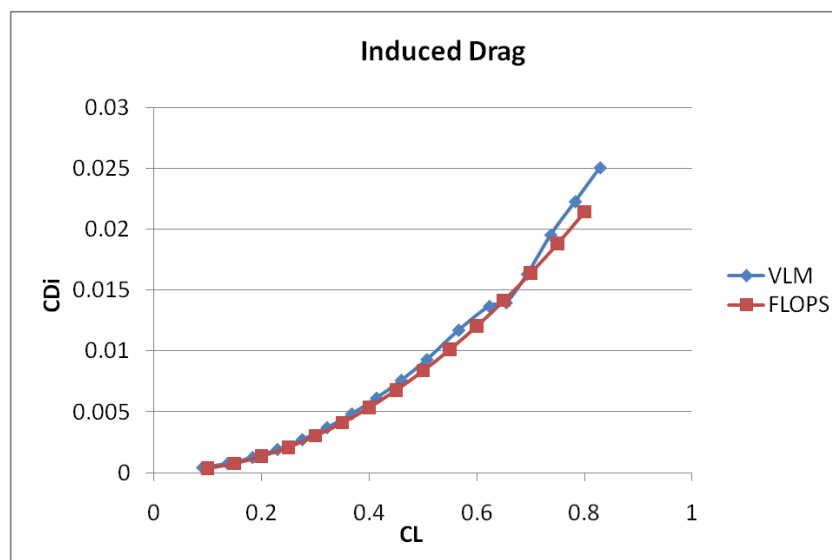


Figure 5-11 A320 lift and induced drag coefficients with VLM and FLOPS

The lift and drag coefficients from the USMAC, FLOPS, and VLM are presented in Table 5.4, the VLM result is integrated with Mason's drag model (Gur *et al.*, 2010) for comparison purpose. In this work, the drag polar is calculated at the design Mach number of 0.797 with the lift coefficient of 0.5411. As seen from, this table, all methods show good agreement in each component of drag.

Table 5.4 Aerodynamic Coefficients Comparison

Coefficients	Remarks	USMAC	FLOPS	VLM +
--------------	---------	-------	-------	-------

				Mason
CL	lift	0.5411	0.5411	0.5411
CDi	induced drag	0.0098	0.0098	0.0108
CDf	friction drag	0.0170	0.0174	0.0170
CDp	pressure drag	0.0025	0.0025	0.0025
CDd	compress drag	-	0.0015	0.0013
CD	total drag	0.0293	0.03135	0.03176

The friction drag model contributes approximately half of the total drag; therefore, it is crucial to determine the friction drag accurately. The friction drag models used in USMAC and FLOPS are based on the same formulation used in the friction model by Mason (Section 5.3) and therefore yield similar results.

The pressure drag and compressibility drag are calculated based on the empirical models to form the total drag. In the USMAC test case, compressibility drag model is not included; therefore the total drag coefficient from USMAC aerodynamic models is slightly lower than the other methods. The results from FLOPS and VLM + Mason methods are both within 1%. The VLM has slightly higher due to a higher value of induced drag.

The drag polar from FLOPS' aerodynamic models and VLM with friction drag model at transition Mach numbers from 0.7 to 0.825 are presented in Figure 5-12. The results from VLM show good agreement with FLOPS for all Mach numbers considered. This result is expected since the FLOPS' aerodynamic models are validated on the conventional configuration and therefore should yield very accurate solution. The integration of VLM will become beneficial as the design moves toward unconventional configuration, which is presented in the next section.

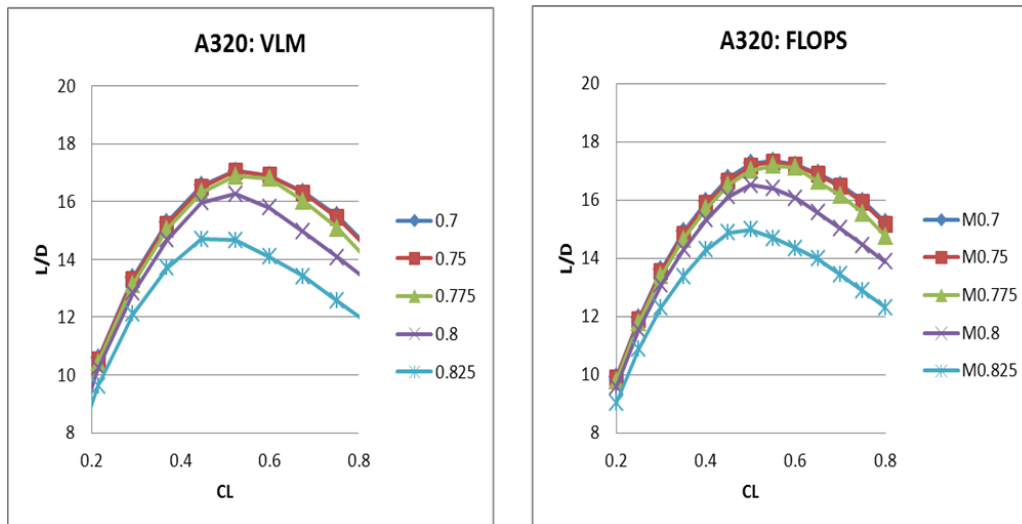


Figure 5-12 A320 Drag Polar at various Mach numbers with VLM (left) and FLOPS (right)

5.6.2 Blended Wing Body

The current version of FLOPS contains models for more accurate weight estimations of the blended wing body. However only a single set of models are provided in aerodynamic analysis module which are based on validated data with conventional aircraft. Bradley (2004) has modified the module which calculates the wetted area by using the detailed wing station data and used an interpolation between curve fits of average friction vs. Reynolds number to determine the skin friction coefficient. In this thesis, the actual aircraft surfaces have been generated which provide more realistic wetted area values.

The Blended Wing Body configuration of the MOB project (Morris *et al.*, 2004) is analysed. An induced drag is calculated through FLOPS and VLM, of which the VLM yields higher values. In order to validate the results, the total drag is required. The induced drag coefficients are added to the friction drag and pressure drag, resulting in the total drag coefficient. These coefficients are then compared with the Reynolds Averaged Navier-Stokes (RANS) results from Qin *et al.*(2002) as presented in Table 5.5.

Table 5.5 Lift and drag coefficients for BWB

CL	CDi		CDf	CDp	CD total		
	FLOPS	VLM			FLOPS	VLM+ Mason	RANS
0.0363	0.0001	0.0020	0.0080	0.0006	0.0087	0.0106	0.0181
0.1427	0.0009	0.0039	0.0079	0.0003	0.0091	0.0121	0.0197
0.2493	0.0029	0.0075	0.0078	0.0013	0.0120	0.0166	0.0223
0.3560	0.0059	0.0130	0.0076	0.0024	0.0159	0.0230	0.0290
0.4625	0.0099	0.0202	0.0074	0.0061	0.0235	0.0337	0.0395
0.5687	0.0150	0.0293	0.0073	0.0114	0.0337	0.0480	0.0620

The drag polar of the BWB calculated with FLOPS, VLM and RANS are presented in Figure 5-13. Comparing FLOPS and VLM methods, the latter yields overall results closer to the RANS.

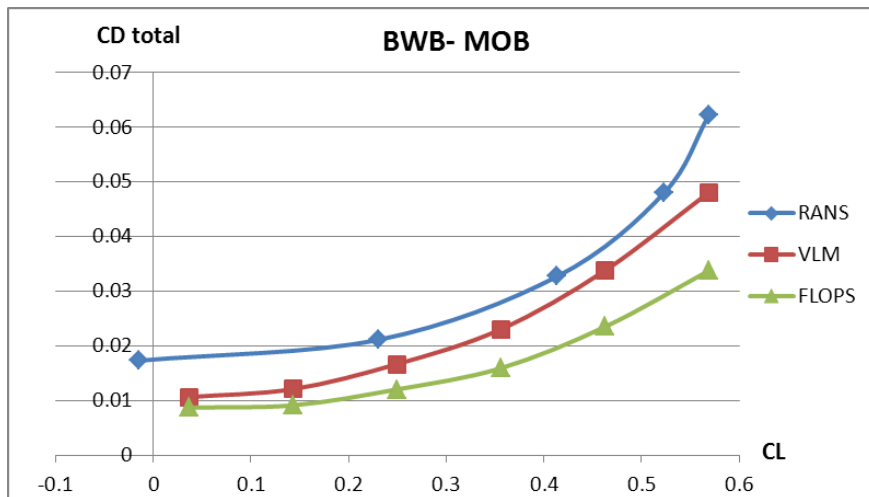


Figure 5-13 BWB Drag Polars with RANS, FLOPS, and VLM

5.6.3 Unmanned Aerial Vehicle: UAV-KU4

The unmanned aerial vehicle: UAV KU-4 model is analysed with TORNADO VLM and compared with the wind tunnel results (In-noi *et al.*, 2004). The model is tested at their speed of 30 m/s (Reynolds number of 249,000). The lift coefficients with respect to angles of attack are presented in Figure 5-14. The result from VLM is in good agreement with the wind tunnel result in the region of low angle of attack up to 8 degrees where the separation occurs. Even though VLM cannot capture the stall behaviour, it is still applicable for the analysis at conceptual stage where the design is

mostly evaluated at the cruise condition. FLOPS result is excluded since it does not take the angle of attack into the calculation.

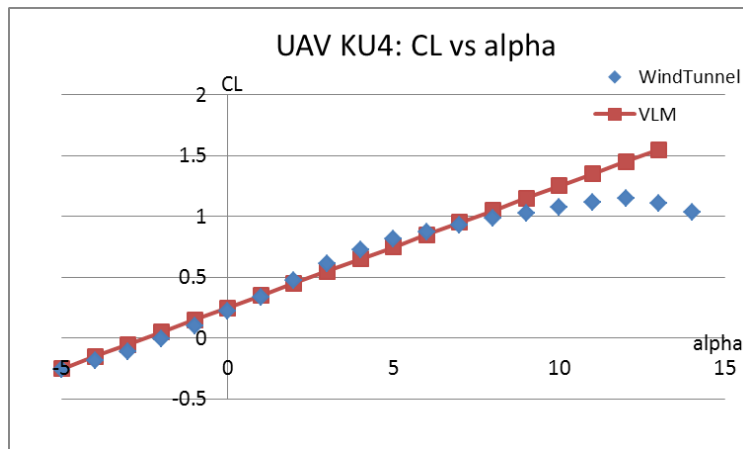


Figure 5-14 Lift coefficient versus angle of attack

As seen from Figure 5-15, both VLM, combined with the Mason method, and FLOPS under-predict the drag coefficients compared to the wind tunnel results. The combined VLM Mason method performs slightly better than FLOPS due to the actual modelling of aircraft components for the friction drag calculation, yet still yields high error from the wind tunnel results. This demonstrates the limitations of the model for analysing a small aircraft due to the scaling effect which results in less accurate Reynolds number for friction drag predictions.

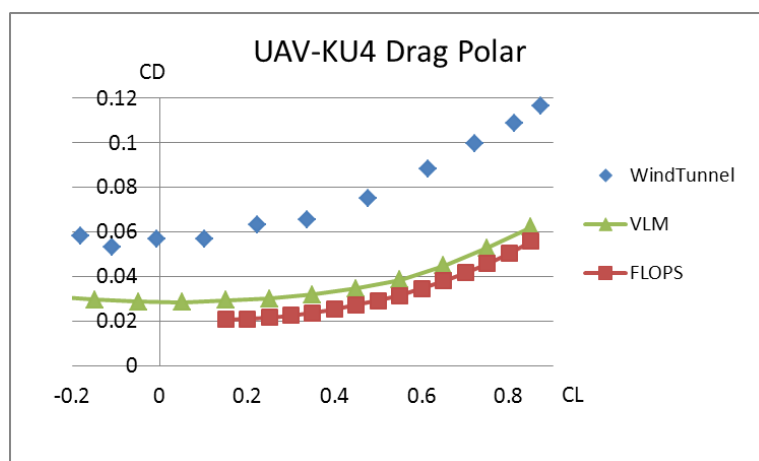


Figure 5-15 UAV-KU4 Drag Polar

5.7 Conclusions

In this chapter, the parameterised aircraft model has been tested in the medium fidelity aerodynamics analysis. The results have been compared with high fidelity or wind tunnel test data according to availability. There are two improvements in the aerodynamic calculation:

1. Potential flow solver can be introduced for lift and drag calculation, which takes the geometry into calculation. For a conventional configuration, VLM show non-significant improvement comparing to the empirical models which have been validated with data of actual aircraft. However, for unconventional design, such as Blended Wing Body, the VLM yields more accurate results than the empirical model. For the modelling of the UAV which has unconventional arrangements of the empennage, the VLM combined with Mason method slightly improves the drag prediction, but not as satisfactory as the blended wing body.
2. The friction drag calculation uses the empirical formula based on the wetted area of the parameterised surface. The modelling of actual surfaces gives more accurate wetted area and therefore improves the accuracy of the friction drag coefficients.

The combined VLM and Mason's friction drag model show promise for practical applications and will be evaluated in an industrial test case in the Chapter 6.

6 Applications in Optimisation and Evaluations

This chapter aims to evaluate the trade-offs of the proposed approach in terms of computational efficiency and extension of scope with regard to geometrical description at early design stage.

The work in this chapter is based on the following tasks:

1. Integration of the prototype geometry generation tool ,VS Geo (developed by the author) and Vortex Lattice Method (TORNADO)
2. Match design parameters required by the VS Geo model and VLM with two different test cases of industrial relevance, both for the geometry and aerodynamic modules. If a certain parameter is not used in the test cases, such as wing twist or CST airfoil coefficients, that particular parameter is added to the computational workflow
3. Disconnect the original geometry and aerodynamic models which calculate the lift and induced drag coefficients in the test case, replacing them with VS Geo model and VLM model, respectively
4. Perform optimisation studies based on the geometrical design parameters

6.1 Integration into Model-Based Design Tool

The considered test cases have been integrated in the program developed by Cranfield University Advanced Engineering Design Group: “Aircadia” (Guenov *et al.*, in preparation). This is a model-based design tool which dynamically assembles computational processes. It has the capabilities of performing multi-disciplinary, multi-objective deterministic and / robust design optimisation. It also allowsthe integration of models written in different languages such as MATLAB or C#.

Figure 6-1 demonstrates the flow of the design variables in the USMAC test case (Guenov *et al.*, 2010) with the two main modules: geometry (orange square), and aerodynamic analysis (blue square). The models which are associated with the calculation of lift and induced drag are replaced by the geometry generator VS Geo and the VLM solver.

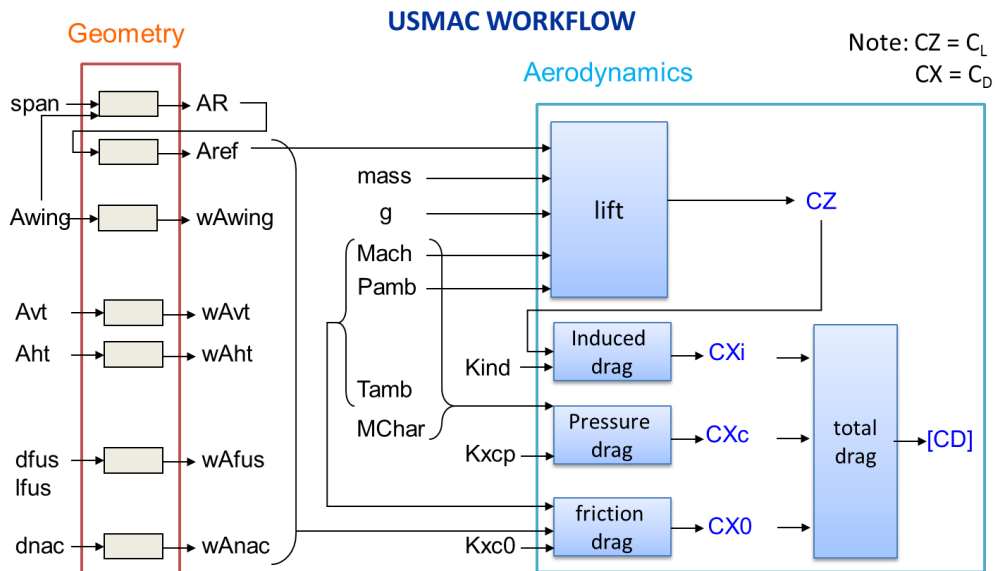


Figure 6-1 Subset of USMAC Workflow (geometry and aerodynamic modules)

After replacing the geometry and aerodynamic models, the computational flow in Figure 6-1 is transformed into the one presented in Figure 6-2. The VS Geo model replaces 7 models in the geometry module and keeps all the corresponding inputs and outputs. The VLM model replaces the lift and induced drag models, with angle of attack (AoA) and thickness to chord ratio (TuC) included as additional design parameters.

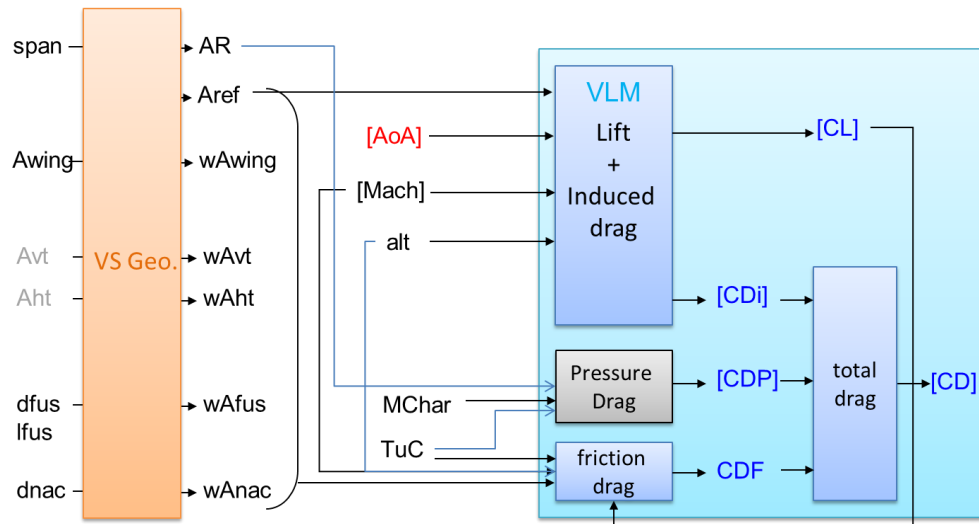


Figure 6-2 Integration of USMAC with VS Geo and VLM

The wetted area calculations for the original USMAC model and VSGeo, and the aerodynamic analysis results between the USMAC model and VLM have been compared, as presented in Chapter 5. The integration of VSGeo and VLM provides an example of integrating foreign models which are written in different languages. Such integration requires model merging which involves matching of variables. This process forms the basis for the integration of the FLOPS test case which contains a larger number of models and design variables.

The aircraft sizing code FLOPS was provided by the code developer. The subset of FLOPS related to transport aircraft configurations was translated to a C# library by Riaz, (2012) to be used in the Aircadia environment. This subset contains 171 models, and 317 design variables. The aerodynamic analysis module of this test case was replaced with the VLM solver in a MATLAB.dll format.

The design inputs and outputs of the VLM model, which is considered as ‘foreign’ model (requires merging), are mapped to the original design variables in FLOPS through the Aircadia architecture, as summarised in Table 6.1. The remaining three drag components: friction, pressure, and compressible drag, are calculated through the original FLOPS model. The total drag polars are stored in a matrix form which will be used in the performance analysis module.

Table 6.1 Induced Drag Models Merging

	model	inputs	outputs
Disable:	LiftCoeffs	desLiftCoeff	noOfLiftCoeff, liftCoeffs[]
Disable:	IndDragCoeff	noOfLiftCoeff, liftCoeffs[], noOfMachNum, machNums[], wingAsp_R, wingTap_R, wingSweep, EO	CDITAB[]
Merge:	VLM	noOfLiftCoeff, wingSpan, horTailAspR, horTailArea, verTailAspR, verTailArea, wingSwp, wingTapR, dihedral, wingRefArea, horTailSwp, horTailTapR, verTailSwp, verTailTapR, fuseDia, desMachNum, maxCrzAlt	liftCoeffs[], CDITAB[]

The main difference between FLOPS and VLM is the way the lift coefficient is calculated. While VLM uses the angle of attack as input for lift and drag, FLOPS does not consider the angle of attack, but uses a set of predefined lift coefficients instead. This vector of lift coefficients is then used to calculate the induced drag coefficients.

In the FLOPS mission analysis module, the drag polars at various Mach numbers are required. The standard model therefore computes a predefined vector of eight Mach numbers. In this analysis the VLM computes lift and induced drag polar at a single Mach number, which reduces computational time.

6.1.1 Optimisation Study #1

The formulation of the first optimisation study is based on Nunez *et al*, (2011). The aim of the study is to minimise both ramp weight (rampW) and landing field length (IndFldLen). The take-off field length and approach speed are chosen as constraints. The Mach number and altitude at cruise are set as constants. The design variables are chosen to reflect the inputs of the VLM model in Table 6.1.

objectives: min rampW [lbs]
 min IndFldLen [ft]

subject to: takoffFldLen ≤ 9000 [ft]
 apprSpeed ≤ 140 [kts]

constant: crzMachNum = 0.8
 maxCrzAlt = 35000 [ft]

variables:	wingSpan = [90,130] [ft]	wingAspR = [8,8.5]
	wingTapR = [0.25, 0.28]	wingThkChdR = [0.09, 0.12]
	wingSwp = [30,32] [deg]	dihedAng = [0,4] [deg]
	thrust = [29200, 32000] [lbf]	desRange = [2500,3000] [nm]

The optimiser used in Aircadia is based on a genetic algorithm (Deb *et al.*,2002). In this study, the optimisation setup consists of a population size of 40 individuals and 20 generations. The recommended population size and number of generations are based on previous studies. The results of the optimisation study are visualised in Aircadia and the screenshots are presented in Figure 6-3 and Figure 6-4. The top-left plot of these two figures presents the objective space, where in this case, the ramp weight and landing field length are both to be minimised. The green dots represent the design points that satisfy all the constraints. The grey dots represent the design points which violate at least one of the constraints. The yellow squares around green dots identify the non-dominated solutions. The values of each design parameter are also plotted in the parallel coordinates plot in the bottom section of the screen by considering the same aforementioned colour notation. For this particular optimisation study, the total computation time for the original FLOPS test case is 14 minutes while the time for the integrated test case FLOPS+VLM is 4 hours and 10 minutes.

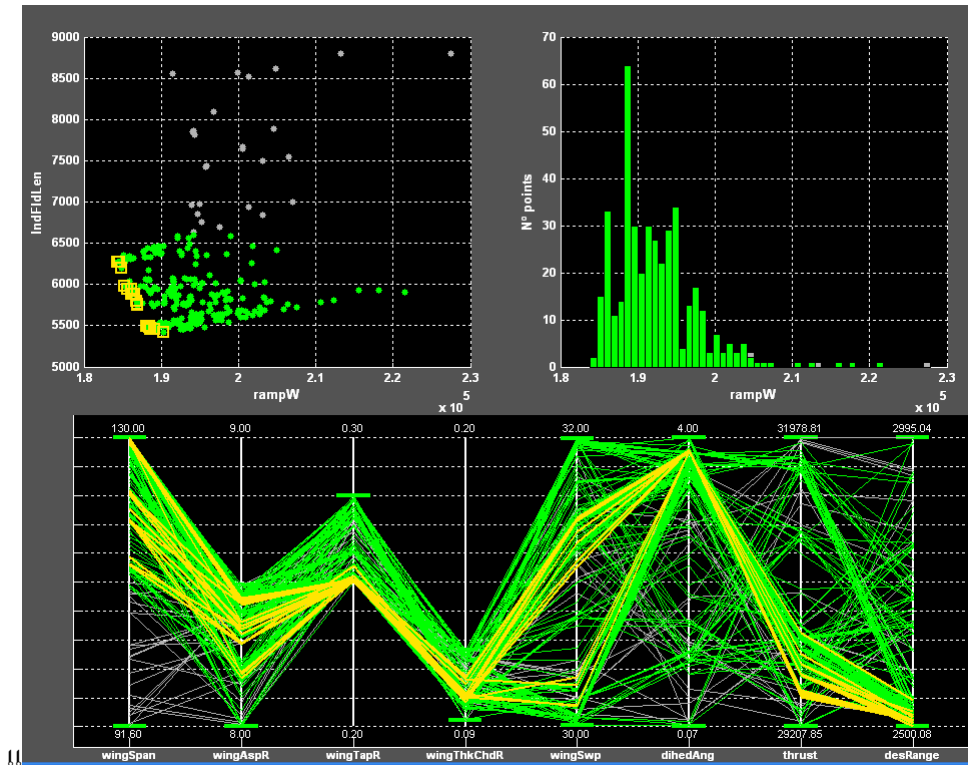


Figure 6-3 Optimisation study: FLOPS

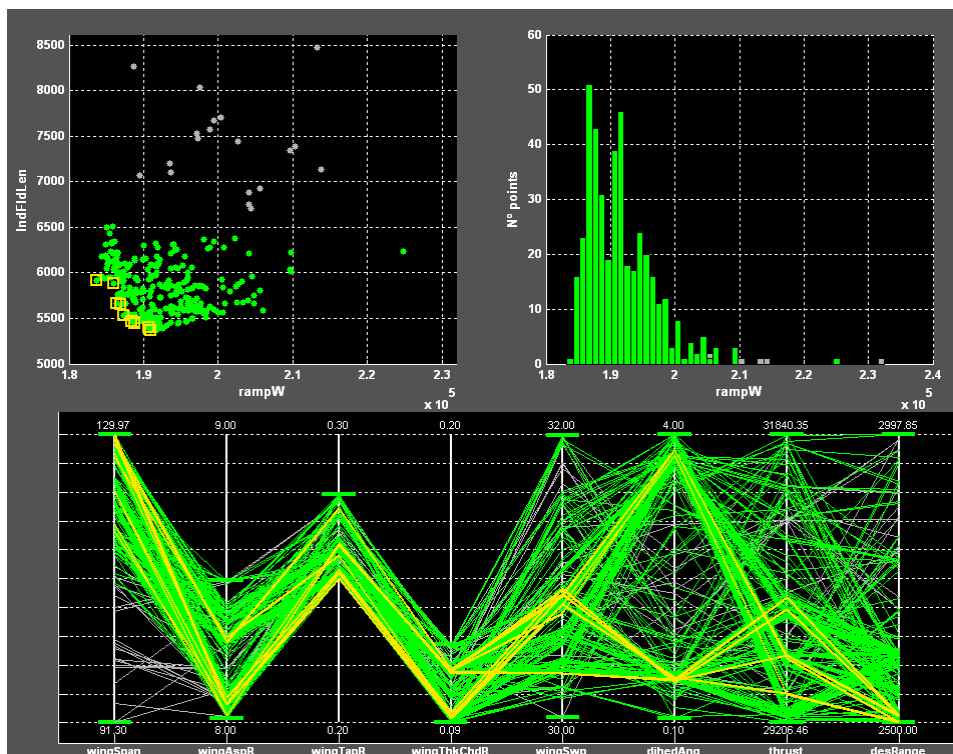


Figure 6-4 Optimisation study: FLOPS + VLM

For comparison purposes, Figure 6-5 presents only the Pareto front of the original FLOPS (in white squares) and FLOPS integrated with VLM (in yellow squares). The

Pareto front of FLOPS+VLM has progressed further left from the results of the original FLOPS. Three sample points (white-filled) have been highlighted to demonstrate that with the same design parameters, the model with VLM results in lower ramp weight and landing field length, approximately 1000 lbs and 20 ft, respectively.

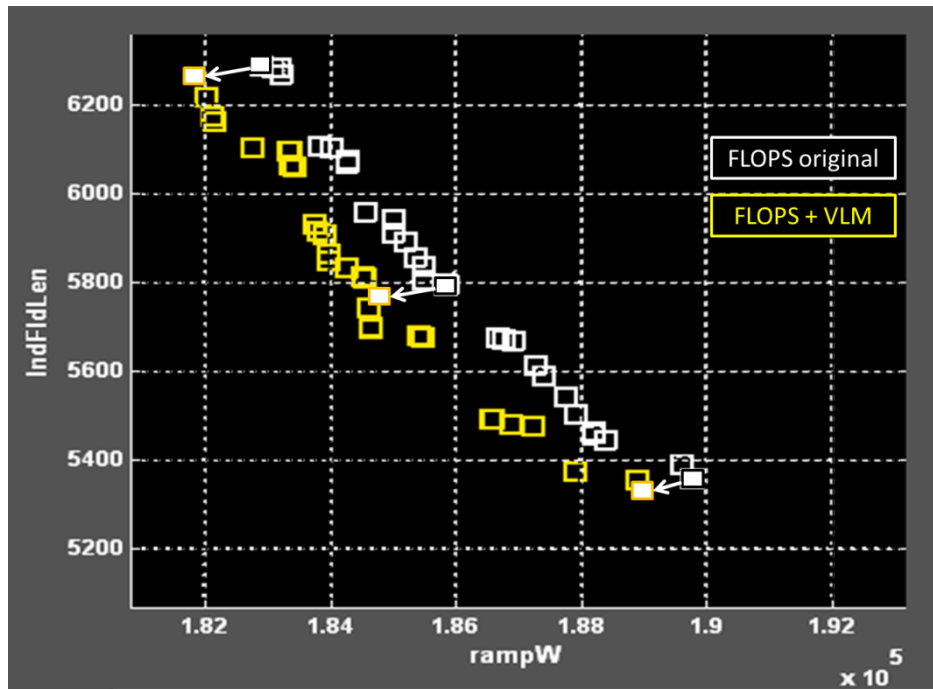


Figure 6-5 Objectives Space [Original FLOPS (white) and FLOPS+VLM (yellow)]

~~The difference in the Pareto fronts results from the introduction of a more detailed geometrical representation of airfoils to estimate the corresponding aerodynamic coefficients to a higher accuracy, as discussed in Chapter 5.~~

As demonstrated in Section 5.6.1, FLOPS aerodynamic analysis model and FLOPS+VLM have similar level of accuracy for aircraft with conventional configuration. Therefore, the results from both optimisation cases should be similar. The possible explanation to the different in results is that in FLOPS+VLM case, detailed drag is calculated based on wetted area from the generated geometry. By this, the results from FLOPS+VLM case should be more accurate.

In order to demonstrate the importance of introducing higher fidelity tools at conceptual stage to assess the impact of design estimations on different design levels, the airfoil description has been included in the next study.

6.1.2 Optimisation study #2

The first optimization study has been extended to demonstrate the capability to include lower level design parameters (e.g., airfoil parameterisation variables) in the formulation of design studies at early stage. This study follows the same formulation in optimisation study #1, but the design parameter “wingThkChdR”(thickness to chord ratio) has been replaced by the CST airfoil coefficients.

It is important to note that the VLM computes the lift and drag coefficient of airfoils based on the normal vectors of the corresponding camber surfaces, without accounting for the entire profile geometry. Therefore, in order to demonstrate the effect of airfoil parameterisation in the aircraft conceptual design process, the upper profile is chosen to be fixed while the lower profile is allowed to change within the defined range. The airfoil profile of Boeing 737 at root is chosen as the baseline airfoil to be optimised. The CST airfoil coefficients for the upper profile (A1-A6) are set as constants, while the design bounds for the lower airfoil (A7-A12) are defined as $\pm 20\%$ of the baseline airfoil.

objectives:	min rampW [lbs]	
	min IndFldLen [ft]	
subject to:	takoffFldLen ≤ 9000 [ft]	
	apprSpeed ≤ 140 [kts]	
constants:	crzMachNum = 0.8	A4 = 0.885
	maxCrzAlt = 35000 [ft]	A4 = 0.885
	A1 = 0.480	A5 = 0.635
	A2 = 0.682	A6 = 0.859
variables:	wingSpan = [90,130] [ft]	wingAspR = [8,8.5]
	wingTapR = [0.25, 0.28]	wingSwp = [30,32] [deg]
	dihedAng = [0,4] [deg]	thrust = [29200, 32000][lbf]
	desRange = [2500,3000] [nm]	A7 = [0.037, 0.460]

$$A8 = [0.557, 0.582]$$

$$A9 = [0.342, 0.933]$$

$$A10 = [0.922, 0.964]$$

$$A11 = [0.589, 1.053]$$

$$A12 = [0.481, 0.667]$$

The results of this study are presented in Figure 6-6. The values of the 6 parameters describing the lower airfoil profile (A7-A12) and the two objectives are shown in the parallel coordinates plot in the bottom part of the figure.

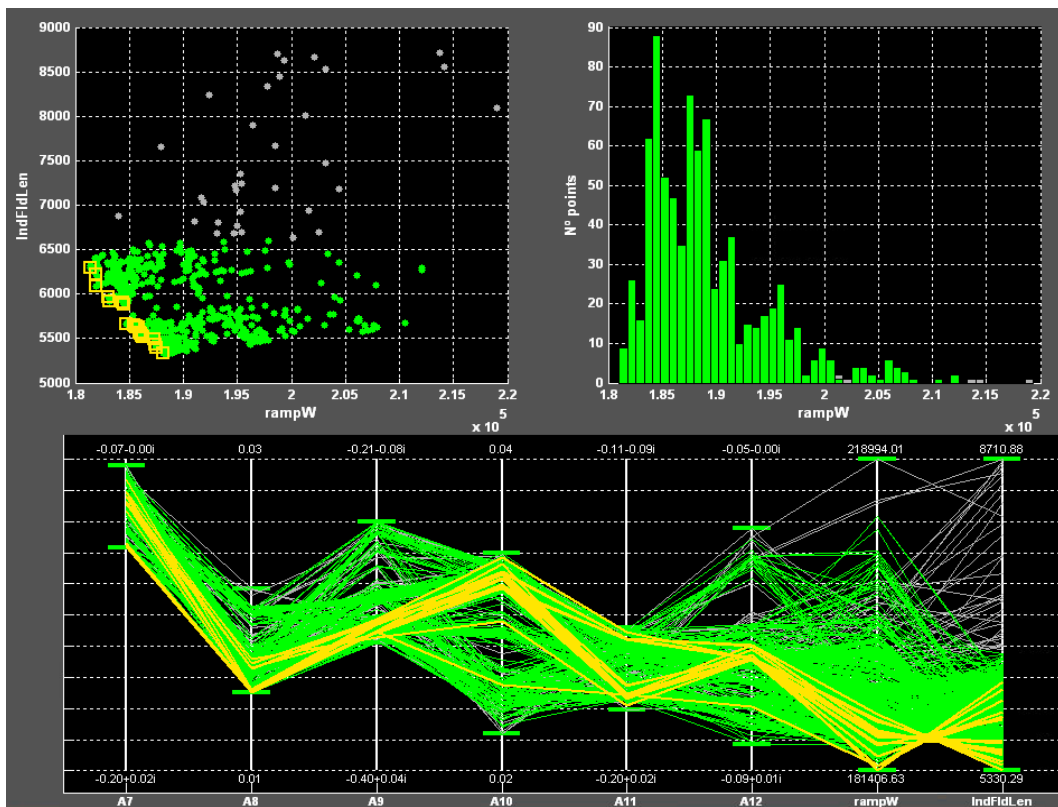


Figure 6-6 FLOPS + VLM with airfoil design parameters (A7-A12)

Examples of airfoil profiles on the Pareto front are presented in Figure 6-7. The baseline airfoil is plotted in red, while the non-dominated solutions are plotted in green, blue, cyan, and pink.

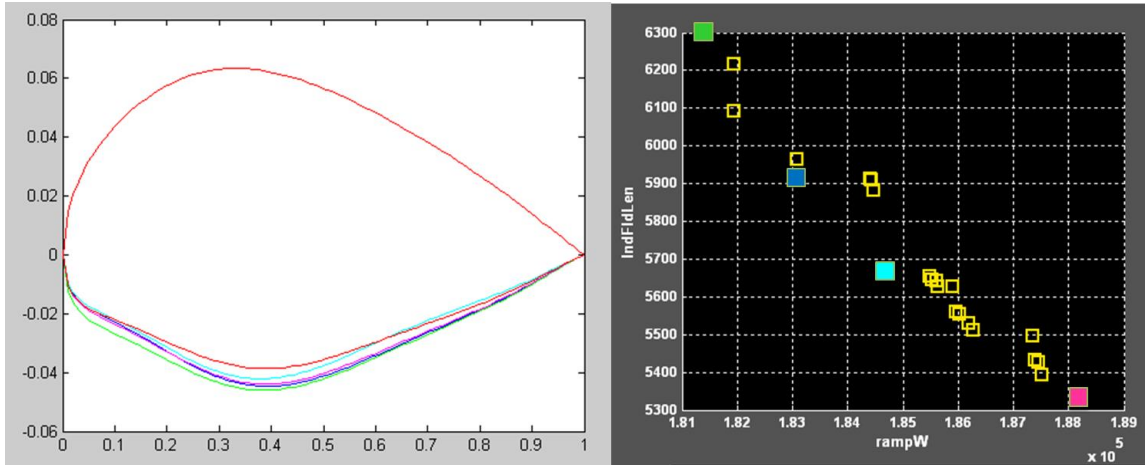


Figure 6-7 Airfoil shapes at the Pareto front

The above optimisation study provides an example of the benefits derived by enabling higher fidelity geometrical representation of the aircraft and its components at early design stage. In this particular case, the full exploitation of such benefits can be achieved through the deployment of alternative airfoil analysis methods (such as Viscous Garabedian-Korn: VGK (ESDU, 2004)) capable of capturing more detailed airfoil representations, as discussed in Padulo *et al.* (2009).

6.1.3 Optimisation study #3

The optimization problem for UAV-KU4 (In-noi *et al.*, 2004), with a focus on wing root and tip airfoil has been studied as follow:

objective: min C_D
subject to: (t/c) tip ≥ 0.09
 (t/c) root ≥ 0.12
constant: CL = 0.66
 Re = 249,000

variables:

A1 =	[0.0859,	0.2148]
A2 =	[0.0780,	0.1950]
A3 =	[0.1290,	0.3224]
A4 =	[0.0948,	0.2371]
A5 =	[0.0890,	0.2224]
A6 =	[0.0740,	0.1851]

$$\begin{aligned}
A7 &= [-0.0748, -0.0075] \\
A8 &= [-0.2308, -0.0923] \\
A9 &= [-0.1110, -0.0333] \\
A10 &= [-0.0597, -0.0239] \\
A11 &= [0.0317, 0.0792]
\end{aligned}$$

The objective is to minimize drag for a given lift, in other word, to maximise lift to drag (L/D). The optimisation problem deals with unit chord airfoil section with fixed lift coefficient of 0.66 at cruise condition or at Reynolds number of 249,000.

The prototype UAV uses NACA63₂-415 airfoil for the wing. The structural limitations result in two constraints i.e. wing thickness to chord ratio at root chord and tip chord must be greater than or equal to 9% and 12%, respectively.

Since the objective is to minimise drag, the eleven CST design parameters uses airfoil in the prototype as the baseline airfoil and also upper bound,where as the lower bound is chosen to be 40%less to accommodate the reduced thickness from 15% to 9% chord length.

The optimisation problem uses XFOIL solver at a constant lift coefficient of 0.66 for viscous analysis, and FMINCON function in MATLAB® is used as an optimiser. The thickness is evaluated at each run to ensure the structural constraints have been met.

The results for optimal airfoil at root and tip are shown in Figure 6.8. The drag coefficient of the optimal airfoils at root and tip are lower than the prototype airfoil (63₂-415) about 10% and 30%, respectively.

Airfoil Section	A1	A2	A3	A4	A5	A6
Prototype	0.2148	0.1950	0.3224	0.2371	0.2224	0.1851
Root	0.1770	0.1600	0.2792	0.1943	0.2020	0.1759
Tip	0.1305	0.0865	0.2230	0.1125	0.1417	0.0996

Airfoil Section	A7	A8	A9	A10	A11	CD
Prototype	- 0.0748	- 0.2308	- 0.1110	- 0.0597	0.0792	0.0122
Root	- 0.0218	- 0.2013	- 0.0688	- 0.0287	0.0766	0.0110
Tip	- 0.0084	- 0.1962	- 0.0371	- 0.0284	0.0328	0.0086

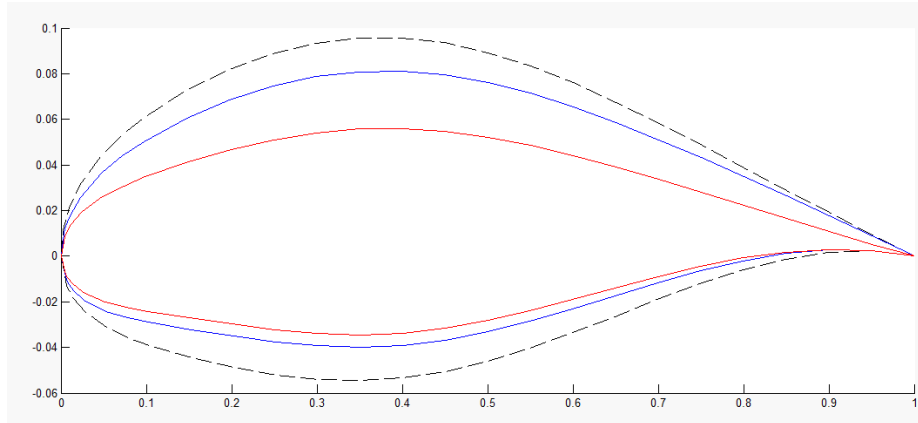


Figure 6-8 Prototype 63₂-415 airfoil (black-dashed), optimal section at root (blue), and optimal section at tip (red) with the corresponding CST parameters

6.2 Evaluations

The trade-offs of introducing more detailed geometry representations in the early design stages are summarised in Table 6.2. The computational speed decreases due to the deployment of higher fidelity aerodynamics analyses (with a factor of ~17 when adopting the VLM solver), although this is still acceptable as more robust solutions can be identified due to a higher level of analysis at the conceptual design stage.

Table 6.2 Evaluations of the integration of VSGeo and VLM into FLOPS

Criteria	Gain / Loss	Achievements
Computation Speed	↘	For 40 individuals 10 generations FLOPS:14 mins / FLOPS+VLM:4 hrs 10 mins
Accuracy	↗	Improved accuracy of design analyses at conceptual stage
Scope	↗	Expanded scope of early design studies by: - including airfoil design coefficients - providing more detailed information at various design levels (e.g. flight mission, aircraft performance, weights, etc.)

It has also been demonstrated in this chapter how the airfoil design coefficients can now be considered at the conceptual design stage, which extends the scope of the study. In general, the detail of geometrical representations can be increased both at an aircraft and components level, depending on the design study to be conducted. An example based on an optimisation problem was considered to demonstrate the possibility of conducting multilevel design activities (e.g., maximisation of aircraft performance at a higher level, while optimising the airfoil aerodynamics at a lower level), as well as supporting decision making processes (e.g., obtaining a better estimation of the ramp weight resulting from the use of a given airfoil). In both cases, the expected benefits are the reduction of subsequent design iterations and/or rework through more informed decisions at early stage.

The key original contributions of this chapter are integrating high fidelity aerodynamic analysis tool and more accurate wetted area calculation to conceptual design analysis tools. For the conventional aircraft, the test case may not obviously show improvement in accuracy, since the low fidelity model has been validated with the similar aircraft configuration. However, this demonstrates possibility to extend and included more detail such as airfoil description into early design stage. Combining with the findings from Chapter 5 that the high fidelity analysis tools: Tornado VLM, in this thesis, yields better accuracy for more unconventional configurations such as Blended Wing Body (BWB), or UAVs, the expected optimisation results should reflects more reality. These contributions are applicable in engineering design practice by use the integrated FLOPS+VLM module to perform design space exploration in conceptual design stage, or replacing some of the empirical models in FLOPS for accommodate unconventional design with the developed geometrical model proposed in this thesis.

7 Summary and Conclusions

Presented in this thesis is an advance shape parameterisation framework for aircraft geometrical representations which captures sufficient detail suitable for efficient aerodynamic analysis and optimisation at early design, in order to close the gap between conceptual and preliminary design stages.

This chapter presents the summary and conclusions from the literature review, followed by the comparison of airfoil parameterisation methods, the proposed surface parameterisation method and its evaluation and integration with aerodynamic analysis tools. Finally, current limitations and suggested future work are discussed.

7.1 Literature Survey

Widely cited shape parameterisation methods were selected for comparison. Since the most basic level of geometry description in this research is airfoil, the literature survey has been focused on airfoil parameterisation. The mathematical formulation for each method has been summarised and the relevant advantages and disadvantages have been discussed. The previous comparison studies have been compiled and it was shown that most of these studies were pair-wise and hence the results were inconclusive. From this, a systematic comparison of airfoil parameterisation was conducted.

7.2 Airfoil Parameterisation Method Comparison

Five widely-cited airfoil shape parameterisation methods have been selected for a comparison study. Relevant metrics have been selected and an assessment methodology corresponding to each criterion has been proposed. The main assessments have been based on airfoil fitting tests which determine the least number of design parameters

necessary for shape generation at the required accuracy and the number of airfoils configurations that each method is able to capture. The Class-Shape function Transformation (CST) method has been chosen for curve generation due to its simplicity and required parsimonious number of design parameters.

7.3 Surface Parameterisation Method

The geometrical model has been generated with design parameters describing the geometry and configurations at the level of detailed considered in the conceptual design stage. The CST functions have been combined with a set of lofting functions proposed by the author, resulting in analytical surfaces representing aircraft components. A joining algorithm between two intersecting surfaces has also been proposed. An object-oriented structure has been proposed for the surface generation and assembly of components.

Three aircraft configurations, a twin-jet airliner, an unmanned aerial vehicle and a blended wing body, have been generated with the proposed parameterisation method, covering both conventional and unconventional configurations. The proposed parameterisation method is able to generate these aircraft geometrical representations with a parsimonious number of design parameters. The baseline configurations can be directly manipulated through a set of design parameters.

7.4 Aerodynamics analysis

The selected potential flow solver has been used to perform the aerodynamic analyses due to its affordable computational cost for early design. The vortex lattice method provides lift and induced drag based on the potential flow solution. The induced drag coefficient is added with friction, pressure, and compressible drag coefficients to provide the total drag coefficient.

The aerodynamic analysis results have been validated with high fidelity models and/or wind-tunnel data according to availability. Introducing higher fidelity tools in the aerodynamic module of an aircraft sizing code, such as FLOPS, improves the accuracy of design analyses, especially for the unconventional design.

7.5 Evaluation

The ultimate aim has been to close the gap between conceptual and preliminary design stages. This has been demonstrated by the integration of a higher fidelity models into industrially validated test cases to identify more robust solutions, which will lead to a reduction of subsequent design iterations and/or rework.

7.6 Novelty and Contribution to knowledge

In order to assess airfoil parameterisation methods, analysis metrics have been proposed. The methodology to assess each desirable property has been developed based on airfoil fitting tests.

Also proposed in this research is an efficient surface parameterisation framework for aircraft surface geometry which includes the following components developed by the author:

1. An object-oriented structure of aircraft components which are based on geometry-related design parameters
2. A surface generation method through a set of proposed and selected distribution and lofting functions

The method is able to capture the surface descriptions of main aircraft components e.g. wing, tail, fuselage, and nacelle with a parsimonious number of design variables, for both conventional and unconventional configurations.

A potential flow solver has been integrated with the proposed geometry generation tools. This provides more reliable aerodynamic analysis results for unconventional design at affordable computational cost.

The proposed framework has been integrated into industrially relevant test cases which demonstrate extending its scope, such as airfoil analysis in the early design stage.

7.7 Current Limitations

1. Geometrical Parameterisation

- The proposed framework has been developed for surface parameterisation only. The object-oriented structure allows fast computation, but currently supports only the main aircraft components such as wing, horizontal and vertical tail, fuselage and nacelle.

-The parameterisation has been developed for surfaces and currently produces quadrilateral panels “on-the-fly” which are suitable only for the panel method analysis.

2. Aerodynamic analysis

- The selected aerodynamic solvers are based on the potential flow which allows fast computation compared to the high-fidelity tools which are based on Navier-Stokes or Euler equations. However, this is limited only to inviscid, incompressible flow. Therefore, the friction drag cannot be determined and still requires empirical models to calculate its value.

7.8 Future work

1. Geometrical Parameterisation

The current framework for geometry generation can be extended in the following directions:

- The definition of more detailed parts and components such as control surfaces can be introduced for more detailed mission analysis, e.g., at take-off and landing which is required for noise prediction.

- The geometric parameterisation method can provide the basis for structural modelling and aeroelastic analysis.

- The geometry can be exported as “cloud of points” in the standard “.stl” format (McDonald, 2004) which requires a mesh generation tool for CFD analysis.

2. Programming / Architecture

Currently, the geometry generation tool and visualisation is developed in MATLAB® environment. An interactive graphical user interface will enhance the capabilities of the geometry generation tool, e.g. allow easy integration and operation between various disciplines.

3. Aerodynamic analysis

In order to improve the accuracy of aerodynamic analysis, alternative solvers which consider the viscosity effects could be introduced, providing that it is computationally affordable for design exploration in the early design stages.

REFERENCES

- Airbus, (2010), “A320 Aircraft Characteristics Airport and Maintenance Planning”. Technical Report. Rev. Sept 01/10
- Anderson, W.K., Karman, S.L., and Burdyslaw, C., (2009), “Geometry Parameterization Method for Multidisciplinary Applications”, AIAA/ISSMO Multidisciplinary Analysis and Optimization, Victoria, British Columbia, Vol.47, No.6, pp.1568-1578.
- Armstrong, C.G., Monaghan, D.J., Price, M.A., Ou, H., and Lamont, J., (2002) “Integrating CAE concepts with CAD geometry”, Topping, B.H.V., Bittnar, Z, editors, Engineering Computational Technology, Stirling: Saxe-Coburg Publications: 75–104.
- Azamatov, A.I., Lee, J.W, Byun, H.W., and Kim, S.H.(2008), “Advanced Configuration Generation Technique for the Complex Aircraft Geometry”, IEEE/ASME International Conference on Advanced Intelligent Mechatronics, China.
- Blair, J.C., Ryan, R.S., Schutzenhofer, L.A., Humphries, W.R., (2001) “Launch vehicle design process: characterization, technical integration, and lessons learned”, NASA, TP-2001-210992.
- Bloor, M.I.G. and Wilson, M.J., (1989a), “Generating Surfaces using Partial Differential Equations”, Computer Aided Design, Vol. 21, No.3, pp. 165-171.
- Bloor, M.I.G. and Wilson, M.J., (1989b), “Using Partial Differential Equation to Generate Free-Form Surfaces” Computer Aided Design, Vol. 22, No.4, pp. 202-212.
- Bradley K., (2004), “ A sizing methodology for the conceptual design of Blended-Wing-Body transports”, NASA/CR-2004-213016; Sept.
- Castonguay, P. and Nadarajah, S., (2007), “Effect of Shape Parameterization on Aerodynamic Shape Optimization”, 45th AIAA Aerospace Science Meeting and Exhibit, Reno, NV.

Ceze, M., Hayashi, M., and Volpe, E., (2009), “A Study of the CST Parameterization Characteristics”, “27th AIAA Applied Aerodynamics Conference”, San Antonio, Texas, AIAA 2009-3767.

Choi, S., Alonso, J., Kim, S. and Kroo, I., (2005), “Two-level multi-fidelity design optimization studies for supersonic jets”, 43rd AIAA Aerospace Science Meeting and Exhibit, Reno, NV, January. AIAA Paper 2005-531

Coxeter, H.S.M., Greitzer, S.L., (1967), “Geometry Revisited”, Mathematical Association of America, pp. 53

Deb, K., Pratap, A., Agerwal, S, and Meyarivan, T., (2002), “A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II”, IEEE Transaction on Evolutionary Computation, Vol.6, No. 2, April.

Fredericks, W. J., Antcliff, K. R., Costa, G., Deshpande, N., Moore, M. D., San Miguel, E., & Snyder, A. N. (2010). “Aircraft conceptual design using vehicle sketch pad”. In 48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition

Guenov, M.D., Nunez, M., Molina-Cristobal, A., Sripawadkul, V, Datta, V, and Riaz, A.(In preparation), “Composition, Management and Exploration of Computational Studies at Early Design Stage”, AIAA Book Chapter

Gur, O., Mason, W.H., Schetz, J.A., (2010), “Full-Configuration Drag Estimation”, Journal of Aircraft, Vol. 47, No. 4, July–August 2010

Hahn, A., (2010) “Vehicle Sketch Pad: A Parametric Geometry Modeler for Conceptual Aircraft Design,” 48th AIAA Aerospace Sciences Meeting and Exhibit, January 4-7, Orlando, Florida.

Hicks, R. and Henne, P., (1978), “Wing Design by Numerical Optimization”, Journal of Aircraft, Vol. 15, No.7, pp. 407-413.

Holz, S, (2006), “curveintersect”, MATLAB file exchange <http://www.mathworks.com/matlabcentral/fileexchange/8908-curve-intersect-2/content/CurveIntersect/curveintersect.m> [Updated Apr 2006]

Filkovic, D, 2009, “APAME”, Graduate Work, University of Zagreb <http://www.3dpanelmethod.com/documentation.html> [accessed 10 Oct 2009]

In-noi, P., Sripawadkul, V., Monthienthong, A., (2004), Performances of UAV-KU4, Engineering Senior Project, Department of Aerospace Engineering, Kasetsart University, Bangkok, Thailand, (in Thai).

Jacobs, E. N., Ward, K. E., and Pinkerton, R. M., (1933), “The characteristics of 78 related airfoil sections from tests in the variable-density wind tunnel”, NACA Report No. 460.

Keane, A.J., and Nair, P.B., (2005), “Computational Approaches for Aerospace Design”, John Wiley & Sons, Sussex, England, pp.434-435

Knight, N.F., Stone, T.J., (2002), “Rapid modeling and analysis tools: evolution, status, needs and directions”, NASA, CR-2002-211751, NASA Langley Research Center.

Kroo, I., Altus, S., Braun, R., Gage, P., Sobieszczanski-Sobieski, J., (1994) “Multidisciplinary optimization methods for aircraft preliminary design”, AIAA94-4325.

Kulfan, B.M. and Bussoletti, J.E., (2006), “Fundamental Parametric Geometry Representations for Aircraft Component Shapes”, AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Portsmouth, Virginia, AIAA 2006-6948.

Kulfan, B.M., (2007), “Universal Parametric Geometry Representation Method - CST”, 45th AIAA Aerospace Science Meeting and Exhibition, Reno, Nevada, AIAA-2007-62.

Kumano, T., Jeong, S., Obayashi, S., Ito, Y., Hatanaka, K., and Morino, H., (2006) “Multidisciplinary Design Optimization of Wing Shape for a Small Jet Aircraft Using Kriging Model,” AIAA paper 2006-0932.

La Rocca, G., Krakkers, L., van Tooren, M.J.L., (2002), "Development of an ICAD Generative Model for Blended Wing Body Aircraft Design", 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization", Atlanta, Georgia, AIAA 2002-5447.

Leifsson, L.T., Ko, A., Mason, W.H., (2005), "Multidisciplinary Design Optimization for a Blended Wing Body Transport Aircraft with Distributed Propulsion", Multidisciplinary Analysis and Design Center for Advanced Vehicles, Virginia Tech.

March, A., (2008), "Influence of Low-speed Aerodynamic Performance on Airport Community Noise", MSc Thesis, Massachusetts Institute of Technology.

Mastin, C.W., Smith, R.E., Sadreghighi, I., Wiese, M.R., (1996), "Geometric Model for a Parametric Study of the Blended-Wing-Body Airplane", AIAA96-2416.

Mason, J.C., Rodriguez, G., Seatzu, S., (1993), "Orthogonal splines based on B-splines -with applications to least squares, smoothing and regularisation problems", Numerical Algorithms, Vol. 5, Issue 1, pp 25-40.

Mason, W.H., (2011), "FRICTman", Virginia Tech Aerodynamics and Design
http://www.dept.aoe.vt.edu/~mason/Mason_f/FRICTman.pdf [accessed February 2011]

McCullers, L. A., (2011), "Flight Optimization System", User's Guide, Release 8.23, NASA Langley Research Center.

McDonald, B., (2004), "SURF2STL", MATLAB file exchange,
<http://www.mathworks.com/matlabcentral/fileexchange/4512> [Updated 24 Feb 2004]

Melin, T., (2001), "User's guide TORNADO 1.0: Reference manual", Royal Institute of Technology (KTH), Stockholm,
<http://www.redhammer.se/tornado/manual.pdf> [Accessed 29 May 2010]

Miranda, L. R., Elliot, R. D., Baker, W. M., (1977), A generalized vortex lattice method for subsonic and supersonic flow applications. NASA Contractor Report 2865.

Morris, A., Arendsen, P., LaRocca, G., Laban, M., Voss, R., Honlinger, H., (2004) "MOB - A European Project On Multidisciplinary Design Optimisation", 24th International Congress of the Aeronautical Sciences,

Nickol C., (2004) "Conceptual design shop", Presentation to conceptual aircraft design working group (CADWG21), September.

Padulo, M., Maginot, J., Guenov, M. and Holden, C., (2009), "Airfoil Design under Uncertainty with Robust Geometric Parameterization", AIAA paper 2009-2270

Pasadas, M., and Rodriguez, M.L., (2008), "Construction of blending surfaces by parametric discrete interpolation PDE splines", *Mathematics and Computers in Simulation*, Vol.77, pp.282-290.

Price, M., Raghunathan, S., and Curran, R., (2006), "An Integrated Systems Engineering Approach to Aircraft Design", *Progress in Aerospace Sciences*, 42: 331-376.

Qin, N., Vavalle, A., Le Moigne, A., Laban, M., Hackett, K., Weinerfelt, P., (2002). "Aerodynamic Studies for Blended Wing Body Aircraft", 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, 4-6 Sept 2002, Atlanta, Georgia.

Raymer, D.P., (1992), "Aircraft Design: A Conceptual Approach", 2nd ed., *AIAA Education Series*, AIAA, Washington.

Riaz, A. (2012), Library of FLOPS models (in C#), Advanced Engineering Design Group, Cranfield University.

Robinson, T.D., (2007), "Surrogate-Based Optimization using Multifidelity Models with Variable Parameterization", Ph.D. Dissertation, Massachusetts Institute of Technology, May.

Rodriguez, D. L. and Sturdza, P., (2006), A Rapid Geometry Engine for Preliminary Aircraft Design, 44th AIAA Aerospace Sciences Meeting and Exhibit, AIAA-2006-0929, Reno, Nevada, Jan.

Sadrey, M., (2012), "Aircraft Design: A Systems Engineering Approach", Wiley, pp.184

Samareh, J.A., (1999), "A Survey of Shape Parameterization Technique", CEAS/AIAA/ACASE/NASA Langley International Forum on Aeroelasticity and Structural Dynamics, Williamsburg, Virginia. NASA/CP-1999-209136, pp. 333-343.

Samareh, J. A., (2004), "Aerodynamic Shape Optimization Based on Free-Form Deformation," AIAA 2004-4630, Sept.

Sensmeier, M.D., and Samareh, J.A., (2005), "Automatic Aircraft Structural Topology Generation for Multidisciplinary Optimization and Weight Estimation", 46th AIAA/ASME/ASCEAHS/ASC Structures, Structural Dynamics, and Material Conference, April, Austin, Texas.

Smith, H., (1999), Blended Wing Body Preliminary Design Study BW-98, Aerogram, Vol. 9, No. 4.

Sobester, A. and Barrett, T., (2008), "The Quest for Truly Parsimonious Airfoil Parameterisation Scheme", 8th AIAA 2008 ATIO Conference, Anchorage, USA, AIAA 2008-8879.

Sobester, A. and Keane, A., (2007), "Airfoil Design via Cubic Splines - Ferguson's Curves Revisited", AIAA infotech@Aerospace 2007 Conference and Exhibit, Rohnert Park, USA, AIAA 2007-2881.

Sobieczky, H., (1998), "Parametric Airfoils and Wings", Notes on Numerical Fluid Mechanics, Vol.16, pp.71-88.

Trapp, J., Sobieczky,H., (1999), "Interactive Parametric Geometry Design", "37th AIAA Aerospace Sciences Meeting and Exhibit, Reno,AIAA-99-0829.

Ugail, H., (2003), "On the Spine of a PDE Surface", Mathematics of Surfaces, Springer, pp. 366-376.

University of Illinois at Urbana-Champaign, UIUC Applied Aerodynamics Group, UIUC Airfoil Coordinates Database, [online database], URL: <http://www.ae.illinois.edu/m-selig/> [accessed 22 JUL 2009]

Wisnoe, W., Nasir, R.E.M., Kuntjoro, W., Mamat, A.M.I., (2009), “Wind Tunnel Experiments and CFD Analysis of Blended Wing Body (BWB) Unmanned Aerial Vehicle (UAV) at Mach 0.1 and Mach 0.3”, 13th International Conference on Aerospace Science & Aviation Technology, ASAT- 13, 26 – 28 May 2009. Cairo, Egypt.

Wu, H.Y., Yang, S., Liu, F., (2003), “Comparison of Three Geometric Representations of Airfoils for Aerodynamic Optimization”, 16th AIAA Computational Fluid Dynamics Conference, Orlando, FL, AIAA 2003-4095

APPENDICES

Appendix A PARSEC-CST Equivalent form

PARSEC airfoil is constructed from the polynomial:

$$Z_k = \sum_{n=1}^6 a_{n,k} X_k^{\frac{n-1}{2}} \quad [\text{B-1}]$$

For upper surface, the equation can be expanded to

$$Z_u = a_1 X^{0.5} + a_2 X^{1.5} + a_3 X^{2.5} + a_4 X^{3.5} + a_5 X^{4.5} + a_6 X^{5.5} \quad [\text{B-2}]$$

where a_1, a_2, a_3, a_4, a_5 are polynomial coefficients

For CST with Bernstein Polynomial order 4 and class function of airfoil, $N1=0.5$, $N2=1.0$:

$$\begin{aligned} Z_u = & b_1 \sqrt{x}(1-x) \binom{4}{0} x^4 + b_2 \sqrt{x}(1-x) \binom{4}{1} x^3(1-x) + b_3 \sqrt{x}(1-x) \binom{4}{2} x^2(1-x)^2 + \dots \\ & + b_4 \sqrt{x}(1-x) \binom{4}{3} x(1-x)^3 + b_5 \sqrt{x}(1-x) \binom{4}{4} (1-x)^4 \end{aligned} \quad [\text{B-3}]$$

Expanding the class function terms, Equation B-3 becomes:

$$\begin{aligned} Z_u = & b_1 X^{0.5} + (4b_4 - 5b_5) X^{1.5} + (6b_3 - 16b_4 + 10b_5) X^{2.5} + (4b_2 - 18b_3 + 24b_4 - 10b_5) X^{3.5} + \dots \\ & + (b_1 - 8b_2 + 18b_3 - 16b_4 + 5b_5) X^{4.5} + (b_1 + 4b_2 - 6b_3 + 4b_4 - b_5) X^{5.5} + (\Delta Z_{TE}) \cdot X \end{aligned} \quad [\text{B-4}]$$

It can be seen from equation, CST parameterization with Bernstein polynomials of order 4 results in the polynomial highest degree of 5.5 which equals to the highest degree of polynomials in PARSEC. This means PARSEC can reach the same level of accuracy as the CST with Bernstein Polynomials of order 4. and the coefficients b_i of CST airfoil in equation B-4 can be transferred to coefficients a_i of PARSEC in Equation B-2.

Appendix B Empirical Model from Industrial Cases

Table B.1 USMAC Aerodynamic models

module	model	Inputs	Outputs
Lift	level_flight_crz	mass_crz, g_crz, Mach_crz, Pamb_crz, Aref	cz_crz
Friction drag	fric_drag_factor	-	Kcx0
	friction_drag_crz	cz_crz, Mach_crz, Pamb_crz, Tamb_crz, wAwing, wAht, wAvt, wAfus, wAnac, lfus, Aref, Lref, ne, Kcx0	cx0_crz
Induced drag	ind_drag_factor	-	Kind
	induced_drag_crz	cz_crz, ar, Kind	cxi_crz
Pressure Drag	press_drag_factor	-	Kcxp
	pressure_drag_crz	Mach_crz, Mchar, Kcxp	cxc_crz
Total drag	drag_factor_crz	cx0_crz, cxi_crz, cxc_crz	cx_crz

Nomenclatures

span	span	mass_crz	total mass at cruise
Awing	wing planform area	g_crz	gravitational acceleration at
dfus,	fuselage diameter	cruise	
lfus	fuselage length	Mach_crz	Mach number at cruise
Aht	horizontal tail area	Pamb_crz	Ambient pressure at cruise
Avt	vertical tail area	Tamb_crz	Ambient temperature at
dnac	nacelle diameter	cruise	
ar	aspect ratio	cx0_crz	friction drag coefficient
ne	number of engine	cxi_crz	induced drag coefficient
wAwing	wing wetted area	cxc_crz	pressure drag coefficient
Aref	wing reference area	Kind	induced drag factor
Lref	reference length	Kcx0	friction drag factor
wAfus	fuselage wetted area	Kcxp	pressure drag factor
wAht	horizontal tail area		
wAvt	vertical tail area		
wAnac	nacelle wetted area		

Table B.2 Aerodynamic analysis models in FLOPS

module	Models	Inputs	Outputs
Lift	DesLiftCoeff	wingThkChd_R, wingAsp_R, wingSweep, wingCamber	desLiftCoeff
	LiftCoeffs	desLiftCoeff	noOfLiftCoeff, liftCoeffs[]
Mach number	DesMachNum	wingAsp_R, wingThkChd_R, wingTap_R, wingSweep, AITEK, desLiftCoeff, maxV	desMachNum
	MachNums	desMachNum, wingThkChd_R	noOfMachNum, machNums[]
Induced drag	IndDragCoeff	noOfLiftCoeff, liftCoeffs[], noOfMachNum, machNums[], wingAsp_R, wingTap_R, wingSweep, EO	CDITAB[]
Friction drag	WingFormFac	wingFine_R, AITEK	wingFormFac
	WingSkinFricCoeff	alt, crzMach, wingCharacLen, TRUW, TRLW	wingSkinFricCoeff, wingReynoldsNum
	WingSkinFricDrag Coeff	wingWetArea, wingSkinFricCoeff, wingFormFac, wingRefArea	wingSkinFricDragCoeff
	(Repeat above 3 models: WingFormFac, WingSkinFricCoeff, WingSkinFricDragCoeff, for horTail, verTail, fuse, nac, can)		
	SkinFricDragCoeff	alt, noOfMachNum, machNums, wingWetArea, wingCharacLen, TRUW, TRLW, horTailWetArea, horTailCharacLen, TRUH, TRLH, perVerTailWetArea, verTailCharacLen, TRUV, TRLV, perFuseWetArea, fuseCharacLen, TRUB, TRLB, perNacWetArea, nacCharacLen, TRUN, TRLN, canWetArea, canCharacLen, TRUC, TRLC, wingFormFac, horTailFormFac, verTailFormFac, fuseFormFac, nacFormFac, canFormFac, wingRefArea, noOfVerTail, noOfFuse, noOfEng	CDFTAB[]
Pressure drag	PressDragCoeff	wingAsp_R, wingThkChd_R, wingCamber, liftCoeff, desLiftCoeff, machNum, desMachNum	pressDragCoeff
	PressDragCoeffs	noOfMachNum, machNums[], desMachNum noOfLiftCoeff, liftCoeffs[], desLiftCoeff	CDPTAB[,]
Compressible drag	CompDragCoeff	wingAsp_R, wingThkChd_R, wingTap_R, wingSweep, wingCamber, wingRefArea, fuseCrossSecArea, fuseLenDia_R, fuseDia_wingSpan_R, aircraftBaseArea, MachNum, desMachNum	pressDragCoeff
	CompDragCoeffs	noOfMachNum, machNums[], wingAsp_R, wingThkChd_R, wingTap_R, wingSweep, wingCamber, wingRefArea, fuseCrossSecArea, fuseLenDia_R, fuseDia_wingSpan_R, aircraftBaseArea, desMachNum	CDCTAB[]

Nomenclature

aircraftBaseArea	aircraft base area
apprSpeed	maximum allowable landing approach velocity
canFormFac	canard form factor
canWetArea	canard wetted area
canCharacLen	canard characteristic length
CDITAB[]	vector of induced drag coefficients
crzMachNum	cruise Mach number
designRange	design range
desLiftCoeff	design lift coefficient
desMachNum	design Mach number
dihedAng	dihedral angle
fuseCrossSecArea	fuselage cross-section area
fuseFormFac	fuselage form factor
fuseDia	fuselage diameter
fuseDia_wingSpan_R	fuselage diameter/wingspan ratio
fuseLenDia_R	fuselage length /diameter ratio
horTailCharacLen	horizontal tail characteristic length
horTailFormFac	horizontal tail form factor
horTailWetArea	horizontal tail wetted area
horTailSwp	horizontal tail sweep
horTailTapR	horizontal taper ratio
ldnFldLen	maximum allowable landing field length
liftCoeffs []	vector of lift coefficients
machNums[]	vector of Mach numbers
maxCrzAlt	maximum cruise altitude
nacFormFac	nacelle form factor
nacWetArea	nacelle wetted area
nacCharacLen	nacelle characteristic length
noOfLiftCoeff	number of lift coefficients
noOfMachNum	number of Mach numbers
noOfVerTail	number of vertical tail
noOfFuse	number of fuselage
noOfEngine	number of engine
rampW	ramp weight
takoffFldLen	maximum allowable take-off field length
verTailCharacLen	vertical tail characteristic length
verTailFormFac	vertical tail form factor
verTailSwp	vertical tail sweep
verTailTapR	vertical tail taper ratio
verTailWetArea	vertical tail wetted area
wingAsp_R	wing aspect ratio
wingCamber	wing camber
wingCharacLen	wing characteristic length
wingFine_R	wing finess ratio
wingFormFac	wing form factor
wingRefArea	wing reference area
wingSkinFricCoeff	wing skin friction coefficient

wingSweep	wing sweep
wingTap_R	wing taper ratio
wingThkChd_R	wing thickness to chord ratio
wingWetArea	wing wetted area

Appendix C MATLAB code for Geometry Generation

```
function mainVSgeo
% VSgeo - generating aircraft surface geometry
% through design parameters
% Vis Sripawadkul,Cranfield University MAR 2012
% Cross-sections based on CST method (B.M.Kulfan, 2007)
clearall
clc

%% global design parameters
%fuselage
lfus=37.25;
wfus=3.92;
hfus=wfus; % default
%wing
span=35.8;
wingxyz=[11.5 0 -1.50];
chordroot=5.8;
chordcrank=4.0;
taper=0.2;
sweep=24.7;
inc=3;
twist=-1;
dih=3.32;
tuc=0.085;
crankpos=4.36; % from centreline
%htail
htailxyz= [31 0.6583 0.9875];
htailchord= 4;
htailtaper= 0.40;
htailsweep= 27.500;
htaildih= 3.3200;
htailtuc= 0.0850;
htailspan= 11.44;
%vtail
vtailxyz= [29.67 0 1.90];
vtailchord= 5;
vtailtaper= 0.4130;
vtailsweep= 34;
vtailtuc= 0.0850;
vtailspan= 6.26;
%nacelle
lnac=4.44;
dnac=2.0;
nacxyz=[0 5.4 -2.7];

%% VSgeo init
% fuselage distribution by default
fusecross(1).ncu= 0.5000;
fusecross(1).ncl= 0.5000;
fusecross(1).x= 0;
fusecross(1).z= -0.14*hfus; %-0.5488;
fusecross(1).W= 0.5*wfus; % 1.9600;
fusecross(1).H= 0.5*hfus; % 1.9600;

fusecross(2).ncu= 0.5000;
fusecross(2).ncl= 0.5000;
fusecross(2).x= 0.04*lfus; % 1.4900;
fusecross(2).z= -0.14*hfus; %-0.5488;
fusecross(2).W= 0.5*wfus; % 1.9600;
fusecross(2).H= 0.5*hfus; % 1.9600;

fusecross(3).ncu= 0.5000;
fusecross(3).ncl= 0.5000;
fusecross(3).x= 0.14*lfus; % 5.2150;
fusecross(3).z= 0;
fusecross(3).W= wfus; %3.9200;
fusecross(3).H= hfus; %3.9200;

fusecross(4).ncu= 0.5000;
fusecross(4).ncl= 0.0500;
```

```

fusecross(4).x= 0.34*lfus; %12.6650;
fusecross(4).z= 0;
fusecross(4).W= wfus; %3.9200;
fusecross(4).H= hfus; %3.9200;

fusecross(5).ncu= 0.5000;
fusecross(5).ncl= 0.0500;
fusecross(5).x= 0.5*lfus; %18.6250;
fusecross(5).z= 0;
fusecross(5).W= wfus; %3.9200;
fusecross(5).H= hfus; %3.9200;

fusecross(6).ncu= 0.5000;
fusecross(6).ncl= 0.5000;
fusecross(6).x= 0.675*lfus; % 25.1438;
fusecross(6).z= 0;
fusecross(6).W= wfus; %3.9200;
fusecross(6).H= hfus; %3.9200;

fusecross(7).ncu= 0.4000;
fusecross(7).ncl= 0.5000;
fusecross(7).x= 0.765*lfus; %28.4963;
fusecross(7).z= 0.05*hfus; %0.1960;
fusecross(7).W= 0.9*wfus; %3.5280;
fusecross(7).H= 0.9*hfus; %3.5280;

fusecross(8).ncu= 0.4000;
fusecross(8).ncl= 0.5000;
fusecross(8).x= 0.94*lfus; %35.0150;
fusecross(8).z= 0.28*hfus; %1.0976;
fusecross(8).W= 0.5*wfus; %1.9600;
fusecross(8).H= 0.45*hfus; %1.7640;

fusecross(9).ncu= 0.5000;
fusecross(9).ncl= 0.5000;
fusecross(9).x= lfus; %37.2500;
fusecross(9).z= 0.34*hfus; %1.3328;
fusecross(9).W= 0.25*wfus; %0.9800;
fusecross(9).H= 0.15*hfus; %0.5880;

% main wing
if wingxyz(1)==0
wing(1).x_pos= 0.308*lfus; % wingxyz(1);
else
wing(1).x_pos =wingxyz(1);
end
wing(1).y_pos= wfus/2.2;
wing(1).z_pos= wingxyz(3);
wing(1).chord= chordroot;
wing(1).taper= chordcrank/chordroot;
wing(1).sweep= sweep;
wing(1).inc= inc;
wing(1).twist= twist;
wing(1).dih= dih;
wing(1).tuc= tuc;
wing(1).station= 0;
wing(1).Bu1= 1;
wing(1).Bu2= 1;
wing(1).Bu3= 1;
wing(1).Bu4= 1;
wing(1).Bu5= 1;
wing(1).Bu6= 1;
wing(1).Bl1= 1;
wing(1).Bl2= 1;
wing(1).Bl3= 1;
wing(1).Bl4= 1;
wing(1).Bl5= 1;
wing(1).Bl6= 1;

wing(2)=wing(1);
wing(2).x_pos= 0;
wing(2).z_pos= 0;
wing(2).chord= chordcrank;
wing(2).taper= taper/wing(1).taper;
wing(2).sweep= sweep;

```

```

wing(2).inc= wing(1).inc+wing(1).twist;
wing(2).twist= twist;
wing(2).dih= dih;
wing(2).tuc= tuc;
wing(2).station= crankpos;

wing(3)=wing(2);
wing(3).x_pos= 0;
wing(3).z_pos= 0;
wing(3).chord= wing(2).taper*wing(2).chord;
wing(3).taper= 0;
wing(3).sweep= 0;
wing(3).inc= wing(2).inc+wing(2).twist;
wing(3).twist= 0;
wing(3).dih= 0;
wing(3).tuc= tuc;
wing(3).station= span/2;

[wing]=xzpos3(wing);

if htaixyz(1)==0
htail(1).x_pos= 0.832*lfus; %htaixyz(1); %31;
else htail(1).x_pos= htaixyz(1);
end
htail(1).y_pos= htaixyz(2); %0.6583;
htail(1).z_pos= htaixyz(3); %0.9875;
htail(1).chord= htailchord; %4;
htail(1).taper= htaitaper; %0.4000;
htail(1).sweep= htailsweep; %27.500;
htail(1).inc= 0;
htail(1).twist= 0;
htail(1).dih= htaildih; % 3.3200;
htail(1).tuc= htaituc; % 0.0850;
htail(1).station= 0;
htail(1).Bu1= 1;
htail(1).Bu2= 1;
htail(1).Bu3= 1;
htail(1).Bu4= 1;
htail(1).Bu5= 1;
htail(1).Bu6= 1;
htail(1).Bl1= 1;
htail(1).Bl2= 1;
htail(1).Bl3= 1;
htail(1).Bl4= 1;
htail(1).Bl5= 1;
htail(1).Bl6= 1;

htail(2)=htail(1);
htail(2).x_pos= 0;
htail(2).y_pos= 0;
htail(2).z_pos= 0;
htail(2).chord= htail(1).taper*htail(1).chord;0.8;
htail(2).taper= 0;
htail(2).sweep= 0;
htail(2).inc= 0;
htail(2).twist= 0;
htail(2).dih= 0;
htail(2).tuc= htaituc;
htail(2).station= htailspar/2; %5.72;

if vtailxyz(1)==0
vtail(1).x_pos= 0.797*lfus; %vtailxyz(1); %29.67;
else vtail(1).x_pos= vtailxyz(1);
end
vtail(1).y_pos= vtailxyz(2); %0;
vtail(1).z_pos= vtailxyz(3); %1.90;
vtail(1).chord= vtailchord; %5;
vtail(1).taper= vtailtaper; %0.4130;
vtail(1).sweep= vtailsweep; %34;
vtail(1).inc= 0;
vtail(1).twist= 0;
vtail(1).dih= 0;
vtail(1).tuc= vtailtuc; %0.0850;
vtail(1).span= vtailspan; %6.26;

```

```

nacin.l= lnac; % 4.4400;
nacin.d= dnac; % 2;
nacin.x= nacxyz(1); %5.4000;
if nacxyz(1)==0
    nacin.x=wing(2).x_pos-0.5*nacin.l;
end
nacin.y= nacxyz(2); %5.4000;
nacin.z= nacxyz(3); %-2.7000;

pylin.x_pos=nacin.x;
pylin.y_pos=nacin.y;
pylin.z_pos=nacin.z+0.42*nacin.d; % was 0.5
pylin.span=wing(2).z_pos-pylin.z_pos;
pylin.chord=nacin.l;
pylin.sweep=atand((wing(2).x_pos-pylin.x_pos)/pylin.span);
pylin.tuc=0.085;
pylin.taper=1-(wing(2).x_pos-pylin.x_pos)/pylin.chord;

%% fuselage
[fusesec,fusewidth,wAreaFus]=fuse_surf3(fusecross);

%% wing
[wingsurf,wAwing,wRootarea]=CST_Wing4(wing);
% search for fuselage section where main wing join
for i=1:length(fusesec)
if fusesec(i).x(1,1)> wingsurf(1).x(1,1)
    njointsec=i-1;
break
end
end
[wingsurfmod,wAwingdel]=
fus_wing_join(fusesec(njointsec),fusesec(njointsec+1),wingsurf);

%% htail and vtail
[htailsurf,wAhtail,htRootArea]=CST_Wing4(htail);
% search for fuselage section where htail and vtail join
for i=1:length(fusesec)
if fusesec(i).x(1,1)> htailsurf.x(1,1)
    njointsec=i-1;
break
end
end
[htailsurfmod,wAhtaildel]=
fus_tail_join(fusesec(njointsec),fusesec(njointsec+1),htailsurf);
[vtailsurf,wAvtail,vtRootArea]=VTail4(vtail); % width & height of fuselage

%% nacelle and pylon
[nacsurf,nacintsurf,wAnac]=nacelle3(nacin);
[pylsurf,wApyl,pylrootarea]=VTail4(pylin);

%% plotting
figure
axisequal
axisauto
holdon
fus_plot(fusesec)
wingplot(wingsurf)
wingplot(htailsurf)
vtailplot(vtailsurf)
nacplot(nacsurf,nacintsurf)
pylplot(pylsurf)
holdoff

%% wetted area
wAfus=wAreaFus-wRootarea-htRootArea-vtRootArea
wAnacs=2*(wAnac-pylrootarea)
wApyls=2*wApyl
end

```

```

function [fus,wfus,farea]= fuse_surf3(cross)
% generate fuselage surfaces
farea=0;
wfus=0;
nsec=length(cross)-1;
for i=1:nsec % number of sections
C1=cross(i);
C2=cross(i+1);

if C2.W>wfus
wfus=C2.W;
end
secL=C2.x-C1.x;
nx=round(secL/0.5);

if i==1
nd1=0.5; % nose term
nx=nx*2;
else
nd1=0; % open front
end

if i==nsec
nd2=0.05; % aft term
nx=nx*2;
else
nd2=0; % open aft
end
%fuseside cosine spacing
phi = linspace(0,1,nx); % phi=linspace(0,1,50);
etacons = linspace(0,1,30);
eta=0.5*(1-cos(etacons*pi));
[phi,eta]=meshgrid(phi,eta);

ncu=blend(C1.ncu,C2.ncu,phi);
ncl=blend(C1.ncl,C2.ncl,phi);
scu=1;
ccu=eta.^ncu.*(1-eta).^ncu;
scl=1;
ccl=eta.^ncl.*(1-eta).^ncl;
% normalised to unit shape
ccu=normal(ccu);
ccl=normal(ccl);
% determine max position
if i==2
W=halfblend2(C1.W/2,C2.W/2,phi);
H=halfblend2(C1.H/2,C2.H/2,phi);
H2=linear(C1.H/2,C2.H/2,phi);
y1=-(1-2*eta).*(W);
z1=(scu*ccu).*(H);
z11=-(scl*ccl).*(H2);
elseif i==nsec
W=linear(C1.W/2,C2.W/2,phi);
H=linear(C1.H/2,C2.H/2,phi);
H2=linear(C1.H/2,C2.H/2,phi);
y1=-(1-2*eta).*(W);
z1=(scu*ccu).*(H);
z11=-(scl*ccl).*(H2);
else
W=linear(C1.W/2,C2.W/2,phi);
H=linear(C1.H/2,C2.H/2,phi);
y1=-(1-2*eta).*(W);
z1=(scu*ccu).*(H);
z11=-(scl*ccl).*(H);
end

%% distribution part
sd=1;
cd=phi.^nd1.*(1-phi).^nd2;
zshift=linear(C1.z,C2.z,phi);
fus(i).x=phi*secL+C1.x;
fus(i).y=-(sd.*cd).*y1;
fus(i).zu=(sd.*cd).*z1+zshift;
fus(i).zl=(sd.*cd).*z11+zshift;

```

```

area_i=area3dsum(fus(i).x,fus(i).y,fus(i).zu)+area3dsum(fus(i).x,fus(i).y,fus(i).zl);
farea=farea+area_i;
end
end

function linvec=linvar(initvec,finvec,grid)
linvec=ones(length(initvec),length(grid));
for i=1:length(initvec)
for j=1:length(grid)
linvec(i,j)=(1-grid(j))*initvec(i)+grid(j)*finvec(i);
end
end
end

function [nc]=linear(nc1,nc2,phi)
nc= (1-phi).*nc1+phi.*nc2;
end

function [nc]=halfblend2(nc1,nc2,phi)
tannc1=1.5;
nc=(1-3*phi.^2+2*phi.^3).*nc1+(-2*phi.^3+3*phi.^2).*nc2+(phi-2*phi.^2+phi.^3).*tannc1;
end

function [nc]=blend(nc1,nc2,phi)
nc=(1-3*phi.^2+2*phi.^3).*nc1+(-2*phi.^3+3*phi.^2).*nc2;
end

function [cnorm]= normal(cn)
[ row, col]=size(cn);
cnorm=zeros(row,col);
for i=1:col
height=max(cn(:,i));
cnorm(:,i)=cn(:,i)/height;
end
end

function area=area3d(A,B,C,D)
AB=B-A;
AD=D-A;
CB=B-C;
CD=D-C;
area=0.5*norm(cross(AB,AD))+0.5*norm(cross(CB,CD));
end

function areasum=area3dsum(x,y,z)
% compute surface area
[ni,nj]=size(x);
area=zeros(ni-1,nj-1);
areasum=0;
for i=1:ni-1
for j=1:nj-1
A=[x(i,j),y(i,j),z(i,j)];
B=[x(i,j+1),y(i,j+1),z(i,j+1)];
C=[x(i+1,j+1),y(i+1,j+1),z(i+1,j+1)];
D=[x(i+1,j),y(i+1,j),z(i+1,j)];
area(i,j)=area3d(A,B,C,D);
areasum=areasum+area(i,j);
end
end
end

function [wingpan,wAwing,xrootarea]=CST_Wing4(wing)
nsec=length(wing)-1;
wAwing=0;
nx=round(wing(1).chord/0.5);
for i=1:nsec
SPAN = wing(i+1).station-wing(i).station;
CR = wing(i).chord; % root chord (m)
DIH = wing(i).dih; % dihedral angle (deg)
SWEEP = wing(i).sweep; % wing sweep angle (deg)
TWIST = wing(i).twist; % twist anngle (deg)
INC = wing(i).inc; % wing incidence angle (deg)
TCA = wing(i).tuc; % thickness to chord ratio
TR = wing(i).taper; % taper ratio

```



```

x_pos = wing(i).x_pos;           % root setting x-position
y_pos = wing(i).y_pos;
z_pos = wing(i).z_pos;           % root setting z-position

ny=round(SPAN);
phicons = linspace (0,1,nx);     % phi = linspace (0,1,20);
phi=0.5*(1-cos(phicons*pi));     % eta = linspace (0,1,20);
eta = linspace (0,1,ny);

%% modified variable for calculation
L=CR*(1-(1-TR)*eta);
H=TCA*L/2;
Zt=tand(DIH)*SPAN;               % wingtip up distance
Zle=(tand(INC+TWIST*eta)).*L;    % incidence angle
swle=tand(SWEEP)*SPAN;          % xle sweep distance
NC1=0.5;
NC2=1.0;
% airfoil cross-section
n=5;
bp=BPO(n,phi);

%% inputs for CST design parameters
% designer's choice of airfoil section
Au=[wing(i).Bu1 wing(i).Bu2 wing(i).Bu3 wing(i).Bu4 wing(i).Bu5 wing(i).Bu6];
Al=[wing(i).Bl1 wing(i).Bl2 wing(i).Bl3 wing(i).Bl4 wing(i).Bl5 wing(i).Bl6];

Auf=[wing(i+1).Bu1 wing(i+1).Bu2 wing(i+1).Bu3 wing(i+1).Bu4 wing(i+1).Bu5
wing(i+1).Bu6];
Alf=[wing(i+1).Bl1 wing(i+1).Bl2 wing(i+1).Bl3 wing(i+1).Bl4 wing(i+1).Bl5
wing(i+1).Bl6];

Au_lin=linvar(Au,Auf,eta);
Al_lin=linvar(Al,Alf,eta);

Scu=bp*Au_lin;
Scl=bp*Al_lin;

x=zeros(nx,ny);
y=zeros(nx,ny);
Cc=zeros(nx,1);
psiu=zeros(nx,ny);
psil=zeros(nx,ny);
for k=1:nx
for j=1:ny
x(k,j)=(phi(k)*L(j))+swle*eta(j);
y(k,j)=eta(j)*SPAN;
Cc(k)=(phi(k)^(NC1))*((1-phi(k))^(NC2));
psiu(k,j)=Scu(k,j)*Cc(k);
psil(k,j)=-Scl(k,j)*Cc(k);
end
end

maxu=max(psiu);
maxl=-min(psil);

Zu=psiu/max(maxu);
Zl=psil/max(maxl);

for m=1:nx
for j=1:ny
Zu(m,j)=Zle(j)*(1-phi(m))+Zu(m,j)*H(j)+Zt*eta(j);
Zl(m,j)=Zle(j)*(1-phi(m))+Zl(m,j)*H(j)+Zt*eta(j);
end
end

% translate the wing to ref position (x_pos,z_pos)
wingpan(i).x = x+x_pos;
wingpan(i).y = y+y_pos;
wingpan(i).zu = Zu+z_pos;
wingpan(i).zl = Zl+z_pos;

wingpan(i).wAwU=area3dsum(wingpan(i).x,wingpan(i).y,wingpan(i).zu);
wingpan(i).wAwL=area3dsum(wingpan(i).x,wingpan(i).y,wingpan(i).zl);

```

```

wAwing=wAwing+(wingpan(i).wAwU+wingpan(i).wAwL)*2;
end
rootx=wingpan(1).x(:,1);
rootzu=wingpan(1).zu(:,1);
rootzl=wingpan(1).zl(:,1);

xrootareau=polyarea(rootx,rootzu);
xrootareal=polyarea(rootx,rootzl);
xrootarea=2*(xrootareau+xrootareal);
end

function [B]=BPO(n,x)
K=zeros(n);
B=zeros(length(x),n);
for r=1:n+1
    K(r)=mfun('binomial',n,r-1);
B(:,r)=K(r)*x.^(r-1).*((1-x).^(n-r+1));
end
end

function [vtailsurf,wAvt,vtrootarea]=VTail4(vtail)
SPAN=vtail(1).span;           % wing span (m)
CR=vtail(1).chord;           % root chord (m)
DIH=0;                        % dihedral angle (deg)
SWEEP=vtail(1).sweep;        % wing sweep angle (deg)
TWIST=0;                      % twist anngle (deg)
INC=0;                        % wing incidence angle (deg)
TCA=vtail(1).tuc;            % thickness to chord ratio
TR=vtail(1).taper;           % taper ratio
x_pos=vtail(1).x_pos;
y_pos=vtail(1).y_pos;
z_pos=vtail(1).z_pos;

nx=round(CR/0.5);
ny=round(SPAN);
if SPAN<1
    nx=nx*2;
    ny=2;
end
phicons = linspace(0,1,nx);    % phi = linspace(0,1,20);
phi=0.5*(1-cos(phicons*pi));
eta = linspace(0,1,ny);

L=CR*(1-(1-TR)*eta); % chord length spanwise
H=TCA*L/2;

Yt=tand(DIH)*SPAN;           % wingtip up distance
Yle=(tand(INC+TWIST*eta)).*L; % incidence angle
swle=tand(SWEEP)*SPAN;      % xle sweep distance

NC1=0.5;
NC2=1.0;

n=5;
bp=BPO(n,phi);
Aui=ones(6,ny);
Sc=bp*Aui;

x=zeros(nx,ny);
Z=zeros(nx,ny);
Cc=zeros(nx,1);
psiu=zeros(nx,ny);
psil=zeros(nx,ny);
for i=1:nx
    for j=1:ny
        x(i,j)=(phi(i)*L(j))+swle*eta(j);
        Z(i,j)=eta(j)*SPAN;
        Cc(i)=(phi(i)^(NC1))*((1-phi(i))^(NC2));
        psiu(i,j)=Sc(i)*Cc(i);
        psil(i,j)=-psiu(i,j);
    end
end

maxu=max(psiu);

```

```

maxl=-min(ψil);

Yu=ψiu/max(maxu);
Yl=ψil/max(maxl);
yr=zeros(nx,ny);
yl=zeros(nx,ny);
for i=1:length(phi)
for j=1:length(eta)
    yr(i,j)=Yle(j)*(1-phi(i))+Yu(i,j)*H(j)+Yt*eta(j);
    yl(i,j)=Yle(j)*(1-phi(i))+Yl(i,j)*H(j)+Yt*eta(j);
end
end

% translate
x = x+x_pos;
yr= yr+y_pos;
yl= yl+y_pos;
z = Z+z_pos;

wAvtR=area3dsum(x,yr,z);
wAvtL=area3dsum(x,yl,z);
wAvt=wAvtR+wAvtL;

vtailsurf.x=x;
vtailsurf.yr=yr;
vtailsurf.yl=yl;
vtailsurf.z=z;

rootx=vtailsurf.x(:,1);
rootyr=vtailsurf.yr(:,1);

vrootareau=polyarea(rootx,rootyr);
vrootarea=2*(vrootareau);
end

function [nac,int,nacArea]=nacelle3(nacin)

%% inputs
xnac=nacin.x;
ynac=nacin.y;
znac=nacin.z;
lnac=nacin.l;
dnac=nacin.d;
nx=round(lnac/0.25);
phi=linspace(0,1,nx);
ny=round(dnac/0.2);
etacons=linspace(0,1,ny);
eta=0.5*(1-cos(etacons*pi));
% default nacelle shape
rout=0.6*phi.^3-1.8*phi.^2+1.2.*phi+1;
nyi=ones(ny,1);
sd=nyi*rout/max(rout); %%mod here

%% cross-section shapes
C1.ncu=0.5;
C1.ncl=0.25;
C2.ncu=0.5;
C2.ncl=0.5;
[phi,eta]=meshgrid(phi,eta);
ncu=blend(C1.ncu,C2.ncu,phi);
ncl=blend(C1.ncl,C2.ncl,phi);

scu=1;
ccu=eta.^ncu.*(1-eta).^ncu;
scl=1;
ccl=eta.^ncl.*(1-eta).^ncl;

% normalised to unit shape per length
ccu=normal(ccu);
ccl=normal(ccl);

y1=-(1/2-eta);
z1=(scu*ccu);
z11=-(scl*ccl);

```

```

%% distribution part
nac.x=phi*lnac+xnac;
nac.y=(sd).*y1*dnac+ynac;
nac.zu=(sd).*z1*dnac/2+znac;
nac.zl=(sd).*z11*dnac/2+znac;
nacArea=area3dsum(nac.x,nac.y,nac.zu)+area3dsum(nac.x,nac.y,nac.zl);

%% inner tube
lnacint=lnac/4;
phi=linspace(0,1,(nx/4));
etacons=linspace(0,1,ny);
eta=0.5*(1-cos(etacons*pi));
rint=-0.5*phi.^3+phi.^2-0.5*phi+1;
sdint=nyi*rint/max(rout);
C1.ncu=0.5;
C1.ncl=0.25;
C2.ncu=0.5;
C2.ncl=0.5;

[phi,eta]=meshgrid(phi,eta);
ncu=blend(C1.ncu,C2.ncu,phi);
ncl=blend(C1.ncl,C2.ncl,phi);

scu=1;
ccu=eta.^ncu.*(1-eta).^ncu;
scl=1;
ccl=eta.^ncl.*(1-eta).^ncl;

% normalised to unit shape
ccu=normal(ccu);
ccl=normal(ccl);

y1=-(1/2-eta);
z1=(scu*ccu);
z11=-(scl*ccl);

%% distribution par
int.x=phi*lnacint+xnac;
int.y=-(sdint).*y1*dnac+ynac;
int.zu=(sdint).*z1*dnac/2+znac;
int.zl=(sdint).*z11*dnac/2+znac;
end

function af=xzpos3(af)
for i=1:length(af)-1
    af(i+1).x_pos=af(i).x_pos+(af(i+1).station-af(i).station)*tand(af(i).sweep);
    af(i+1).y_pos=af(i).y_pos+(af(i+1).station-af(i).station);
    af(i+1).z_pos=af(i).z_pos+(af(i+1).station-af(i).station)*tand(af(i).dih);
end
end

function fus_plot(fus)
for i=1:length(fus)
    surf(fus(i).x,fus(i).y,fus(i).zu)
    surf(fus(i).x,fus(i).y,fus(i).zl)
end
end

function wingplot(wing)

for i=1:length(wing)
    surf(wing(i).x,wing(i).y,wing(i).zu)
hold on
    surf(wing(i).x,wing(i).y,wing(i).zl)
    surf(wing(i).x,-wing(i).y,wing(i).zu)
    surf(wing(i).x,-wing(i).y,wing(i).zl)
end
end

function vtailplot(vtailsurf)
surf(vtailsurf.x,vtailsurf.yr,vtailsurf.z)
hold on
surf(vtailsurf.x,vtailsurf.yl,vtailsurf.z)

```

```

end

function nacplot(nac,int)
surf(nac.x,nac.y,nac.zu)
surf(nac.x,nac.y,nac.zl)
surf(nac.x,-nac.y,nac.zu)
surf(nac.x,-nac.y,nac.zl)

surf(int.x,int.y,int.zu)
surf(int.x,int.y,int.zl)
surf(int.x,-int.y,int.zu)
surf(int.x,-int.y,int.zl)
end

function pylplot(pylsurf)
surf(pylsurf.x,pylsurf.yr,pylsurf.z)
surf(pylsurf.x,pylsurf.yl,pylsurf.z)
surf(pylsurf.x,-pylsurf.yr,pylsurf.z)
surf(pylsurf.x,-pylsurf.yl,pylsurf.z)
end

function [wingsurfmod,wAwingmod]= fus_wing_join(fus1,fus2,wingsurf)
wingsurfmod=wingsurf;

xjoin=[fus1.x(2,:), fus2.x(2,:)];
yjoin=[fus1.y(2,:), fus2.y(2,:)];

wingx=wingsurf(1).x;
wingy=wingsurf(1).y;
wingzu=wingsurf(1).zu;
wingzl=wingsurf(1).zl;

nxint=length(wingx);
xint=zeros(nxint,1);
yint=zeros(nxint,1);
zuint=zeros(nxint,1);
zlint=zeros(nxint,1);

for i = 1:nxint
xwing=[wingx(i,1) wingx(i,2)];
ywing=[wingy(i,1) wingy(i,2)];
[xint(i),yint(i)]=curveintersect(xjoin,yjoin,xwing,ywing);
zuint(i)=(yint(i)-wingy(i,1))/(wingy(i,2)-wingy(i,1))*(wingzu(i,2)-
wingzu(i,1))+wingzu(i,1);
zlint(i)=(yint(i)-wingy(i,1))/(wingy(i,2)-wingy(i,1))*(wingzl(i,2)-
wingzl(i,1))+wingzl(i,1);
end
%original geo
wingxmod=wingx;
wingymod=wingy;
wingzumod=wingzu;
wingzlmod=wingzl;

%% delete inside
for i=1:nxint
wingxmod(i,1)=xint(i);
wingymod(i,1)=yint(i);
wingzumod(i,1)=zuint(i);
wingzlmod(i,1)=zlint(i);
end
wingsurfmod(1).x=wingxmod;
wingsurfmod(1).y=wingymod;
wingsurfmod(1).zu=wingzumod;
wingsurfmod(1).zl=wingzlmod;

wareau=area3dsum(wingsurfmod(1).x,wingsurfmod(1).y,wingsurfmod(1).zu);
wareal=area3dsum(wingsurfmod(1).x,wingsurfmod(1).y,wingsurfmod(1).zl);
wAwingmod=2*(wareau+wareal);

for n=2:length(wingsurf)
wingsurfmod(n).wAwU=area3dsum(wingsurfmod(n).x,wingsurfmod(n).y,wingsurfmod(n).zu);
wingsurfmod(n).wAwL=area3dsum(wingsurfmod(n).x,wingsurfmod(n).y,wingsurfmod(n).zl);
wAwingmod=wAwingmod+(wingsurfmod(n).wAwU+wingsurfmod(n).wAwL)*2;

```

```

end
end

function [tailsurfmod,hWarea]= fus_tail_join(fus1,fus2,tailsurf)

xjoin=[fus1.x(1,:), fus2.x(1,:)];
yjoin=[fus1.y(1,:), fus2.y(1,:)];

htx=tailsurf.x;
hty=tailsurf.y;
htzu=tailsurf.zu;
htzl=tailsurf.zl;

nxint=length(htx);
xint=zeros(nxint,1);
yint=zeros(nxint,1);
zuint=zeros(nxint,1);
zlint=zeros(nxint,1);

for i = 1:nxint
xht=[htx(i,1) htx(i,2)];
yht=[hty(i,1) hty(i,2)];
[xint(i),yint(i)]=curveintersect(xjoin,yjoin,xht,yht);
zuint(i)=(yint(i)-hty(i,1))/(hty(i,2)-hty(i,1))*(htzu(i,2)-htzu(i,1))+htzu(i,1);
zlint(i)=(yint(i)-hty(i,1))/(hty(i,2)-hty(i,1))*(htzl(i,2)-htzl(i,1))+htzl(i,1);
end
%original geo
htxmod=htx;
htymod=hty;
htzumod=htzu;
htzlmod=htzl;

%% delete inside
for i=1:nxint
htxmod(i,1)=xint(i);
htymod(i,1)=yint(i);
htzumod(i,1)=zuint(i);
htzlmod(i,1)=zlint(i);
end

hareau=area3dsum(htxmod,htymod,htzumod);
hareal=area3dsum(htxmod,htymod,htzlmod);
hWarea=2*(hareau+hareal);

tailsurfmod.x=htxmod;
tailsurfmod.y=htymod;
tailsurfmod.zu=htzumod;
tailsurfmod.zl=htzlmod;

end

```