

# *Geometric Parameterisation and Aerodynamic Shape Optimisation*



Name: Feng Zhu

Supervisor: Prof Ning Qin

Email: [f.zhu@sheffield.ac.uk](mailto:f.zhu@sheffield.ac.uk)

Address: Department of Mechanical Engineering  
University of Sheffield  
S1 3JD

PhD Thesis

# Abstract

Aerodynamic optimisation plays an increasingly important role in the aircraft industry. In aerodynamic optimisation, shape parameterisation is the key technique, since it determines the design space. The ideal parameterisation method should be able to provide a high level of flexibility with a low number of design variables to reduce the complexity of the design space. In this work, the Class/Shape Function Transformation (CST) method is investigated for geometric representation of an entire transport aircraft for the purpose of aerodynamic optimisation. It is then further developed for an entire passenger transport aircraft, including such components as the wing, horizontal tail plane, vertical tail plane, fuselage, belly fairing, wingtip device, nacelle, flap tracking fairing and pylon. This work presents the parameterisation of these components in detail using the CST methods for the reference of future aerodynamic optimisation work. The intersection line calculation method between CST components is presented for future entire aircraft optimisation. The performance of the CST has been tested as well, and it found a few drawbacks of the CST methods; for example, it cannot provide some key intuitive design parameters and can lose the accuracy in the wing leading edge area. Therefore, two derivatives of the CST method are proposed: one is called the intuitive CST method (iCST), which is to transform the CST parameters to intuitive design parameters; the other is called the RCST method, which is able to increase the fitting accuracy of the original CST method with fewer design variables. Their performances are studied by comparing them regarding their accuracy in inversely fitting a wide range of aerofoils. Finally, the CST method is also developed to represent the shock control bump, which has better curvature continuity than cubic polynomials.

The aerodynamic optimisation study based on adjoint approaches is carried out using the above parameterisation methods. Optimisation was performed on two-dimensional cases to make a preliminary investigation of the performances of the above parameterisation methods. The results showed that all of CST, iCST and RCST parameterisation methods are able to successfully reduce the drag. The results of the CST methods showed the

lower order CST is able to provide fast convergence, and the high order CST is able to provide more flexibility and more local control of the shape to reach better optimal solution. The iCST providing intuitive parameters is improving the process of setup constraints, which is useful for aerofoil optimisation. The RCST showed good performance in aerodynamic optimisation in terms of convergence rate, number of design variables, low order of polynomials and smoothness of the shape. This work provides a reference to designer for choosing suitable parameterisation method in these three methods regarding specific requirement. The shock control bump optimisation on 2D aerofoil is performed to compare three shock control bump parameterisation methods. The results showed the CST parameterisation method is promising for shock control bump optimisation.

Three-dimensional optimisation tests, including wing and winglet drag minimisation, were performed using the above parameterisation methods. The results showed that the CST methods are able to handle three-dimensional wing optimisation. It also investigated the effect of the order of CST method in optimisation. The results showed the lower order CST already performed well in optimisation in terms of optimal results and convergence rate. The optimisation also discussed the importance of using  $C_{mx}$  constraint in aerodynamic optimisation. In the winglet test cases, it showed the CST methods and adjoint approach are able to perform winglet optimisation. The drag of four winglets are successfully reduced. The downward winglet showed the potential benefits in terms of lower wing root bending momentum. At the end, the shock control bump optimisation using CST method on 3D wing has been performed. The results showed the mesh adjoint methods is able to identify the sensitive area for deploying shock control bumps and the CST shock control bump successfully reduced the wave drag.

## Acknowledgements

I sincerely appreciate my supervisor Prof. Ning Qin for his guidance. He offered me this opportunity and led me into a very interesting aerodynamic design and optimisation area, and has provided me with endless helpful advice to overcome all difficulties throughout the years. I also would like to thank present and former colleagues in the Aerodynamics Group at the University of Sheffield for their fruitful discussions.

This work was funded by a Scholarship from Airbus within the CFMS programme. I would like to thank Stefano Tursi, Murray Cross, Francois Gallard and Kasidit Leoviriyakit from Airbus. This work would have been impossible without their support and help. I would also like to thank DLR(German Aerospace Research Center) for providing the experimental version of the TAU solver with flow and mesh adjoint capability. I am indebted to Dr Caslav Illic for his availability and all the help and explanations he gave me on the TAU solver.

Finally, my gratitude goes to my family for always supporting me through difficult times over the past few years.

# List of Contents

Abstract .....	i
Acknowledgements .....	iii
List of Contents .....	iv
List of Figures .....	viii
List of Tables .....	xvii
Nomenclature .....	xviii
Chapter 1 Introduction .....	1
1.1 Background .....	1
1.2 Outline of thesis .....	4
PART I	
Chapter 2 Literature Review of Geometric Parameterisation .....	6
2.1 Discrete methods .....	7
2.2 Analytical methods .....	8
2.3 Polynomial, spline methods, CAD-based and free-form deformation .....	10
2.4 PARSEC parameterisation methods .....	23
2.5 Class/shape function transformation (CST) methods .....	34
2.6 Comparison of parameterisation methods .....	41
Chapter 3 Development of CST and PARSEC Methods in Two-Dimensional Aerofoils	48
3.1 Combination of CST and PARSEC: the intuitive CST method .....	48
3.2 Geometric inverse fitting test and results of iCST methods .....	52
3.2.1. Geometric inverse fitting .....	53
3.2.2. Inverse fitting test results .....	53
3.3 CST method with rational function (RCST) .....	58
3.4 Geometric fitting results of RCST .....	60

3.5 Conclusion .....	64
<b>Chapter 4 CST Parameterisation Method for the Entire Aircraft .....</b>	<b>66</b>
4.1 Parameterisation for wing type geometries.....	66
4.1.1. Standard CST for wing type geometries .....	66
4.1.2. Fitting accuracy of the standard CST method for a wing.....	70
4.1.3. RCST method for wing type geometries .....	86
4.1.4. Fitting accuracy of the RCST for a wing .....	87
4.2 CST parameterisation method for wing tip device.....	93
4.3 CST Parameterisation for fuselage (simplified forward, mid and tail cone parts) .....	111
4.3.1 Cylindrical fuselage .....	112
4.3.2 Nose fuselage.....	113
4.3.3 Rear fuselage.....	116
4.4 CST parameterisation for belly-fairing.....	117
4.5 CST parameterisation for the nacelle .....	123
4.6 CST parameterisation for flap tracking fairing (FTF) and pylon.....	126
4.7 CST parameterisation for three-dimensional shock bump local modification.....	130
4.8 Calculation of intersection line.....	136

## PART II

<b>Chapter 5 Governing Equation and Numerical Solver .....</b>	<b>147</b>
5.1 Governing equation .....	148
5.2 Reynolds-averaged Navier-Stokes (RANS) simulation and turbulence model .....	151
5.2.1 Spalart-Allmaras turbulence model .....	153
5.3 Finite volume method .....	154
5.4 Central convective fluxes.....	157
5.5 Construction of gradient.....	160
5.6 Temporal discretisation .....	162

Chapter 6 Discrete Adjoint Approach and Numerical Optimisation .....	165
6.1 Common methods to calculate sensitivities.....	167
6.2 Discrete adjoint methods .....	173
6.2.1 Discrete adjoint equation .....	173
6.2.2 Discrete adjoint solver with mesh deformation .....	177
6.3 Numerical optimisation .....	182
6.4 Mesh deformation.....	185
6.5 Optimisation framework .....	195
Chapter 7 Optimisation in Two-Dimensions .....	198
7.1 Two-dimensional aerofoil optimisation .....	198
7.2 Shock bump optimisation in the two-dimensional aerofoil .....	210
Chapter 8 Optimisation in Three-Dimensions .....	217
8.1 Wing optimisation using CST methods.....	218
8.1.1 Influence of different order of the CST methods on wing optimisation .....	218
8.1.2 Wing optimisation with rolling momentum constraint .....	231
8.2 Wing optimisation using RCST methods.....	239
8.3 Winglet optimisation .....	247
8.4 Shock bump optimisation on the wing.....	262
Chapter 9 Conclusion and Future Work .....	270
9.1 Summary.....	270
9.2 Future work .....	275
References.....	277
Publication List .....	301
Appendices.....	302
Appendix A: Derivatives of Bezier curve .....	302
Appendix B: Value of rational shape function of RCST method at the trailing edge .....	303

Appendix C Value at boundary of CST function with class parameters N1=1.0 and N2=1.0 ..	305
Appendix D Partial differentiation of geometry of fuselage and belly-fairing .....	305
Appendix E Objective function using target lift iteration.....	308

# List of Figures

Figure 1.1 Transport aircraft fuel efficiency (from Penner 1999) .....	2
Figure 2.1 Bezier curve with control points.....	12
Figure 2.2 The control box of FFD for ONERA M6 wing (Widhalm <i>et al.</i> 2007) .....	22
Figure 2.3 PARSEC method parameters definition.....	24
Figure 2.4 PARSEC parameters for DTE (Sobieczky 1998).....	27
Figure 2.5 PARSEC parameters for local control bump (Sobieczky 1998) .....	28
Figure 3.1 The intuitive CST parameterisation method.....	49
Figure 3.2 Geometric fitting for RAE 2822 using iCST, CST 12 and PARSEC 12 .....	54
Figure 3.3 Geometric fitting for RAE 5214 using iCST, CST 12 and PARSEC 12 .....	54
Figure 3.4 Geometric fitting for SC-20714 using iCST, CST 12 and PARSEC 12 .....	54
Figure 3.5 Geometric fitting for NLF 414F using iCST, CST 12 and PARSEC 12.....	55
Figure 3.6 Geometric fitting for NLF 416 using iCST, CST 12 and PARSEC 12.....	55
Figure 3.7 Geometric fitting for HSNLF 213 using iCST, CST 12 and PARSEC 12.....	56
Figure 3.8 Geometric fitting for S805A using iCST, CST 12 and PARSEC 12 .....	56
Figure 3.9 Geometric fitting for S809 using iCST, CST 12 and PARSEC 12 .....	57
Figure 3.10 Geometric fitting for S825 using iCST, CST 12 and PARSEC 12 .....	57
Figure 3.11 Geometric fitting for RAE 2822 using 6th order RCST.....	60
Figure 3.12 Geometric fitting for RAE 5214 using 6th order RCST.....	61
Figure 3.13 Geometric fitting for NASA SC-20714 using 6th order RCST .....	61
Figure 3.14 Geometric fitting for NLF 414F using 6th order RCST .....	61
Figure 3.15 Geometric fitting for NLF 416 using 6th order RCST .....	62
Figure 3.16 Geometric fitting for HSNLF 213 using 6th order RCST .....	62
Figure 3.17 Geometric fitting for S805A using 6th order RCST .....	62
Figure 3.18 Geometric fitting for S809 using 6th order RCST .....	63
Figure 3.19 Geometric fitting for S825 using 6th order RCST .....	63
Figure 4.1 Wing aerofoil section definition in the CST method.....	67
Figure 4.2 Leading edge x coordinates distribution and error .....	71
Figure 4.3 Leading edge height distribution and error .....	72

Figure 4.4 Non-dimensional trailing edge thickness distributions leading edge and error .....	72
Figure 4.5 Tangential value of twist angle distributions and error .....	72
Figure 4.6 Local chord distributions and error .....	73
Figure 4.7 The error contour of wing inverse fitting with <i>BPOX 6-BPOY 6</i> (left figure in metre) and <i>BPOX 6-BPOY 10</i> (right figure in metre) .....	74
Figure 4.8 The error contour of wing inverse fitting with <i>BPOX 6-BPOY 12</i> (left figure in metre) and <i>BPOX 10-BPOY 6</i> (right figure in metre) .....	75
Figure 4.9 The error contour of wing inverse fitting with <i>BPOX 10-BPOY 10</i> (left figure in metre) and <i>BPOX 12-BPOY 6</i> (right figure in metre) .....	75
Figure 4.10 The error contour of wing inverse fitting with <i>BPOX 12-BPOY 10</i> (left figure in metre) and <i>BPOX 12-BPOY 12</i> (right figure in metre) .....	75
Figure 4.11 The hybrid mesh of F6 wing for CFD study .....	77
Figure 4.12 The sections index and position on the wing .....	78
Figure 4.13 Comparisons of pressure distribution and wing shape on section 1 .....	79
Figure 4.14 Comparisons of pressure distribution and wing shape on section 2 .....	80
Figure 4.15 Comparisons of pressure distribution and wing shape on section 3 .....	80
Figure 4.16 Comparisons of pressure distribution and wing shape on section 4 .....	81
Figure 4.17 Comparisons of pressure distribution and wing shape on section 5 .....	82
Figure 4.18 Comparisons of pressure distribution and wing shape on section 6 .....	83
Figure 4.19 Comparisons of pressure distribution and wing shape on section 7 .....	83
Figure 4.20 Comparisons of pressure distribution and wing shape on section 8 .....	84
Figure 4.21 The HTP model using the CST methods .....	85
Figure 4.22 The VTP model using the CST methods .....	86
Figure 4.23 The error contour of wing inverse fitting with RCST <i>BPOX 6-BPOY 6</i> (in metre) .....	87
Figure 4.24 Comparisons of pressure distribution and wing shape at 10% span .....	89
Figure 4.25 Comparisons of pressure distribution and wing shape at 20% span .....	89
Figure 4.26 Comparisons of pressure distribution and wing shape at 30% span .....	90
Figure 4.27 Comparisons of pressure distribution and wing shape at 40% span .....	90
Figure 4.28 Comparisons of pressure distribution and wing shape at 50% span .....	90

Figure 4.29 Comparisons of pressure distribution and wing shape at 60% span .....	91
Figure 4.30 Comparisons of pressure distribution and wing shape at 70% span .....	91
Figure 4.31 Comparisons of pressure distribution and wing shape at 80% span .....	91
Figure 4.32 Comparisons of pressure distribution and wing shape at 90% span .....	92
Figure 4.33 The wing with winglet-1: wing in blue, transition part in green, winglet in red .....	96
Figure 4.34 The leading edge lines control points.....	97
Figure 4.35 The leading edge lines of transition part and winglet on the y-z plane.....	98
Figure 4.36 The extension of the wing on the y-z plane.....	99
Figure 4.37 The planform parameters for wing extension on the x-y plane.....	100
Figure 4.38 The leading and trailing edge lines and control point and polygon for wing extension on the x-y plane .....	101
Figure 4.39 The surface of wing extension.....	104
Figure 4.40 The translation relationship between wing extension and winglet-1 .....	105
Figure 4.41 The rotation relationship.....	106
Figure 4.42 The translation view on the x-y plane .....	106
Figure 4.43 The winglet-2 leading edge on the y-z plane and planform parameters.....	108
Figure 4.44 The planform parameters of wing extension for winglet-2 on the x-y plane .....	108
Figure 4.45 The surface of wing extension for winglet-2 on the x-y plane.....	109
Figure 4.46 The translation of wing extension to winglet-2.....	109
Figure 4.47 The surface of winglet-2.....	110
Figure 4.48 The winglet-3.....	111
Figure 4.49 The winglet-4.....	111
Figure 4.50 The three main parts of the fuselage.....	112
Figure 4.51 The CST parametric mid-part fuselage component.....	112
Figure 4.52 The forward part fuselage of an F4 aircraft with cabin .....	114
Figure 4.53 The CST parametric forward part fuselage .....	115
Figure 4.54 A CAD tail cone model .....	116
Figure 4.55 The CST parametric tail cone.....	117
Figure 4.56 Belly fairing model and its profile.....	118

Figure 4.57 The CST belly-fairing parametric model .....	121
Figure 4.58 The error contour of belly-fairing inverse fitting with <i>BPOX 6-BPOY 6</i> (left figure) and <i>BPOX 8-BPOY 8</i> (right figure).....	122
Figure 4.59 The error contour of belly-fairing inverse fitting with <i>BPOX 10-BPOY 10</i> (left figure) and <i>BPOX 12-BPOY 12</i> (right figure).....	122
Figure 4.60 The nacelle inlet using PARSEC intuitive parameters.....	124
Figure 4.61 The CST parametric nacelle model .....	125
Figure 4.62 The CST parametric model of FTF .....	127
Figure 4.63 A pylon CAD model.....	128
Figure 4.64 A simplified pylon model without root fairing.....	129
Figure 4.65 CST parametric model of a simplified pylon .....	130
Figure 4.66 Parameterisation for 2D shock control bump using piecewise polynomials (Wong 2006) .....	132
Figure 4.67 The bump curve using the CST methods .....	133
Figure 4.68 The 1 <sup>st</sup> derivative distribution of bump curve using the CST method .....	134
Figure 4.69 The 2 <sup>nd</sup> derivative distribution of bump curve using the CST method.....	134
Figure 4.70 The intersection lines of two high free-form surfaces.....	138
Figure 4.71 The intersection line between wing and belly-fairing .....	139
Figure 4.72 The approximation of step length.....	142
Figure 4.73 The important points on intersection line (Huang and Zhu 1997) .....	144
Figure 4.74 The example of intersection line between fuselage and belly-fairing .....	146
Figure 5.1 The control volume <i>i</i> in the finite volume method .....	155
Figure 5.2 The flux between cells <i>i</i> and <i>j</i> .....	156
Figure 5.3 The cell-vertex finite volume: black nodal points and grey lines form the primary grid, black lines form the secondary grid .....	157
Figure 5.4 Surrounding points used for the least square algorithm .....	161
Figure 6.1 The vector of $\lambda$ mesh over the surface of RAE 2822 aerofoil for drag .....	179
Figure 6.2 The vector of $\lambda$ mesh over the surface of RAE 2822 aerofoil for lift.....	179
Figure 6.3 The pressure distribution of RAE 2822.....	180
Figure 6.4 Validation of gradient of Cd for RAE 2822 using CST 7 <sup>th</sup> order.....	181
Figure 6.5 Validation of gradient of Cl for RAE 2822 using CST 7 <sup>th</sup> order .....	181

Figure 6.6 Blending function for grid node deformation computation, including the parameter radius full weight (RFW) and radius zero weight (RZW) .....	194
Figure 6.7 Adjoint optimisation framework of Surfgard .....	196
Figure 7.1 Mesh of RAE 2822.....	200
Figure 7.2 Cd (left) and Cl (right) optimisation history of 7 <sup>th</sup> order CST .....	201
Figure 7.3 Cd (left) and Cl (right) optimisation history of 10 <sup>th</sup> order CST .....	201
Figure 7.4 The contour of pressure coefficients of initial aerofoil (left) and optimum aerofoil (right) obtained by 7 <sup>th</sup> order CST .....	202
Figure 7.5 The contour of pressure coefficients of initial aerofoil (left) and optimum aerofoil (right) obtained by 10 <sup>th</sup> order CST .....	203
Figure 7.6 Cp distributions of initial aerofoil and optimal aerofoil obtained by 7 <sup>th</sup> order CST and 10 <sup>th</sup> order CST.....	203
Figure 7.7 Cd (left) and Cl (right) optimisation history of iCST .....	204
Figure 7.8 The contour of pressure coefficients of initial aerofoil (left) and optimum aerofoil (right) obtained by iCST.....	205
Figure 7.9 Cp distributions of initial and optimal aerofoils obtained by iCST method..	206
Figure 7.10 Cd (left) and Cl (right) optimisation history of RCST .....	207
Figure 7.11 The contour of pressure coefficients of initial aerofoil (left) and optimum aerofoil (right) obtained by RCST .....	208
Figure 7.12 Cp distributions of initial and optimal aerofoils obtained by RCST method .....	208
Figure 7.13 Comparison of the initial aerofoil and optimal aerofoils obtained by various parameterisation methods .....	209
Figure 7.14 Optimisation of drag using CST bump.....	212
Figure 7.15 Optimisation of drag using PARSEC bump.....	212
Figure 7.16 Optimisation of drag using standard cubic bump.....	212
Figure 7.17 Contour of pressure coefficient of aerofoil without bump (left) and aerofoil with optimal CST bump (right).....	213
Figure 7.18 Contour of pressure coefficient of aerofoil without bump (left) and aerofoil with optimal PARSEC bump (right).....	213

Figure 7.19 Contour of pressure coefficient of aerofoil without bump (left) and aerofoil with optimal standard cubic bump (right).....	214
Figure 7.20 Comparison of Cp distribution .....	214
Figure 7.21 Comparison of bump shape .....	215
Figure 8.1 Optimisation history of drag (left) and the 10 <sup>th</sup> torsion box volume (right) using CST with <i>BPOX 6-BPOY 6</i> .....	221
Figure 8.2 Optimisation history of drag (left) and the 10 <sup>th</sup> torsion box volume (right) using CST with <i>BPOX 6-BPOY 8</i> .....	221
Figure 8.3 Optimisation history of drag (left) and the 10 <sup>th</sup> torsion box volume (right) using CST with <i>BPOX 6-BPOY 10</i> .....	222
Figure 8.4 Optimisation history of drag (left) and the 10 <sup>th</sup> torsion box volume (right) using CST with <i>BPOX 10-BPOY 10</i> .....	222
Figure 8.5 The Cp contour plot of initial wing surface, lower (left) and upper (right) ..	223
Figure 8.6 The Cp contour plot of optimal wing surface, lower (left) and upper (right), obtained by CST with <i>BPOX 6</i> and <i>BPOY 6</i> .....	223
Figure 8.7 The Cp contour plot of optimal wing surface, lower (left) and upper (right), obtained by CST with <i>BPOX 6</i> and <i>BPOY 8</i> .....	224
Figure 8.8 The Cp contour plot of optimal wing surface, lower (left) and upper (right), obtained by CST with <i>BPOX 6</i> and <i>BPOY 10</i> .....	224
Figure 8.9 The Cp contour plot of optimal wing surface, lower (left) and upper (right), obtained by CST with <i>BPOX 10</i> and <i>BPOY 10</i> .....	225
Figure 8.10 Cp distribution (left) and aerofoil shapes (right) at 10% span of wing .....	226
Figure 8.11 Cp distribution (left) and aerofoil shapes (right) at 30% span of wing .....	226
Figure 8.12 Cp distribution (left) and aerofoil shapes (right) at 50% span of wing .....	226
Figure 8.13 Cp distribution (left) and aerofoil shapes (right) at 70% span of wing .....	227
Figure 8.14 Cp distribution (left) and aerofoil shapes (right) at 80% span of wing .....	227
Figure 8.15 Cp distribution (left) and aerofoil shapes (right) at 90% span of wing .....	227
Figure 8.16 Twist distribution .....	230
Figure 8.17 The spanwise lift distribution .....	231
Figure 8.18 Optimisation history of drag (left) and Cmx (right) using CST with <i>BPOX 6-BPOY 6</i> .....	233

Figure 8.19 Optimisation history of the 10 <sup>th</sup> torsion box volume using CST with <i>BPOX 6-BPOY 6</i> .....	233
Figure 8.20 The Cp contour plot of optimal wing surface, lower (left) and upper (right), obtained by CST with <i>BPOX 6</i> and <i>BPOY 6</i> in optimisation with Cmx constraint .....	234
Figure 8.21 Cp distribution (left) and aerofoil shapes (right) at 10% of wingspan .....	234
Figure 8.22 Cp distribution (left) and aerofoil shapes (right) at 30% of wingspan .....	235
Figure 8.23Cp distribution (left) and aerofoil shapes (right) at 50% of wingspan .....	235
Figure 8.24 Cp distribution (left) and aerofoil shapes (right) at 70% of wingspan .....	236
Figure 8.25 Cp distribution (left) and aerofoil shapes (right) at 80% of wingspan .....	236
Figure 8.26 Cp distribution (left) and aerofoil shapes (right) at 90% of wingspan .....	236
Figure 8.27 Twist distribution .....	237
Figure 8.28 The lift distribution along span.....	238
Figure 8.29 Optimisation history of drag (left) and Cmx (right) using RCST with <i>BPOX 6-BPOY 6</i> .....	240
Figure 8.30 Optimisation history of the 10 <sup>th</sup> torsion box volume using RCST with <i>BPOX 6- BPOY 6</i> .....	240
Figure 8.31 The Cp contour plot of Initial wing surface, lower (left) and upper (right), which is represented by RCST with <i>BPOX 6</i> and <i>BPOY 6</i> .....	241
Figure 8.32 The Cp contour plot of optimal wing surface, lower (left) and upper (right), obtained by RCST with <i>BPOX 6</i> and <i>BPOY 6</i> in optimisation with Cmx constraint.....	241
Figure 8.33 Cp distribution (left) and aerofoil shapes (right) at 10% of wingspan .....	242
Figure 8.34 Cp distribution (left) and aerofoil shapes (right) at 30% of wingspan .....	242
Figure 8.35Cp distribution (left) and aerofoil shapes (right) at 50% of wingspan .....	243
Figure 8.36 Cp distribution (left) and aerofoil shapes (right) at 70% of wingspan .....	243
Figure 8.37 Cp distribution (left) and aerofoil shapes (right) at 80% of wingspan .....	243
Figure 8.38 Cp distribution (left) and aerofoil shapes (right) at 90% of wingspan .....	244
Figure 8.39 Twist distribution .....	245
Figure 8.40 The lift distribution along span.....	245
Figure 8.41 Optimisation history of drag of winglet-1 (left) and winglet-2 (right).....	252
Figure 8.42 Optimisation history of drag of winglet-3 (left) and winglet-4 (right).....	252

Figure 8.43 The Cp contour and zoomed in view with Cf lines of initial design of winglet-1 .....	254
Figure 8.44 The Cp contour and zoomed in view with Cf lines of optimised design of winglet-1 .....	255
Figure 8.45 The Cp contour and zoomed in view with Cf lines of initial design of winglet-2 .....	255
Figure 8.46 The Cp contour and zoomed in view with Cf lines of optimised design of winglet-2 .....	256
Figure 8.47 Cl (left) and Cd wave drag (right) distribution along span of winglet-1 ....	256
Figure 8.48 Cl (left) and Cd wave drag (right) distribution along span of winglet-2....	257
Figure 8.49 Cp Contour of initial and optimised winglet-3.....	259
Figure 8.50 Cp Contour of initial and optimised winglet-4.....	260
Figure 8.51 Cl (left) and Cd wave drag (right) distribution along span of winglet-3....	260
Figure 8.52 Cl (left) and Cd wave drag (right) distribution along span of winglet-4....	261
Figure 8.53 Cd pressure drag (left) and friction drag (right) of optimal results of four types of winglet.....	261
Figure 8.54 Sensitivities of Cd to surface point Z direction and boundaries of bumps..	263
Figure 8.55 Optimisation history of Cd .....	264
Figure 8.56 Optimised 12 bumps on wing.....	265
Figure 8.57 Surface Cp contour in range -1.2 to -0.4 of initial wing without bump(left) and optimised bumps (right) .....	265
Figure 8.58 Cp Contour and Cf lines plot on upper surface of initial wing without bump (left) and optimised bumps (right) .....	266
Figure 8.59 Cp distribution and aerofoil section cut at middle of bump 1 .....	266
Figure 8.60 Cp distribution and aerofoil section cut at middle of bump 5 .....	267
Figure 8.61 Cp distribution and aerofoil section cut at middle of bump 9 .....	267
Figure 8.62 Cp distribution and aerofoil section cut at middle of bump 12 .....	267
Figure 8.63 Cd wave along span of initial wing and optimal bumps .....	268



# List of Tables

Table 2.1 Various types of geometries using different class parameters.....	36
Table 4.1 Total $L_2$ norm error of inverse fitting ( $\times 10^{-2}$ m).....	74
Table 4.2 Lift coefficient of original geometry and approximated geometry.....	77
Table 4.3 Drag coefficient of original geometry and approximated geometry.....	77
Table 4.4 Lift coefficient of original geometry and RCST approximated geometry.....	88
Table 6.1 Common radial basis function .....	189
Table 7.1 Aerodynamic coefficients and constraint values .....	202
Table 7.2 Aerodynamic coefficients and constraint values .....	205
Table 7.3 Aerodynamic coefficients and constraint values .....	207
Table 7.4 Aerodynamic coefficients .....	215
Table 8.1 Torsion box volume of initial geometry .....	220
Table 8.2 Aerodynamic coefficients (drag units in drag count) .....	229
Table 8.3 Aerodynamic coefficients of optimal results (drag units in drag count) .....	238
Table 8.4 Aerodynamic coefficients of optimal results using RCST (drag units in drag count) .....	246
Table 8.5 Planform parameters of winglet-1 .....	248
Table 8.6 Planform parameters of winglet-2 .....	249
Table 8.7 Planform parameters of winglet-3 .....	250
Table 8.8 Planform parameters of winglet-4 .....	251
Table 8.9 Aerodynamic coefficients of winglet-1 (drag unit in drag count) .....	253
Table 8.10 Aerodynamic coefficients of winglet-2 (drag unit in drag count) .....	253
Table 8.11 Aerodynamic coefficients of winglet-3 (drag unit in drag count) .....	258
Table 8.12 Aerodynamic coefficients of winglet-4 (drag unit in drag count) .....	258
Table 8.13 Aerodynamic coefficients (drag unit in drag count) .....	268

# Nomenclature

## Roman Symbols

$A_i$	= Weighting coefficient of shape function
$A_{up,lo}, B_{up,lo}$	= Matrices and vectors of design parameters for upper and lower surfaces
$a_i, b_i$	= Coefficients of PARSEC polynomials
$b$	= Wing span
$Bu_{i,j}, Bl_{i,j}$	= Upper and lower wing surface weight factors
$BPO$	= Order of the Bernstein polynomial
$BPOX$	= Order of the Bernstein polynomial for chordwise
$BPOY$	= Order of the Bernstein polynomial for spanwise
$c$	= Chord length
$C_{Local}$	= Local chord length
$CFD$	= Computational Fluid Dynamics
$CST$	= Class/Shape function transformation
$C(\psi)$	= Class function
$Cd$	= Drag coefficient
$Cl$	= Lift coefficient
$Cmx$	= Rolling momentum coefficient
$C_p$	= Pressure coefficient
$C1, C2$	= First and second surface derivatives
$c_i^{ie}$	= The i-th inequality constraint function
$c_j^e$	= The j-th equality constraint function
$D$	= Design variables
$\bar{D}_{ij}$	= The artificial dissipation term
$E$	= Total energy per control volume
$f$	= Body force in linear elasticity
$\mathbf{F}^c$	= Convective flux
$\mathbf{F}^v$	= Viscous flux

FFD	= Free-form deformation
FTF	= Flap track fairing
HTP	= Horizontal tail plane
$H$	= Total enthalpy
$H_{\text{Fuselage}}$	= Height of fuselage
$H_{\text{nose}}$ $H_{\text{tail}}$	= Local crown height of fuselage nose profile (relate with crown line)
$iCST$	= Intuitive CST
$I$	= Cost function
$K_{\text{nose}}$ $K_{\text{tail}}$	= Local keel height of fuselage nose and tail cone profile (relate with Keel line)
$K_{r,n}$	= Binomial coefficient
$\mathbf{K}$	= Stiffness matrix of the mesh deformation system
$k_{ij}$	= Spring stiffness model between two adjacent grid nodes
$k$	= kinetic energy
$L_2$	= $L_2$ -norm
$L_{\text{root}}$	= Bump length at beginning of boundary side
MDO	= Multi-disciplinary optimisation
$n$	= Order of Bernstein polynomials
$N_1, N_2, N_c$	= Class function exponents
NDV	= Number of design variables
$N_x$	= Order of Bernstein Polynomial in chordwise
$N_y$	= Order of Bernstein Polynomial in spanwise
NURBS	= Non-uniform Rational B-Spline
PARSEC	= Parametric aerofoil section
$P_i$	= Control points vector of Bezier Curve, control points of winglet plant form
$p$	= Pressure
$q$	= Heat flux
$R$	= Radius on cylindrical coordinate, Flow residual in flow and adjoint equation
$RCST$	= Rational CST

$R_{crown}, R_{keel}$	= Radius on cylindrical coordinate when $\theta = 1.5\pi$ and $\theta = \pi$
$R_{le,upper}, R_{le,lower}$	= Leading edge radius on upper lobe and lower lobe of nacelle
$R_{le}$	= Leading edge radius
$s$	= Approximated function in Radial Basis Function mesh deformation
$s_k$	= Search direction in the design space
$S(\psi)$	= Shape function
$SR(\psi)$	= Rational Shape function
$Sy, Sx$	= Bernstein polynomials in span and chordwise
$T$	= Temperature, the residual vector of mesh deformation in the mesh adjoint approach
$t$	= Time
$T_{End, up}, T_{End, lo}$	= Tangential value of upper and lower ends of inlet
$T_{length}$	= Total length of component
$u$	= Non-dimensional coordinate in Bezier Curve, velocity component in x direction
$v$	= Velocity component in y direction
$VTP$	= Vertical tail plane
$V_k$	= Volume of cell k
$W$	= Conservative state vector
$w$	= Velocity component in z direction
$W_i$	= Weights coefficients in rational equation
$W_{Fuselage}$	= Width of fuselage
$W_{nose}, W_{tail}$	= Local width of fuselage nose and tail cone profile, (relate with side line)
Winglet-1	= Upward winglet with a transition part and straight winglet part
Winglet-2	= Upward winglet only with a smooth winglet part
Winglet-3	= Downward winglet with a transition part and straight winglet part
Winglet-4	= Downward winglet only with a smooth winglet part
$X$	= Grid variable
$X_s$	= Surface mesh points
$X_v$	= Volume mesh points

$X_{LE}$	= Leading edge position in x coordinate in wing CST representation
$X_{LO}$	= Lower throat x position of nacelle inlet
$X_{UP}$	= Upper throat x position of nacelle inlet
$X_{End, up}$	= Upper end station of nacelle inlet
$X_{End, lo}$	= Lower end station of nacelle inlet
$Z_{End, up}$	= Upper end z position of nacelle inlet
$Z_{End, lo}$	= Lower end z position of nacelle inlet
$Z_{LO}$	= Lower throat height of nacelle inlet, z coordinate on aerofoil lower surface
$Z_{UP}$	= Upper throat height of nacelle inlet, z coordinate on aerofoil lower surface
$X_{up}$	= X coordinates of aerofoil crest on upper surface
$X_{lo}$	= X coordinates of aerofoil crest on lower surface
$X_{1,up}X_{2,up}$	= X coordinates of two iCST design points on upper surface
$X_{3,lo} X_{4,lo}$	= X coordinates of two iCST design points on lower surface
$V_{up,lo}$	= Vectors of coefficients of polynomials
$Z_{up}$	= Z coordinates of crest on upper surface
$Z_{lo}$	= Z coordinates of crest on lower surface
$Z_{xx,up}$	= Second derivative at crest on upper surface, (at throat in nacelle)
$Z_{xx,lo}$	= Second derivative at crest on lower surface, (at throat in nacelle)
$Z_{te}$	= Z coordinate of aerofoil trailing mid-point
$Z_{1,up} Z_{2,up}$	= Z coordinates of two iCST design points on upper surface
$Z_{3,lo} Z_{4,lo}$	= Z coordinates of two iCST design points on lower surface
$Z_{x,1,up} Z_{x,2,up}$	= First derivative at two iCST design points on upper surface
$Z_{x,3,lo} Z_{x,4,lo}$	= First derivative at two iCST design points on lower surface
$Z_{xx,1,up} Z_{xx,2,up}$	= Second derivative at two iCST design points on upper surface
$Z_{xx,3,lo} Z_{xx,4,lo}$	= Second derivative at two iCST design points on lower surface
$Z_{Centre}$	= Coordinate of local profile centre of fuselage nose and tail cone
$\Delta Z_{te}$	= Trailing edge thickness

## Greek Symbols

$\alpha_{twist}$	$\Delta\alpha$	= Local wing twist angle
$\alpha$		= Angle of attack
$\alpha_i$		= Coefficient of Radial Basis Function mesh deformation
$\alpha_{LE\_tran}$		= Sweep angle of leading edge of transition part of winglet 1
$\alpha_{TE\_tran}$		= Sweep angle of trailing edge of transition part of winglet 1
$\alpha_{LE\_winglet}$		= Sweep angle of leading edge of winglet part of winglet 1
$\alpha_{TE\_winglet}$		= Sweep angle of trailing edge of winglet part of winglet 1
$\alpha_{dihedral\_winglet}$		= Winglet tip dihedral angle of winglet 1
$\alpha_{leading}$		= Leading edge sweep angle of bump
$\alpha_{trailing}$		= Trailing edge sweep angle of bump
$\alpha_{LE\_tip}$		= Sweep angle of leading edge at winglet tip of winglet 2
$\alpha_{TE\_tip}$		= Sweep angle of trailing edge at winglet tip of winglet 2
$\alpha_{te}$		= Trailing edge angle
$\beta_{te}$		= Trailing edge wedge angle
$\beta_i$		= Coefficient of Radial Basis Function mesh deformation
$\theta_{up,lo}$		= Trailing edge angle of upper or lower surface
$\psi$		= Non-dimensional coordinate in chordwise
$\eta$		= Non-dimensional spacewise coordinate
$\lambda$		= Local laminar bulk viscosity
$\kappa$		= Thermal conductivity
$\lambda_{flow}$		= Adjoint operator for flow equation
$\lambda_{mesh}$		= Adjoint operator for mesh deformation
$\rho$		= Flow density
$\theta_{Z,axis}$		= Rotation angle of winglet section along z-axis
$\theta_{X,axis}$		= Rotation angle of winglet section along x-axis
$\bar{\theta}$		= Non-dimensional angle on cylindrical coordinate
$\theta$		= Angle on cylindrical coordinate, trailing edge tangential angle
$\lambda_i$		= Hicks-Henne function design variables

$\gamma$	= Constant specific heat
$\xi_N$	= Non-dimensional local aerofoil installation height
$\xi$	= Non-dimensional coordinate in direction normal to chordwise
$\Delta\xi_{te}$	= Trailing edge thickness ratio in CST
$\xi_U$	= Non-dimensional upper surface coordinate
$\xi_L$	= Non-dimensional lower surface coordinate
$\Delta\xi_{TE}$	= Non-dimensional trailing edge thickness
$\tau$	= Shear stress
$\sigma$	= Stress tensor in linear elasticity
$\phi$	= Radial basis function
$\Omega_i$	= Volume of control element
$\psi$	= Non-dimensional chordwise coordinate (in aerofoil or wing), non-dimensional coordinate along body axis (in non-wing type components)

# Chapter 1 Introduction

## 1.1 Background

Computational aerodynamics has been employed to assist aircraft design for more than six decades. With the development of high performance computers, computational aerodynamic flow solutions have become much less expensive than large-scale experiments for the right Reynolds and Mach numbers. Therefore, computational aerodynamics has been widely employed in the aircraft industry, and is playing an increasingly important role in aircraft design.

Computational aerodynamics tools have developed from the simple low-fidelity panel method to the more complex high-fidelity Reynolds averaged Navier-Stokes (RANS) solution methods. Nowadays, the low-fidelity methods are able to provide results in a very short time and are used for the concept design process; they are effective for assisting the designer in the analysis of technical and economic feasibility for future projects. The high-fidelity methods, such as the Euler equation, RANS and large eddy simulation (LES), are able to provide more accurate results and are normally used in the preliminary and detail design stages.

The pursuit of excellent design is invariably the goal for aircraft designers. Based on Figure 1.1, it has been estimated that the fuel efficiency of a current civil jet transport aircraft, e.g. Airbus A330-300, has been reduced by 70% from the Comet 4 of the 1950s, with 30% coming from advanced airframe design and 40% due to improvement of aero-engines (Mann and Elsholz 2005). The Strategic Research Agenda (ACARE 2002; Mann and Elsholz 2005), prepared by ACARE (Advisory Council for Aeronautical Research in Europe), set the direction for European research to reduce the environmental impact of aircraft and to improve safety and operational efficiency. ‘Vision2020’ requires a step change in aircraft performance, such as 50% CO<sub>2</sub> emission reduction and perceived noise reduction (ACARE 2002). This is a huge challenge to aircraft designers, since modern aircraft comprise a large number of highly complicated systems. The traditional manual

approach would find it very hard, if not impossible, to satisfy future design requirements. Hence, numerical optimisation techniques based on computational flow solutions have become a critical tool for the aircraft industry to help designers to meet future design challenges.

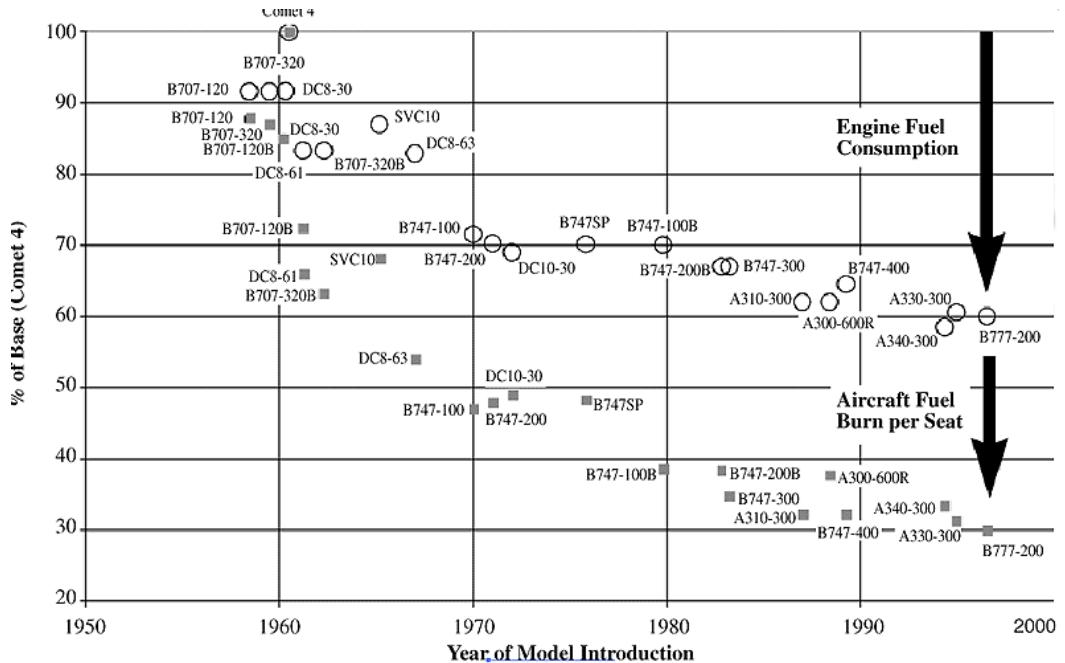


Figure 1.1 Transport aircraft fuel efficiency (from Penner 1999)

Optimisation is a well-established topic, and its essence is to find the maximum or minimum value of the objective function, which mathematically represents the relationship between the input variables and the objective values. Aerodynamic optimisation requires an automatic design process which is able to take geometric parameters to modify the geometry, to run numerical methods to obtain an objective value and to search for the best design shape. Rapid aerodynamic solution methods such as the penal methods and the lifting surface methods are normally employed for conceptual design and multi-disciplinary optimisation (MDO). The high-fidelity CFD methods are used for the detailed design stage; however, they can result in unaffordable computational cost when applied to aerodynamic optimisation. Therefore, although high-fidelity aerodynamic optimisation was proposed at almost the same time as CFD, high-fidelity aerodynamic optimisation has developed slowly along with the increase in the performance of digital computers. Jameson (1988) successfully applied a method, called

the ‘adjoint’ method, to aerodynamic optimisation. With this method, the computational cost was dramatically decreased and this considerably improved the feasibility of high-fidelity aerodynamic optimisation for the aircraft industry. At present, aerodynamic optimisation is widely employed as an automated tool, from the two-dimensional aerofoil to complex three-dimensional configurations (Anderson 1997; Jameson 2004).

For aerodynamic optimisation, the objective values are normally the aerodynamic performance parameters obtained from the numerical methods, such as lift and drag coefficients, pressure distribution, pitching/bending momentum and others. The input variables are also called design variables and normally represent the geometry using various parameterisation methods. Consequently, the parameterisation methods used have a profound effect on the design space, determining the complexity of the design space and the optimum geometries obtainable. Therefore, the shape parameterisation method is a key technique for the designer in the numerical optimisation process.

The two main objectives of this thesis are to find and develop a parameterisation method for the entire modern civil transport aircraft and to apply it to high-fidelity aerodynamic optimisation using the adjoint approach.

The first task is to develop a geometric parameterisation method for the entire modern civil transport aircraft. There are already many geometric parameterisation methods implemented in aerodynamic optimisation. However, most parameterisation methods can only be applied to individual aircraft components rather than the entire aircraft. A few methods, such as computer-aided-design (CAD) and free form deformation (FFD) parameterisation, can potentially be used to parameterise the entire aircraft. However, these are either too complicated to build into the optimisation framework or they struggle to satisfy some designers’ preferred requirements, such as intuitiveness and generality. Therefore, the author of this thesis has further developed a parameterisation method for the entire civil transport aircraft based on Kulfan’s Class/Shape function transformation methods (CST). This method is able to represent most aircraft aerodynamic components in a universal and efficient way.

The second task is to build an aerodynamic optimisation framework and apply this parameterisation method in an adjoint-based optimisation for industrial application. Both two-dimensional aerofoil section and three-dimensional geometry optimisation are conducted in the thesis. The performance of this geometric parameterisation will be examined in the aerodynamic optimisation investigation.

## 1.2 Outline of thesis

This thesis is split into two parts based on the two main tasks, the first focusing on geometric parameterisation and the second on aerodynamic optimisation using the adjoint approach.

In Part I, current geometric parameterisation methods are reviewed and investigated in Chapter 2. This presents the ideal properties of a parameterisation method which is good for aerodynamic design and optimisation, and gives a review of the most common geometric parameterisation methods employed in aerodynamic optimisation and design. This will also explain why the CST method has been selected for this study.

The following chapters, Chapters 3 and 4, present two new geometric parameterisation methods, iCST method and RCST methods, for two-dimensional aerofoils and the development of the CST methods for the entire aircraft. The PARSEC method, CST method, iCST which is able to parameterise aerofoil with full intuitiveness, and RCST which employs the Rational Bernstein polynomials to improve the accuracy of standard CST methods are presented, and their performance according to accuracy in representing existing aerofoils is also discussed in Chapter 3. The CST method is then extended to parameterise three-dimensional civil transport aircraft components, including wing, horizontal tail plane (HTP), vertical tail plane (VTP), winglet, fuselage, belly-fairing, flap tracking fairing (FTF), pylon and nacelle. The parameterisation methods for each component are discussed in detail in Chapter 4. The performance of CST and RCST methods for 3D wing are investigated in Chapter 4. The CST method for shock control

bump is then presented. At the end of Chapter 4, the intersection line calculation method is presented for future entire aircraft optimisation.

In Part II, Chapter 5 presents the flow governing equation and the numerical solver for high-fidelity CFD optimisation based on the RANS equations. The optimisation method is presented in Chapter 6 with a literature review of related aerodynamic optimisation methods, the discrete adjoint methodology, the mesh deformation strategy and the optimisation framework. The review focuses on current optimisation techniques applied in aerodynamic optimisation. Chapter 7 shows the two-dimensional optimisation test and results, and includes aerofoil optimisation and aerofoil with bump optimisation for transonic conditions. It examines the performance of the CST parameterisation methods in optimisation, leading to the later three-dimensional optimisation. Chapter 8 shows the three-dimensional optimisation results, including optimisation for the F6 wing, the F6 wing with different types of winglets and shock control bumps on 3D wing optimisation. Finally, Chapter 9 gives the conclusion of this thesis and provides some suggestions for future work.

## **Chapter 2 Literature Review of Geometric Parameterisation**

This chapter presents the various typical geometric parameterisation methods and their application in aerodynamic optimisation by previous researchers. This gives a background of current state-of-the-art geometric parameterisation methods and an understanding of the basic methodology of parameterisation. The advantages and disadvantages of the various methods will also be discussed.

‘Parameterisation’ is the representation of the specifications of a model as a set of parameters. In aerodynamic optimisation, parameterisation is usually applied to the representation of geometry. These geometric parameters are then employed as design variables for the designer or as input of an optimisation to find a desirable geometry which satisfies required performance.

Samareh (2001) and Kulfan (2006) have pointed out that a well-behaved parameterisation method should have the following properties:

- 1) To provide high flexibility to cover the potential optimal solution in the design space,
- 2) To give as small number of design variables as possible,
- 3) To produce smoothness and realisability of the shapes,
- 4) To provides intuitiveness of the design parameters for geometrical and physical understanding by the design engineers in exploring the design space and setting up optimisation constraints,
- 5) To provides grid sensitivity derivatives of grid respect to design variable, which is important for gradient-based optimisation.

In actual applications, a balance needs to be struck for parameterisation, as it is unlikely that all the requirements can be satisfied. For example, parameterisation methods with a

high number of design variables are normally able to provide a highly flexible design space; however, the high number of design variables will increase the complexity of the design space and will require that the optimiser makes an extra effort to find the optimum solution. In general, the cost of optimisation based on high-fidelity CFD computation is still very high; it will cause unaffordable computational expense. Even if the adjoint method is applied to be numerically efficient in calculating the sensitivities in gradient-based optimisation (Jameson 1988; Jameson *et al.* 1997; Le Moigne and Qin 2004; 2006), finding global optimum from a highly complex design space is still a challenging issue. On the other hand, for example, the NACA 4-series aerofoil definition only uses three parameters (maximum camber, position of maximum camber and maximum thickness) to represent an aerofoil (Ladson *et al* 1996), which are unable to provide sufficient design space to satisfy the desired aerodynamic performance.

Samareh (2001) reviewed and compared some of these methods and classified the shape parameterisation methods into the following eight categories: the basis vector, domain element, partial differential equation, discrete (mesh point), polynomial and spline, analytical, CAD-based and free-form deformation (FFD) methods. Among these methods, the discrete, analytical, polynomial, spline, CAD and FFD are the most common. They are studied and reviewed in the following sections. Another two methods, the parametric aerofoil section method (PARSEC) and the class/shape function transformation method (CST), are presented at the end.

## 2.1 Discrete methods

The discrete approach, which is the simplest way to do parameterisation, uses the mesh points as design variables. The discrete methods are able to provide a large design space since there is not any natural limit of design space, and theoretically it is possible to represent any shape. It is also easy to set up for any kind of geometry. Therefore, many researchers have tried to use discrete methods in aerodynamic optimisation (Jameson 1988; Campbell 1992; Jameson *et al.* 1997; Mousavi *et al.* 2007; Wu *et al.* 2003; Castonguay and Nadarajah 2007).

However, there are two main drawbacks of discrete methods. The first is that it is hard to maintain smoothness of geometry since each surface point is moved individually. Therefore, a smoothing algorithm is required to maintain the smoothness of the geometry. Moreover, in gradient-based optimisation, the gradients along grid points are normally not smooth. As a consequence, a smoothing algorithm will also be required to obtain a smooth gradient (Jameson 1988).

The most important drawback of the discrete method is that it results in a large number of design variables. As stated at the beginning, this will generate a very high dimensional design space. As a result, the complexity of the design space could reduce the efficiency of the optimiser in searching for global optimum and lead to unaffordable computational cost in the aerodynamic optimisation, although local optimum can be obtained efficiently with the adjoint method. Additionally, another drawback of discrete methods is difficult to provide intuitive parameters, for example, sweep angle, thickness, twist and so on.

## 2.2 Analytical methods

As presented above, although the discrete method is able to provide the most flexible design space, it is not good for reducing the complexity of design space as a large number of design variables are used. A parameterisation method with a small number of design variables that produces a smooth shape is preferred for aerodynamic optimisation. The most efficient way is to put a set of mathematical functions on the geometry surface, which is defined as Equation 2.1:

$$y(x, \lambda) = y(x)_{initial} + \sum_{i=1}^n \lambda_i f_i(x) \quad 2.1$$

where  $\lambda_i$  is used as design variable,  $n$  is the number of design variables and  $f_i(x)$  is the shape function.

The shape functions could be Hicks-Henne functions, Wagner functions, Legendre functions, Bernstein functions or NACA series aerofoil functions. The method is able to support direct representation function without adding any initial geometry.

The most common analytical method is called the ‘Hicks-Henne’ shape function method. It was first introduced by Hicks and Henne in 1978. It employs a set of bump basis shape functions, as defined in Equation 2.2:

$$f_i = [\sin(\pi x^{\beta_i})]^w \quad 2.2$$

where

$$\beta_i = \frac{\ln 0.5}{\ln x_{Mi}}$$

$x_{Mi}$  defines the position of maximum peak point of the  $i$ -th bump function, and  $w$  controls the width of the bump function.

Khurana *et al.* (2008) conducted a study of analytical parameterisation methods by comparing five different shape functions, including the Hicks-Henne, Wagner, Legendre, Bernstein, and NACA normal modes. In order to examine their impact on design space. The first work was to carry out an aerofoil geometric fitting study using these five functions, with the NACA 0015 aerofoil being used as a baseline shape. The NASA LRN (1)-1007, NASA LS-0417, NASA NLF (1)-1015 aerofoils were employed as target aerofoils. Five shape functions were used to fit three target aerofoils under a Particle Swarm Optimisation (PSO) algorithm and a linear search method. The speed of convergence and accuracy of the approximation were compared and the design variables, from 4 to 20, were examined for each method. The results showed that the optimisation could converge very fast when using four variables. However, the accuracy with four variables was less than that with 20 variables. The results which were obtained using the five functions were compared. The Hicks-Henne function provided the highest fitting accuracy of all the types of function within relative convergence speed. Thus, the Hicks-Henne function was found to be a better aerofoil shape representation method than the others. The performance of the Hicks-Henne shape function in aerodynamic optimisation was also examined in the second part of this work by carrying out an inverse design process. The results demonstrated that the Hicks-Henne function would provide good results when it is applied to an inverse design for an aerofoil at high Reynolds number condition. However, it generated some oscillations on a Cp distribution for the case at a low Reynolds number due to unsMOOTH shape. Eyi and Lee (1997) employed the Hicks-Henne and Wagner functions as smooth perturbations to the initial geometry in a two-

dimensional aerofoil inverse design optimisation. The test showed that both the Hicks-Henne and Wagner methods could achieve the target aerofoil and the convergence speed of the Wagner functions is slightly faster than the Hicks-Henne functions.

The Hicks-Henne method has been widely employed in aerodynamic optimisation studies. Sung and Kwon. (2001) employed 15 Hicks-Henne functions to modify an aerofoil, and used 5 sections with 10 Hicks-Henne functions on each section to modify a wing shape. Kim, Sasaki *et al.* (2001) carried out a wing-body-nacelle and a wing-body aerodynamic optimisation study with Hicks-Henne functions. The wing is defined as 5 sections with 20 Hicks-Henne functions for each section plus planform height parameters. The total number of design variables is only 106 for this three-dimensional configuration. After optimisation, the shock wave was greatly reduced. Kim *et al.* (2002) performed an aerodynamic optimisation test for a high-lift device. A two-dimensional multi-element aerofoil was represented by 157 design variables, of which 50 were Hicks-Henne functions for each master element, three rigging variables for the slat and flap element, and one for angle of attack. Nakayama *et al.* (2006) carried out a similar two-dimensional multi-element aerofoil optimisation. The total number of design variables was decreased to 71, of which 65 were design variables for aerofoil, slat and flap, eight variables for position of slat and flap and two variables for slat and flap angle. Kim and Nakahashi (2005) carried out a high-lift device optimisation with the unstructured adjoint method. A multi-element aerofoil with vane and flap was modified and the Hicks-Henne shape functions were employed to parameterise the geometry. The total number of design variables was 37, with 10 functions for vane upper and lower surfaces and flap leading edge area respectively. More Hicks-Henne applications can be found in Hageri *et al.* (1994), Kim and Alonso (2002a, 2002b), Zuo *et al.* (2006), Reuther *et al.* (1996; 1999), Elliott and Peraire (1996), Kim *et al.* (1999), Nadarajah and Jameson (2000), Eyi *et al.* (1996) and others.

### **2.3 Polynomial, spline methods, CAD-based and free-form deformation**

Other techniques to represent geometry shapes with reduced number of design variables are polynomials and splines. The polynomial method is the basic method to represent

curves with an easy mathematical power form function and high computational efficiency.

The polynomial curves can be written as:

$$R(u) = \sum_{i=0}^n A_i u^i \quad 2.3$$

where the polynomial of  $n$ -th order,  $R(u)$ , is the value of the polynomial function,  $A_i$  are coefficients of polynomials and normally used as design variables to control curves.

It can be in either implicit or explicit form. The low order polynomial form performs well for representing a simple curve. For a complex curve, a high order polynomial is required to provide more flexibility. However, high order polynomials will easily produce oscillations and cause numerical instability issues. In addition, the power form basis polynomial provides less intuitive information to the designer, such as starting and ending point positions and tangential values. Therefore, it is normally employed to parameterise simple curves in aerodynamic components, such as leading edge and twist distribution function (Le Moigne 2002).

For more complex curves, Bezier and B-spline curves are preferred. The Bezier curve was originally used in the design of automotive bodies and has been widely employed in the aerospace industry. A Bezier curve in  $n$ -th order for a single segment is described as:

$$R(u) = \sum_{i=0}^n P_i B_{i,n}(u) \quad 2.4$$

$$0 \leq u \leq 1$$

where  $n$  is the order of Bezier curve (thus the total number of control points is  $n+1$ ),  $R(u)$  is the vector value of the polynomial function,  $P_i$  are the control point vectors which are normally used as design variables in optimisation and  $B_{i,n}(u)$  is the  $i$ -th term of an  $n$ -th order Bernstein polynomial, which is defined as:

$$B_{i,n}(u) = K_{i,n} \cdot u^i (1-u)^{n-i} \quad 2.5$$

where  $K_{i,n}$  is the binomial coefficient defined as:

$$K_{i,n} \equiv \binom{n}{i} \equiv \frac{n!}{i! (n-i)!} \quad 2.6$$

Once the control points are determined, the Bezier curve can be established. The Bezier curve is bounded by a ‘control polygon’ which is formed of all the control points: see Figure 2.1. The first and last lines of the control polygon coincide with the tangential direction of this Bezier curve at starting and ending points. The details can be found in Appendix A (Piegl and Tiller 1997).

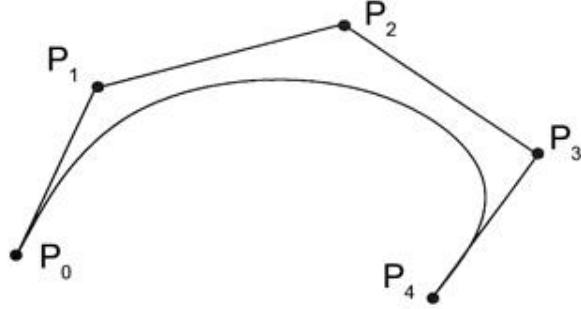


Figure 2.1 Bezier curve with control points

The three-dimensional Bezier surface can be defined by a tensor product form, and shown as:

$$R(u, v) = \sum_{i=0}^m \sum_{j=0}^n P_{i,j} B_{i,m}(u) B_{j,n}(v) \quad 2.7$$

$$0 \leq u \leq 1 \quad 0 \leq v \leq 1$$

where  $m, n$  is the order of Bezier surface in  $i$  and  $j$  direction,  $R(u)$  is the vector value of the polynomial function, and  $P_{i,j}$  are the control point vectors. Many researchers have used the Bezier curve in shape optimisation (Cosentino and Holst 1986; Désidéri *et al.* 2004) and have presented the Bezier curve as efficiently representing an aerofoil-like curve and providing designers with more interactive control. However, for very complex curves, the Bezier curve is less efficient and more control points are required, as more control points will increase the order of the polynomial. Similarly for the power basis form, higher order polynomials will produce oscillations and cause numerical instability issues. Thus, it is inefficient for representing a very complex curve. Furthermore, any coefficient or control point could affect the entire curve and, therefore, it lacks local control. To overcome these shortages, piecewise polynomials are employed. The ‘B-

spline curve' is developed based on this and can be considered as a curve which comprises a few Bezier curves. A  $p^{\text{th}}$  order B-spline curve can be written as:

$$R(u) = \sum_{i=0}^n P_i N_{i,p}(u) \quad 2.8$$

$0 \leq u \leq 1$

where  $p$  is the order of the B-spline curve,  $R(u)$  is the coordinate of geometry vector,  $P_i$  is the B-spline control points vector and  $N_{i,p}(u)$  is the B-spline basis function, which is normally obtained by a recurrence formula, from Boor (1972 and 1977) :

$$N_{i,0} = \begin{cases} 1 & \text{if } u_i \leq u \leq u_{i+1} \\ 0 & \text{otherwise} \end{cases} \quad 2.9$$

and

$$N_{i,p} = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \quad 2.10$$

where  $u_i$  are the breakpoints, so called 'knot' in B-spline methods. For a  $p$ -th order B-spline curve with  $n$  control points,  $m=n+p+1$  knots are required. A detailed description and background of B-spline is given in Piegl and Tiller (1997).

B-spline is able to provide an excellent overall shape control, and it also can provide a high capability for local shape control, because the control points only affect the curve on local zone  $[u_i, u_{i+p+1}]$ . Like the Bezier methods, it can also be extended to represent a three-dimensional surface, defined as

$$R(u, v) = \sum_{i=0}^m \sum_{j=0}^n P_{i,j} N_{i,p}(u) N_{j,q}(v) \quad 2.11$$

$0 \leq u \leq 1 \quad 0 \leq v \leq 1$

Therefore, B-splines have been widely employed in curve and surface design and aerodynamic optimisation research. It provides very high flexible design space with a relatively low number of design variables. Furthermore, because the B-spline has excellent performance in interpolating dataset, it could be employed to interpolate

through a few key points on a curve or surface to provide intuitive control of curve or surface.

However, there is a shortcoming of the Bezier and B-spline curves: they are essentially still polynomial-based and cannot naturally accurately represent implicit conic shapes, such as circles, ellipses and hyperbolas. Therefore, there is a special modification, namely ‘rational curves’, which is introduced to overcome this issue by employing another polynomial, the so-called *weights*. In the B-spline curve, the non-uniform rational B-spline (NURBS) has been developed (Versprille 1975; Tiller 1983; Piegl and Tiller 1997).

The NURBS form is written as:

$$R(u) = \frac{\sum_{i=0}^n P_i N_{i,p}(u) W_i}{\sum_{i=0}^n N_{i,p}(u) W_i} \quad 2.12$$

$$0 \leq u \leq 1$$

where, similarly with B-spline,  $p$  is the order of NURBS curve,  $R(u)$  is the coordinate of the geometry vector,  $P_i$  is the NURBS control points vector and  $W_i$  are the *weights*.  $N_{i,p}(u)$  is the B-spline basis function that is the same as the above B-spline curve basis function.

The design variables could be selected either from  $P_i$  or  $W_i$ . Therefore, the NURBS inherits the benefits of B-spline, and overcomes B-spline’s shortcomings. The NURBS is capable of accurately representing the quadratic primitives; it is also able to represent the three-dimensional surface, and the definition is:

$$R(u, v) = \sum_{i=0}^m \sum_{j=0}^n R_{i,p;j,q}(u, v) P_{i,j} \quad 2.13$$

$$0 \leq u \leq 1 \quad 0 \leq v \leq 1$$

where 2.14

$$R_{i,p;j,q} = \frac{N_{i,p}(u) N_{j,q}(v) W_{i,j}}{\sum_{i=0}^m \sum_{j=0}^n N_{i,p}(u) N_{j,q}(v) W_{i,j}}$$

$$0 \leq u \leq 1$$

Because of B-spline, NURBS have very good performance for curve manipulation; most CAD systems have employed them as a key tool to generate curves and surfaces. The CAD system has a powerful capability for handling complex geometry and has been widely used in the design process. Therefore, using commercial CAD software directly in

the optimisation design process as parameterisation, in so-called *CAD-based* methods, is increasingly interesting for industrial designers and researchers. The benefits of *CAD-based* methods are stated as:

- 1) The CAD software is powerful for manipulating very complex geometries, which reduces the researchers' development time for complex geometries,
- 2) CAD can provide many intuitive parameters, such as camber, thickness, slots, twist, etc. (Samareh 2001),
- CAD has become a standard design tool in different areas, including aerodynamic, structure and system design. Thus, CAD software is also able to provide connectors for different purposes. This would be significant for multi-disciplinary optimisation (MDO).

However, it is still a challenge to embed CAD in an optimisation loop; the main difficulties are:

- 1) For the most part, CAD can provide accurate and smooth geometry, but it is not perfect. Normally, there are some blemishes, such as gaps, unwanted wiggles and free edges, in the CAD surface model. These may be ignorable in solid design, but are not acceptable for update or regenerate CFD (Samareh 2001).
- 2) In the gradient-based optimisation technique, the sensitivities of surface points with respect to design variable are required. However, this information is not provided in most current CAD tools (Townsend *et al.* 1998). Hence, gradient-based high-fidelity optimisation is hardly applied in CAD-based methods. An alternative way is to calculate surface sensitivities by finite difference (He *et al.* 1998). However, this requires that the surface topology does not change. Another promising method is developed by Armstrong *et al.* (2009) using design velocity.
- 3) The surface topology may be changed when the design variable is updated (Samareh 2001; Fudge and Zingg 2005). This could cause failure when updating surface CFD mesh and calculating surface sensitivities.

- 4) As a practical issue, the number of CAD licences will be a hurdle for parallel optimisation processes.

Many researchers have used Bezier, B-spline, NURBS and CAD-based methods in aerodynamic optimisation (Lambert 1995; Tang and Désidéri 2002; Li and Krist 2005; Painchaud-Ouellet *et al.* 2006; Grasso 2012; Nelson *et al.* 2005; Fudge and Zingg 2005). Samareh (2001) asserted that B-spline and NURBS are best suited for the two-dimensional optimisation case, because the three-dimensional complex geometry requires a large number of control points.

Sasaki and Obayashi (2003) used the B-spline and Bezier surface to represent the wing-fuselage configuration with a total number of 131 design variables. Nemec and Zingg (2002) performed a study of a multi-point and multi-objective aerodynamic optimisation for the design of a two-dimensional single-element and multi-element aerofoil. Fifteen control points were employed to represent a simple NACA 0012 aerofoil, and 10 of 15 control points were employed as design variables. The RAE 2882 aerofoil was represented as 25 control points and 19 control points were used as design variables. Later on, Nemec *et al.* (2004) carried out an aerodynamic optimisation study based on a CAD system. In order to address the issues of CAD parameterisation methods, a non-commercial CAD library, called *Cart3D*, was employed rather than using commercial CAD software directly. This CAS library employs the B-spline curve and surface to represent two-dimensional and three-dimensional geometries. A gradient-based optimisation method was used, and the gradient was calculated by finite-difference. A two-dimensional aerofoil optimisation and a complex configuration (fuselage, wing and canard) optimisation were tested. They showed B-spline methods have successfully provide optimal solutions on both 2D and 3D cases and also presented an alternative way of implementing CAD-based parameterisation.

Song and Keane (2004) made a comparative study of two parameterisation methods, which are the basis function derived by Robinson and Keane (2001) and B-spline interpolation methods. Three aerofoils, NACA 0406, NACA 0610 and RAE 2822, were chosen as the target aerofoils. Inverse design optimisation was carried out to examine the

performance of two parameterisation methods. The difference of the B-spline method here is that it employed 34 points on the aerofoil surface as the design variable. B-spline was used to generate the entire surface by interpolating through all design points. The study shows that the B-spline methods provided enough accuracy to inverse design all tested aerofoils; however, it would require more computational cost. The basis function dramatically reduced the number of design variables to five; but it was not able to provide a high level of flexibility and obtained low accuracy results.

Zingg and Elias (2006) and Zingg and Billing (2007) tested a two-dimensional multi-point aerodynamic shape optimisation using B-spline curves. The B-spline employed 15–25 control points for representing an aerofoil. One control point was frozen at the leading edge and two were frozen at the trailing edge, while the rest of the control points were set as design variables. The results showed that the B-spline was able to provide very high flexible design space to satisfy the wide range of multi-point optimisation. Driver and Zingg (2007) investigated a two-dimensional aerofoil shape optimisation incorporating laminar-turbulence transition prediction using the Newton-Krylov gradient-based method. The research tried to find a good performance aerofoil with maximum lift-to-drag ratio, maximum endurance factor and maximum lift coefficient. The NACA 0012 was selected as an initial aerofoil. The aerofoil was represented using B-spline curve with 15 control points, of which 12 were used as design variables. The results demonstrated that the B-spline had excellent performance for obtaining a new laminar flow aerofoil. Truong *et al.* (2008) employed a similar method in an aerodynamic optimisation in conjunction with a discret adjoint with Newton-Krylov optimisation algorithm. The NACA 0012 was chosen as the initial aerofoil with 15 control points, of which 13 were used as design variables. The results demonstrated that the performance of B-spline was good enough to provide a reasonable and smooth shape under restricted geometric constraints.

Giammichele and Trépanier (2007) performed a test of multi-resolution B-spline control shape optimisations with constraint. A novel B-spline control shape method was investigated, which decomposed the curve representation to a multi-level type. At the highest level, the high number of control points resulted in high accuracy. It also allowed

local shape modification. At the lowest level, the low number of control points resulted in low accuracy. The purpose of using low level is to give a good global shape modification. Two studies were carried out to investigate this method. Firstly, a geometric fitting test was run to examine the capability of this method to represent existing aerofoils. It was found that the multi-resolution B-spline provided more accuracy than the direct B-spline with the same number of control points. This indicated that this method gives a good compromise between the efficiency of low number of control points and the high flexibility of a high number of design variables. The aerodynamic simulation and optimisation test demonstrated that this method was also able to smooth shapes in the design space, and the efficiency of the optimisation method could be improved.

Lepine *et al.* (2001) offer an excellent presentation of the NURBS method in geometric representation and aerodynamic optimisation. The properties of NURBS for shape representation in aerodynamics were investigated. In the first step, the method was used to approximate several existing aerofoils. In the second step, NURBS was applied in an inverse aerodynamic design. To approximate to an aerofoil, the process of which could be defined as an optimisation issue to minimise the approximate  $L_2$  norm error, a second order quasi-Newton method, Broyden–Fletcher–Goldfarb–Shannon (BFGS), was selected to search a minimum cost function. Five aerofoils, NACA 2412, RAE 2822, Bombardier–Canadair (BC), Boeing A4 and Boeing A8, were employed in this test as target aerofoils. The results demonstrated that NURBS was able to accurately represent most of the aerofoils using 13 control points or fewer. The approximate error could reach below  $8 \times 10^{-5}$ , which is lower than the normal tolerance. This means that NURBS was able to provide an excellent curve control and gave a precise curve fitting; it also impressively reduced the number of control points without losing flexibility. In addition, the original target aerofoil data are given by discrete points data; the noise normally appears in curvature space, especially in the area of high curvature. The first experiment showed that the NURBS approximation was able to filter these noises.

In the second test by Lepine *et al.* (2001), inverse aerodynamic design with a target pressure distribution was performed using the quasi-Newton BFGS method. The gradient

of the objective function was obtained using finite differences. Eleven design variables were employed to control each side of the aerofoil surface. In order to compare the performance of parameterisation methods, the Hicks-Henne parameterisation method with a maximum of 11 design variables was selected for comparison. The results showed that the Hicks-Henne method could reach convergence faster than NURBS. However, because some non-physical shapes were generated by the Hicks-Henne method, it required manual intervention to restart the optimiser; otherwise, it was difficult to obtain the ideal shape. Furthermore, Hicks-Henne had very high noise compared with the result of the NURBS method; the obtained unsMOOTH shape could lead to aerofoil failure at off-design conditions. In a three-dimensional wing optimisation case, a wing was represented as seven control sections, each section being controlled by NURBS with 13 control points. The total number of design variables was up to 77. The results showed that NURBS had a faster convergence rate than the Hicks-Henne method, the reason being that the three-dimensional optimisation was highly sensitive to surface noise. The optimised shape and final pressure distribution showed that NURBS significantly reduced the shock strength and provided a smooth shape. The research demonstrated that NURBS has three main advantages. Firstly, NURBS is able to efficiently reduce the number of design variables; secondly, it could provide a wide range of design spaces for a small number of design variables; thirdly, it could provide a natural smooth and noise-free shape, which is important for aerodynamic performance.

Bentamy and Guibault (2005) used NURBS for parameterisation in a realistic aircraft wing. The aircraft wing was divided into five sections, namely: root section, mid-section between root and break section, break section, mid-section between break section and tip section and tip section. Each section was parameterised by a NURBS curve with 24 control points. The three-dimensional surface is then generated by interpolation of five control sections using NURBS surface. An inverse design case and direct aerodynamic optimisation were performed to test the performance of the NURBS. The results showed that NURBS was able to efficiently represent a smooth and complex geometry of a wing within a restricted number of design variables in aerospace design work.

Fudge and Zingg (2005) developed a CAD-free and a CAD-based system for aerodynamic optimisation. In the CAD-free system, the B-spline patch method was employed to represent the fuselage-wing geometry, and the B-spline surface control net was used as design variable; furthermore, the design variables were divided into three layers: control net, drive nodes and global variables. The control nets implemented geometry control; the driving nodes constrained the control net; and the global variables were used to drive the control net displacement. In the CAD-based system, they successfully integrated CATIA V5 into an optimisation loop by utilising the Computational Analysis Programming Interface (CAPRI), which is an application programming interface (API) whose purpose is to provide a seamless bridge between CAD systems and computational engineering analysis (such as CFD). They posited that the CAD-based variables are generic and may be applied to any CAD loft created with the appropriate parameters. The CAD-free system is able to calculate the surface point perturbation in a straightforward manner compared with the CAD-based system. In their work, Fudge and Zingg (2005) made extra effort to overcome this issue in the CAD-based system by employing an internal representation of geometries. Both methods have been applied to a DLR F6 wing-fuselage geometry and prepared for future optimisation work.

A few more recent studies using CAD-based methods have been done by DLR. Ronzheimer *et al.* (2010) have employed CATIA V5 to parameterise a transonic transport aircraft, and applied a comprehensive MDO process to maximise the range of this aircraft. A limited number of design variables, including wing aspect ratio, sweep angle and twist of four-wing control section, were employed. In the MDO process, the geometry was first updated from CATIA V5. The Euler CFD was then run to provide aerodynamic load data for the structure sizing process. Afterwards, the CFD-CSM coupling process was performed repeatedly to obtain correct drag and mass under aero-elasticity. The Breguet range was then calculated as an objective value of optimisation. The optimisation method employed a gradient-free Simplex method. This work showed the potential ability of CAD-based methods in a practical case, and demonstrated that CAITA had the capability to provide an interface between CFD surfacing and CSM processing. Brezillon *et al.*

(2012) subsequently extended this work to optimise the rear-fuselage and engine position, and applied MDO to include aero-acoustic analysis. However, they maintained that the use of CAD in optimisation is still a challenge because of the costs of complexity (IT infrastructure, coupling with CFD mesh), stability (Windows operating system) and extendibility (number of design parameters).

More recently, researchers are increasingly interested in free-form deformation methods. These FFD methods are based on a powerful surface morphing technique which was originally applied in industrial animation motion and deformed of solid models (Watt and Watt 1992; Barr 1984; Sederberg and Parry 1986). In design optimisation, the small perturbation may lead to a significant change in the design object's performance. The idea of FFD is to parameterise the geometric perturbation with small number of design variables instead of parameterising the entire geometry. The basic idea of FFD is first to map the geometry into a square control box, as Equation 2.15.

$$\mathbf{R}_{initial}(x, y, z) \rightarrow \mathbf{R}_{initial}(u, v, w) \quad 2.15$$

where  $\mathbf{R}_{initial}(x, y, z)$  is the coordinate vector of initial/baseline geometry with Cartesian coordinates and  $\mathbf{R}_{initial}(u, v, w)$  is the vector of initial/baseline shape on coordinate inside control box.

The FFD methods is then modifying  $\mathbf{R}(u, v, w)$  using Bezier, B-spline or a NURBS basis function to get the new coordinates(Ronzheimer 2005; Yamazaki *et al.* 2008; Lamousin and Waggenspack 1994). The vertices of control box are the control points. The FF formula using Bezier polynomial functions (Yamazaki *et al.* 2008) is presented below:

$$\mathbf{R}(u, v, w) = \sum_{i=0}^l \sum_{j=0}^m \sum_{k=0}^n B_{i,l}(u) B_{j,m}(v) B_{k,n}(w) P_{i,j,k} \quad 2.16$$

Because the technique is based on Bezier and spline techniques, the author included FFD in this subsection. The control points  $P_{i,j,k}$ , forming a lattice box, also called ‘control box’, are used as design variables to generate new surface (see Figure 2.2).

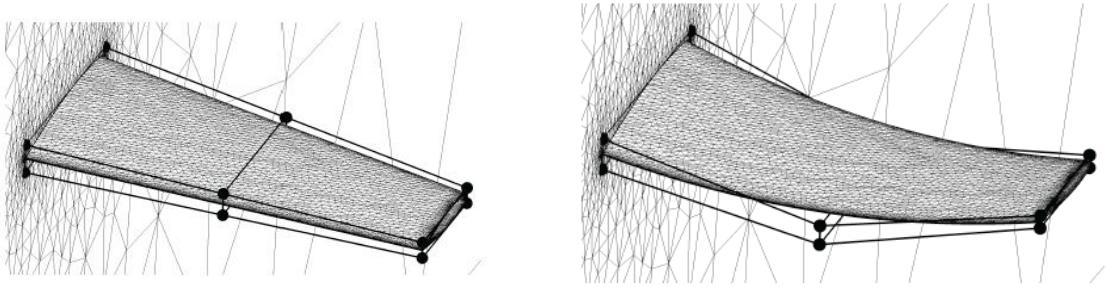


Figure 2.2 The control box of FFD for ONERA M6 wing (Widhalm *et al.* 2007)

Generally, the FFD process can be split into the following steps (Widhalm *et al.* 2007; Samareh 2004):

- 1) Selecting the region that needs to be modified or optimised,
- 2) Set up control box surrounding the design region and map all grid points on initial surface into this control box,
- 3) Change the control points of the control box and apply the perturbation over the control volume space to obtain a new surface.

The FFD methods have a major impact on the number of design variables and provide high flexible surface deformation at the same time. The second benefit of FFD methods is a mathematically algebraic scheme to perturb every point within the control volume, so that it is possible to provide analytical sensitivities of surface mesh points with respect to design variables. The details can be found in Samareh (2004) and Yamazaki *et al.* (2008). This is significant for gradient-based optimisation. The third advantage is that, because FFD methods directly perturb the original geometry, it could be applied in complex geometries. Another advantage is that FFD can maintain the grid topology. Therefore, FFD can be linked with mesh deformation or can regenerate the mesh when the deformation is not too great. (Additionally here, the reason and benefits of using mesh deformation rather than mesh regeneration in optimisation and mesh deformation methods will be discussed in chapter 6.)

However, there are some main disadvantages, which have been presented by Samareh (2005). FFD hardly provides the parameters with physical meaning. This would mean that designers would have less intuitive control of geometry surface, and it would be hard to specify design and optimisation bounds and constraints.

A few researchers have employed the FFD methods in aerodynamic optimisation or MDO. Samareh (2004) applied FFD based on NURBS in aerodynamic optimisation using Trivariate Volume-Based Deformation to provide more intuitive control for designers, and used it for many different configurations such as wing-fuselage leading edge fillet and blended-wing-body drone. Widhalm *et al.* (2010) embedded the FFD methods into gradient-free and gradient-based adjoint optimisation frameworks; M6 wing and a wing-body configuration were successfully tested. Yamazaki *et al.* (2008) developed the FFD methods for 2D aerofoil optimisation with physics-based direct manipulation where designers could have more freedom to specify the location of control points. The FFD method was then extended to deform flow field mesh, and results showed that FFD could efficiently preserve the mesh quality, better than spring analogy-based mesh deformation methods.

Ronzheimer (2006) presented a few examples of applications of the FFD methods in aerodynamic optimisation and successfully applied FFD in belly-fairing optimisation with a gradient-free suplex method. He also presented the potential capability of the FFD method to extend to the computational structure mechanics (CSM) model and CFD coupling design, where the FFD method is employed to deform the structure model and aerodynamic surface model simultaneously. Furthermore, Ronzheimer *et al.* (2010) further developed the FFD method and applied it to optimise a passenger transport aircraft's aerodynamic components, including fuselage tail part, pylon and nacelle, with only 20 parameters. The gradient-free optimisation method was employed, and results showed that the total drag was successfully reduced about 5 drag counts. More details can be found in Nielsen and Park (2005), Anderson *et al.* (2008; 2009), Andreoli *et al.* (2003), Désidéri *et al.* (2004), Duvigneau *et al.* (2006), Ronzheimer (2005).

## 2.4 PARSEC parameterisation methods

The PARSEC method was originally developed by Sobieczky (1998). In his work, explicit mathematical functions were introduced to represent a two-dimensional aerofoil. Intuitive parameters were used in this method. The purpose of the method is to find a

minimum number of variables to address the special aerodynamic, geometric and flow features. Sobieczky posited that the aerofoil curvature distribution is strongly linked to the desirable pressure distribution. Therefore, some parameters of curvature were employed to represent an aerofoil. In this method, two sixth order polynomials were used to control the upper and lower surfaces of the aerofoil, respectively:

$$Z_{up} = \sum_{n=1}^6 a_n X^{n-\frac{1}{2}} \quad 2.17$$

$$Z_{lo} = \sum_{n=1}^6 b_n X^{n-\frac{1}{2}} \quad 2.18$$

Eleven intuitive parameters were employed to explicitly represent an aerofoil, as illustrated in Figure 2.3.

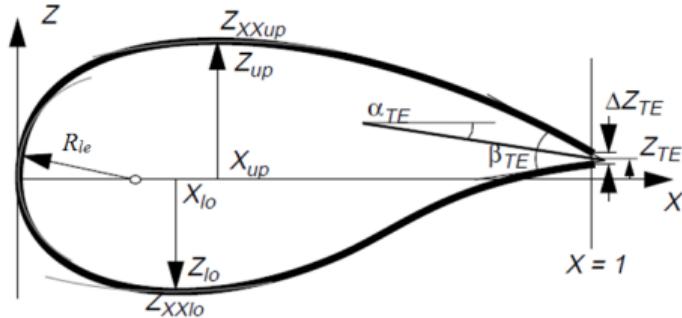


Figure 2.3 PARSEC method parameters definition

The parameters are: the leading edge radius ( $R_{le}$ ), upper crest position ( $X_{up}, Z_{up}$ ), upper crest curvature ( $Z_{xxup}$ ), lower crest position ( $X_{lo}, Z_{lo}$ ), lower crest curvature ( $Z_{xxlo}$ ), trailing edge position ( $Z_{te}$ ), trailing thickness ( $\Delta Z_{te}$ ) and trailing edge angle and trailing edge wedge angle ( $\alpha_{te}$  and  $\beta_{te}$ ). The relationships between the parameters and polynomials are shown as:

$$R_{le} = \left[ \frac{\left[ 1 + \left( \frac{dZ}{dX} \right)^2 \right]}{\frac{d^2Z}{dX^2}} \right] = \left| \frac{(a_1, b_1)^2}{2} \right| \quad 2.19$$

$$Z(X_{te})_{up,lo} = Z_{te} \pm \Delta Z_{te} \quad 2.20$$

$$Z_{up,lo} = \sum_{n=1}^6 (a_n, b_n) X_{up,lo}^{n-\frac{1}{2}} \quad 2.21$$

$$\left[ \frac{dZ}{dX} \right]_{X=X_{te}} = \tan(\theta_{up,lo}) = \tan\left(\alpha_{te} \pm \frac{1}{2}\beta_{te}\right) = \sum_{n=1}^6 (n - \frac{1}{2})(a_n, b_n) X_{te}^{n-\frac{3}{2}} \quad 2.22$$

$$Z_{xup,xlo} = \left[ \frac{dZ}{dX} \right]_{X=X_{up},X_{lo}} = \sum_{n=1}^6 \left( n - \frac{1}{2} \right) (a_n, b_n) X_{up,lo}^{n-\frac{3}{2}} = 0 \quad 2.23$$

$$Z_{xxup,xxlo} = \left[ \frac{d^2Z}{dX^2} \right]_{X=X_{up},X_{lo}} = \sum_{n=1}^6 \left( n - \frac{3}{2} \right) \left( n - \frac{1}{2} \right) (a_n, b_n) X_{up,lo}^{n-\frac{5}{2}} \quad 2.24$$

Two linear systems of equations for the upper and lower surfaces are then established:

$$\begin{aligned} A_{up} \cdot V_{up} &= B_{up} \\ A_{lo} \cdot V_{lo} &= B_{lo} \end{aligned} \quad 2.25$$

where

$$A_{up} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ X_{TE}^{\frac{1}{2}} & X_{TE}^{\frac{3}{2}} & X_{TE}^{\frac{5}{2}} & X_{TE}^{\frac{7}{2}} & X_{TE}^{\frac{9}{2}} & X_{TE}^{\frac{11}{2}} \\ X_{UP}^{\frac{1}{2}} & X_{UP}^{\frac{3}{2}} & X_{UP}^{\frac{5}{2}} & X_{UP}^{\frac{7}{2}} & X_{UP}^{\frac{9}{2}} & X_{UP}^{\frac{11}{2}} \\ \frac{1}{2} X_{TE}^{-\frac{1}{2}} & \frac{3}{2} X_{TE}^{\frac{1}{2}} & \frac{5}{2} X_{TE}^{\frac{3}{2}} & \frac{7}{2} X_{TE}^{\frac{5}{2}} & \frac{9}{2} X_{TE}^{\frac{7}{2}} & \frac{11}{2} X_{TE}^{\frac{9}{2}} \\ \frac{1}{2} X_{UP}^{-\frac{1}{2}} & \frac{3}{2} X_{UP}^{\frac{1}{2}} & \frac{5}{2} X_{UP}^{\frac{3}{2}} & \frac{7}{2} X_{UP}^{\frac{5}{2}} & \frac{9}{2} X_{UP}^{\frac{7}{2}} & \frac{11}{2} X_{UP}^{\frac{9}{2}} \\ -\frac{1}{4} X_{TE}^{-\frac{3}{2}} & \frac{3}{4} X_{UP}^{-\frac{1}{2}} & \frac{15}{4} X_{UP}^{\frac{1}{2}} & \frac{35}{4} X_{UP}^{\frac{3}{2}} & \frac{53}{4} X_{UP}^{\frac{5}{2}} & \frac{99}{4} X_{UP}^{\frac{7}{2}} \end{bmatrix} \quad 2.26$$

$$A_{lo} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ X_{TE}^{\frac{1}{2}} & X_{TE}^{\frac{3}{2}} & X_{TE}^{\frac{5}{2}} & X_{TE}^{\frac{7}{2}} & X_{TE}^{\frac{9}{2}} & X_{TE}^{\frac{11}{2}} \\ X_{LO}^{\frac{1}{2}} & X_{LO}^{\frac{3}{2}} & X_{LO}^{\frac{5}{2}} & X_{LO}^{\frac{7}{2}} & X_{LO}^{\frac{9}{2}} & X_{LO}^{\frac{11}{2}} \\ \frac{1}{2}X_{TE}^{-\frac{1}{2}} & \frac{3}{2}X_{TE}^{\frac{1}{2}} & \frac{5}{2}X_{TE}^{\frac{3}{2}} & \frac{7}{2}X_{TE}^{\frac{5}{2}} & \frac{9}{2}X_{TE}^{\frac{7}{2}} & \frac{11}{2}X_{TE}^{\frac{9}{2}} \\ \frac{1}{2}X_{LO}^{-\frac{1}{2}} & \frac{3}{2}X_{LO}^{\frac{1}{2}} & \frac{5}{2}X_{LO}^{\frac{3}{2}} & \frac{7}{2}X_{LO}^{\frac{5}{2}} & \frac{9}{2}X_{LO}^{\frac{7}{2}} & \frac{11}{2}X_{LO}^{\frac{9}{2}} \\ -\frac{1}{4}X_{LO}^{-\frac{3}{2}} & \frac{3}{4}X_{LO}^{-\frac{1}{2}} & \frac{15}{4}X_{LO}^{\frac{1}{2}} & \frac{35}{4}X_{LO}^{\frac{3}{2}} & \frac{53}{4}X_{LO}^{\frac{5}{2}} & \frac{99}{4}X_{LO}^{\frac{7}{2}} \end{bmatrix} \quad 2.27$$

and

$$V_{up} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{bmatrix} B_{up} = \begin{bmatrix} \sqrt{2R_{le}} \\ Z(X_{te})_{up} \\ Z_{up} \\ \tan(\theta_{up}) \\ 0 \\ \left[ \frac{d^2Z}{dX^2} \right]_{X=X_{lo}} \end{bmatrix} \quad V_{lo} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \end{bmatrix} B_{lo} = \begin{bmatrix} -\sqrt{2R_{le}} \\ Z(X_{te})_{lo} \\ Z_{lo} \\ \tan(\theta_{lo}) \\ 0 \\ \left[ \frac{d^2Z}{dX^2} \right]_{X=X_{lo}} \end{bmatrix} \quad 2.28$$

The coefficients  $a_n$  can then be obtained by solving these two systems of equations:

$$\begin{aligned} V_{up} &= A_{up}^{-1} \cdot B_{up} \\ V_{lo} &= A_{up}^{-1} \cdot B_{lo} \end{aligned} \quad 2.29$$

Once the vectors  $V_{up}$  and  $V_{lo}$  are solved, the polynomials for PARSEC are defined. The aerofoil coordinates are then calculated from the polynomials. The coefficients need to be updated as soon as any PARSEC intuitive parameter is changed.

Sobieczky also presented that the PARSEC method was able to modify the trailing edge in order to enhance the aerodynamic efficiency. For an aerofoil with blunt trailing edge, a convex upper surface contour shape and a concave lower surface were generated by increasing and decreasing the curvature. This resulted in a minimum thickness change of the aerofoil a few percent upstream of the TE. This modification was called ‘divergent trailing edge’ (DTE). Modification based on a hodograph was employed in the PARSEC

method. A few parameters were used to control the DTE shape, which are shown in Equation 2.30 and Figure 2.4.

$$\Delta Z = \frac{L \cdot \tan \Delta \alpha}{\mu \cdot n} \cdot [1 - \mu \xi^n - (1 - \xi^n)^\mu] \quad 2.30$$

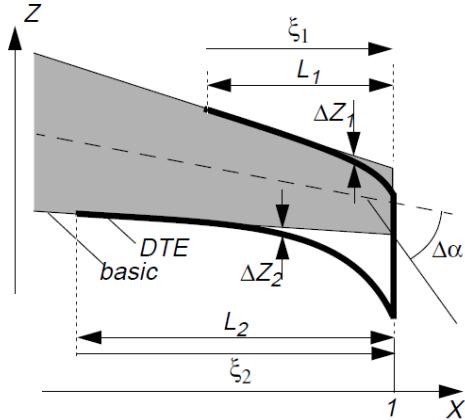


Figure 2.4 PARSEC parameters for DTE (Sobieczky 1998)

where the range of  $L$  was limited between 20 and 50 percent for upper and lower surfaces, respectively. The exponent  $n=3$  and  $\mu$  is between 1.3 and 1.8.

Modern civil transport aircraft normally fly in high subsonic conditions where the flow is transonic. The shock wave is usually generated on the upper (suction) surface. There have been many attempts to reduce the shock wave in order to improve the aerodynamic performance. Tai *et al* (1988) developed a humped aerofoil, which placed a compression ramp to decelerate the local Mach number before the shock wave. Ashill *et al.* (1993) presented a two-dimensional bump placed at the foot of the shock wave to achieve isentropic compression, which significantly weakens the shock strength and hence reduces wave drag without the significant viscous drag penalty. Sobieczky and Seebass (1984) introduced two suitable bumps on the aerofoil to weaken the shock wave. The first bump near the leading edge induced a set of expansion waves, and the second bump absorbed recompression waves which blended near the sonic recompression.

Local surface bump design has become more and more interesting in high subsonic wave drag reduction design. This design requires local curvature modification in the critical

areas; therefore, the parameterisation methods have to be capable of local surface control. As above mentioned in Samareh's review (2001), some of the parameterisation methods, such as the discrete method, domain element, B-spline and FFD, are able to produce the local shape variation. In the PARSEC method, this issue is considered as an extra bump function to be added on to the original shape. The definition of this bump function is shown in Equations 2.31-2.33 and Figure 2.5 presents the definition of PARSEC local surface bump control.

$$\Delta Z = Z_m \cdot \sin^{g(\xi)}(f(\xi)) \quad 2.31$$

$$f(\xi) = a\xi + b\xi^2 \quad 2.32$$

$$g(\xi) = (P - Q\xi) \cdot (1 - (1 - c) \cdot \sin(\xi)) \quad 2.33$$

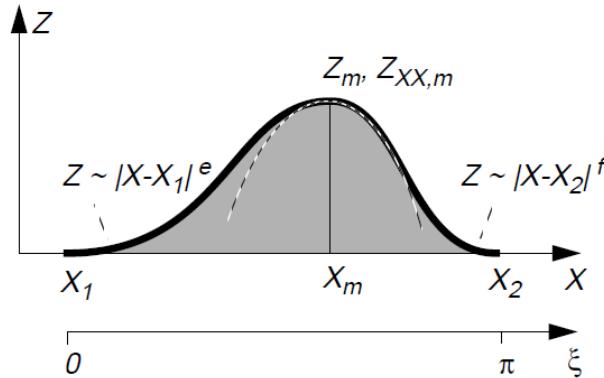


Figure 2.5 PARSEC parameters for local control bump (Sobieczky 1998)

The chordwise coordinate is converted to a local coordinate controlled by variable  $\xi$ .  $X_m$  and  $Z_m$  describe the position of the bump crest which is controlled by the coefficients  $a$  and  $b$ . The bump crest curvature  $Z_{XX,m}$  and the bump ramps curvature are controlled by  $P$ ,  $Q$  and  $c$  in function  $g(\xi)$ .

Thus far, PARSEC for 2D aerofoil design has been presented. For 3D real wing optimisation, two strategies could be employed to describe the wing surface. One way is to utilise many aerofoils as control sections (also called support sections or master sections) located along the wing in the spanwise direction. Each control section is

parameterised by the PARSEC method. The surface grid could be then calculated either using linear interpolation between these control sections or B-spline surface to lifting. This strategy is normally employed in wing optimisation work; however, it always requires many support sections for complex wing configurations. Otherwise, inaccuracy and non-smoothness in the spanwise direction could occur.

The other way is to analytically define the wing as a distribution of aerofoils in the spanwise direction. Therefore, each PARSEC parameter is then replaced by a distribution function along the spanwise direction. This strategy could overcome the drawback of using a control section and provide smoothness with only a few parameters. In Sobieczky's work, an Oblique Flying Wing design was carried out using both of these wing parameterisation strategies and the results showed that PARSEC with spanwise distribution for wing surface could be a promising method.

Many researchers have employed the PARSEC method in their work. Fuhrmann (2005) employed PARSEC to investigate a low Reynolds number, high Mach number aerofoil under Martian atmosphere conditions. Many test cases were performed to find the key parameters. It was found that the location of the upper crest position  $X_u$  was related with lift-to-drag ratio and the upper crest curvature  $Y_{xxu}$  and trailing edge angle had an effect on lift-to-drag ratio, but not as significant as the upper crest position  $X_u$ . The results showed that PARSEC was able to provide flexibility and robust results. However, Fuhrmann also pointed out that the PARSEC aerofoil definition is good for a slight modification of an existing aerofoil, but not good for generating a new aerofoil by randomly combining the parameters, since rippling and the intersection of the upper and lower surfaces could be occurred.

Khurana *et al.* (2008) studied characteristics of the PARSEC method. The design variables sensitivity analysis was performed to investigate the effect of each design variable. In this sensitivity analysis, the test started at a basic arbitrary aerofoil with 11 PARSEC variables. One variable was then perturbed while the rest of the variables were fixed. The perturbation was limited within a defined test domain to generate a series of

aerofoils. The geometric features and aerodynamic characteristics, such as thickness, camber, trailing edge angle, lift coefficient and drag coefficient, were calculated. Four independent aerofoils were used as a basic aerofoil. In order to visualise the process information, a self-organising map (SOM) technique was employed to reduce the results from high to low dimensional space. The results showed that the leading edge radius had a slight effect on aerofoil camber, and had no effect on thickness and trailing edge angle. Aerodynamic characteristics of the leading edge radius were tested as well, under subsonic flow condition. Results showed that the leading edge radius had a high effect on lift and drag coefficients, but not on moment coefficient. The crest position variable  $y_u$  had an effect on thickness and camber, but not on trailing edge angle. Aerodynamic performance of  $y_u$  indicated that it greatly affected the lift and drag coefficients. The SOM analysis was carried out over all PARSEC parameters and the test showed that the PARSEC parameters were able to provide independent one-to-one control over aerofoil geometry. The crest positions  $X_u$  and  $X_l$  were sensitive and needed to be limited carefully within an appropriate domain, otherwise an un-realistic geometry would be generated. In general, the PARSEC variables satisfied the requirement for parameterisation in aerodynamic optimisation. By this variable sensitivity study, the design limit bounds were discovered corresponding to geometric constraints; these constraints could assist in improving the optimiser efficiency. In the later work of Khurana *et al.* (2009), PARSEC was employed to take part in aerofoil shape optimisation. A test was carried out to search for a feasible design bound which would have a minimum number of un-realistic shapes. Four different aerofoils with disparate flight performances were used as test samples. An inverse fitting of PARSEC variables was implemented to find the design variables of each aerofoil. A particle swarm optimisation algorithm was employed to carry out this process with  $L_2$  norm as an objective function. Then, the design bound was determined using a certain map technique. This design bound was proofed in order that it could provide enough flexibility with minimum non-feasible aerofoils in a subsequent optimisation experiment.

Another similar research study was performed by Jeong *et al.* (2005) under transonic flow conditions. A transonic aerofoil design was parameterised using 11 PARSEC

variables. Analysis of variance (ANOVA) and SOM were employed to investigate the effect of each variable. The results of ANOVA showed that the crest position  $Y_u$  and  $Y_l$  had a large effect on the lift coefficient, and a similar effect on drag coefficient; it also presented that the leading edge radius could affect the drag coefficient. The SOM test gave the same conclusion. It found that  $Y_u$  was related to the drag coefficient, and  $Y_l$  was related to the lift coefficient. The small leading edge radius could decrease the drag coefficient. The conclusion was similar to the previous work of Khurana *et al.* (2008). The effect of each variable was identified and they also emphasised that some parameters should be limited in an appropriate region to avoid unrealistic geometry.

Padulo *et al.* (2009) tried to run a conceptual design under geometry uncertainty conditions, aiming to model geometry variability due to manufacturing or environmental variants. In this research, shape parameterisation played a very important role in determining the quality of the design space. The PARSEC-11 parameterisation method was selected to parameterise the aerofoil because it could provide intuitive geometric parameters. In their review work, they found there are some regions with flawed aerofoils in the entire design space provided by the PARSEC method. Some inherent problems of the PARSEC method were examined in further research. PARSEC used 11 parameters to solve a linear algebraic system of 12 equations. Subsequently, the coefficients of the polynomial were obtained; however, high order polynomials could lead to the appearance of more than one local maximum in the curve. In other words, although crest position parameters, such as  $X_u$ ,  $X_l$ ,  $Y_u$ ,  $Y_l$ , can be employed to control crest position, they cannot guarantee that the crest position is absolutely maximised. The other drawback is that there was no constraint imposed in PARSEC to guarantee avoiding the intersection between upper and lower surfaces. To identify the erroneous region of the PARSEC method, a parameterisation study was carried out on a large LP-Tau design of experiment (DOE) to screen the design space in a large initial search domain. In this study, LP-Tau sampling of 131072 points was used, and SOM was employed to visualise the data information. Finally, 2052 poor-condition profiles were identified in the design space. The results demonstrated that the erroneous profiles were related with values of  $X_u$ ,  $X_l$  and high values of  $Y_{xu}$ ,  $Y_{xl}$  and  $Z_{te}$  and inappropriate trailing edge angle. The reduced

region was then established, which could guarantee that the parameterisation method provided robust and meaningful results. The technique could also increase the efficiency of the optimisation. Thus, this work demonstrates that the crest position and curvature value should be selected extra carefully.

Winnemöller and van Dam (2007) performed a design and optimisation study on a thick aerofoil with blunt trailing edge. The blunt trailing edge has been shown to be good for the aerodynamic performance and structural cost of thick aerofoils; the blunt trailing edge could be able to decrease the sensitivity of lift to premature boundary-layer transition, but on the other hand it could contribute to the drag at the same time. In order to reduce the drag penalty, the PARSEC method was employed in aerofoil design work. Results showed that PARSEC was able to handle the thickness and trailing edge very well and provided enough flexibility to address the Pareto front for later aerodynamic performance study.

Pehlivanoglu (2009) performed a representation method effect study. PARSEC and Bezier parameterisation methods were compared using a vibrational genetic algorithm in a two-dimensional aerofoil design. Two test cases were implemented: an inverse optimisation was carried out to test two parameterisation methods in low speed flow conditions, and the second case was to test the performance of two parameterisation methods on drag reduction in transonic flow conditions. In the first case, PARSEC-11 provided better results than Bezier methods with 22 control points. PARSEC also attained convergence faster than Bezier methods. The second case also showed that the Bezier method provided slightly more drag reduction than the PARSEC method. The study concluded that the PARSEC method is more efficient than the Bezier method; however, the PARSEC method has less flexibility.

In Vavall and Qin's (2007) work, an improved response surface based optimisation for a two-dimensional aerofoil design under transonic flow conditions was carried out. PARSEC method was employed because it has low number of design variables. Two test cases were investigated to validate this response surface method. The results showed that

the PARSEC method was able to provide a large design space; however, unconventional profiles may be included. Shahrokhi and Jahangirian (2007) studied the effect of the PARSEC parameterisation method for viscous transonic flows. In their work, the original PARSEC method was exposed, in that it was unable to offer good control over the trailing edge area. To overcome this disadvantage, the Sobieczky DTE trailing edge method, which has been introduced earlier in this section, was used to increase the local curvature towards the trailing area. However, Shahrokhi and Jahangirian demonstrated that the Sobieczky DTE trailing edge could generate the intersection between upper and lower surfaces. Thus, a modified Sobieczky DTE trailing edge was introduced to avoid the intersection. Moreover, increasing the upper surface curvature in the trailing edge area may cause unfavourable pressure gradients on the upper surface, which could reduce the aerodynamic performance. Therefore, a new method was developed to overcome shortage by flattening the upper surface and, to compare this method, three test aerofoils with different methods were set up. In the first, the Sobieczky DTE trailing edge was used for both surfaces. In the second one, the new method was used for both surfaces. In the last one, the Sobieczky DTE trailing edge was used for the lower surface, and the new method was used for the upper surface. The results showed that the first aerofoil obtained the highest lift; however, the highest drag was obtained as well, and the lift-to-drag ratio was the lowest in all of the test cases. The second aerofoil could make a maximum reduction in the drag coefficient; however, the lift coefficient was decreased at the same time, because of the decrease of the trailing edge curvature on the lower surface. The third aerofoil could give a reasonable result with high drag reduction and slight lift reduction. This achieved the highest lift-to-drag ratio of the three aerofoils. Some further aerodynamic optimisation tests were carried out using genetic algorithm. The first case was to test the effect of the original PARSEC and the Sobieczky DTE trailing edge methods under transonic viscous flow conditions. RAE 2822 was employed as initial aerofoil, and the objective was to find the maximum lift-to-drag ratio. The results showed that both methods were able to weaken the shock wave. However, the Sobieczky DTE could provide further modification near the trailing edge area and therefore, the Sobieczky DTE trailing edge method could obtain more lift and more drag reduction. The second test case involved the new parameterisation method. The results showed that the

new method provided better pressure distribution on the upper and lower surfaces than the other methods, and the shock was almost eliminated. The highest lift-to-drag ratio was obtained by the new method as well. The convergence rate of the new method was much faster than the other methods. This demonstrated that the new method, based on the PARSEC method, provided more flexibility than the conventional PARSEC method in the aerofoil optimisation.

## 2.5 Class/shape function transformation (CST) methods

As stated at the beginning, an ideal parameterisation method is required which is able to provide a high level of flexibility and be compact in number of design variables and also provide physically meaningful design variables. The above reviewed methods either focus on the first two properties, such as Hicks-Henne, Bezier and B-spline, or focus on providing intuitive parameters, like PARSEC. Therefore, a universal method is desired which could not only satisfy most ideal properties of the geometric parameterisation method but also specifically represent aircraft aerodynamic components. A new approach, the so-called ‘class/shape function transformation’ method proposed by Kulfan (2006 - 2010), is increasingly used in aerofoil/aircraft optimisation. The purpose of this method is to develop a universal parameterisation method for complex aircraft configurations which is not limited just to aerofoils.

The CST method is initially derived from a mathematical representation of an aerofoil with round leading edge and aft-end. For this type of aerofoil, the difficulties in representing it mathematically are due to the infinite slope and second derivative requirement at the leading edge and large variations of curvature over the shape. The CST method was intended to overcome these limits and represent the different type of geometries in a generic way. It starts at a general mathematical expression for a two-dimensional aerofoil as:

$$\frac{z}{c} = \sqrt{\frac{x}{c}} \cdot \left(1 - \frac{x}{c}\right) \cdot \sum_{i=0}^N \left[ A_i \cdot \left(\frac{x}{c}\right)^i \right] + \frac{x}{c} \cdot \frac{\Delta z_{te}}{c} \quad 2.34$$

where  $\sqrt{\frac{x}{c}}$  describes the round nose,  $(1 - \frac{x}{c})$  describes the sharp trailing edge,  $\frac{\Delta z_{te}}{c}$  presents the trailing edge thickness and  $\sum_{i=0}^N \left[ A_i \cdot \left(\frac{x}{c}\right)^i \right]$  is a general function to describe the detailed shape.  $\sqrt{\frac{x}{c}}$ ,  $(1 - \frac{x}{c})$  and  $\frac{\Delta z_{te}}{c}$  terms are associated with the basic characteristics of aerofoils.

Therefore, this representation form can be rewritten as:

$$\xi(\psi) = C_{N_2}^{N_1}(\psi) \cdot S(\psi) + \psi \cdot \Delta \xi_{te} \quad 2.35$$

where

$$C_{N_2}^{N_1}(\psi) = \psi^{N_1} \cdot (1 - \psi)^{N_2} \quad 2.36$$

$$S(\psi) = \sum_{i=0}^N \left[ A_i \cdot \left(\frac{x}{c}\right)^i \right] \quad 2.37$$

where the  $C_{N_2}^{N_1}(\psi)$  is the class function,  $N_1$  and  $N_2$  are called class parameters,  $S(\psi)$  the shape function and  $\Delta \xi_{te}$  the trailing edge thickness ratio.

For the general aerofoil with a round nose and an aft-end trailing edge, the class parameters  $N_1$  and  $N_2$  are set to 0.5 and 1.0, respectively. In Kulfan's paper, the class function has been demonstrated to have a powerful capability in representing a large number of geometrical types. Table 2-1 illustrates the different types of geometries that are represented by the class function using various class parameters when the shape function  $S(\psi) = 1$ .

Table 2.1 Various types of geometries using different class parameters

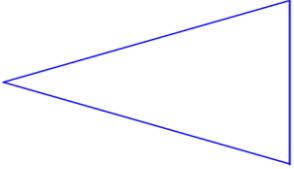
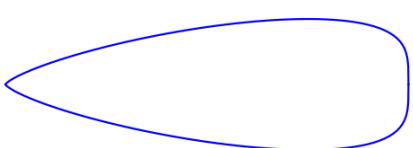
N <sub>1</sub>	N <sub>2</sub>	Shape	N <sub>1</sub>	N <sub>2</sub>	Shape
0.5	1.0		1.0	1.0	
0.5	0.5		0.75	0.75	
1.0	1.0001		0.75	0.25	

Table 2.1 shows that the supersonic aerofoil with sharp leading and trailing edges can be represented by using the class parameters, both of  $N_1$  and  $N_2$ , equal to 1.0. The circle and ellipse type of geometries could be represented when the class parameters both of  $N_1$  and  $N_2$  are set to 0.5. The wedge and bullet types of geometry could also be represented. These shapes are the most common shapes appearing in aerodynamic design.

Any kind of algebraic polynomial can be employed as the shape function. In the CST methods, the Bernstein polynomial is preferred for use as the shape function, since Bernstein polynomials have the mathematical property of ‘partition of unity’ and are more numerical stable than power form polynomials. The form of Bernstein polynomials is shown as:

$$S_i(\psi) = \sum_{r=0}^n K_{r,n} \psi^r (1 - \psi)^{n-r} \quad 2.38$$

where K is the binomial coefficient which is defined as

$$K_{i,n} \equiv \binom{n}{i} \equiv \frac{n!}{i! (n-i)!} \quad 2.39$$

The aerofoil shape can be represented using the Bernstein polynomial with different weight coefficients. These weight coefficients are then employed as design variables in optimisation. The total number of design variables depends on the order of the Bernstein polynomial, i.e.  $n+1$ . Eventually, the completed mathematical equation of the CST aerofoil could be written as:

$$\xi(\psi) = \psi^{0.5} \cdot (1 - \psi)^{1.0} \cdot \sum_{i=0}^n \left[ A_i \cdot \binom{n}{i} \cdot \psi^i \cdot (1 - \psi)^{n-i} \right] + \psi \cdot \Delta\xi_{te} \quad 2.40$$

The first weight coefficient of the Bernstein polynomial  $A_0$  corresponds to the leading edge radius:

$$S(0) = A_0 = \sqrt{\frac{2R_{le}}{c}} \quad 2.41$$

The last weight coefficient of the Bernstein polynomial corresponds to the trailing edge angle and trailing edge vertical position:

$$S(1) = A_n = -\tan(\theta) + \frac{\Delta Z_{te}}{c} \quad 2.42$$

The details of the derivation of this relation can be found in Kulfan (2006). Therefore, the CST method for aerofoils includes two intuitive parameters; the other coefficients in the CST method are non-intuitive. Some properties of the CST method for representing aerofoils have been summarised in Kulfan's paper (2006), as follows:

- a) Any aerofoil can be represented;
- b) This aerofoil representation technique provides a large design space of smooth aerofoils;
- c) Every aerofoil in the entire design space can be derived from the unit shape function aerofoil.

The convergence characteristics of the CST method were carried out to investigate the relationship between the CST method order and approximation accuracy in Kulfan's research. The CST method was firstly employed to inverse fit existing aerofoils with increasing Bernstein order (BPO) of the shape function; the convergence characteristics

of fitting accuracy of the CST method for various aerofoils were studied. A CFD solver called TRANAIR was secondly employed to test the convergence characteristics of fitting accuracy for aerodynamic performance. Three aerofoils, namely NACA 0012, RAE 2822 and NASA 02-714, were reported in the paper.

For the convergence studies of a symmetric NACA 0012 aerofoil, the geometric and aerodynamic characteristics of an approximate aerofoil were investigated, with 14 approximations obtained with Bernstein Polynomial Order (BPO) 2 to 15. The results showed that the approximated aerofoil with low BPO order was very close to the actual aerofoil. The error reduced rapidly with increasing BPO order, and almost vanished when BPO was greater than 9. The investigation of aerofoil 1<sup>st</sup> and 2<sup>nd</sup> derivatives demonstrated that the CST shape function provided strong smoothing control over the aerofoil shape. The TRANAIR flow solver was then employed to test the convergence of aerodynamics of the approximated aerofoil. Results showed that there was a significant error of the approximated aerofoil with the low BPO order, even if the geometric difference was slight. The approximated aerofoil with at least 6<sup>th</sup> order Bernstein polynomial could satisfy the error requirement in aerodynamics.

The convergence studies of the RAE 2822 cambered supercritical aerofoil indicated that it was similar to the NACA 0012 case, in that the aerodynamic and geometric characteristics were tested with 14 approximations obtained with different BPO orders. The results demonstrated that, for the approximated aerofoil with 6<sup>th</sup> order or greater, the Bernstein polynomial could make a very accurate fitting for this aerofoil. The aerodynamic test presented that the results of pressure distribution obtained with BPO 4 were very close to the results obtained by the original aerofoil; however, a BPO order of 8 or more is required to eliminate the difference of lift and drag coefficients between approximate and original aerofoils.

A NASA 02-714 cambered supercritical aerofoil was also presented in their report. As with the above two cases, the NASA 02-714 aerofoil was represented using BPO 2 to 15. The results of geometric fitting accuracy showed that BPO orders of 8 or higher were

required to accurately approximate geometric 1<sup>st</sup> and 2<sup>nd</sup> derivatives distribution of this aerofoil within wind tunnel tolerance. The results of the aerodynamic test showed that BPO 10 or higher was required to accurately capture the aerodynamic characteristics of this aerofoil. Kulfan's test clearly showed that for the CST to be able to represent the aerofoil very accurately. The BPO is normally required to be around 9.

Ceze *et al.* (2009) performed a further study to investigate the characteristics of the CST parameterisation method. The geometric representation error analysis was studied first; it was found that the  $L_2$  norm which described the difference the CST approximation and original shape decreased with increasing BPO order. The conclusion was the same as Kulfan's, in that BPO 9 was required to fit the existing aerofoil -. Afterwards, the numerical uniqueness of the CST parameterisation method was investigated: this was to check whether the set of parameters represent a unique shape and also whether perturbing an input parameter within a defined domain will result in a perturbed shape within a required domain. This is a significant influence on the efficiency of optimisation, since non-uniqueness parameterisation methods could generate noise in the design space. In order to investigate the numerical uniqueness, the CST method with  $n$ -th order BPO was re-written in matrix form when passed through  $n+1$  control points on an aerofoil.

$$\begin{bmatrix} S_0(x_0) & S_1(x_0) & \dots & S_n(x_0) \\ S_0(x_1) & S_1(x_1) & \dots & S_n(x_1) \\ \vdots & & \ddots & \\ S_0(x_n) & S_1(x_n) & \dots & S_n(x_n) \end{bmatrix} \cdot \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_n \end{bmatrix} = \begin{bmatrix} y(x_0) \\ y(x_1) \\ \vdots \\ y(x_n) \end{bmatrix} \quad 2.43$$

where

$$S_i(x) = C(x_i) \cdot K_{i,n} \cdot x^i \cdot (1-x)^{n-i} \quad 2.44$$

The matrix with S function on the left was specified as matrix  $M$ . The spectral condition number of matrix  $M$  was employed to examine the numerical uniqueness. The large value of spectral condition number presents the wide eigenvalues, which indicates the matrix  $M$  is ill-conditioned. If the matrix  $M$  is ill-conditioned, a small variation of  $y$  could cause a large variation of  $b$ . In other words, it implies that similar geometries could be

represented by widely different variables. The test of numerical uniqueness of the CST method demonstrated that the spectral condition number becomes large when increasing the BPO order. The matrix  $M$  approaches to be virtually singular when the BPO order is over 30. Another test for numerical uniqueness was to check if a shape could be represented by two different sets of variables. A 36<sup>th</sup> order BPO shape function was employed to fit an aerofoil with a unit shape function. For a large dimensional matrix, the exact inversion of the matrix would be hardly obtained or eventually non-existent. Thus, an approximate inverse with least-square sense was utilised to solve this linear system. The solved parameters showed that the two sets of parameters were largely different. However, the results of geometric error showed that there were small differences for both geometric shapes, and the  $L_2$  norm was  $7.7545 \times 10^{-7}$ , which is extremely small. The study concludes that high order shape function leads to shape numerical non-uniqueness, which should be avoided in optimisation. The results suggested that the BPO should not be greater than 15. Thus, care is needed to choose the order of CST parameterisation in optimisation.

Ceze *et al.* (2009) also performed an aerodynamic optimisation based on the adjoint method to validate the CST method. The pressure distribution of a NACA 0009 aerofoil was employed as target, and the NACA 0012 was used for initial geometry. For the first two test cases, the geometries were parameterised by the CST method with 2-parameters and 6-parameters respectively. The results showed that both tests could achieve the target geometry for a given tolerance. The comparison of the convergence rate demonstrated that low order parameterisation was faster than high order parameterisation. The second case employed the RAE 2822 as target aerofoil using higher order CST parameterisation with 11 parameters. The results showed that the high order CST method could converge to the target, and match the shock position. However, there were still differences, in the vicinity of the leading and the trailing edge areas, between the approximated and original aerofoil. The reason could be the steep descent method and inaccurate sensitivities solved by adjoint solver in those areas. The results demonstrated that the CST method was a promising parameterisation method in aerodynamic optimisation. Similar tests were done by Lane and Marshall (2010), who performed inverse aerofoil design to find an aerofoil

which could match the target pressure distribution using the CST methods. The results showed that the CST method could achieve the target aerofoil in accuracy and smoothness. Some other works which employed the CST method can be found in Bogue and Crist (2008), Haderlie and Crossley (2009) and Lane and Marshall (2010).

## 2.6 Comparison of parameterisation methods

Some researchers have carried out comparison studies of the different geometric parameterisation methods. Some of the more interesting works are summarised in this subsection in order to compare the characteristics of different parameterisation techniques.

Mousavi *et al.* (2007) performed a study of the effect of shape parameterisation techniques on three-dimensional aerodynamic optimisation. Three parameterisation methods, which were the mesh points, B-spline and CST methods, were employed. Inverse design and drag minimisation optimisation were performed for the study. The effects of the number of design variables and convergence rate were also investigated. The gradient-based optimiser with an adjoint approach was employed for optimisation. The first case was an inverse design with target pressure distribution taken from an ONERA M6 wing section. The aerofoil from M6 wing has high curvature at the leading edge, which is a challenge to parameterisation methods. For the mesh point methods, the smoothing algorithm was necessary to smooth the gradient. The low order CST method could be considered as a free ill-condition and high smooth shape; thus, there was no smoothing algorithm for the CST method. The results of B-spline showed that at least 32 control points were needed to achieve the high level of accuracy area. The highest level of accuracy was obtained by the mesh point method. The CST method started from the 5<sup>th</sup> order, and the results showed that increasing the order could improve the level of accuracy; however, the level of accuracy would be decreased with increasing order when the order is greater than 11. This is because the high order of polynomial developed high frequency and generated oscillations. For this case, the CST could not provide a good agreement with target geometry, especially in the leading edge area, although it could still capture the location of shock. The convergence showed that the most accurate result was obtained by the mesh point method. B-spline could provide a high level of accuracy,

which was close to that of the mesh point. However, the level of accuracy of the B-spline was independent of the number of control points. For the CST method, the best result was obtained by the 11<sup>th</sup> order. Although the level of accuracy of the CST method was lower than other methods, it employed quite a small number of design variables: in other words, the CST method could reach the higher level of accuracy with a lower number of design variables. The second case was to perform two-dimensional drag minimisation. The initial geometry was NACA 0012, and the objectives were to minimise the drag coefficient and maintain the current lift coefficient. The results showed that all three methods were able to reduce the drag coefficient from current 0.0153 to an average value of 0.0092. The best result was obtained by B-spline with 16 control points. The CST method also provided the similar amount of drag reduction as the other methods, but the number of design variables is much lower than the others. The third case was a three-dimensional drag minimisation. The M6 wing with NACA 0012 aerofoil was used as initial geometry. The results of B-spline showed that the shock was almost eliminated close to the root region. Although the shock still appeared close to the tip region, the strength was reduced. The drag coefficient obtained by B-spline was as low as the mesh point method. The results of the CST method showed that the shock wave was reduced close to the tip region. However, the shock wave close to the root region was not successfully eliminated. Improved shock reduction could be obtained by increasing the order of polynomial in the spanwise direction since more flexibility is provided using higher order. A further test was carried out to test the performance of the CST method on an unswept wing to ignore the effect of spanwise surface variation. The results showed that the CST method had a good performance for the unswept wing with reduced weak shock drag. Comparing all results, the mesh point method could provide the most drag reduction. However, the B-spline and the CST methods could dramatically reduce the number of design variables and obtain acceptable results in reduced design space even if the drag reduction was not as high as the mesh point method. It is noteworthy that the CST method could only use five variables to reduce the shock wave in the 2D drag minimisation case. Although the CST method could not successfully reduce the shock wave for a three-dimensional swept wing, it could successfully remove the shock wave

for a three-dimensional unswept wing and therefore it is a promising parameterisation method and some further investigation is necessary.

Another comparison study was carried out by Wu *et al.* (2003). In their study, the Hicks-Henne shape function method, the mesh-point method and the PARSEC parameterisation method were compared. Three transonic two-dimensional test cases were performed to investigate the performance of each parameterisation method. Adjoint method was used to obtain sensitivity information for a gradient-based optimiser. An inverse design test was carried out at first. Three pressure distributions from NACA 0015 cascade, 10<sup>th</sup> standard configuration compressor blade and the VKE turbine nozzle blade were used as targets. For the NACA 0015 cascade, all three parameterisation methods were able to handle its case. The Hicks-Henne had a very fast convergence compared with other methods. The mesh-point method had the highest accuracy. For the second and third test cases, both the Hicks-Henne shape function and the mesh point method were able to achieve the target aerofoil. Although the accuracy of the Hicks-Henne was slightly lower than the mesh-point method, the Hicks-Henne method could provide a fast convergence. The high accuracy of the Hicks-Henne method could be reached by increasing the number of variables. Compared with the mesh-point method, the Hicks-Henne method could provide smoothness without any smoothing algorithm. The results showed that the PARSEC method was good for processing the NACA 0015 cascade. However, the convergence rate was slower than the Hicks-Henne method even though the number of variables of the PARSEC method was lower. The results also clearly showed that the PARSEC method completely failed to achieve the target of blade cascade cases. This test demonstrated that there are some limitations of the PARSEC methods.

Castonguay and Nadarajah (2007) studied four parameterisation methods: the mesh point, the Hicks-Henne shape functions, B-spline and PARSEC. An inverse design was carried out with a target pressure distribution obtained from an ONERA M6 wing section. A NACA 0012 aerofoil was employed as initial aerofoil for each method. The test showed that B-spline was able to achieve the target aerofoil; however, at least 32 control points were required to expand the design space to provide sufficient flexibility. However, the

results also illustrated that the B-spline method could cause unrealistic results when a very high number of control points were employed. The results of the Hicks-Henne functions demonstrated that at least 30 functions were needed in order to provide high accuracy, but the level of accuracy was still lower than the mesh-points and B-spline methods. The reason was that the Hicks-Henne functions are not orthogonal. Orthogonality guarantees that each aerofoil shape corresponds to a unique set of input parameters. This property is particularly relevant to the parameterization methods which construct an aerofoil by combining existing sets of aerofoils or other types of base functions. This in turn can lead to spurious modality which arises where similar aerofoil geometries are created by different combinations of the base functions and is considered undesirable for optimisation processes.

The effect of the smoothing gradient algorithm was tested as well in Castonguay and Nadarajah's work (2007): it was able to accelerate the convergence rate of the B-spline and the Hicks-Henne. The highest inverse accuracy results were still obtained by the mesh-points method, but the efficiency of the mesh point method was lower than other methods. The B-spline and Hicks-Henne functions successfully performed drag reduction optimisation and reduced the drag to the same level using a similar number of design variables. In general, B-spline provided slightly smoother results than the Hicks-Henne functions. The PARSEC method was tested in an inverse design case. The final pressure distribution demonstrated that the PARSEC method was able to move the shock wave to match the target pressure distribution; however, the final pressure distribution at the leading edge area failed to match with the target. Convergence history of  $L_2$  norm showed that the PARSEC method only gave a slight improvement; this means that the PARSEC intuitive method is unable to perform in the inverse design case.

Furthermore, Antunes *et al.* (2009) implemented an aerodynamic optimisation study and three parameterisation techniques, Bezier, PARSEC and CST, were used to test the effect of parameterisation on optimisation. All of the parameterisation methods could provide excellent optimal solutions. It was obvious that the obtained optimal aerofoils were quite different due to the different parameterisation methods. Thus, the parameterisation

method has a profound effect on the optimisation results although the same geometry is contained in their design space. The results showed that the CST method provides more drag reduction than the other two methods. Sripawadkul and Padulo (2010) studied and compared five aerofoil parameterisation methods, namely Ferguson's curves, Hicks-Henne bump functions, B-Spline, PARSEC and class/shape function transformation (CST) methods, in terms of compact number of design variables, geometric fitting accuracy, orthogonality, realisticness and intuitiveness. The five parameterisation methods were scored to assist in selecting the best method in respect of a specific issue. Silisteau and Botez (2012) compared mesh points, polynomial, Bezier curve, B-spline and CST methods for low speed aerofoil design. They came to the conclusion that B-spline could always get a better optimum result than the other methods. However, the Bezier curve has a better global property as it changes the entire aerofoil.

The most common parameterisation methods have been reviewed in this chapter. As stated at the beginning of this chapter, parameterisation is fundamental to optimisation since it has a profound effect on design space. It inherently determines if the optimal solution is discovered in the design space and affects the optimisation efficiency. The ideal parameterisation method should provide high flexibility on design space; provide a compact number of design variables; and provide a smooth and realistic shape and intuitive physical meaningful design variables.

The discrete method is the simplest method which directly uses the grid points as the design variables. The literature showed that the discrete method could provide the highest flexibility. However, it lacks natural smoothness and, therefore, an additional smooth algorithm is required. The main disadvantage of the discrete method is that it requires a large number of design variables, which is not expected in aerodynamic optimisation. The other methods, such as Hick-Henne, Bezier polynomial and B-spline methods, are therefore preferred. The Hick-Henne method is a simple method which employs various shape functions with weighted numbers and adds them to an initial shape, the weighted numbers being used for the design variables. The main advantage of the Hick-Henne method is that the number of design variables can be reduced compared to the discrete

method. However, these shape functions are not orthogonal; large numbers of design variables are required to provide high flexibility. Furthermore, this method is only suitable for aerofoil and wing design; it is hard to be applied in complex 3D configurations. The Bezier curve and B-spline methods are the common parameterisation methods in aerodynamic design. These methods can greatly reduce the number of design variables and provide high flexibility. Various published research has presented results showing that the B-spline methods have a very good performance for aerodynamic optimisation. B-spline has more flexibility than Bezier curve, since piecewise function is employed; therefore, it is more popular than the Bezier curve methods, especially in a complex geometry case. B-spline is also able to handle a three-dimensional surface: thus, B-spline methods are normally preferred in complex configuration optimisation. However, B-spline could not accurately represent an implicit conic. A modified version of B-spline, NURBS, has been developed and it has been performed very well in recent research and has become an interesting method. B-spline and NURBS are normally employed in a CAD package to handle surface geometry. Using CAD software as a parameterisation method is promising for a complex industrial case. However, several shortcomings of the CAD parameterisation method, such as lack of sensitivity, closure geometries, change of geometry topology, etc. are challenging tasks for optimisation of practical cases. Therefore, the CAD method is not considered in our work.

In general, the traditional methods have been widely used for aerodynamic optimisation. These methods still find it hard to satisfy all the requirements for an ideal parameterisation method. For example, Bezier and B-spline methods are unable to provide the intuitive parameters to assist the engineers and are difficult for applying geometry constraints. Thus, two promising methods, namely PARSEC and CST methods, are studied.

The PARSEC method was developed by Sobieczky (1998). The impressive characteristics of the PARSEC method are that it could provide the intuitive design variables and reduce the number of design variables. The basic aerofoil with a round nose could be represented by only 11 variables. The key parameters, such as leading edge

radius, crest position of upper and lower surfaces, trailing edge angle, etc., appear explicitly in the parameterisation method which could directly assist the engineers to improve their design. Many researchers have employed the PARSEC method in their work. However, some of the literature has reported that PARSEC could not provide as a high flexibility as other common parameterisation methods, and could generate unrealistic shapes. Thus, further development of the PARSEC method could be carried out.

The CST method was developed by Kulfan (2006). The impressive feature of the CST method is that it could provide a universal way to parameterise the most common aircraft geometries. It provides high flexibility and the number of design variables is reasonably small. Some intuitive design variables, such as leading edge radius, trailing thickness etc., appear explicitly in the parameterisation. Some published work has asserted that CST is a promising parameterisation method. It could provide a good result in aerodynamic optimisation, and is very robust in providing a realistic shape. Kulfan (2006) also presented the potential capability of the CST method to represent three-dimensional geometries.

Therefore, the PARSEC and CST methods are selected for the starting-point of this project. The PARSEC method is more on the physical side and the CST method is more on the mathematical side. Some further developments of PARSEC and CST are carried out in this study to overcome their disadvantages. Later on in this research, the CST method is extended to represent different three-dimensional aerodynamic components of an entire civil transport aircraft.

# **Chapter 3 Development of CST and PARSEC Methods in Two-Dimensional Aerofoils**

In this chapter some further developments of the CST and PARSEC methods are demonstrated. In the first part, a new aerofoil parameterisation method is proposed, which tries to combine the flexibility and accuracy of the CST method and the intuitiveness of the PARSEC parameterisation method. The proposed new intuitive CST (iCST) method has been evaluated by comparing it with the CST and PARSEC methods regarding their accuracy in inversely fitting a wide range of aerofoils. In the second part, a further development of the CST method is proposed. In this development, the CST method employs the Rational Bernstein Polynomial to further improve the accuracy of the CST method and reduce the number of design variables.

## **3.1 Combination of CST and PARSEC: the intuitive CST method**

As mentioned in Chapter 2, the PARSEC method is a full intuitive parameterisation method for aerofoils. However, it is unable to provide enough flexibility. The reason is that the standard PARSEC-11 only employs 6<sup>th</sup> order polynomials which are not sufficient to represent some modern complex aerofoils. The advantages of the PARSEC method are obviously its intuitiveness and the small number of the design parameters. Nevertheless, some other key geometrical features between the leading edge and the crest, and the crest and the trailing edge, are uncontrolled, which can impact on the aerofoil's performance. For example, for natural laminar flow aerofoils, the crest position is related to the transition point, and the slope and curvature between the leading edge and crest are important for keeping the flow accelerating and giving a favourable pressure gradient (Paul and Ruxandra 2012). For the supercritical aerofoil, the shock strength and position are very sensitive to the upper surface curvature in order to maintain a near constant pressure over a large part of the surface to be terminated by a weak shock or compression wave. The slope and curvature on the upper surface between crest and trailing are

significant for the pressure recovery to avoid a large adverse pressure gradient and flow separation (Harris 1990).

Based on the above argument, it is proposed in this work to introduce the gradient and curvature at two  $x$ -positions between the leading edge and the crest, and between the trailing edge and the crest with their independent  $z$ -positions, for the upper and lower surfaces, respectively. This is in addition to the original PARSEC parameters. Note that the upper and lower leading edge radii can be different in the present parameterisation to give more flexibility. This extra set of 16-intuitive design variables gives more intuitive control of the aerofoil curves, with a total number of design variables of 28. Making use of the flexibility of the CST parameterisation, this set of intuitive parameters is mapped into the CST polynomial coefficients, which correspond to a 10<sup>th</sup> order CST.

The 16 intuitive design parameters, as shown in Figure 3.1, are therefore:

$$(X_{1,up}, Z_{1,up}, Z_{x,1,up}, Z_{xx,1,up}),$$

$$(X_{2,up}, Z_{2,up}, Z_{x,2,up}, Z_{xx,2,up}),$$

$$(X_{3,lo}, Z_{3,lo}, Z_{x,3,lo}, Z_{xx,3,lo}),$$

$$(X_{4,lo}, Z_{4,lo}, Z_{x,4,lo}, Z_{xx,4,lo}).$$

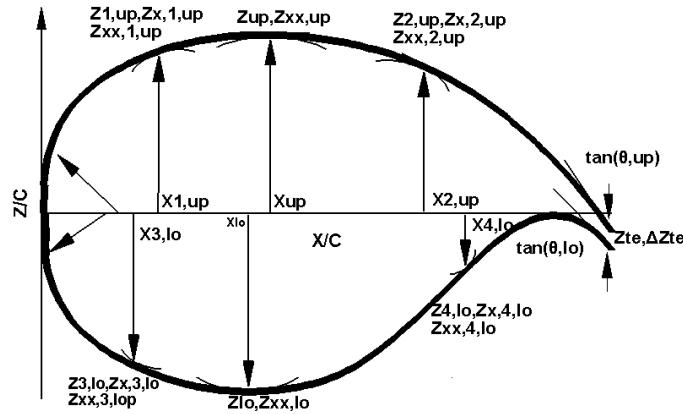


Figure 3.1 The intuitive CST parameterisation method

Different from the standard PARSEC-11 (Figure 2.3), in this method, two different leading edge radius parameters for the upper and lower surface are allowed to provide

more flexibility. Ceze *et al.* (2009) demonstrated the influence of leading edge radius on the geometric property of the entire curve including the leading edge region. Due to the different requirements for the upper and lower surfaces for the aerofoil from the physics flow point of view, independent upper and lower leading edge radii are more beneficial. The importance of the leading edge on aerodynamic drag was analysed theoretically for subsonic and transonic aerofoils with parabolic noses by Rusak (1993; 1994).

Indeed, for some practical aerofoils, the leading edge radii of the upper and lower surfaces are actually different. For example, the NASA phase 3 supercritical aerofoil is obtained by undercutting the lower leading edge surface of the phase 2 supercritical aerofoil, resulting in different leading edge radii between upper and lower surfaces (Harris 1990). The new full intuitive parameters are schematically illustrated in Figure 3.1 (Zhu and Qin 2013).

The total number of design variables is increased to 28 for the aerofoil. We need to build up the transformation between the intuitive parameters and the PARSEC mathematical equation with 11 coefficients. In PARSEC, simple sixth order polynomials, Equations 2.17 and 2.18, are employed. However, these simple polynomials are unstable for higher orders. However, for the CST polynomials with aerofoil class parameters, we can naturally limit the curve to be an aerofoil-like shape, and use the Bernstein polynomials, which are systematically more stable than the power form. Moreover, some intuitive parameters, such as leading edge radius, trailing edge thickness and tangential value of trailing edge angle, have already been included in the CST polynomial equation.

Kulfan and Bussoletti (2006) and Ceze *et al.* (2009) illustrated that the CST parameterisation methods are able to represent most aerofoils with high accuracy when the Bernstein polynomial order is higher than 9. Ceze *et al.* (2009) also pointed out the non-uniqueness issue of the CST parameterisation methods when the Bernstein polynomial is higher than 15 orders. In order to build up the linear system for mapping 28 intuitive parameters to the CST parameters, an 11x11 square matrix is required. Here, the 10<sup>th</sup> order CST polynomials with 24 parameters, 11 weighting coefficients and a trailing

edge position parameter for upper and lower surfaces, respectively, are employed to describe the shape:

$$Z(x) = x^{0.5} \cdot (1-x)^{1.0} \cdot \sum_{i=0}^{10} [A_i \cdot K_{i,10} \cdot x^i \cdot (1-x)^{10-i}] + x \cdot Z(x_{te}) \quad 3.1$$

The linear equations are then established for the upper and lower surfaces as:

$$\begin{aligned} A_{up} \cdot V_{up} &= B_{up} \\ A_{lo} \cdot V_{lo} &= B_{lo} \end{aligned} \quad 3.2$$

where

$$A_{up} = \begin{bmatrix} 1 & 0 & \cdots & \cdots & \cdots & 0 \\ a_{1,0} & a_{1,1} & \ddots & \ddots & \ddots & a_{1,10} \\ a_{2,0} & a_{2,1} & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ a_{9,0} & a_{9,1} & \cdots & \cdots & \cdots & a_{9,10} \\ 0 & 0 & \cdots & \cdots & 0 & -1 \end{bmatrix} \quad 3.3$$

$$V_{up} = [A_0, A_1, A_2 \dots A_{10}]^T \quad 3.4$$

$$B_{up} = \begin{bmatrix} \sqrt{2R_{te}} \\ Z_{1,up} - Z(X_{te})_{up} \cdot X_{1,up} \\ Z_{x,1,up} - Z(X_{te})_{up} \\ Z_{xx,1,up} \\ Z_{up} - Z(X_{te})_{up} \cdot X_{up} \\ -Z(X_{te})_{up} \\ Z_{xx,up} \\ Z_{2,up} - Z(X_{te})_{up} \cdot X_{2,up} \\ Z_{x,2,up} - Z(X_{te})_{up} \\ Z_{xx,2,up} \\ \tan(\theta_{up}) - Z(X_{te})_{up} \end{bmatrix}$$

and

$$Z(X_{te})_{up} = Z_{te} + \Delta Z_{te} \quad 3.5$$

$$a_{1,i} = K_{i,10} \cdot x_{1,up}^{i+0.5} \cdot (1-x_{1,up})^{10-i+1} \quad 3.6$$

$$\begin{aligned} a_{2,i} &= (i+0.5) \cdot x_{1,up}^{i-0.5} \cdot K_{i,10} \cdot (1-x_{1,up})^{10-i+1} \\ &\quad - (10-i+1)(1-x_{1,up})^{10-i} K_{i,10} x_{1,up}^{i+0.5} \end{aligned} \quad 3.7$$

$$a_{3,i} = K_{i,10}[(i+0.5)(i-0.5) \cdot x_{1,up}^{i-1.5} \cdot (1-x_{1,up})^{10-i+1} - (i+0.5)(10-i+1) \cdot x_{1,up}^{i-0.5} \cdot (1-x_{1,up})^{10-i} + (10-i+1)(10-i) \cdot (1-x_{1,up})^{10-i-1} x_{1,up}^{i+0.5} - (10-i+1)(i+0.5)(1-x_{1,up})^{10-i} x_{1,up}^{i-0.5}] \quad 3.8$$

$$a_{4,i} = K_{i,10} \cdot x_{up}^{i+0.5} \cdot (1-x_{up})^{10-i+1} \quad 3.9$$

$$a_{5,i} = (i+0.5) \cdot x_{up}^{i-0.5} \cdot K_{i,10} \cdot (1-x_{up})^{10-i+1} - (10-i+1)(1-x_{up})^{10-i} K_{i,10} x_{up}^{i+0.5} \quad 3.10$$

$$a_{6,i} = K_{i,10}[(i+0.5)(i-0.5) \cdot x_{up}^{i-1.5} \cdot (1-x_{up})^{10-i+1} - (i+0.5)(10-i+1) \cdot x_{up}^{i-0.5} \cdot (1-x_{up})^{10-i} + (10-i+1)(10-i) \cdot (1-x_{up})^{10-i-1} x_{up}^{i+0.5} - (10-i+1)(i+0.5)(1-x_{up})^{10-i} x_{up}^{i-0.5}] \quad 3.11$$

$$a_{7,i} = K_{i,10} \cdot x_{2,up}^{i+0.5} \cdot (1-x_{2,up})^{10-i+1} \quad 3.12$$

$$a_{8,i} = (i+0.5) \cdot x_{2,up}^{i-0.5} \cdot K_{i,10} \cdot (1-x_{2,up})^{10-i+1} - (10-i+1)(1-x_{2,up})^{10-i} x_{2,up}^{i+0.5} \quad 3.13$$

$$a_{9,i} = K_{i,10}[(i+0.5)(i-0.5) \cdot x_{2,up}^{i-1.5} \cdot (1-x_{2,up})^{10-i+1} - (i+0.5)(10-i+1) \cdot x_{2,up}^{i-0.5} \cdot (1-x_{2,up})^{10-i} + (10-i+1)(10-i) \cdot (1-x_{2,up})^{10-i-1} x_{2,up}^{i+0.5} - (10-i+1)(i+0.5)(1-x_{2,up})^{10-i} x_{2,up}^{i-0.5}] \quad 3.14$$

The linear system of equations for the lower surface can be similarly derived. The coefficients of the polynomials can then be obtained by solving the two linear systems of equations:

$$\begin{aligned} V_{up} &= A_{up}^{-1} \cdot B_{up} \\ V_{lo} &= A_{up}^{-1} \cdot B_{lo} \end{aligned} \quad 3.15$$

### 3.2 Geometric inverse fitting test and results of iCST methods

A well-behaved parameterisation method should be able to represent a wide range of existing aerofoils with high accuracy, indicating the flexibility of the parameterisation. In this work, a range of aerofoils, including natural laminar flow aerofoils, supercritical aerofoils and wind turbine aerofoils, have been employed to test the inverse fitting performance of the CST, PARSEC and the proposed iCST methods.

### 3.2.1. Geometric inverse fitting

The inverse curve fitting is an optimisation problem in itself, defined by minimizing the difference between the approximated and the original curves. The  $L_2$ -norm employed to describe this difference is given by:

$$L_2 = \sqrt{\sum_i^{Np} (y_{i,original}(x) - y_{i,approx}(x))^2} \quad 3.16$$

An SLSQP (Nocedal and Wright 2006) optimizer from the Python program library is employed to minimize the  $L_2$  norm which is the objective function. In order to make comparisons, PARSEC-12 is used, allowing different upper and lower leading edge radii. A 12<sup>th</sup> order CST method is employed, which has 28 parameters, 14 parameters for upper and lower surfaces, respectively, to make a fair comparison with the same number of design variables as for the iCST method. The tolerance for wind tunnel geometries, namely  $5 \times 10^{-4}$  relative to the chord length, is employed to examine their fitting accuracy (Kulfan 2006).

### 3.2.2. Inverse fitting test results

Three types of aerofoils are employed for comparing the inverse fitting of three parameterisation methods. The supercritical aerofoils are used for the first test. They were designed to delay the shock wave on the upper surface and reduce wave drag, and have been widely applied in high speed subsonic aircraft, which fly at a Mach number ranging between 0.6~0.85. Performance of shape parameterisation for supercritical aerofoils is significant in aircraft aerofoil design and optimisation. Three typical supercritical aerofoils, RAE 2822, RAE 5214 and NASA SC-20714 (Harris 1990), are chosen for this study, and the results of geometric fitting by three methods and the error distributions are shown in Figure 3.2 to 3.4.

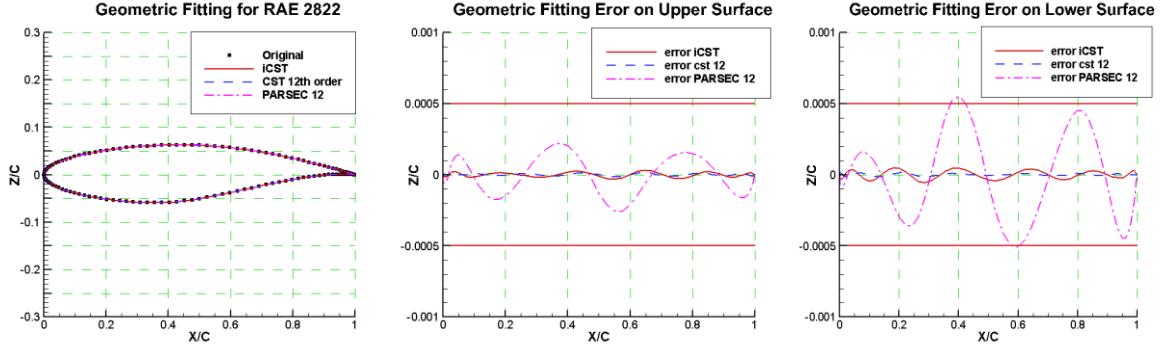


Figure 3.2 Geometric fitting for RAE 2822 using iCST, CST 12 and PARSEC 12

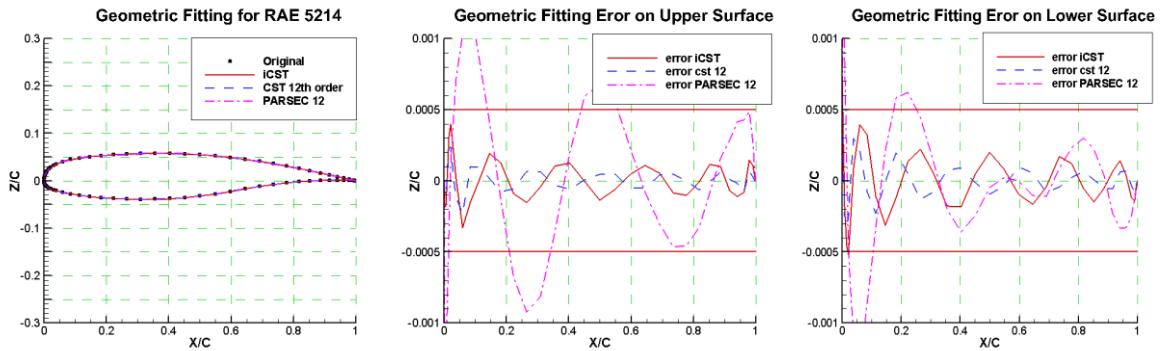


Figure 3.3 Geometric fitting for RAE 5214 using iCST, CST 12 and PARSEC 12

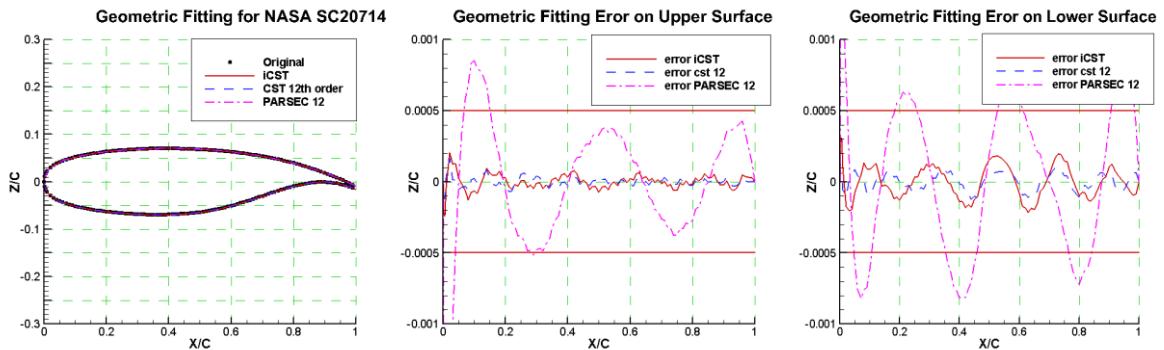


Figure 3.4 Geometric fitting for SC-20714 using iCST, CST 12 and PARSEC 12

The second types of aerofoils tested are the Natural Laminar Flow (NLF) aerofoils. The NLF aerofoils were designed to delay the flow transition from laminar to turbulent boundary layer, and as a result, the skin friction could be dramatically reduced. It is an increasingly interesting topic for aerodynamic designers since it could potentially give

further drag reduction and improve the aircraft performance to satisfy demanding economic and emission requirements in the future. Therefore, the suitability of the shape parameterisation methods for NLF aerofoils is practically important. Three traditional NLF aerofoils, namely NASA NLF-414F, NLF-416 and HSNLF-213, are employed for this study (Viken 1986; Viken *et al.* 1987). NLF-416 is a relatively simple NLF aerofoil but NLF-414F and HSNLF-213 are more complicated with more variation of curvature near the trailing edge area to avoid flow separation. The results of fitting the NLF aerofoils by the three parameterisation methods are shown in Figure 3.5 to 3.7.

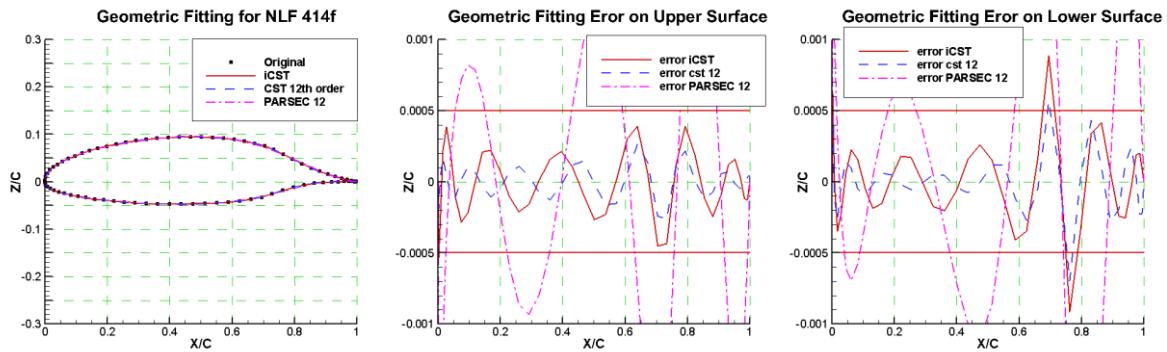


Figure 3.5 Geometric fitting for NLF 414F using iCST, CST 12 and PARSEC 12

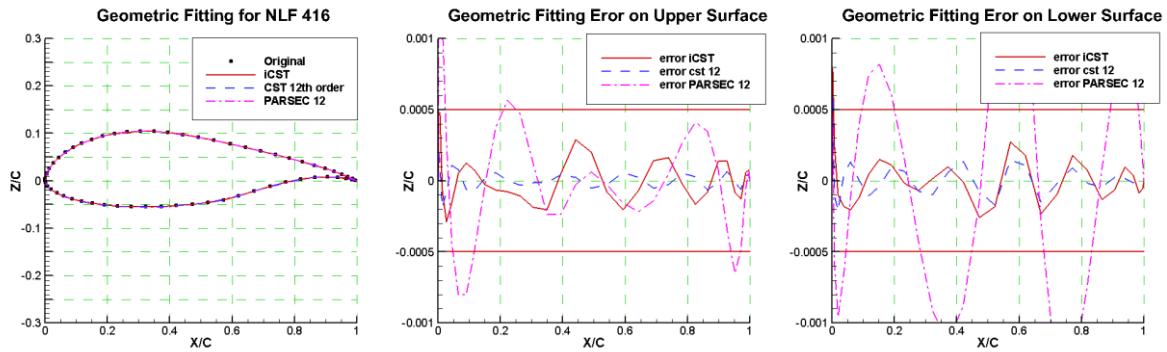


Figure 3.6 Geometric fitting for NLF 416 using iCST, CST 12 and PARSEC 12

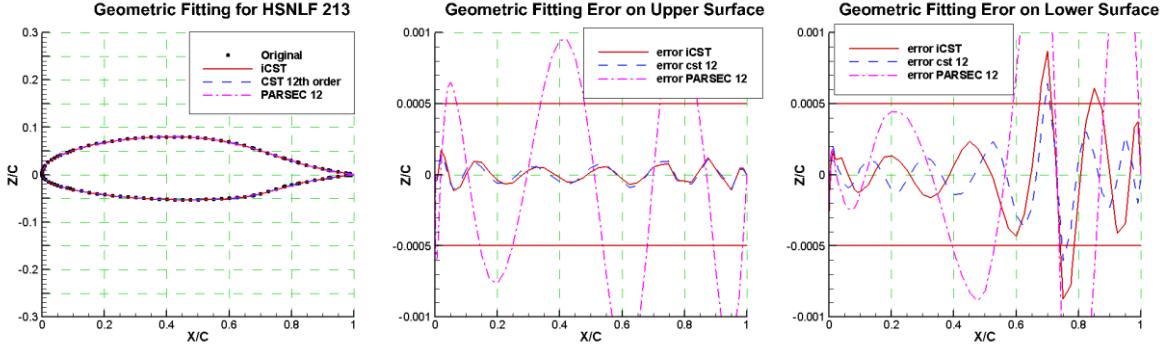


Figure 3.7 Geometric fitting for HSNLF 213 using iCST, CST 12 and PARSEC 12

Some wind turbine aerofoils are employed for the last test. Wind turbines are used to convert wind energy to electrical power and play an increasingly important role in generating clean and renewable energy (Mohammed *et al.* 2012; Timoleon *et al.* 2012). Improvement of the efficiency of wind turbine blades is crucial for the design of wind turbines in the future. Many researchers have employed a numerical optimisation method for wind turbine aerofoils (Fudge and Zingg 2005; Timoleon *et al.* 2012; Wang *et al.* 2009). An effective shape parameterisation to represent wind turbine type aerofoils with high accuracy would be helpful for design. Three wind turbine aerofoils, S805A, S809 and S825, from NREL (Tangler and Somers 1995) are used in this study and the results are shown in Figure 3.8 to 3.10.

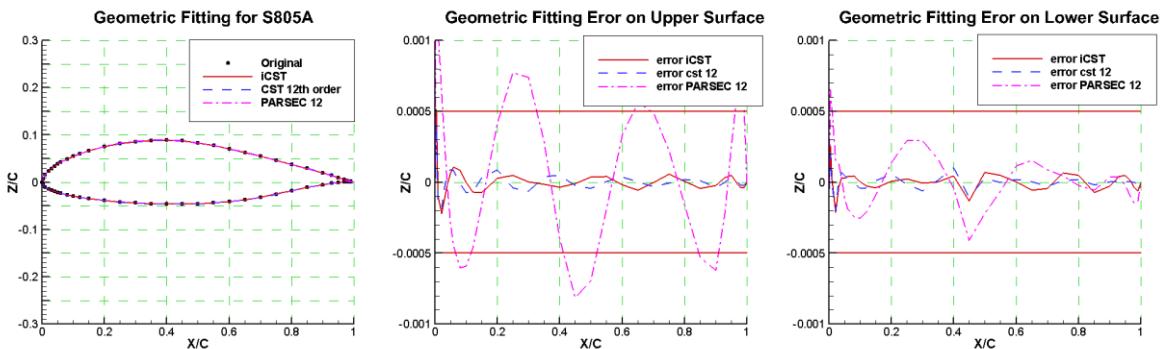


Figure 3.8 Geometric fitting for S805A using iCST, CST 12 and PARSEC 12

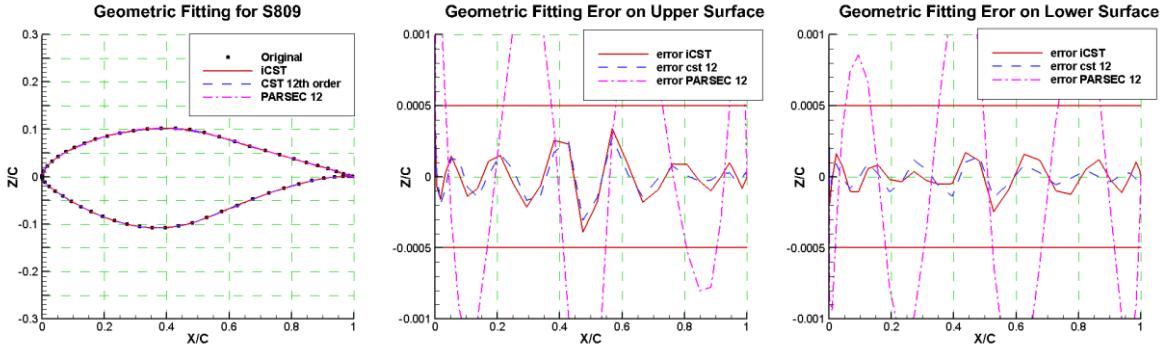


Figure 3.9 Geometric fitting for S809 using iCST, CST 12 and PARSEC 12

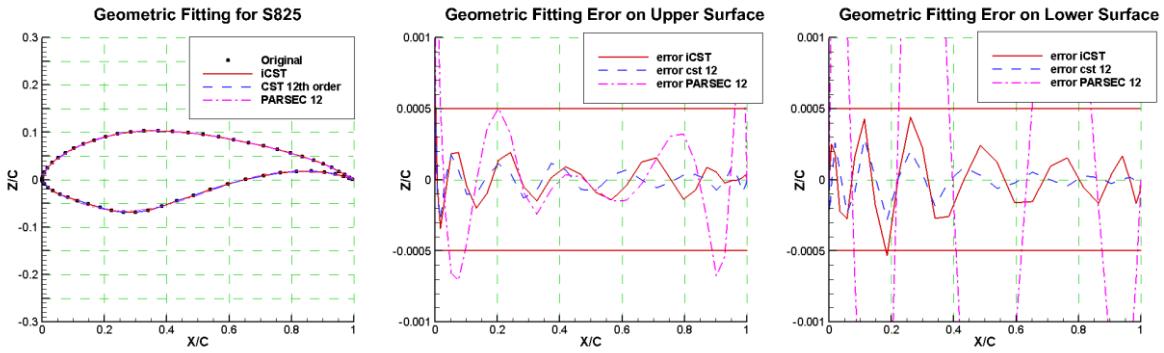


Figure 3.10 Geometric fitting for S825 using iCST, CST 12 and PARSEC 12

The results for supercritical aerofoils show that all methods are able to fit the RAE 2822 aerofoil within the tolerance. The 12<sup>th</sup> order CST and the iCST methods are able to fit RAE 5214 and NASA SC-20714 to a higher accuracy. However, PARSEC-12 cannot fit RAE 5214 and NASA SC-20714 as well as RAE 2822, and a large error occurs at the leading edge.

For the NLF aerofoils, the 12<sup>th</sup> order CST can fit all three within the tolerance. The iCST can fit the NLF 416 as well as the 12<sup>th</sup> order CST. For NLF-414F and HSNLF-213, there are some slightly larger errors in the leading edge area and in the region between 60% and 80% chord on lower surface, where a large curvature change occurs. PARSEC-12 is able generally to fit NLF-416 on the upper surface, but very large error is observed on the entire lower surface and at the leading edge area on the upper surface. Moreover, PARSEC-12 fails to fit the NLF-414F and HSNLF-213, and in the region between 60%

and 80% chord on the lower surface the error is more than twice the tolerance. In general, the PARSEC-12 method shows poor performance in fitting NLF aerofoils.

For the wind turbine aerofoils, the 12<sup>th</sup> order CST is able to fit all three aerofoils. Although a slightly larger error is found in leading edge area on the upper surface of S805A and the lower surface of S825 within the tolerance, the iCST method also fits the three wind turbine aerofoils very well. PARSEC-12 is able to fit the S805A aerofoil within reasonable error. However, it fails to fit the S809 and S825 aerofoils and huge errors are found on both upper and lower surfaces.

The results show that the 12<sup>th</sup> order CST and iCST are able to represent all test aerofoils with high accuracy. In the case of NASA SC20714, NLF 416, S805A and S809, the magnitude of the fitting errors of iCST and the 12th order CST are almost the same.

### **3.3 CST method with rational function (RCST)**

The results of the test comparison study of the CST, PARSEC and iCST methods found that CST with a 12<sup>th</sup> order Bernstein polynomial is generally able to fit all aerofoils within allowable tolerance. However, there are still some areas that could not be represented very well, for example, the leading edge of RAE 5214, NLF 414F and NLF 416, the curvature change area of NLF 414F and HSNLF 213 and the lower surface of S825 aerofoils. Increasing BPO could certainly improve the fitting accuracy and provide large flexibility. However, in the previous test case, the BPO has already reached 12<sup>th</sup> order, which has approached the limitation of BPO 15. Therefore, continually increasing the BPO is not a solution for the CST methods. This means that the CST method with Bernstein polynomials has an inherent drawback for accurate representation of a complex aerofoil in its design space. In certain engineering work, such as preliminary and detailed design, the intuitiveness could be considered as secondary. Therefore, the high accuracy parameterisation method with a low number of design variable methods is still desired.

In order to improve the CST method, Powell and Søbester (2010) have tried to employ genetic programming, which is a technique used to optimise and design a new formula to

satisfy the target objective. In their work, genetic programming is employed to optimise a new class function to minimise the fitting  $L_2$  norm error for an existing aerofoil. However, implementation of genetic programming is not easy. The other issue is that it requires changing the class base function each time when applying to a different aerofoil. This negated the idea of using the class function. Straathof and van Tooren (2011) have applied the B-spline piecewise basis function into the shape function of the CST methods, which is so called the class-shape-refinement-transformation (CSRT). They demonstrated that this could bring a capability of local control to the CST methods and make CST able to provide more flexibility. They also showed the capability of CSRT to control volume constraints. However, using a local control B-spline basis function will make the issue more complicated, for example the knot position will be considered as an input condition. Furthermore, the local control capability will also bring more local optimal in the design space and has potentially negative effects on optimisation efficiency.

Therefore, a further development of the CST method, but keeping its properties, is desired. The standard CST method employs Bernstein polynomials. In Chapter 2, it has been reviewed that the Bernstein polynomials are unable to accurately present the conic, circle, ellipse and hyperbola curves (Versprille 1975; Tiller 1993; Samareh 2001). The rational Bernstein polynomials could overcome this drawback of the standard Bernstein polynomials (Forrest 1968; Farin 1983; Piegl and Tiller 1997). Therefore, the CST method has been developed to couple with rational Bernstein polynomials here, and noted as ‘RCST’. The form of RCST for an aerofoil is written as below:

$$\xi(\psi) = C_{N_2}^{N_1}(\psi) \cdot SR(\psi) + \psi \cdot \Delta\xi_{te} \quad 3.17$$

where

$$C_{N_2}^{N_1}(\psi) = \psi^{N_1} \cdot (1 - \psi)^{N_2} \quad 3.18$$

$$SR(\psi) = \frac{\sum_{i=0}^n A_i \cdot W_i \cdot K_{i,n} \cdot \psi^i \cdot (1 - \psi)^{n-i}}{\sum_{i=0}^n W_i \cdot K_{i,n} \cdot \psi^i \cdot (1 - \psi)^{n-i}} \quad 3.19$$

where the  $C_{N_2}^{N_1}(\psi)$  is the class function,  $\Delta\xi_{te}$  the trailing edge thickness ratio (they are the same as with standard CST methods),  $SR(\psi)$  is the shape function with rational Bernstein polynomials,  $W_i$  is the *weights*,  $A_i$  are the coefficients of polynomials.

For the round nose and aft-end aerofoils, the class parameters are still set to  $N_1 = 0.5$  and  $N_2 = 1.0$ . Furthermore, the employment of rational Bernstein polynomials will not affect the ability of the first and last coefficients to represent their physical meaning. The details can be found in Appendix B.

### 3.4 Geometric fitting results of RCST

RCST with 6<sup>th</sup> order rational Bernstein polynomials is then applied to inverse fitting for all of the aerofoils which are employed in the previous test. The coefficients  $A_i$  and *weights*  $W_i$  are the design variables to be changed by optimiser to find the minimum  $L_2$  norm error. The results of RAE 2822, RAE 5214 and NASA SC 20-714 are shown in Figure 3.11 to 3.13. The results of NLF 414F, NLF 416 and HSNLF213 are shown in Figure 3.14 to 3.16. The results of S805A, S809 and S825 are shown in Figure 3.17 to 3.19.

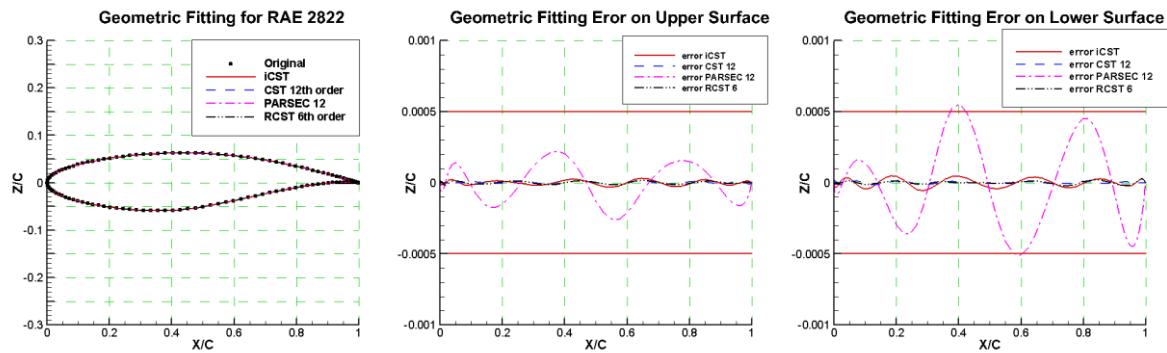


Figure 3.11 Geometric fitting for RAE 2822 using 6th order RCST

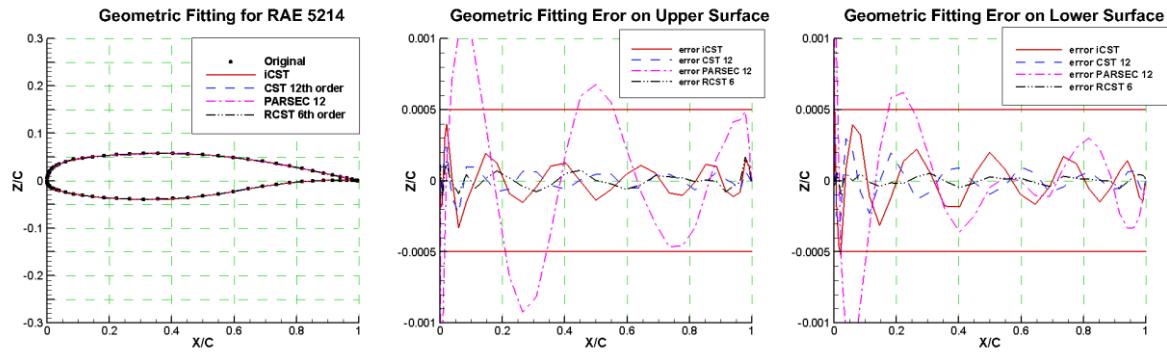


Figure 3.12 Geometric fitting for RAE 5214 using 6th order RCST

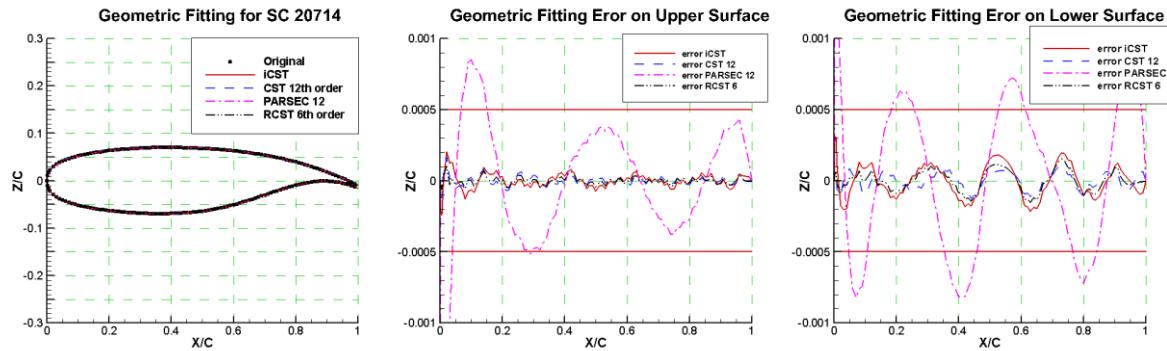


Figure 3.13 Geometric fitting for NASA SC-20714 using 6th order RCST

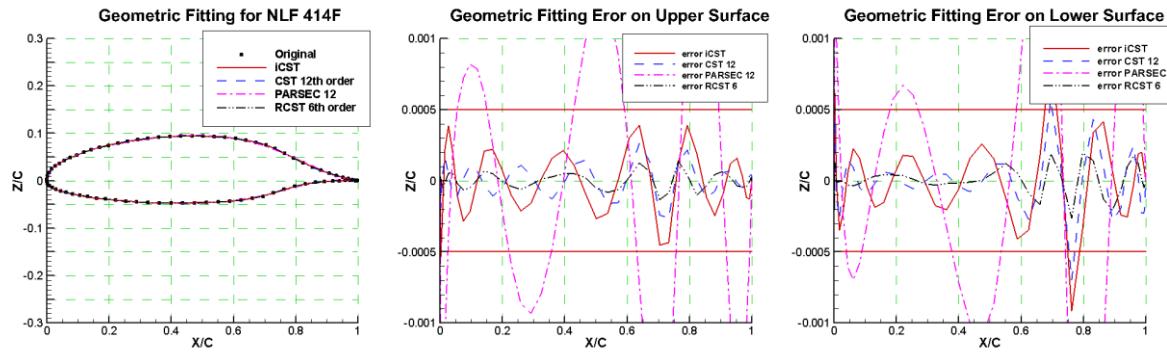


Figure 3.14 Geometric fitting for NLF 414F using 6th order RCST

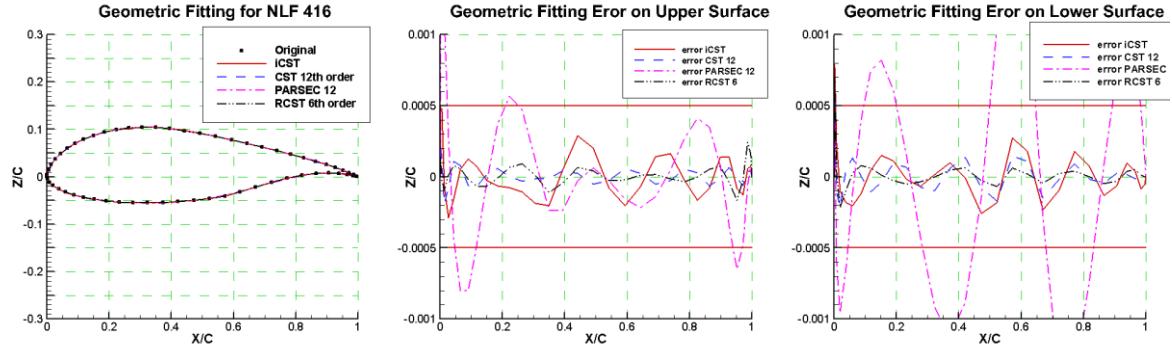


Figure 3.15 Geometric fitting for NLF 416 using 6th order RCST

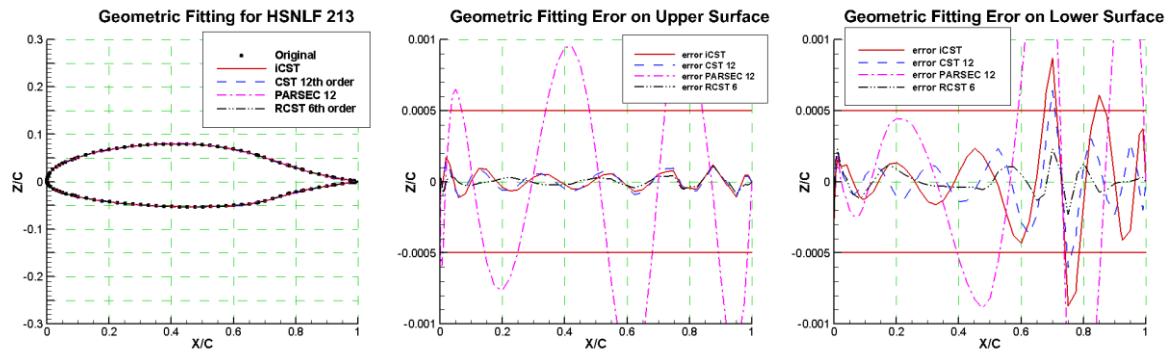


Figure 3.16 Geometric fitting for HSNLF 213 using 6th order RCST

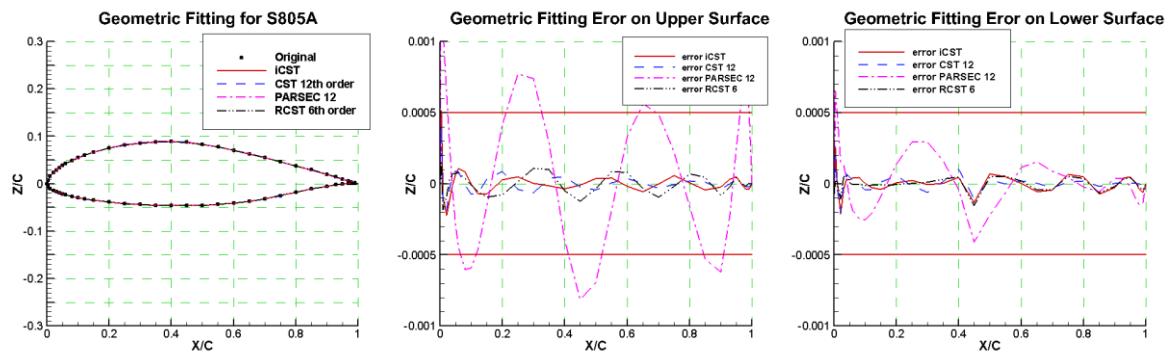


Figure 3.17 Geometric fitting for S805A using 6th order RCST

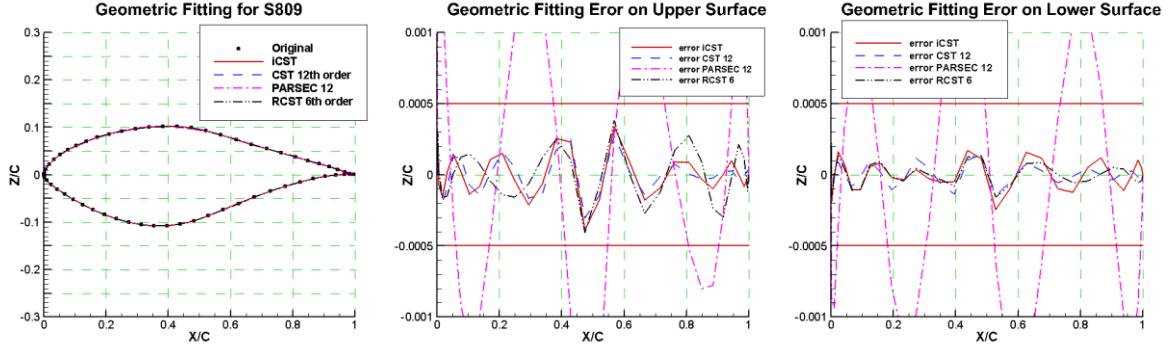


Figure 3.18 Geometric fitting for S809 using 6th order RCST

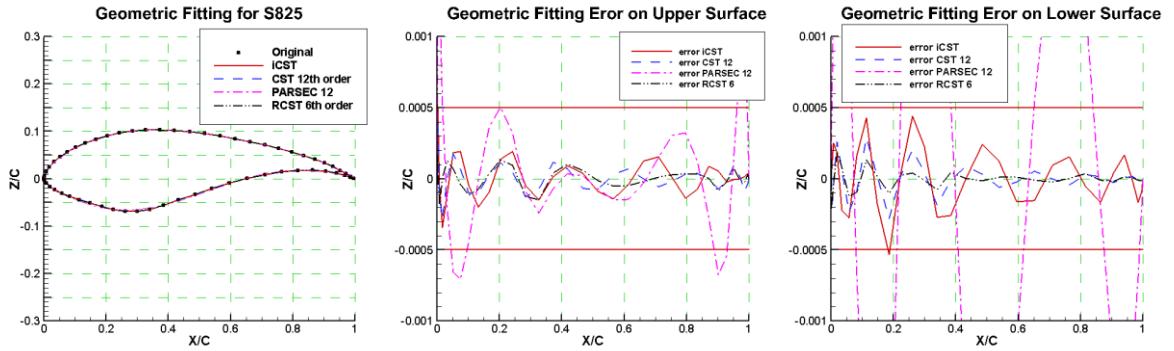


Figure 3.19 Geometric fitting for S825 using 6th order RCST

The results show the impressive improvement of fitting accuracy of RCST. First, the fitting errors of all aerofoil shapes are completely controlled under the allowed tolerance, even if in the curvature change area of the NLF 414F and HSNLF 213. Compared with standard CST 12<sup>th</sup> order, these areas were not able to be fitted accurately by the standard CST method. Second, the fitting errors of the leading edge areas of all aerofoils are controlled within  $2 \times 10^{-5}$ . Compared with standard CST methods, the leading areas of RAE 5214, NLF 414F and NLF 416 are much improved. Considering the sensitivities of leading edge to aerodynamic flow, this improvement is significant.

Furthermore, the results only show the 6<sup>th</sup> order rational polynomials. High accuracy has been already achieved. Therefore, higher order RCST has not been shown since it is able to theoretically increase the accuracy. In optimisation, only the coefficients  $A_i$  will be

employed as design variables in order to avoid non-linear effects. This means that the 6<sup>th</sup> order RCST will only have seven parameters plus one trailing edge parameter on each side of the aerofoil. This is equivalent to the standard CST with 12<sup>th</sup> order BPO. However, the accuracy of RCST is even higher than the standard CST with 12<sup>th</sup> order. Therefore, RCST has improved the standard CST and further reduced the number of design variables.

### 3.5 Conclusion

In the first part of this chapter a new parameterisation method is proposed to combine the features of high flexibility from the CST method and full intuitiveness from the PARSEC method. This new method introduces eight extra intuitive parameters at two x-positions between the crest and the leading edge and between the crest and the trailing edge on each surface of the aerofoils based on the original PARSEC-12 method. Therefore, 28 design variables are used to parameterize the entire aerofoil. The power form polynomials used in PARSEC have been replaced by the 10<sup>th</sup> order CST equation for curve representation. Its performance is examined by checking the accuracy of inverse fitting for a range of aerofoils in comparison with the 12<sup>th</sup> order CST and PARSEC-12 methods.

The results show that the 12<sup>th</sup> order CST and the iCST methods with the same number of design variables are able to represent supercritical and wind turbine aerofoils with a high level of fitting accuracy, while the representation of complex NLF aerofoils is also acceptable but it shows a slightly higher error after the half chord position. With insufficient number of control parameters, the standard PARSEC-12 fails to fit most of the aerofoils tested here, and is particularly problematic for NLF and wind turbine aerofoils.

The iCST method can be viewed as an extension of the PARSEC method with full intuitive parameters. It can also be viewed as a transformation of the originally non-intuitive parameters in the CST method to a full set of intuitive parameters through a transformation matrix.

In comparison with the CST method, the proposed iCST method provides opportunities for the aerodynamic designers to understand the relationship between the aerodynamic properties and the geometric features and to guide the exploration of the design space by selecting proper design variables and setting proper bounds/constraints in the optimisation process.

In the second part, a further modification of the CST method has been developed for application when intuitiveness is not desired. In the new developed version of CST, the rational Bernstein polynomials are employed to replace the standard Bernstein polynomials in standard CST methods, and notated as RCST.

The performance of RCST is examined by inverse fitting a range of aerofoils and compared with the standard CST method from a previous test case. The results clearly showed the significant improvement obtained by the RCST method. All aerofoils can be fitted within tolerance. Especially, the representation of the leading edge area is significantly improved. In addition, the lower 6<sup>th</sup> order RCST could obtain higher accuracy than the standard CST 12<sup>th</sup> order. However, in optimisation, the number of design variables of 6<sup>th</sup> order RCST will be half of the standard CST 12<sup>th</sup> order. This demonstrated that RCST could significantly reduce the number of design variables. The performance of iCST and RCST will be further tested in later optimisation test work.

# **Chapter 4 CST Parameterisation Method for the Entire Aircraft**

In this chapter, the CST method is investigated for geometric representation of an entire transport aircraft for aerodynamic optimisation purposes. As studied in Chapter 2, a class function of the CST methods is employed to define the fundamental geometry, and a shape function is used to describe in detail the shape of the geometry. In this chapter, the inverse fitting accuracy of the CST method for a three-dimensional wing has been examined to investigate firstly the effects of varying the order of the Bernstein polynomials. The new RCST method is extended from a two-dimensional aerofoil to a three-dimensional wing. The performance of inverse fitting accuracy of the RCST method is then examined.

An investigation of the CST method for an entire passenger transport aircraft is then presented, including such components as the wing, horizontal tail plane, vertical tail plane, fuselage, belly-fairing, blended wingtip device, nacelle, flap tracking fairing and pylon. The CST method for local shock bump parameterisation is also developed and presented in this chapter. At the end of this chapter, the methodology for calculating the intersection line is briefly presented. This method employs the Newton-Raphson method based on direct differentiation of the CST geometries.

## **4.1 Parameterisation for wing type geometries**

### **4.1.1. Standard CST for wing type geometries**

Kulfan and Bussoletti (2006) presented the capability of the CST method to represent a three-dimensional wing. They define the wing as a distribution of the aerofoils along the spanwise direction. The CST functions for 2D aerofoils can be directly employed to define the wing aerofoil section. However, because the planform of a realistic wing has dihedral and twist, two extra parameters for wing section height installation position and twist are introduced to the 2D aerofoil CST function, as shown in Figure 4.1.

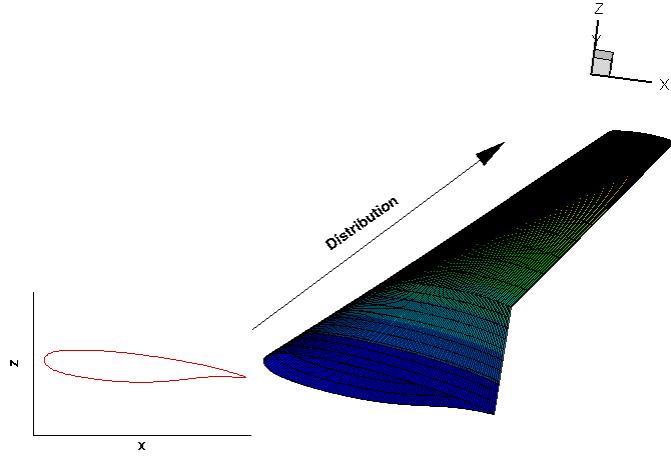


Figure 4.1 Wing aerofoil section definition in the CST method

To define the wing twist, the rotation coordinate transformation function is applied to an aerofoil. Equations 4.1 and 4.2 show the rotation coordinate transformation of an aerofoil on the x-z plane with  $\Delta\alpha_T$  nose up:

$$x' = x \cdot \cos(-\Delta\alpha_T) - z \cdot \sin(-\Delta\alpha_T) \quad 4.1$$

$$z' = x \cdot \sin(-\Delta\alpha_T) + z \cdot \cos(-\Delta\alpha_T) \quad 4.2$$

if the non-dimensional chord length of the aerofoil projected onto the x-coordinate is 1. After the rotation coordinate transformation, the leading edge position and non-dimensional chord length will be changed. Therefore, in the proper wing definition, the new leading edge should be found, and the aerofoil should be then re-scaled to keep the non-dimensional chord length on the projected x-coordinate to maintain the wing planform.

In order to fix the non-dimensional length that the section aerofoil projected onto the x-coordinate, the value of the tangent of the twist angle is subtracted directly from the CST equation rather than applying the rotation equation in Kulfan's definition. Although the camber distribution along the chordwise direction is changed, the shape function will be

applied to compensate for this. The CST equation of a wing section in non-dimensional form is written as:

$$\xi_U(\psi) = \xi_n + C_{N2}^{N1}(\psi) \bullet \sum_i^n Au_i \bullet S_i(\psi) + \psi[\xi_{TE,upper} - \tan[\alpha_{Twist}]] \quad 4.3$$

$$\xi_L(\psi) = \xi_n + C_{N2}^{N1}(\psi) \bullet \sum_i^n Al_i \bullet S_i(\psi) + \psi[\xi_{TE,lower} - \tan[\alpha_{Twist}]] \quad 4.4$$

where  $\xi_n(\eta) = Z_n / C_{local}$  is the non-dimensional local wing section height installation position,  $\Delta\alpha_T$  is the local wing section twist angle,  $Au$  and  $Al$  are the control parameters for the aerofoil and  $\xi_{TE}$  is the non-dimensional trailing edge position.  $S_i(\psi)$  is the shape function which is the same as in Equation 2.37.

In order to extend the 2D aerofoil to a 3D non-dimensional wing surface, 3D Bernstein Shape function which is similar to Equation 2.7 is applied. The CST equations for the 3D non-dimensional wing surface could be obtained as:

$$\xi_U(\psi, \eta) = C_{N2}^{N1}(\psi) \bullet \sum_i^{Nx} \sum_j^{Ny} [Bu_{i,j} \bullet Sy_j(\eta) \bullet Sx_i(\psi)] + \psi[\xi_T(\eta) - \tan[\alpha_{twist}(\eta)]] + \xi_n(\eta) \quad 4.5$$

$$\xi_L(\psi, \eta) = C_{N2}^{N1}(\psi) \bullet \sum_i^{Nx} \sum_j^{Ny} [Bl_{i,j} \bullet Sy_j(\eta) \bullet Sx_i(\psi)] + \psi[\xi_T(\eta) - \tan[\alpha_{twist}(\eta)]] + \xi_n(\eta) \quad 4.6$$

where  $\eta$  are non-dimensional spanwise coordinates. The entire wing in Cartesian coordinates is then represented as the following set of equations:

$$x = \psi \bullet C_{local}(\eta) + x_{LE}(\eta) \quad 4.7$$

$$y = \frac{b}{2} \bullet \eta \quad 4.8$$

$$Z_U = \xi_U(\psi, \eta) \bullet C_{local}(\eta) \quad 4.9$$

$$Z_L = \xi_L(\psi, \eta) \bullet C_{local}(\eta) \quad 4.10$$

where  $b$  is the length of span of the full wing,  $C_{local}(\eta)$  is the local chord distribution, and  $x_{LE}(\eta)$  is the leading edge coordinates distribution in the spanwise direction.

Different from the other parameterisation methods for wing representation, the CST method does not describe the wing in various control sections. It represents the entire wing surface by one set of equations. This can avoid the unsMOOTHNESS caused by interpolation between control sections, and can reduce the number of design variables. The order of the Bernstein polynomial that controls the geometry in the chordwise direction is denoted as ‘*BPOX*’, and the order of the Bernstein polynomial controlling the geometry in the spanwise direction is denoted as ‘*BPOY*’. Depending on the complexity of wing, the number of planform design parameters is flexible. In modern civil transport aircraft, the wing normally has a crank near the wing root. Therefore, piecewise functions with respect to crank are used to represent the distribution of planform parameters along the spanwise direction. Equations 4.11 and 4.12 show the Bernstein polynomial for a twist distribution function for inner and outer wings:

$$\alpha_{Twist,inner}(\eta_{inner}) = \sum_{i=0}^{N_{inner}} [A_{inneri} \cdot K_{i,n} \cdot \eta_{inner}^i \cdot (1 - \eta_{inner})^{n-i}] \quad 4.11$$

$$\alpha_{Twist,outter}(\eta_{outer}) = \sum_{i=0}^{N_{outer}} [A_{outteri} \cdot K_{i,n} \cdot \eta_{outer}^i \cdot (1 - \eta_{outer})^{n-i}] \quad 4.12$$

where

$$\eta_{inner} = y/b_{crank} \quad 4.13$$

$$\eta_{outer} = y/(b_{wing} - b_{crank}) \quad 4.14$$

In order to maintain the geometric continuity and at least 1<sup>st</sup> order geometric derivative continuity, the following constraints should be applied during optimisation:

$$A_{inner_{N_{inner}}} = A_{outer_0} \quad 4.15$$

$$\frac{A_{inner_{N_{inner}}} - A_{inner_{N_{inner}-1}}}{b_{crank}} = \frac{-A_{outer_0} + A_{outer_1}}{(b_{wing} - b_{crank})} \quad 4.16$$

Horizontal Tail Plane (HTP) and Vertical Tail Plane (VTP) are two relatively smaller surfaces located at the empennage of civil passenger transport aircraft. The geometries of VTP and HTP are slightly different, but can be defined in the same way since they are both ‘wing-type geometry’. The CST wing equations can be applied to represent HTP without alteration. Since the VTP is a symmetric geometry along the aircraft’s symmetric plane, there is no twist and dihedral installation position term in the VTP equations. The CST equations for VTP are then rewritten from Equations 4.17 to 4.21:

$$\xi(\psi, \eta) = C_{N2}^{N1}(\psi) \bullet \sum_i^{Nx} \sum_j^{Ny} [B_{i,j} \bullet S y_j(\psi) \bullet S x_i] + \psi[\xi_T(\eta)] \quad 4.17$$

$$x = \psi \bullet C_{local}(\eta) + x_{LE}(\eta) \quad 4.18$$

$$y_U = \xi(\psi, \eta) \bullet C_{local}(\eta) \quad 4.19$$

$$y_L = -\xi(\psi, \eta) \bullet C_{local}(\eta) \quad 4.20$$

$$z = \frac{b}{2} \bullet \eta \quad 4.21$$

At the first stage of the optimisation process, the baseline geometry needs to be parameterised. Therefore, it requires the parameterisation methods to have good capability to fit to the baseline geometry. In the previous chapter, the CST method was employed for representing a 2D aerofoil and it was concluded that CST has good fitting accuracy for this purpose when the higher order Bernstein polynomial is applied. Similarly, the performance of the CST method fitting for a 3D wing is examined in this work.

#### 4.1.2. Fitting accuracy of the standard CST method for a wing

To parameterise an existing wing using the CST method, the process of inverse fitting could be divided into two steps. The first step is to extract the wing planform parameters at the local section, such as twist, leading edge coordinates, leading edge height, local

chord length and trailing edge thickness, from the wing surface point dataset. Any curve fitting technique, such as Bezier curve, polynomial, B-spline *etc.*, could be applied to obtain the planform parameter distribution function which will be employed in aerodynamic optimisation. The second step is to determine the parameters of class and shape equations by minimising the mean of the ‘least-square error’, where the error is measured by the  $L_2$  norm of the difference between the original geometry and the CST approximation. The results from the test case using a pure F6 wing will be presented here. The F6 wing is a realistic wing and has been used for CFD validation for many years (Brodersen 2002; Brodersen *et al.* 2008; Sclafani *et al.* 2008). Various approximations, obtained by different order *BPOX* and *BPOY*, of the CST wing will be shown. Afterwards, the CFD calculation will be carried out on the original geometry and approximated geometries. The effect of *BPOX* and *BPOY* on geometric and aerodynamic change will be examined.

High order polynomials are used to fit the installation height, trailing edge thickness and tangential value of twist angle distribution functions. Figure 4.2 to 4.6 show the results of planform parameter distribution functions:

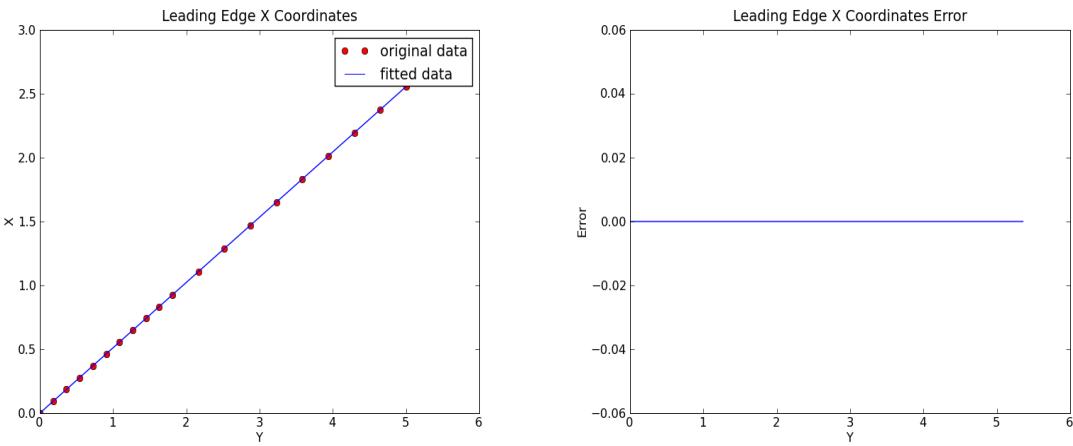


Figure 4.2 Leading edge x coordinates distribution and error

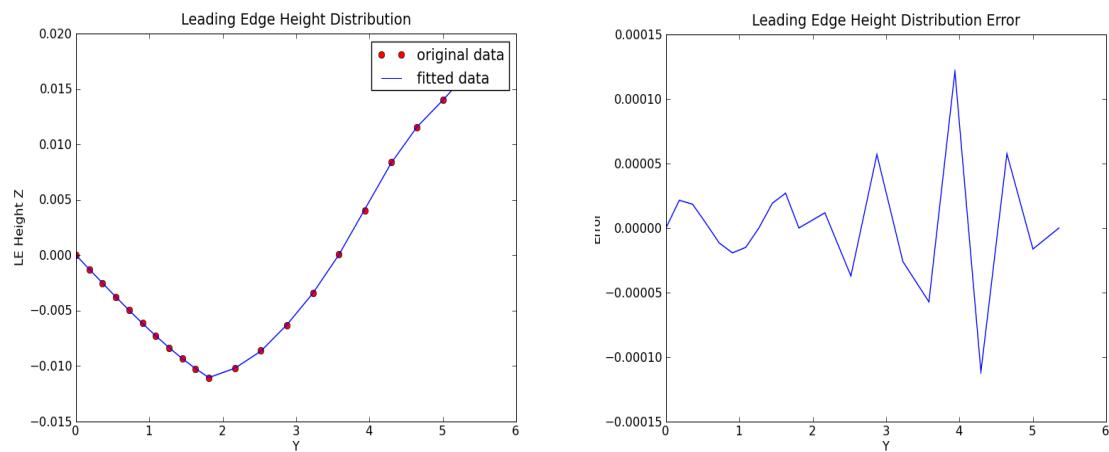


Figure 4.3 Leading edge height distribution and error

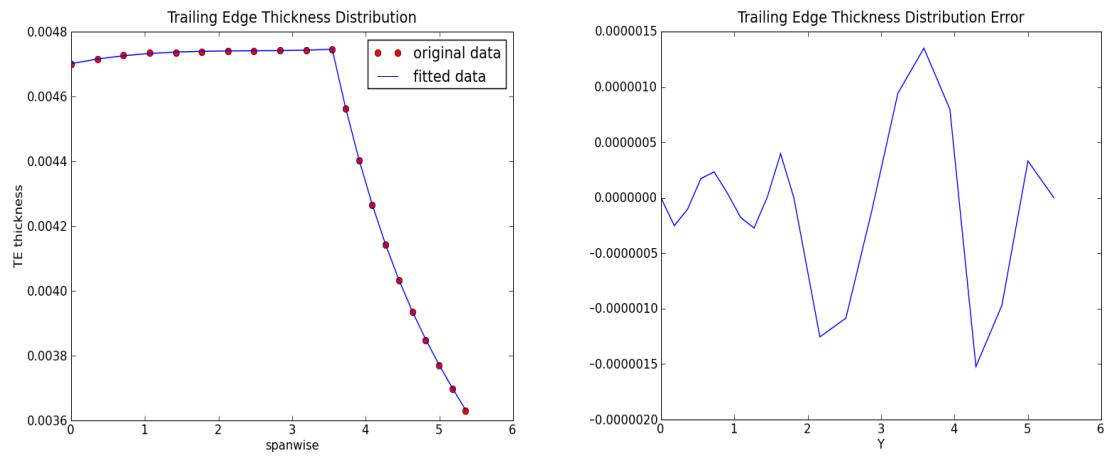


Figure 4.4 Non-dimensional trailing edge thickness distributions leading edge and error

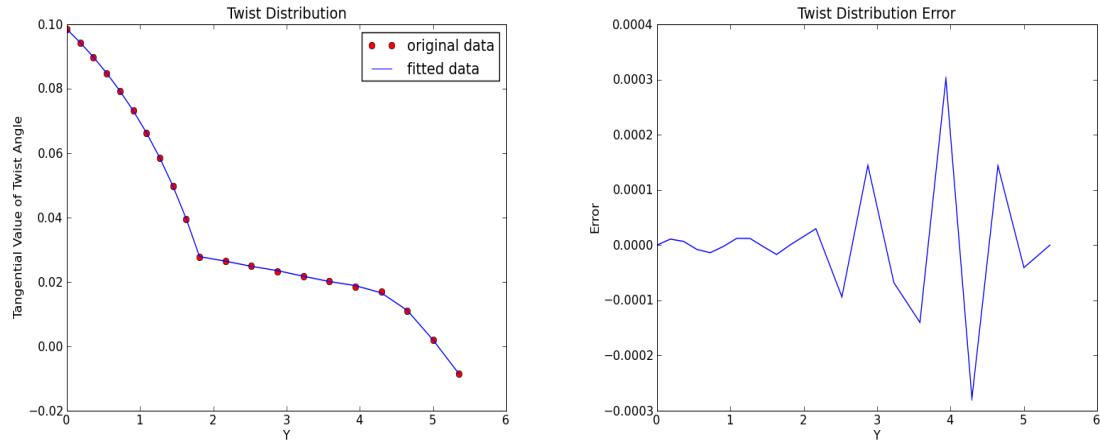


Figure 4.5 Tangential value of twist angle distributions and error

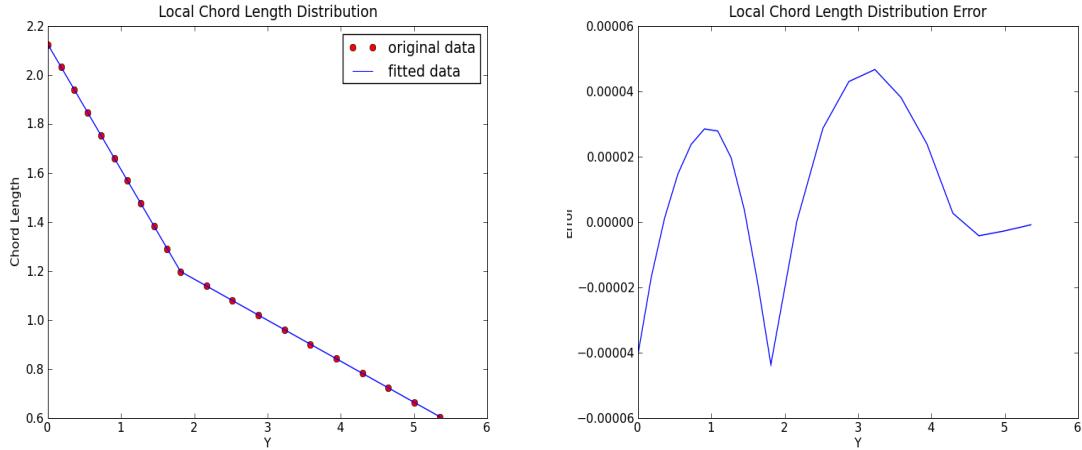


Figure 4.6 Local chord distributions and error

As the above figures show, most planform parameter distribution functions can be addressed accurately. The leading edge  $x$  coordinates can be fit exactly. The trailing edge thickness and local chord can be fitted within error lower than  $4 \times 10^{-5}$ . The maximum error is occurred at tangential value of twist angle with  $3 \times 10^{-4}$  which is much lower than the tolerance. Therefore, this fitted distribution could be kept. The next step is to find the class and shape parameters to fit the entire wing surface. Because this wing has standard aerofoil with round nose and sharp trailing edge, the class parameters are fixed to  $N_1 = 0.5$  and  $N_2 = 1.0$ . As with the demonstration of the results of the CST method for the 2D aerofoil test case, different orders of the Bernstein polynomial could provide various levels of accuracy, and using the different leading edge radius parameters on upper and lower surfaces could increase the accuracy of inverse fitting.

As presented above, two Bernstein polynomials control the entire wing surface. The effects of different Bernstein polynomials order in the chord-wise direction  $BPOX$  and in the span-wise direction  $BPOY$  on fitting accuracy are studied. Table Table 4.1 shows the maximum error of inverse fitting against  $BPOX$  and  $BPOY$ . In Ceze *et al.*'s (2009) paper, the effect of the Bernstein polynomial order of the CST method on the numerical uniqueness was studied, and it is undesirable to use a very high order Bernstein polynomials. Therefore, the range of  $BPOX$  and  $BPOY$  is limited between 6 and 12. The number of design variables of the CST wing is  $2 \times (BPOX + 1) \times (BPOY + 1)$ .

Table 4.1 Total  $L_2$  norm error of inverse fitting ( $\times 10^{-2}$ m)

BPOY \ BPOX	6	7	8	9	10	11	12
6	5.2816	4.0285	3.9978	3.4832	3.2066	3.1271	2.8129
7	5.0620	3.7356	3.7025	3.1396	2.8296	2.7392	2.3740
8	5.0387	3.7038	3.6704	3.1017	2.7875	2.6957	2.3237
9	5.0046	3.6574	3.6235	3.0461	2.7255	2.6315	2.2489
10	4.9741	3.6154	3.5812	2.9955	2.6688	2.5727	2.1798
11	4.9605	3.5967	3.5623	2.9729	2.6434	2.5464	2.1487
12	4.9517	3.5845	3.5500	2.9582	2.6268	2.5292	2.1282

Table 4.1 demonstrates that increasing  $BPOX$  and  $BPOY$  can apparently decrease the error of inverse fitting. Therefore, using higher order  $BPOX$  and  $BPOY$  can effectively improve the geometric fitting and expand the design space. Furthermore, the total  $L_2$  error is more sensitive to  $BPOY$  since very high  $BPOY$  can result in a relatively low  $L_2$  error, even if  $BPOX$  is kept at low order. Hence,  $BPOY$  should use a higher order for representing the wing. Some test case results of the error distribution contour are shown in Figure 4.7 to 4.10.

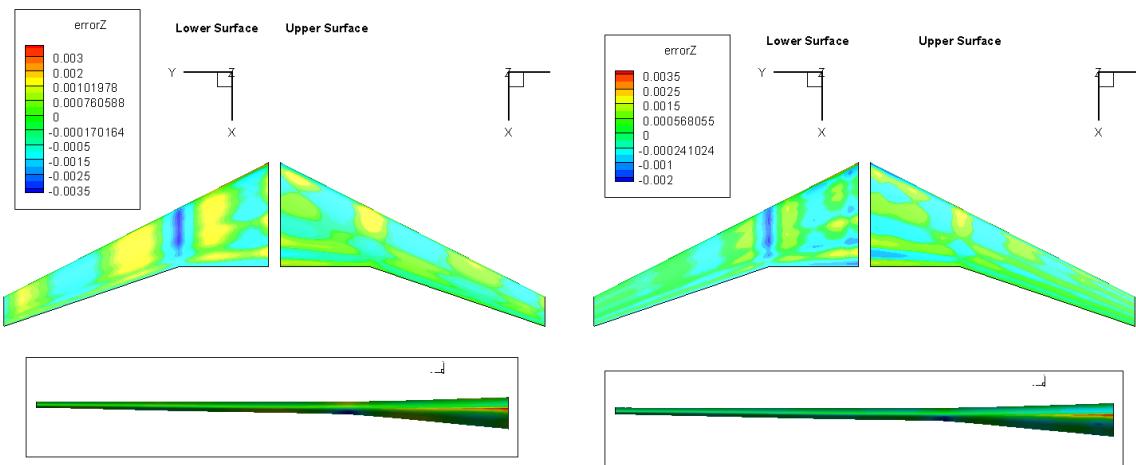


Figure 4.7 The error contour of wing inverse fitting with  $BPOX$  6- $BPOY$  6 (left figure in metre) and  $BPOX$  6- $BPOY$  10 (right figure in metre)

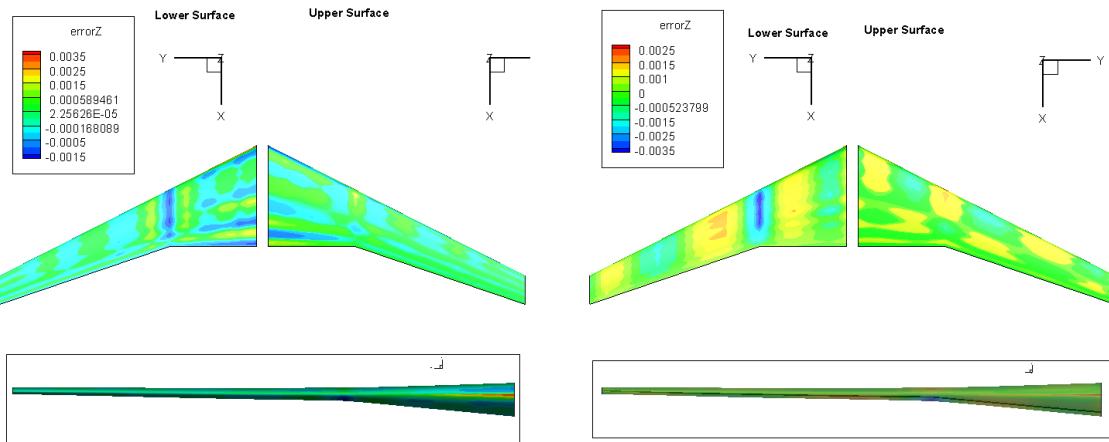


Figure 4.8 The error contour of wing inverse fitting with *BPOX 6-BPOY 12* (left figure in metre) and *BPOX 10-BPOY 6* (right figure in metre)

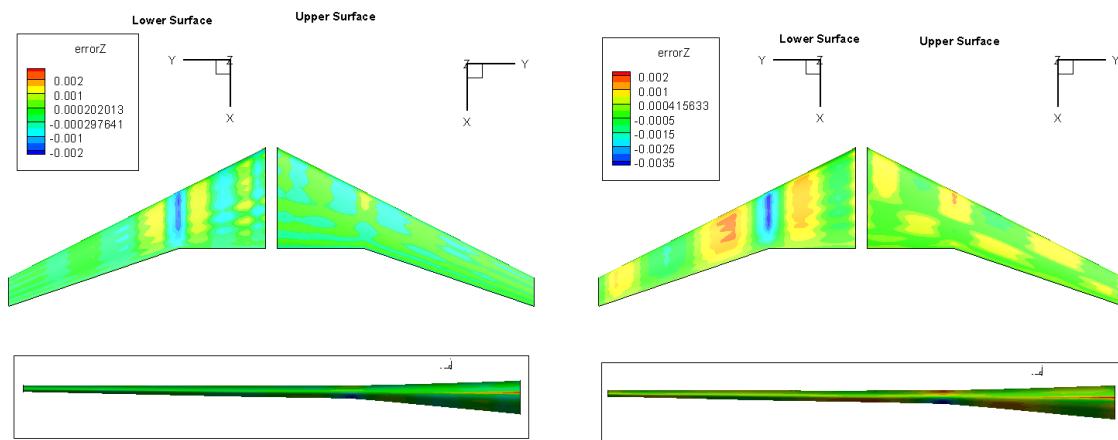


Figure 4.9 The error contour of wing inverse fitting with *BPOX 10-BPOY 10* (left figure in metre) and *BPOX 12-BPOY 6* (right figure in metre)

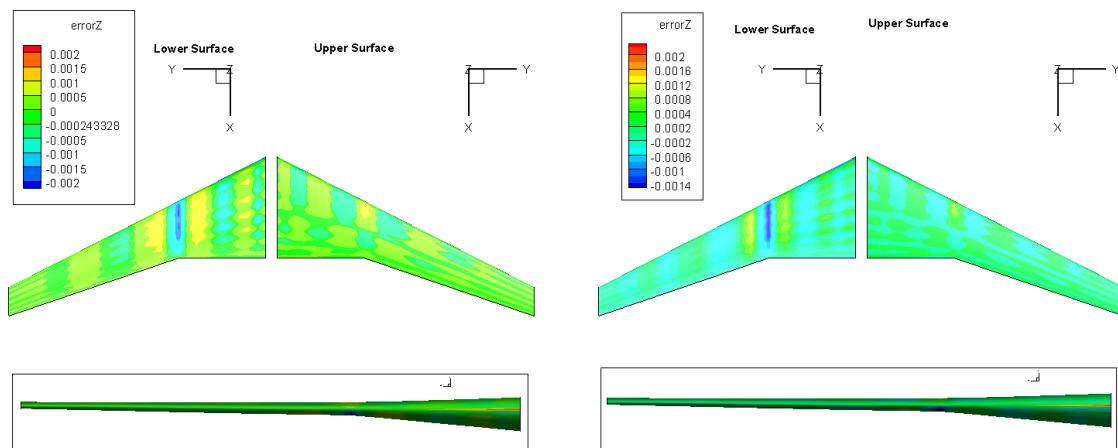


Figure 4.10 The error contour of wing inverse fitting with *BPOX 12-BPOY 10* (left figure in metre) and *BPOX 12-BPOY 12* (right figure in metre)

From the above error contours, it is apparent that the maximum approximated error decreases for increasing order of *BPOX* or *BPOY*. The largest error is at the leading edge of the upper surface near the wing root and the lower side of the wing around the crank area. The reason is that there is a large curvature change in the leading edge of the upper surface, and inverse approximation is difficult to address this curvature change. In the crank area, the original wing shape has curvature discontinuity and the CST approximation will automatically smooth the shape in this area. Moreover, a slight oscillation on the error contour is observed, which means that the approximated wing is oscillating around the original wing. More oscillation is presented when a higher order polynomial is employed. This is because the Bernstein polynomial lacks an orthogonal basis within that interval which, in least-squares approximation problems, complicates the assembly of convergent sequences of approximation terms (Ceze *et al.* 2009; Farouki 2000). Although the approximated geometry is around the original one, it does not mean that the approximated geometry is not a smooth shape. This can be seen in the comparison study on the CFD results.

Further study of the effect of the order of the Bernstein polynomial is made by comparing the CFD results on the original geometry and approximated geometries. The flow condition is Mach=0.75,  $Re=5\times 10^6$ , angle of attack 0.0 and the turbulent model uses Spalart-Allmaras according to Sclafani *et al.* (2008). The meshes of all geometries are using a hexahedral dominant hybrid mesh which is generated by Solar, developed by the Aircraft Research Association, BAE Systems and Airbus (Leatham *et al.* 2000) and the total number of mesh points is around 3 million. Figure 4.11 shows the hybrid mesh of the F6 wing for the CFD study. The flow solver uses TAU developed by DLR (Gerhold *et al.* 1997), which is a hexahedral dominant multi-grid solver. Due to the CFD calculation time, not all of the approximated geometries shown in Table 4.2 are included in the CFD comparison study.

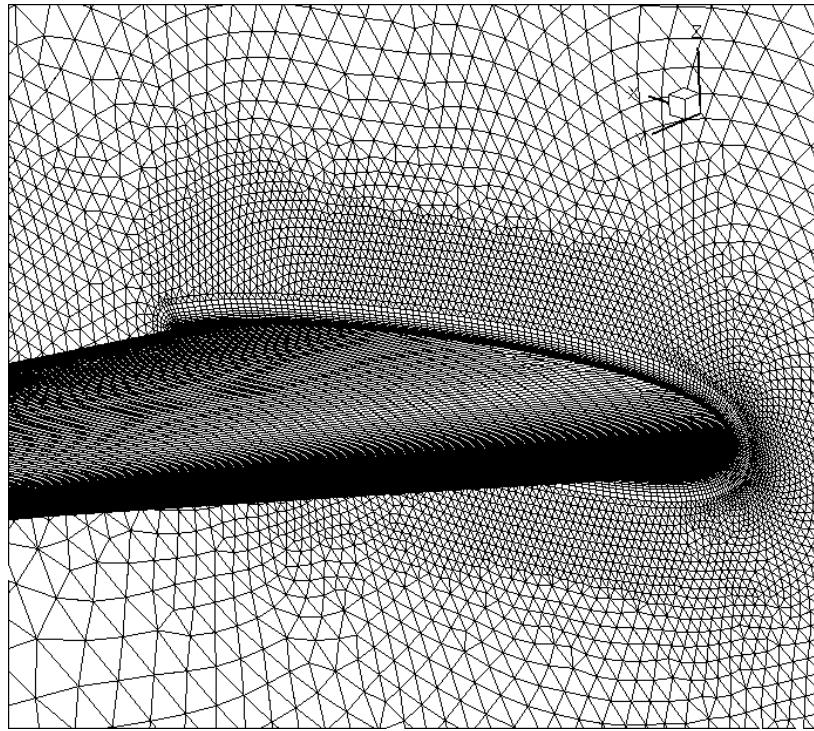


Figure 4.11 The hybrid mesh of F6 wing for CFD study

Table 4.2 lists the lift coefficient of the original and approximated geometry, and Table 4.3 lists the drag coefficients of the original and approximated geometry.

Table 4.2 Lift coefficient of original geometry and approximated geometry

Original	0.56056		
BPOY	6	10	12
BPOX			
6	0.54962	0.54980	0.54978
10	0.55984	0.56002	
12	0.55898	0.55912	0.56015

Table 4.3 Drag coefficient of original geometry and approximated geometry

Original	0.02383		
BPOY	6	10	12
BPOX			
6	0.02371	0.02372	0.02373
10	0.02419	0.02420	
12	0.02417	0.02419	0.02421

The above results clearly show that the approximated models with lower order of Bernstein polynomial will provide high errors on lift coefficients. The approximated model with BPOX 6 will have error about 2 lift counts. By fixing BPOX 6 and increasing BPOY, the error is only reduced slightly. By fixing BPOY 6 and increasing BPOX, the lift coefficient is approaching to original model. By increasing both of *BPOX* and *BPOY* to higher than ten, the lift coefficient error will be reduced to 0.05 lift counts. The drag coefficient of approximated models is a bit more complicated. The approximated model with BPOX has error about 2 drag counts. However, the approximated model with BPOX 10 and 12 has error about 3 drag counts. This is because the drag coefficient is more sensitive on pressure distribution. Although drag coefficients cannot show convergence with increasing Bernstein order, it still can notice that the BPOY does not affect the drag coefficient, and the BPOX has more impact on drag coefficient.

The pressure distributions of the original geometry and the approximated geometries are studied for further understanding of the characteristics of approximated models. Eight sections along the spanwise direction are taken to carry out this study. Figure 4.10 shows the sections index and position. Figure 4.13 to 4.20 show the pressure distribution obtained from the original F6 wing and parametric approximated wing models and their wing profile on each section.

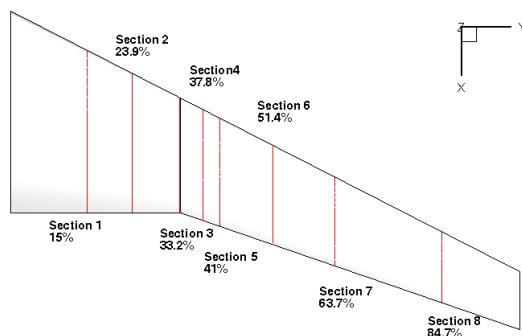


Figure 4.12 The sections index and position on the wing

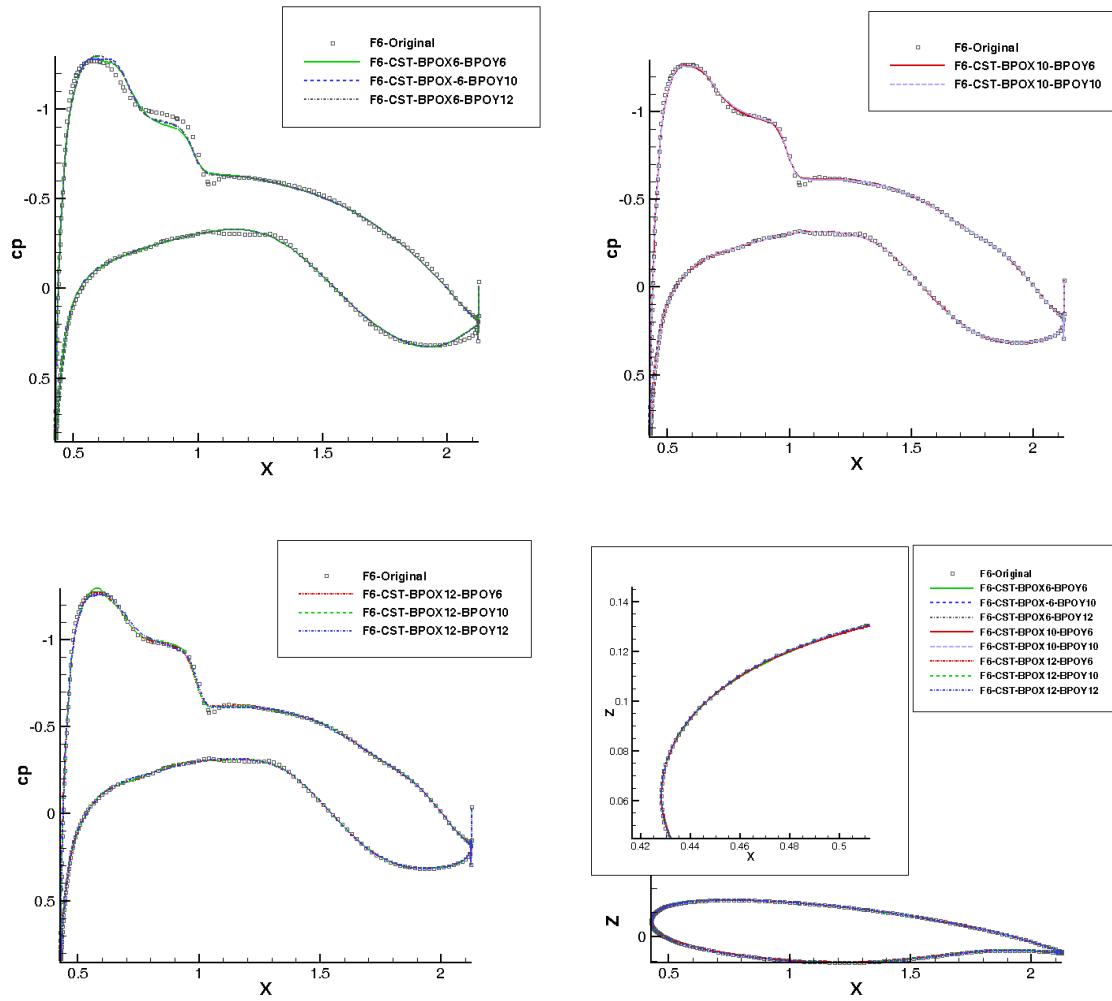
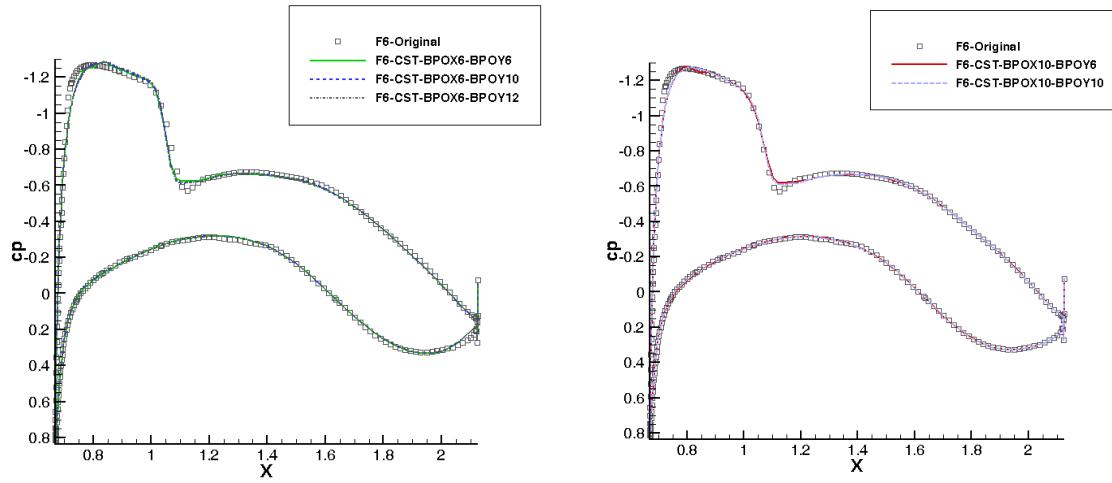


Figure 4.13 Comparisons of pressure distribution and wing shape on section 1



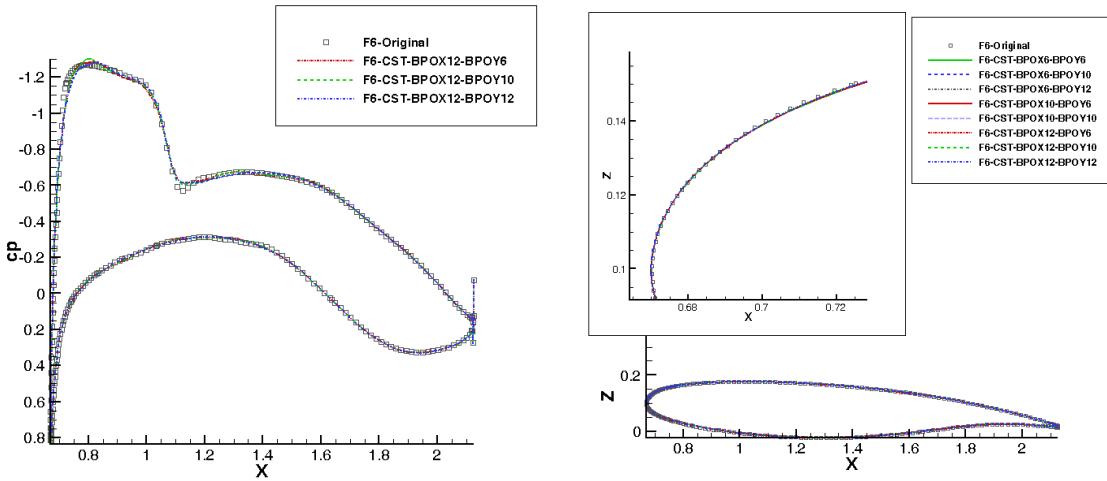


Figure 4.14 Comparisons of pressure distribution and wing shape on section 2

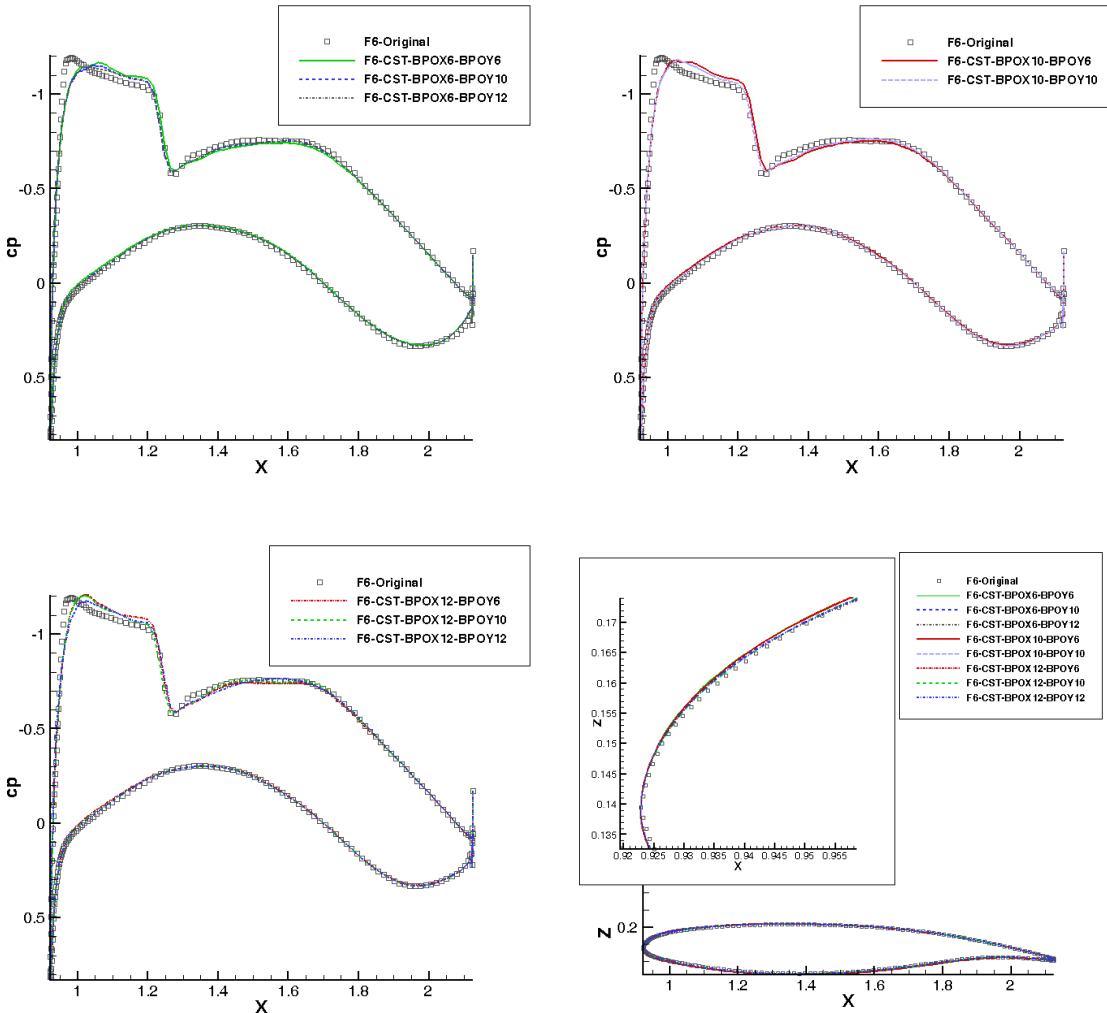


Figure 4.15 Comparisons of pressure distribution and wing shape on section 3

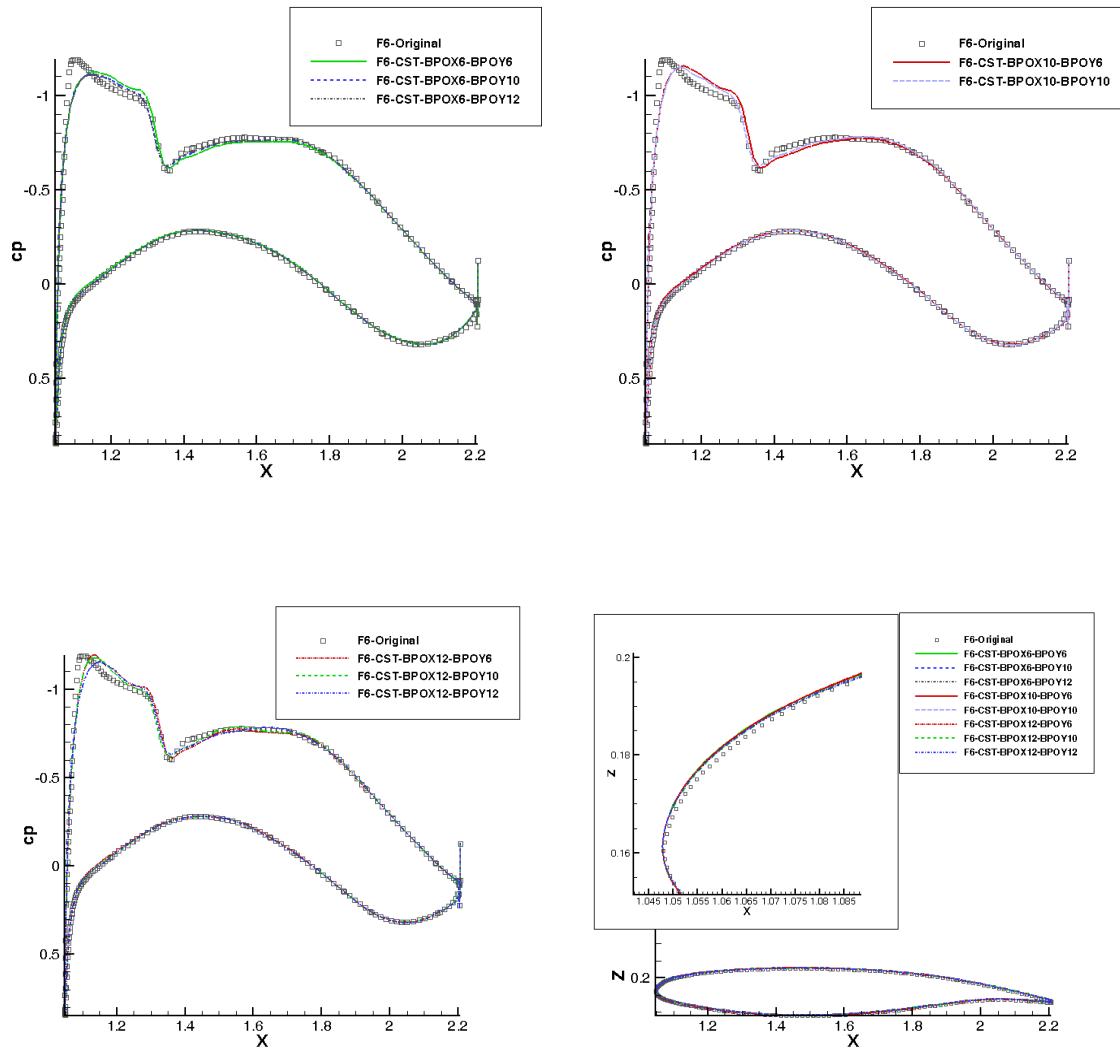
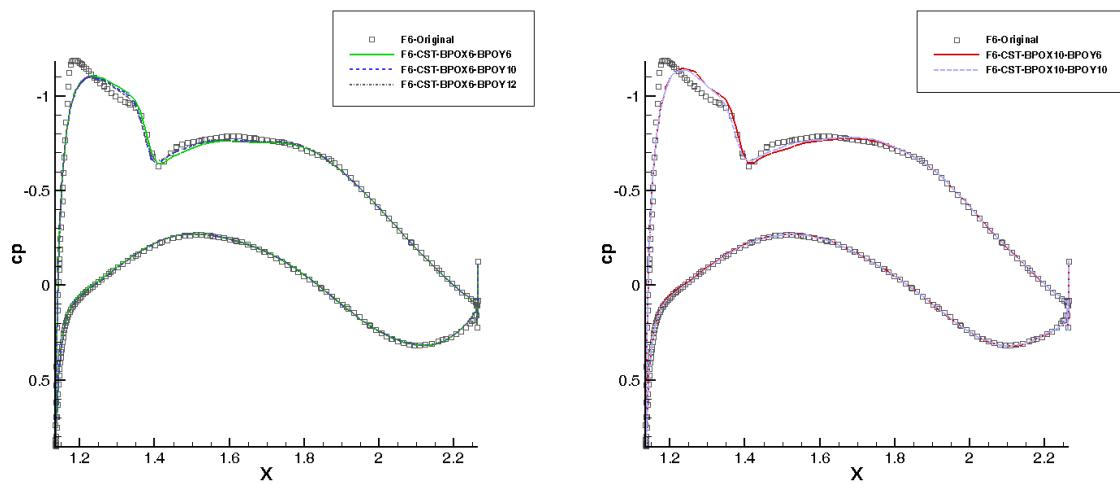


Figure 4.16 Comparisons of pressure distribution and wing shape on section 4



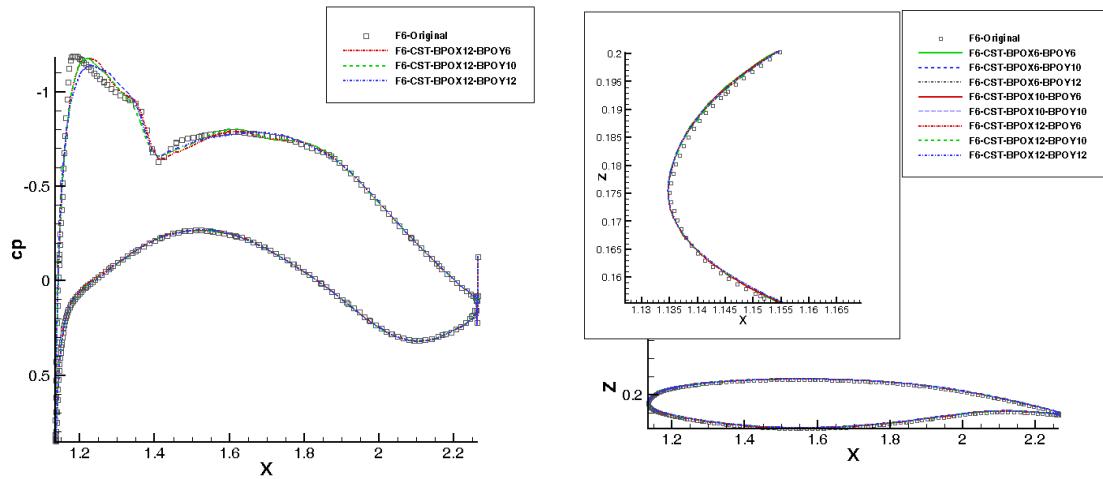


Figure 4.17 Comparisons of pressure distribution and wing shape on section 5

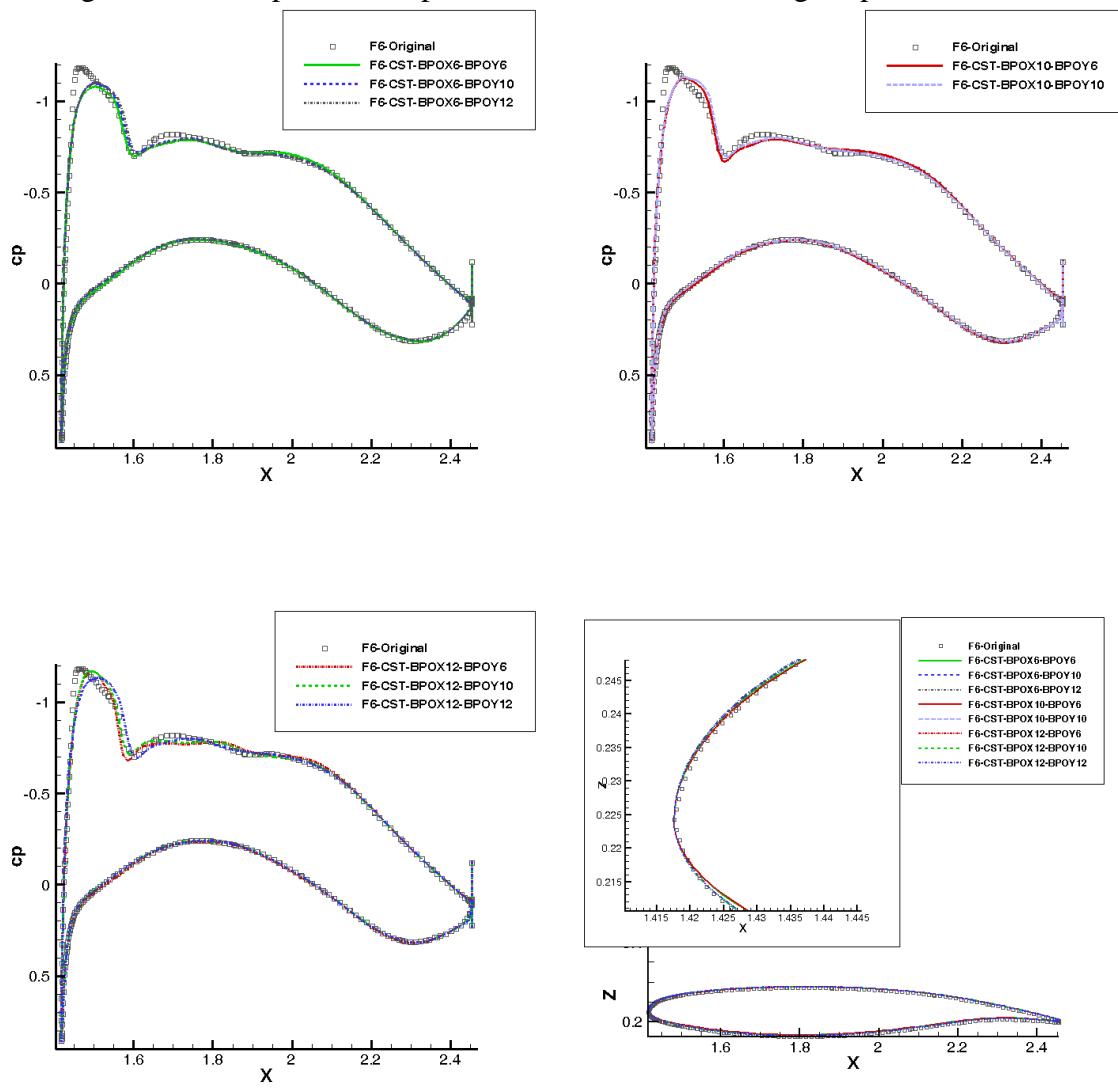


Figure 4.18 Comparisons of pressure distribution and wing shape on section 6

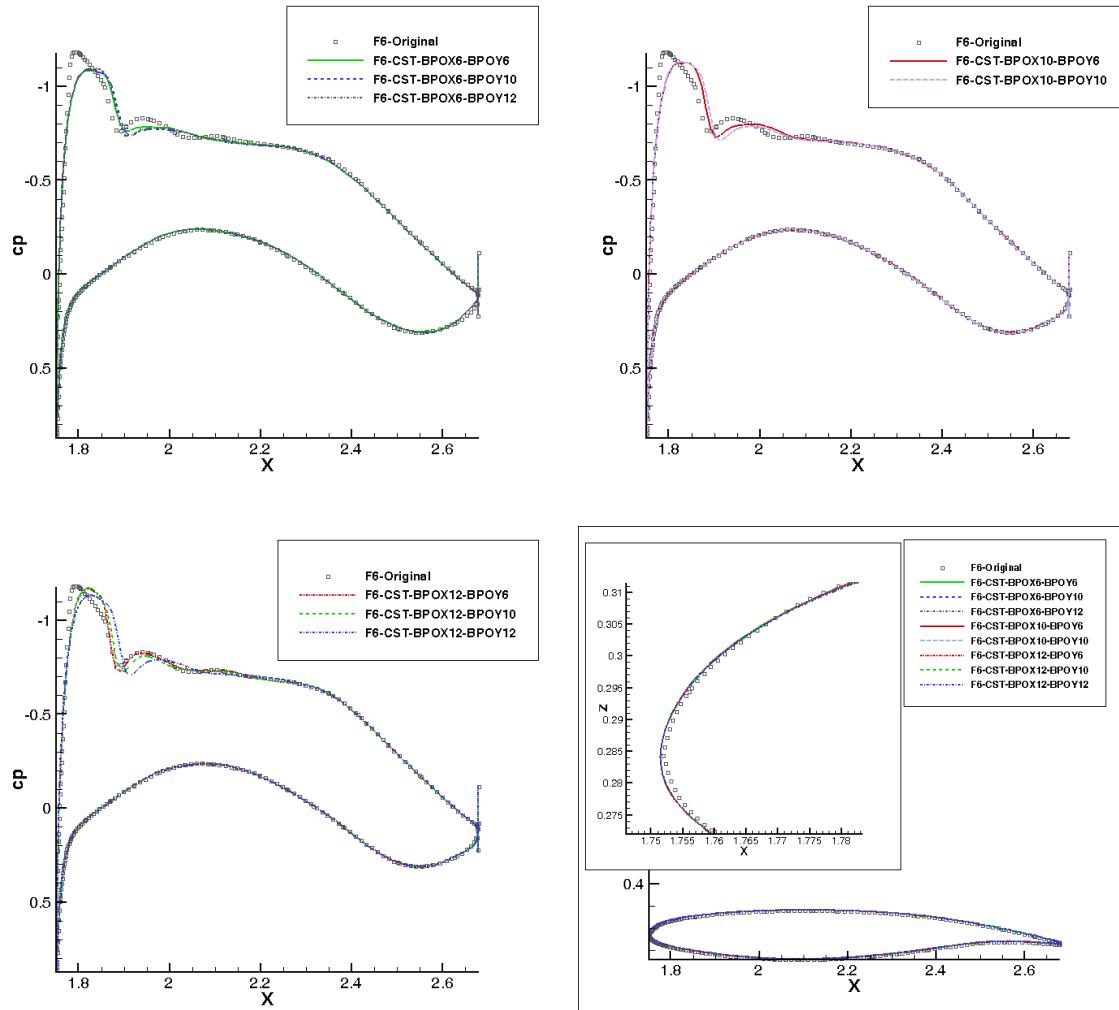
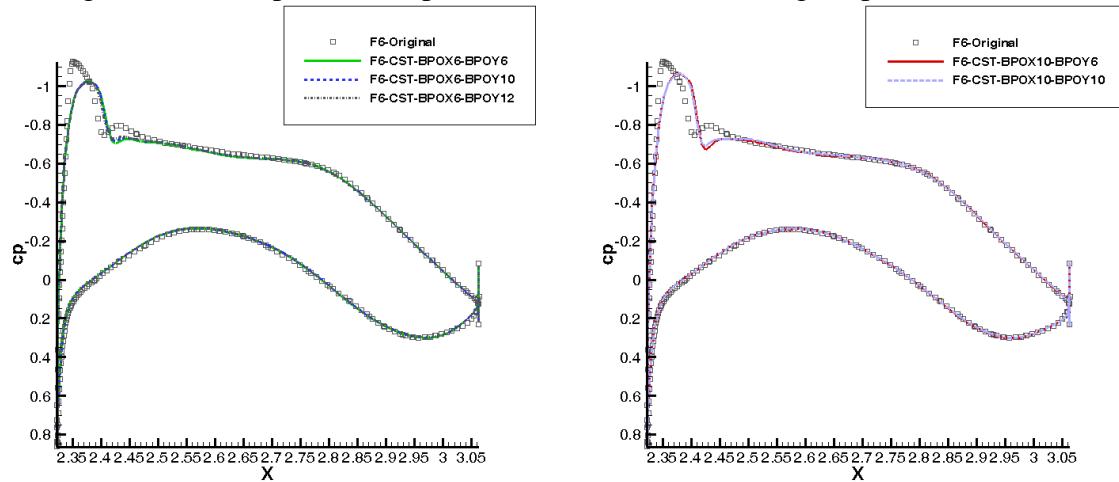


Figure 4.19 Comparisons of pressure distribution and wing shape on section 7



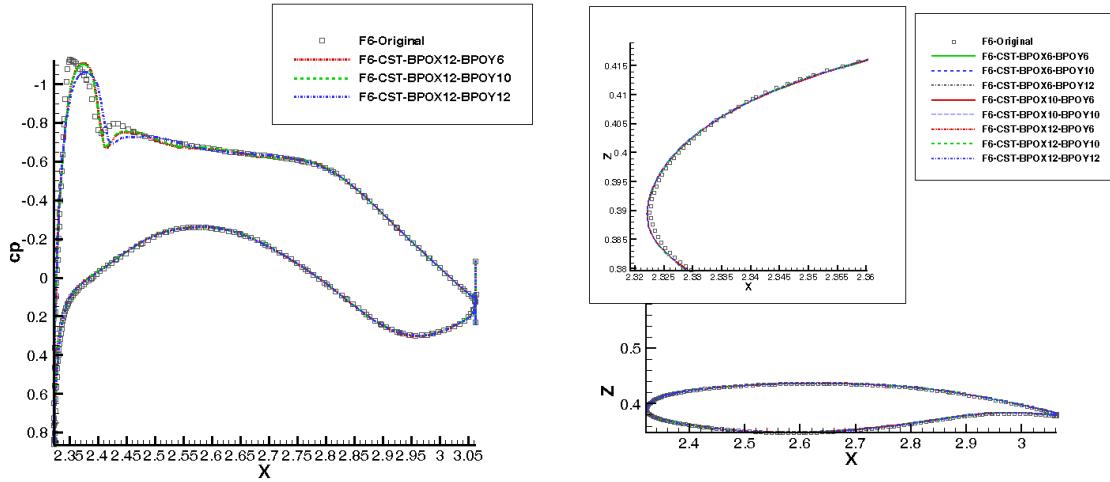


Figure 4.20 Comparisons of pressure distribution and wing shape on section 8

The above figures illustrate that the pressure distribution on the lower side of the approximated models with any *BPOX* and *BPOY* can match well with the original model at every section, even in the crank area and leading edge near wing tip where there is a relatively higher geometric fitting error. This is mainly because the flow on the lower side is not very sensitive, so the lower surface of the wing can be well fitted in both geometric and aerodynamic cases using any *BPOX* and *BPOY*.

However, fitting the upper side is a crucial issue, and the figures show that the pressure distributions on the upper surface are not matched as well as that on the lower surface, because the flow is highly sensitive in the leading edge area. The error of pressure on the upper surface is high in the area between leading edge and the 20% chord length. Before section 4, the pressure distributions on the upper surface can still be matched with reasonable error. However, after section 5 the error of pressure distributions becomes larger. It can be noticed from the comparison of wing profile figures that the surface geometric fitting error, especially in the leading edge area, is getting larger along the spanwise direction towards the wing tip.

From each section, the pressure distribution error could be decreased when using CST with higher  $BPOX$ , especially when  $BPOX$  is higher than 10. However, only increasing  $BPOY$  with fixed  $BPOX$  in a lower order cannot improve the pressure distribution error.  $BPOX$  obviously dominates this error, consequently, when using the CST method for the wing, higher order  $BPOX$  should be considered.

However, even if the high  $BPOX$  12 is applied and the value of peak pressure in the leading edge could be approached, the position of peak pressure of the approximated model does not match with the original model. The main reason is that the current CST parameterisation is hard to fit accurately to the leading edge, and another reason is a slight error occurring in planform parameter distribution fitting should be considered as well. Further study is needed to improve the CST parameterisation fitting in the leading edge.

A similar conclusion can be obtained in HTP and VTP, which will not be shown here. Figures 4.19 and 4.20 show the HTP and VTP models which are generated using the CST methods.

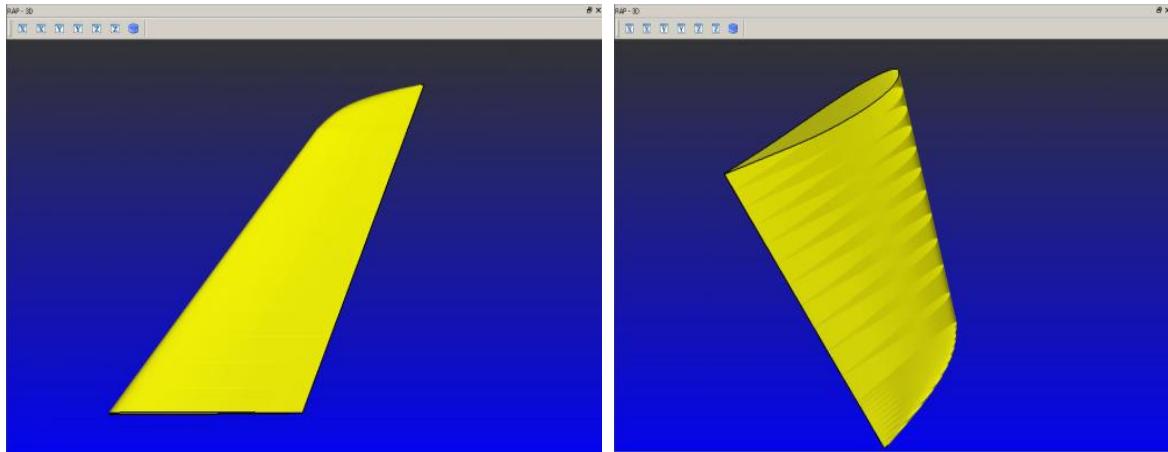


Figure 4.21 The HTP model using the CST methods

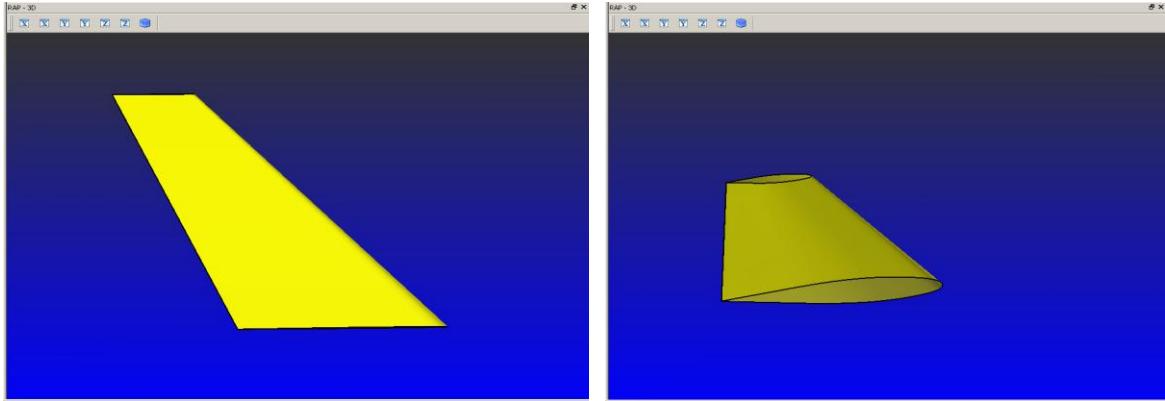


Figure 4.22 The VTP model using the CST methods

#### 4.1.3. RCST method for wing type geometries

From the test of performance of fitting accuracy of the standard CST method, it can be clearly seen that even if high order Bernstein polynomials are applied in both streamwise and spanwise directions in the standard CST, the value and position of peak pressure in leading edge area of the approximated model do not match with the original model. Therefore, modification of the CST method is desired.

In Chapter 3, RCST with rational Bernstein polynomial function has been proposed for the two-dimensional aerofoil. The test results showed that RCST is able to significantly improve the fitting accuracy. Therefore, RCST is extended here to represent a three-dimensional wing. There is no change in the basic definition of the CST wing. The standard Bernstein polynomials are replaced by the rational Bernstein polynomials in Equations 4.3 and 4.4 RCST for a non-dimensional wing surface is then written as:

$$\xi_U(\psi, \eta) = C_{N2}^{N1}(\psi) \cdot \sum_i^{Nx} \sum_j^{Ny} \left[ \frac{Bu_{i,j} \cdot Wu_{i,j} \cdot Sy_j(\eta) \cdot Sx_i(\psi)}{Wu_{i,j} \cdot Sy_j(\eta) \cdot Sx_i(\psi)} \right] + \psi[\xi_T(\eta) - \tan(\Delta\alpha(\eta))] + \xi_{n(\eta)} \quad 4.22$$

$$\xi_L(\psi, \eta) = C_{N2}^{N1}(\psi) \cdot \sum_i^{Nx} \sum_j^{Ny} \left[ \frac{Bl_{i,j} \cdot Wl_{i,j} \cdot Sy_j(\eta) \cdot Sx_i(\psi)}{Wl_{i,j} \cdot Sy_j(\eta) \cdot Sx_i(\psi)} \right] + \psi[\xi_T(\eta) - \tan(\Delta\alpha(\eta))] + \xi_{n(\eta)} \quad 4.23$$

where  $Wu_{i,j}$  and  $Wl_{i,j}$  are the weights for upper and lower surfaces respectively. The non-dimensional wing surface coordinate function is then applied to dimensionalise the wing surface with Equations 4.7 to 4.10. The wing surface in Cartesian coordinates is then established.

#### 4.1.4. Fitting accuracy of the RCST for a wing

The performance of the RCST is examined also by inverse fitting an F6 wing. *BPOX* 6 for streamwise and *BPOY* 6 for spanwise are selected to try to perform inverse fitting. The least-square error  $L_2$  norm is employed as the target objective value. The coefficients  $Bu_{i,j}$ ,  $Bl_{i,j}$ ,  $Wu_{i,j}$  and  $Wl_{i,j}$  are used as design variables to be determined by optimiser. For RCST with *BPOX* 6 and *BPOY* 6, the total number of design variables is 196.

The  $L_2$  norm is the error of the RCST with *BPOX* 6 and *BPOY* 6 which is  $7.1389 \times 10^{-3}$ . The fitting error of the standard CST with *BPOX* 6 and *BPOY* 6 is  $5.2816 \times 10^{-2}$ . Therefore, RCST is dramatically reduced by the  $L_2$  norm error. The error distribution contour is shown in Figure 4.23.

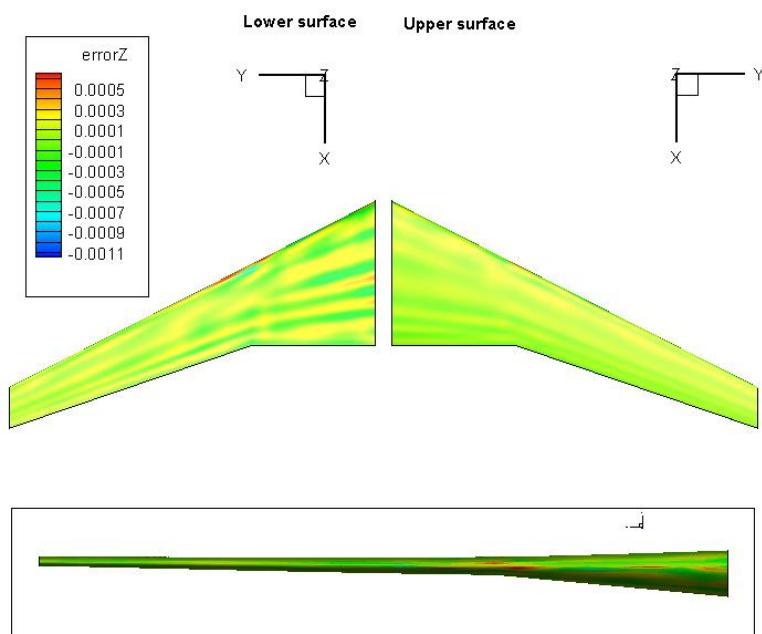


Figure 4.23 The error contour of wing inverse fitting with RCST *BPOX* 6-*BPOY* 6 (in metre)

The error distribution contour shows that the maximum occurs in the leading edge around crank area. However, the value of maximum error is around  $1.1 \times 10^{-3}$ . This value is much lower than the standard CST with *BPOX 6-BPOY 6* where this value was around  $3.5 \times 10^{-3}$ , and it is even lower than the standard CST with *BPOX 12-BPOY 12*.

A CFD study is performed for RCST as well. The flow condition is kept the same as with the previous test case, Mach=0.75, Re= $5 \times 10^6$ , angle of attack 0.0. The lift and drag coefficients of original geometry and approximated geometry are listed in Table 4.4.

Table 4.4 Lift coefficient of original geometry and RCST approximated geometry

	Lift	Drag
Original	0.56056	0.02383
RCST with BPOX6-BPOY6	0.56065	0.02392

Table 4.4 shows that both the lift coefficient error and the drag coefficient error are reduced to within 1 drag count using RCST with *BPOX 6-BPOY 6*. This demonstrated that RCST with fewer design variables is able to accurately represent a wing shape which satisfies the aerodynamic requirement.

Furthermore, the pressure distributions of the original geometry and the approximated geometries are plotted to examine the details. Nine sections evenly distributed along spanwise are taken to carry out this study. Figure 4.24 to 4.32 show the pressure distribution obtained from the original F6 wing and RCST approximated wing models and their wing profile on each section.

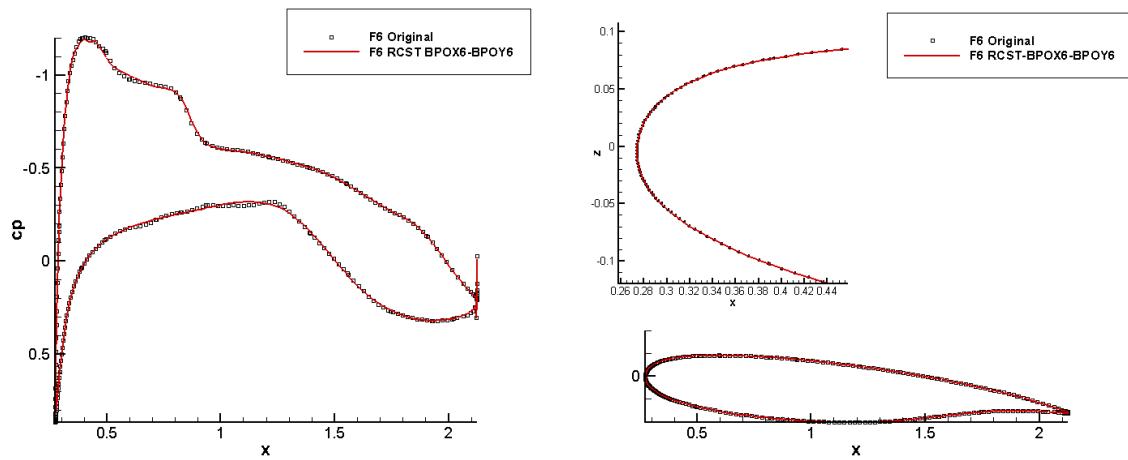


Figure 4.24 Comparisons of pressure distribution and wing shape at 10% span

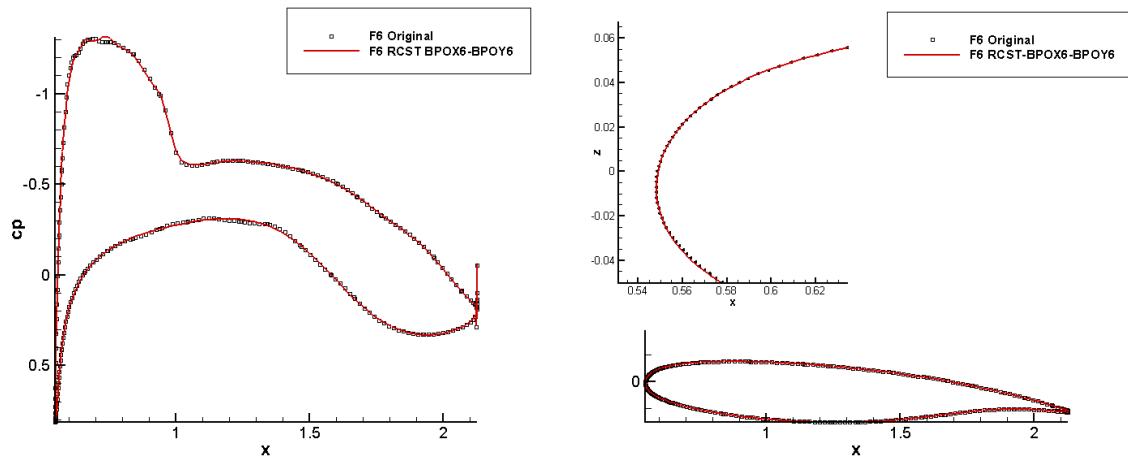


Figure 4.25 Comparisons of pressure distribution and wing shape at 20% span

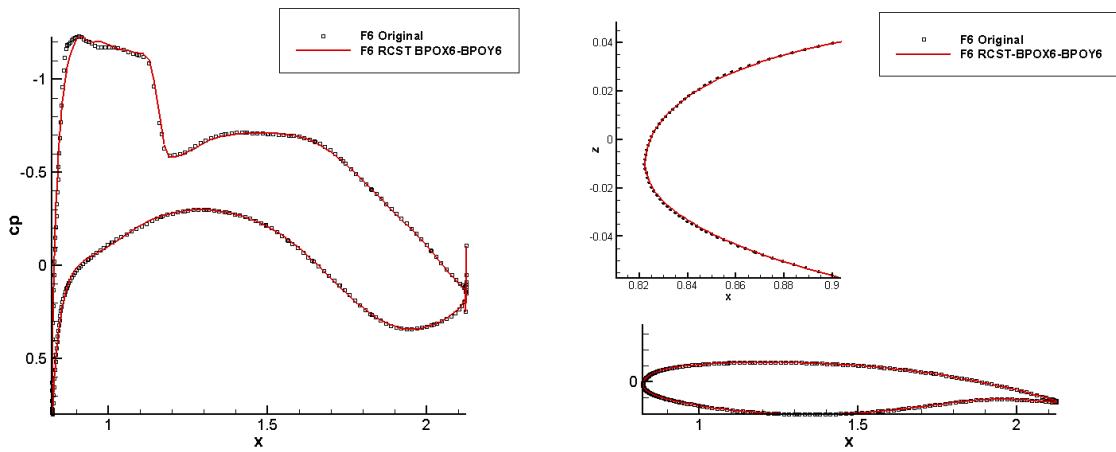


Figure 4.26 Comparisons of pressure distribution and wing shape at 30% span

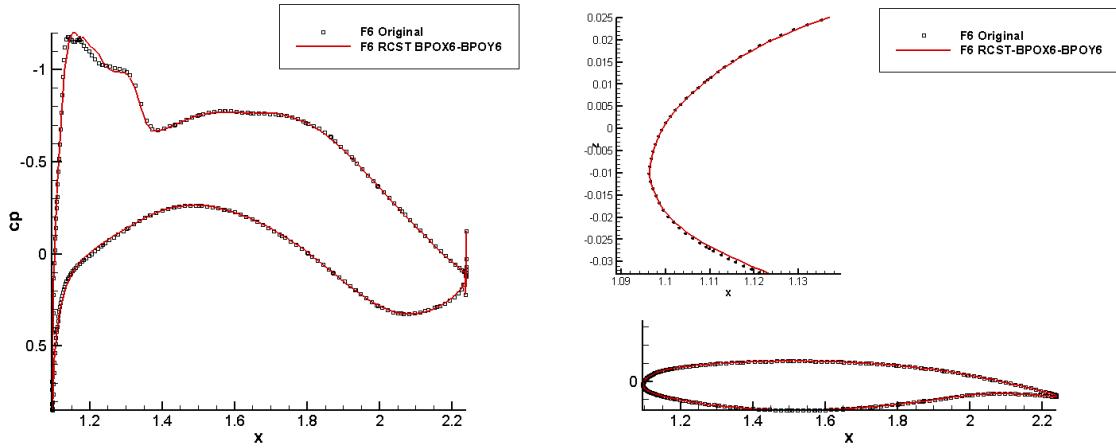


Figure 4.27 Comparisons of pressure distribution and wing shape at 40% span

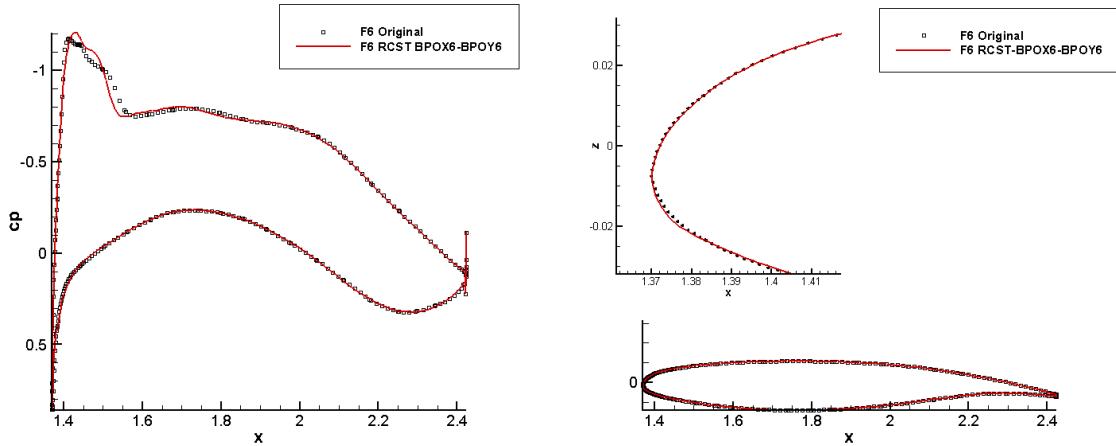


Figure 4.28 Comparisons of pressure distribution and wing shape at 50% span

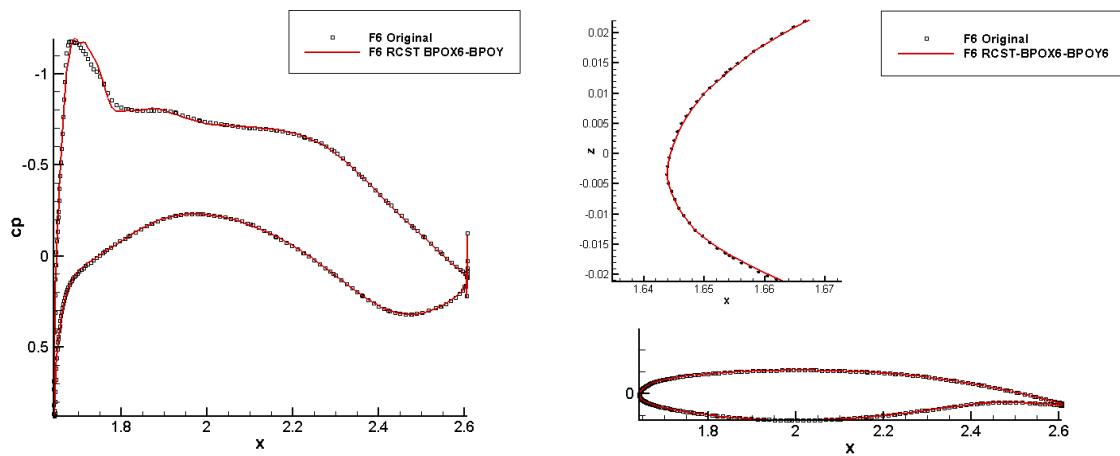


Figure 4.29 Comparisons of pressure distribution and wing shape at 60% span

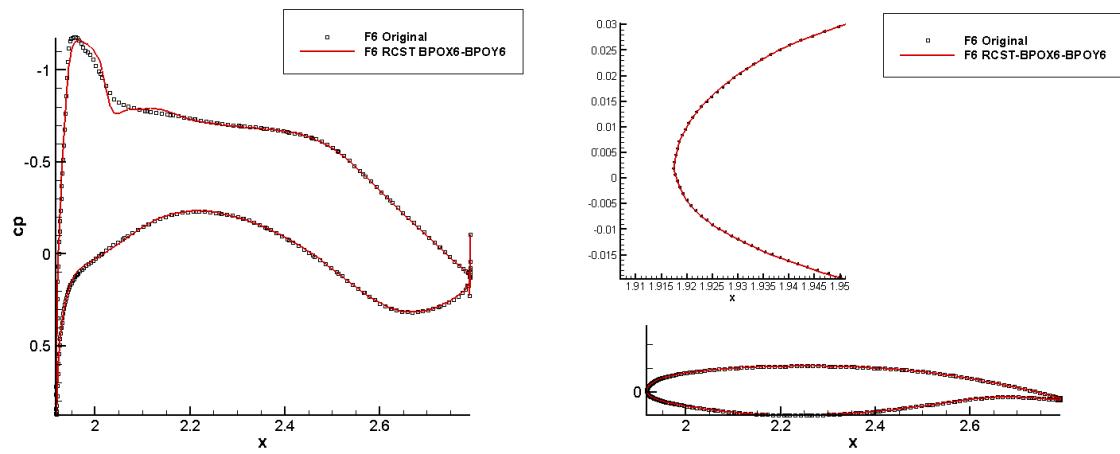


Figure 4.30 Comparisons of pressure distribution and wing shape at 70% span

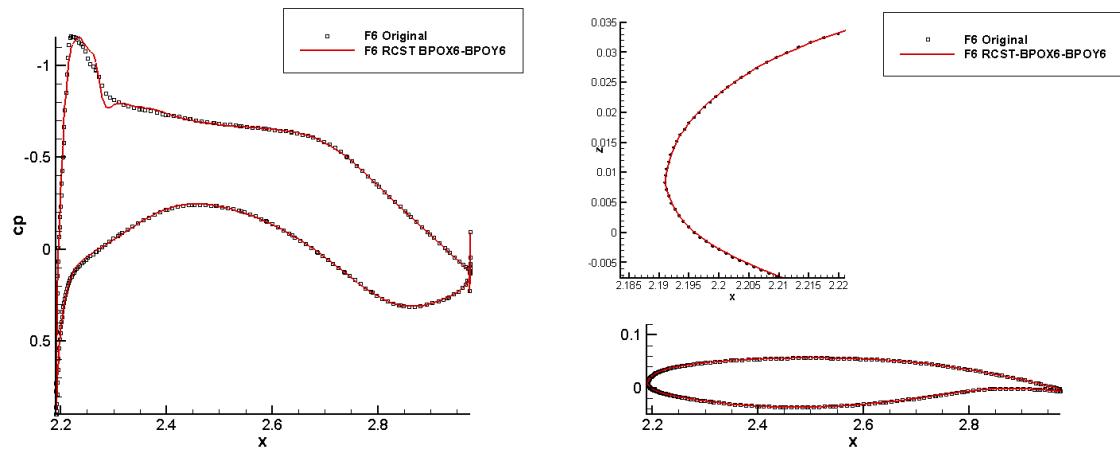


Figure 4.31 Comparisons of pressure distribution and wing shape at 80% span

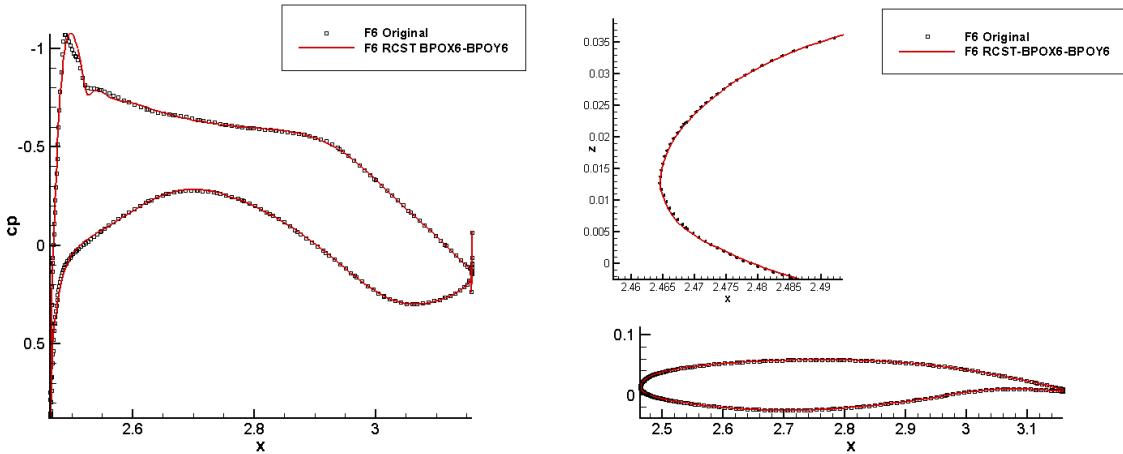


Figure 4.32 Comparisons of pressure distribution and wing shape at 90% span

The above figures illustrate that the pressure distribution on the lower side of approximated model using RCST matches very well. This is because the flow is less sensitive to geometric variation on the lower surface and the improvement of fitting accuracy with the RCST method.

Impressive improvement could be found on the upper surface. The figures show that significantly better pressure distribution on the upper surface are obtained by the RCST method. At 10% and 20% span sections, the pressure distributions are almost the same for the original and RCST approximated model. At 30% and 40% span sections, there is a slight error in the pressure suction peak. However, the value of the pressure suction peak is addressed correctly. The section profile figures show that the fitting leading edge area is significantly improved; the error is even much smaller than the standard CST method with *BPOX* 12 and *BPOY* 12.

The largest error occurred at 50% span section. From the geometric error distribution contour, it is found that the largest geometric error happened around this area. However, despite of the largest error in the leading edge, the value of the suction peak is slightly overestimated. After 50% span, of the remainder of the section, from 60% to 90% span, only slightly mis-matches the position of the suction peak. The value of the suction peak for all of the other sections is addressed accurately.

As a reminder here, this RCST only employs 6<sup>th</sup> order in streamwise and 6<sup>th</sup> order in spanwise direction. It has already reached the high accuracy which the standard CST with BPO 12 and BPO 12 could not obtain. However, if RCST is employed in optimisation, the number of design variables will be only half of that of the standard CST with *BPOX* 12 and *BPOY*. This is an impressive improvement on both accuracy and small number of design variables. The performance of RCST will be further studied in later optimisation work.

## 4.2 CST parameterisation method for wing tip device

As presented in Chapter 1, step changes in aircraft performance, such as 50% CO<sub>2</sub> emission reduction, are set for European aeronautic research. In order to achieve this objective, advanced wingtip device design has been identified as a key technology to provide benefits in drag reduction, reducing emissions and improving the performance of aircraft take-off and landing. Another important feature of the wingtip device is that it is able to retrofit to existing aircraft to provide significant improvement at relatively low cost. Therefore, design and optimisation of the wing tip device is interesting for the aeronautic industry and research.

Wing tip devices, such as winglets, intended to reduce the induced drag, have been applied to aircraft for many years, leading to various types being designed, such as conventional winglet, blended winglet, smoothly blended winglet and Spiroid winglet (Guerrero *et al.* 2012; Hantrais-Gervois *et al.* 2009). The investigation of various types of wing tip device can be found in the literature (de Mattos *et al.* 2003; Reneaux 2004; Mann and Elsholz 2005; Hantrais-Gervois *et al.* 2009; Gerontakos and Lee 2006; Takenaka *et al.* 2008; Meheut *et al.* 2009; Rajendran 2012). Previous literature demonstrates that all types of wing tip device are effectively able to reduce aircraft induced drag. However, the design of a wing tip device is a complex issue, as not only aerodynamic effects but also other constraints such as structure, manufacture, stability and weight penalty should be considered. Hantrais-Gervois *et al.* (2009) and Reneaux (2004) summarised the properties of the most common wing tip devices. They pointed

out that conventional winglet is still the most feasible option for the aircraft industry due to high aerodynamic drag reduction and relatively low structural penalty.

De Mattos *et al.* (2003) investigated the winglet design and application in many historic aircraft and modern aircraft. In their work they pointed out that there is no smooth transition between wing and winglet on the conventional winglet; therefore it is inevitable that an intersection between the wing and wing tip device, extra interference drag and more wave drag could be generated and counteract some benefits of the winglet.

Therefore, a novel winglet, which provides a smooth transition between wing and winglet to give a reduction in the interference drag in this area and provide better aerodynamic performance, has recently been preferred in the aircraft industry (Gratzer 1994). From the point of view of geometric parameterisation, the conventional winglet could be considered as a simple wing that is attached at the wing tip. Therefore, only a few design parameters, such as cant angle, toe angle, taper ratio are needed to describe the entire winglet (Falcão *et al.* 2010; Weierman and Jacob 2010).

Compared with conventional winglets, the novel winglet is a non-planar surface and has more degrees-of-freedom. Therefore, the simple planform parameters used to define the conventional winglets are not sufficient to define this type of winglet. The novel winglet is normally designed using a CAD system. Therefore, CAD-based parameterisation methods are an option, and have been employed in many researchers' work, such as in Takenaka *et al.* (2008), Minnella *et al.* (2010), Pfeiffer (2004) and Rajendran (2012). However, as studied in Chapter 2, a few issues, such as sensitivities and robustness in topology, are still challenging for use of a CAD-based parameterisation method, especially in high-fidelity CFD optimisation. Therefore, a parameterisation method, which is independent of the CAD-system, is desired to describe the novel winglet.

The other promising method is the free-form deformation method (FFD) (Ronzheimer 2005; Anderson *et al.* 2012). However, the FFD method lacks of intuitive control, which would make it difficult to apply constraints to winglet design. Thus, a parameterisation

method with intuitive planform parameters is needed. Furthermore, Gerontakos and Lee (2006) demonstrated that the winglet with negative dihedral cant, which means the winglet points downward, is more efficient in reducing induced drag than the normal winglet with positive dihedral. Therefore, the downward winglet is interesting to the aircraft industry for future aircraft (Hantrais-Gervois *et al.* 2009; Mann 2007; Hicken 2009; Minnella *et al.* 2010). Therefore, a robust parameterisation method is required to represent both upward and downward novel winglets.

In addition, the smooth winglet for large aircraft has been proposed by Felker (2002). The smooth winglet is intended to generate aerodynamic loading at wing tip and to approach to an elliptical distribution. Therefore, Minnella *et al.* (2010) compared the winglet and smooth winglet regarding payload contribution, de-strengthening of vortices, positive traction component, wing flutter, retrofitting capabilities and manufacturing. They concluded that the smooth winglet is more efficient than conventional winglet. Therefore, in this work, the new parameterisation method is also required to be able to represent the smooth winglet.

As mentioned above, the novel winglet has a high degree-of-freedom. The three-dimensional Bezier or B-spline curves are normally employed to directly control the leading edge and trailing edge of winglet (Hantrais-Gervois *et al.* 2009; Meheut *et al.* 2009). However, with direct control of the leading edge in three-dimensions it would be hard to apply constraints and maintain the geometric continuity. Therefore, another way to define winglet is applied. The winglet is considered as a transition mapping from a wing tip extension (Minnella *et al.* 2010).

As shown in the previous section, CST parameterisation has been successfully employed to represent the wing. Therefore, it is extended to parameterise the winglet. In a brief description of the CST method for winglet parameterisation, the CST method is first employed to parameterise the wing tip extension part, and then a transition mapping is applied to translate the wing tip extension part to the winglet. The detailed procedures are presented below.

Two types of winglets have been employed in this thesis. The first type of winglet (winglet-1) composes two parts: the transition part and winglet part, as in Figure 4.33. The transition part is to link between wing and winglet. The winglet in a winglet-1 is a small wing with straight leading edge.

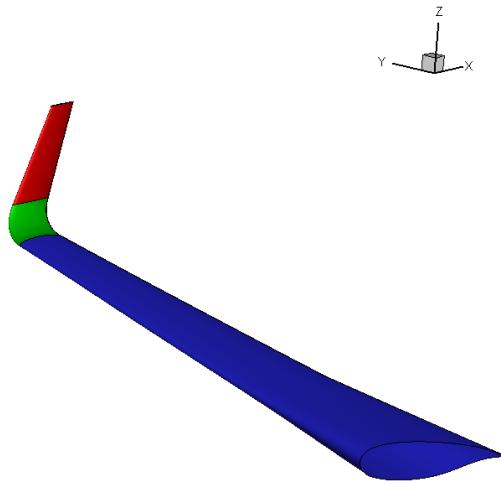


Figure 4.33 The wing with winglet-1: wing in blue, transition part in green, winglet in red

Then, the parameterisation of the winglet-1 is carried out in the following steps:

- 1) The position of the leading edge end point of transition part  $P_2(y,z)$  and the position of the leading edge end point of winglet part  $P_1(y,z)$  are set with respect to the end point of the wing leading edge  $P_0(y,z)$ , as in Figure 4.34. The reason for using the control point position on the y-z plane rather than cant angle or winglet length is that the most concerned constraint for winglet optimisation in practical design is the actual winglet span and height. In this way, the span and height constraints could be set up immediately by limiting  $P_1$  and  $P_2$ .

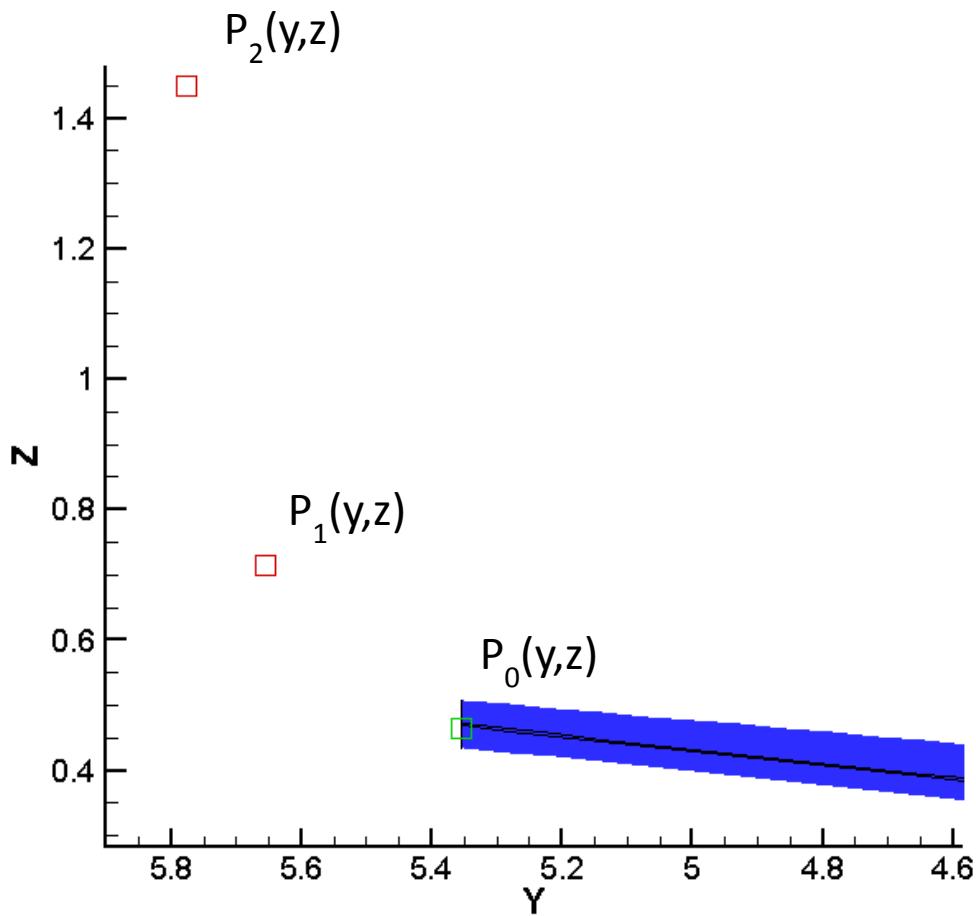


Figure 4.34 The leading edge lines control points

- 2) This step is to set up the leading edge lines of the transition part and winglet projected on the y-z plane. First,  $P_1$  and  $P_2$  are linked linearly as the leading edge line of the winglet. A third order Bezier curve is employed to control the leading edge line of the transition part. Therefore, 4 control points are required for the Bezier curve.  $P_0$  and  $P_1$  are employed as the first two control points. The third control point  $P_3$  is on the extension line of the wing leading edge with respect to the tangential value at the wing tip. The fourth control point  $P_4$  is on the extension line of  $P_1P_2$ . The lengths of  $P_0P_3$  and  $P_1P_4$  are employed as design variables. In order to avoid a non-physical condition, the lengths of  $P_0P_3$  and  $P_1P_4$  are a percent of the length of  $P_0P_1$ . These constraints are aimed at keeping the tangential

continuity between wing and transition, transition and winglet. The leading edge line of the transition part on the y-z plane is then established and notated as  $P_0P_3P_4P_1$ . The leading edge lines and control polygons are shown in Figure 4.35.

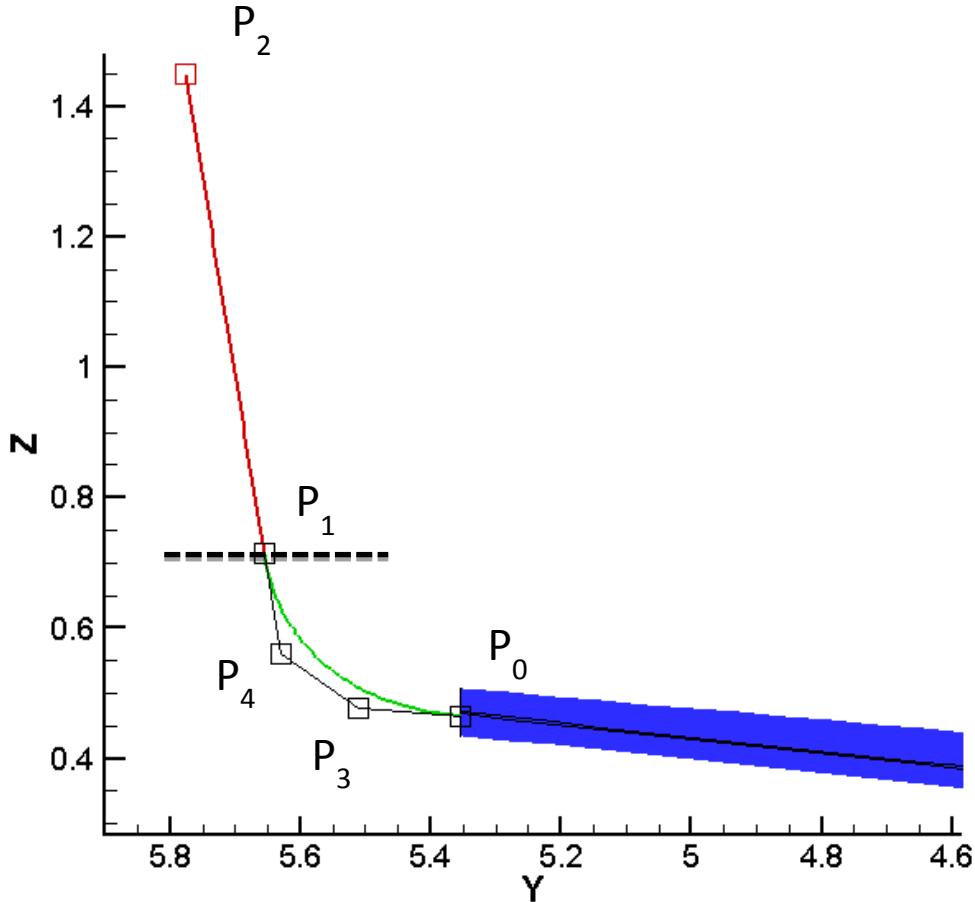


Figure 4.35 The leading edge lines of transition part and winglet on the y-z plane

- 3) In this step, the arc length of leading edge lines on the y-z plane of transition part  $P_0P_3P_4P_1$  and winglet  $P_1P_2$  are calculated at first as  $L_1$  and  $L_2$ . A straight line is then extended from the wing tip with respect to the tangential value on the y-z plane. The straight extension line is then cut at the position with arc length  $L_1$  and  $L_2$ . The span of the wing extension,  $b_{\text{winglet}}$ ,  $b_{\text{transition}}$ ,  $b_{\text{total}}$  is then taken from the projection of the extension line on the y axis, as in Figure 4.45.

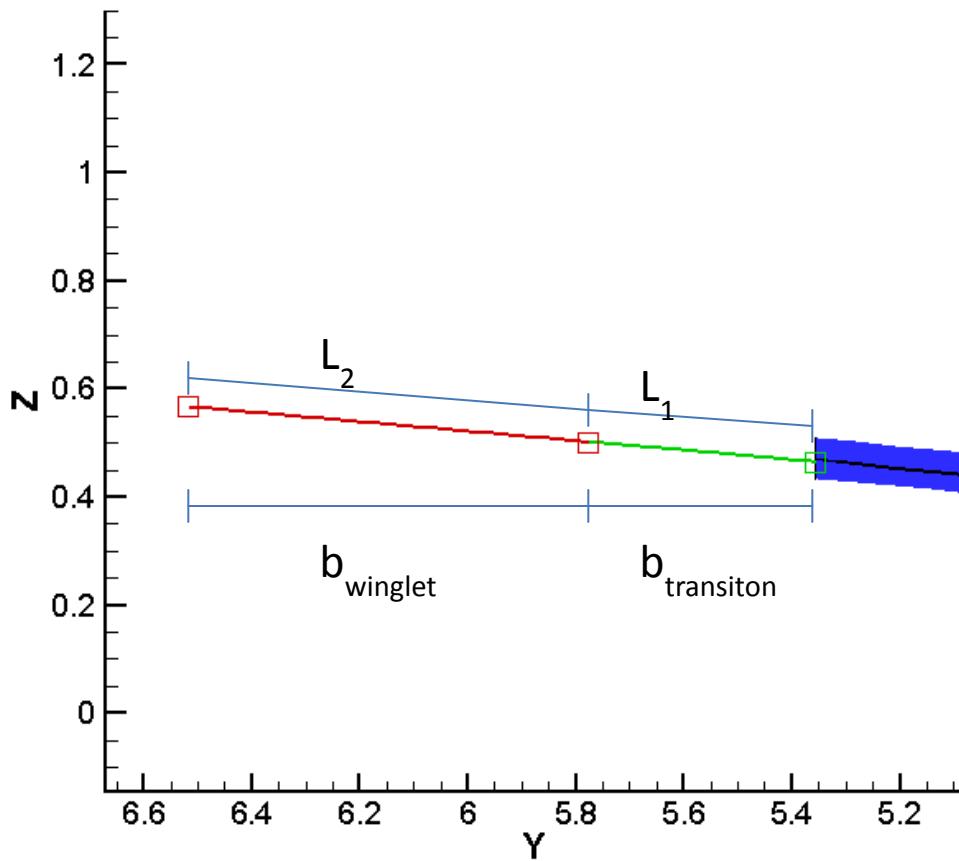


Figure 4.36 The extension of the wing on the y-z plane

- 4) Once the extension line of the wing on the y-z plane is established, the view is changed to the x-y plane. The planform parameters, such as sweep angle of leading edge of transition part  $\alpha_{LE\_tran}$ , sweep angle of trailing edge of transition part  $\alpha_{TE\_tran}$ , sweep angle of leading edge of winglet  $\alpha_{LE\_winglet}$  and sweep angle of trailing edge of winglet  $\alpha_{TE\_winglet}$ , are employed to calculate the positions of the control points  $P'_1, P'_2, P_7, P_8$  of the leading edge and trailing edge of the wing extension on the x-y plane, as shown in Figure 4.35.

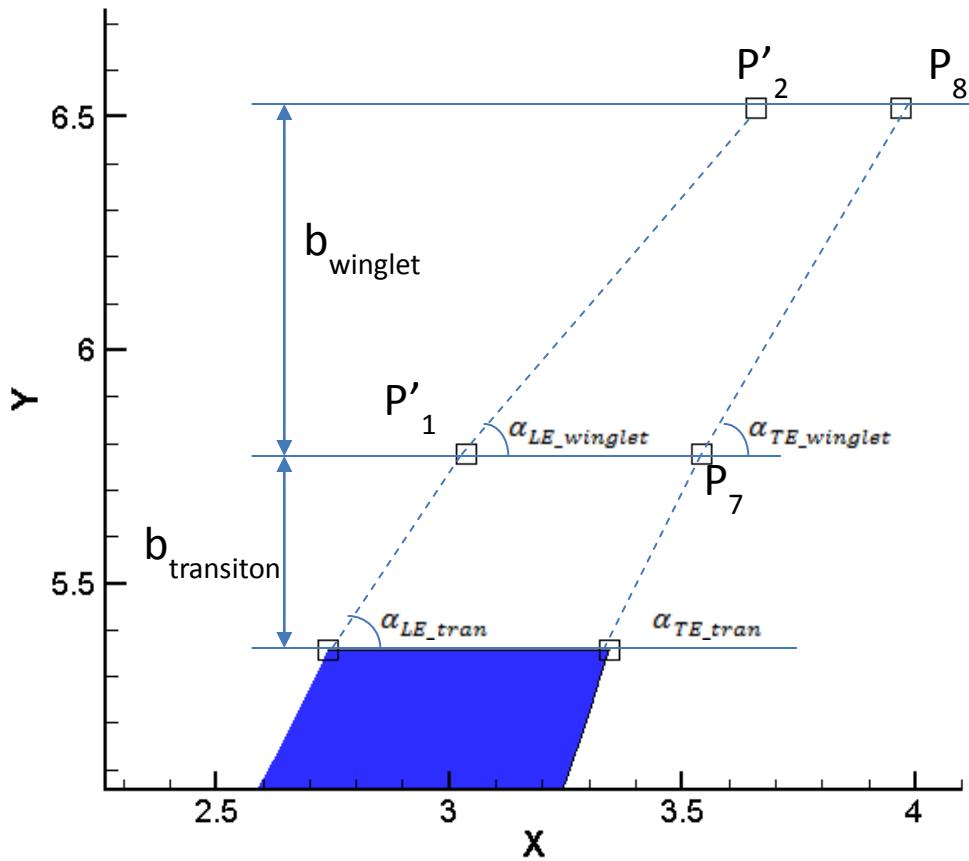


Figure 4.37 The planform parameters for wing extension on the x-y plane

- 5) The leading edge and trailing edge lines of the winglet can be then established by connecting  $P'_1P'_2$  and  $P_7P_8$ . Two fourth order Bezier curves are then employed to control the leading edge and trailing edge of the transition part on the x-y plane. Similar with the leading edge line on the y-z plane, the position of control points  $P'_3P'_4$  and  $P_5P_6$  are located on the extension lines of wing and winglet, and the lengths of the control polygon are employed as design variables. This step is shown in Figure 4.36.

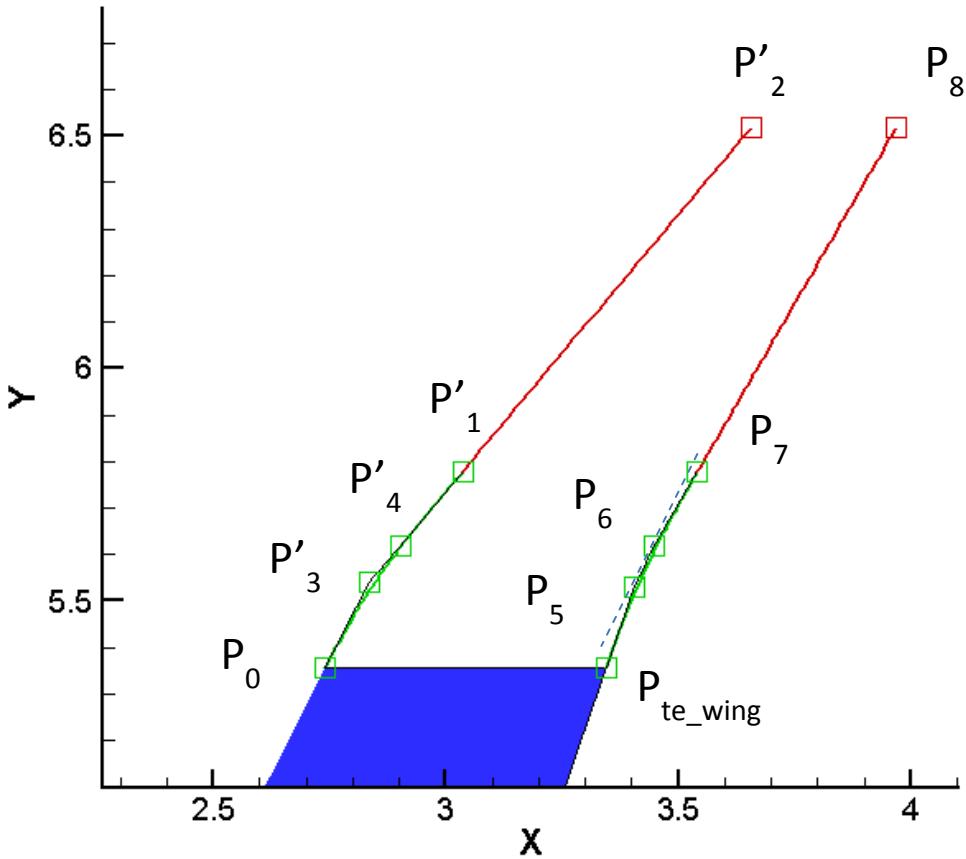


Figure 4.38 The leading and trailing edge lines and control point and polygon for wing extension on the x-y plane

- 6) Once the leading edge and trailing edge lines of the wing extension are generated, the CST parameterisation of the wing is then applied to represent the wing extension part. The definition is same as for the CST wing definition, as in Equations 4.24 to 4.26:

$$\xi_U(\psi, \eta) = C_{N2}^{N1}(\psi) \cdot \sum_i^{Nx} \sum_j^{Ny} [Bu_{i,j} \cdot Sy_j(\eta) \cdot Sx_i] + \psi [\xi_{TE,Up}(\eta) - \tan(\alpha_{Twist}(\eta))] \quad 4.24$$

$$\xi_L(\psi, \eta) = C_{N2}^{N1}(\psi) \cdot \sum_i^{Nx} \sum_j^{Ny} [Bl_{i,j} \cdot Sy_j(\eta) \cdot Sx_i] + \psi [\xi_{TE,Lo}(\eta) - \tan(\alpha_{Twist}(\eta))] \quad 4.25$$

$$x_1(\psi, \eta) = x_{leading}(\eta) + \psi \cdot C_{Local}(\psi, \eta)$$

$$y_1(\psi, \eta) = b_{total} \cdot \eta \quad 4.26$$

$$z_1(\psi, \eta) = z_{leading}(\eta) + \xi(\psi, \eta)$$

where  $\eta = \frac{y}{b_{total}}$  is with respect to the total span of wing extension  $b_{total} = b_{winglet} + b_{transition}$ , and the local chord distribution is calculated from trailing edge and leading edge  $C_{local} = y_{trailing}(\eta) - y_{leading}(\eta)$ .

In order to maintain the geometric continuity and geometric first order derivative continuity between wing and wing extension, a few constraints are introduced. The first condition is the wing and wing extension should be represented with the same *BPOX*. Therefore, the geometric continuity for the CST equation would be easily derived, as in Equations 4.27 to 4.30:

$$Bu_{i,0,winglet} = Bu_{i,BPOY\_wing,wing} \quad 4.27$$

$$Bl_{i,0,winglet} = Bl_{i,BPOY\_wing,wing} \quad 4.28$$

$$\begin{aligned} & \frac{BPOY\_winglet(Bu_{i,0,winglet} - Bu_{i,1,winglet})}{b_{total\_wingextenstion}} \\ &= \frac{BPOY\_wing(-Bu_{i,BPOY\_wing,wing} + Bu_{i,BPOY\_wing-1,wing})}{b_{wing}} \end{aligned} \quad 4.29$$

$$\begin{aligned} & \frac{BPOY\_winglet(Bl_{i,0,winglet} - Bl_{i,1,winglet})}{b_{total\_wingextenstion}} \\ &= \frac{BPOY\_wing(-Bl_{i,BPOY\_wing,wing} + Bl_{i,BPOY\_wing-1,wing})}{b_{wing}} \end{aligned} \quad 4.30$$

where *BPOY\_winglet* is the Bernstein polynomials order for winglet on spanwise, *BPOY\_wing* is the Bernstein polynomials order for winglet on spanwise. Equations 4.27 and 4.28 are for keeping geometric continuity at upper and lower surfaces. Equations 4.29 and 4.30 are for maintaining the 1<sup>st</sup> order  $C^1$  continuity. If the second order continuity  $C^2$  is required, the constraints condition could also be easily derived. Similar

with the CST wing, the planform parameters, such as twist distribution  $\alpha_{Twist}(\eta)$  and trailing edge thickness distribution  $\xi_{TE}(\eta)$ , employ the Bernstein polynomials. Equation 4.31 shows the Bernstein polynomials for twist distribution  $\alpha_{Twist}(\eta)$ :

$$\alpha_{Twist,winglet}(\eta) = \sum_{i=0}^n [A\_winglet_i \cdot K_{i,n} \cdot \eta^i \cdot (1-\eta)^{n-i}] \quad 4.31$$

Similar constraints as with Equations 4.15 and 4.16 should be applied to maintain planform geometric continuity between wing and wing extension. Examples of twist distribution are shown in Equations 4.32 and 4.33:

$$A\_winglet_0 = A\_outter\_wing_{N_{outter}} \quad 4.32$$

$$\frac{A\_outter\_wing_{N_{outter}} - A\_outter\_wing_{N_{outter}-1}}{(b_{wing} - b_{crank})} = \frac{-A\_winglet_0 + A\_winglet_1}{b_{total\_wingextension}} \quad 4.33$$

Similar constraints for trailing edge thickness distribution can be easily derived and applied. Therefore, the surface of wing extension part on the x-y plane is generated, as in Figure 4.39.

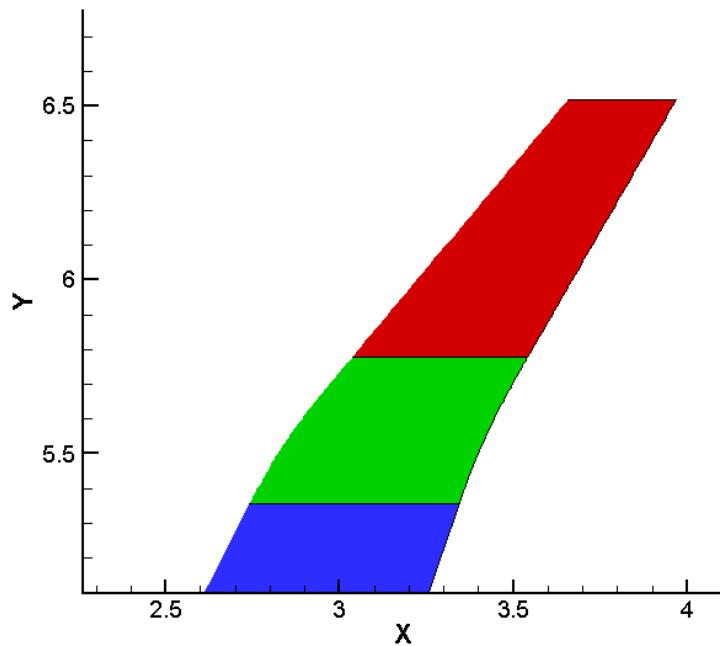


Figure 4.39 The surface of wing extension

- 7) The last step is to translate the surface of the wing extension part from the x-y plane to the y-z plane. The translation is to maintain the same length of leading edge of wing extension and arc length of the leading edge of the winglet-1 on the y-z plane. For example, at the spanwise position  $\eta$ , the leading edge points  $y_{leading}(\eta)$  and  $z_{leading}(\eta)$  can be calculated. The length of the leading edge is then calculated from  $L_1 = \sqrt{(y_{leading} - y_{wingtip})^2 + (z_{leading} - z_{wingtip})^2}$ . Following this, the position  $y'_{leading}(\eta)$ ,  $z'_{leading}(\eta)$  with same arc length  $L_1$  on the leading edge line  $P_0P_3P_4P_1$  of the winglet-1 on the y-z plane, in Figure 4.36, is searched out. The  $y_{leading}(\eta)$ ,  $z_{leading}(\eta)$  of the surface of the wing extension in Equation 4.26 are replaced by  $y'_{leading}(\eta)$ ,  $z'_{leading}(\eta)$ .

$$x_2(\psi, \eta) = x_1(\psi, \eta)$$

$$y_2(\psi, \eta) = y'_{leading}(\eta) \quad 4.34$$

$$z_2(\psi, \eta) = z'_{leading}(\eta)$$

The rotation translation equation is then applied. The rotation angle is with respect to the tangent of the leading edge line  $P_0P_3P_4P_1 \ arc \tan\left(\frac{dz'_{leading}}{dy'_{leading}}\right)$  and the tangent of the leading edge at wing tip,  $\alpha_{wing\_tip}$ .

$$x_3 = x_2$$

$$y_3 = y_2 \cdot \cos(\theta(\eta)) - z_2 \cdot \sin(\theta(\eta)) \quad 4.35$$

$$z_3 = y_2 \cdot \sin(\theta(\eta)) + z_2 \cdot \cos(\theta(\eta))$$

where

$$\theta(\eta) = \arctan\left(\frac{dz'_{leading}(\eta)}{dy'_{leading}(\eta)}\right) - \alpha_{wing\_tip} \quad 4.36$$

The translation relationship is illustrated in Figure 4.40 to 4.42.

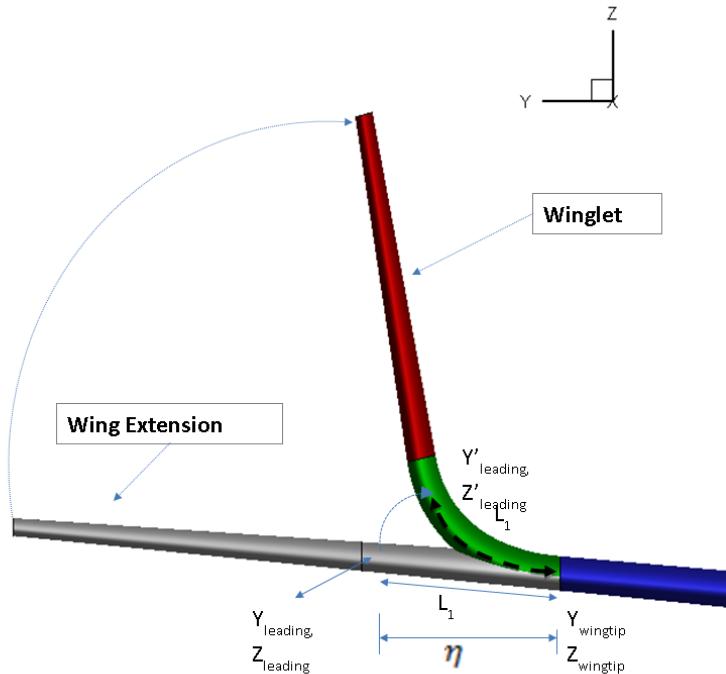


Figure 4.40 The translation relationship between wing extension and winglet-1

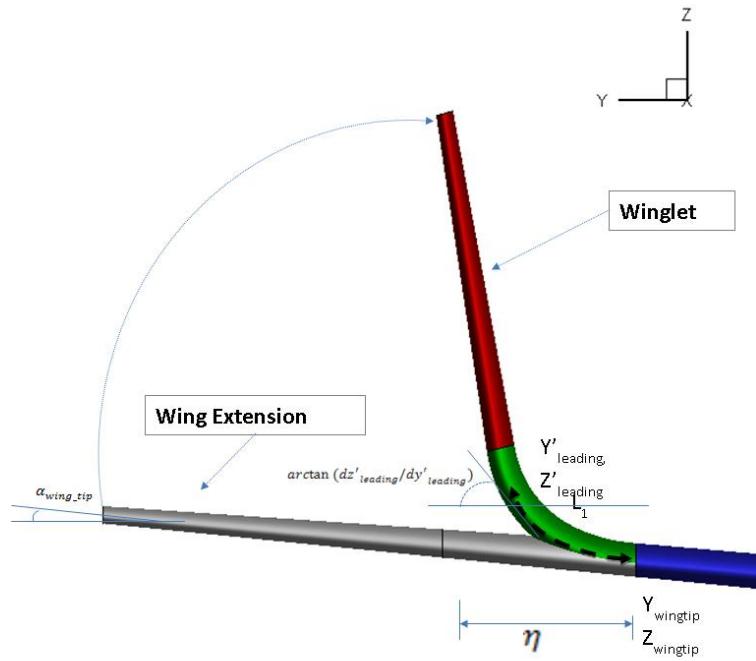


Figure 4.41 The rotation relationship

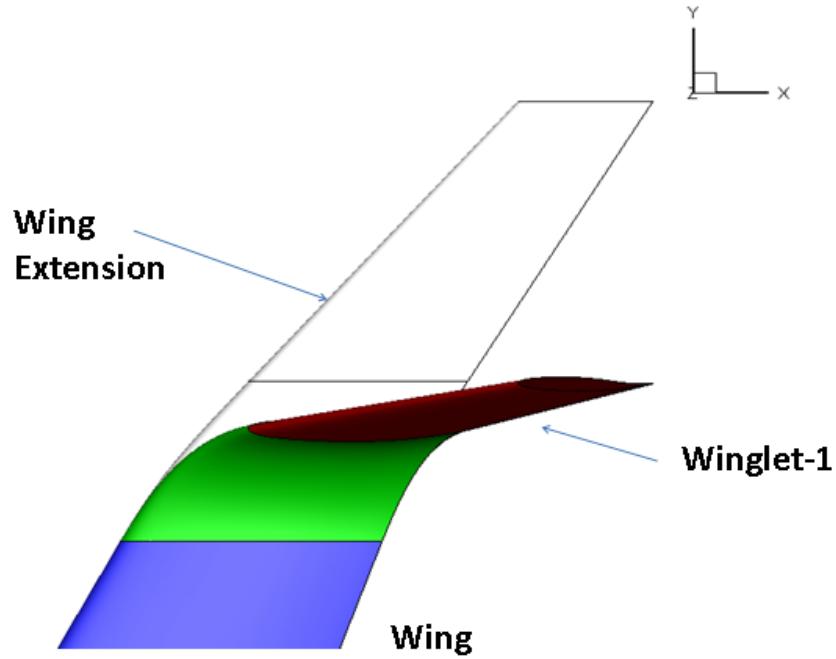


Figure 4.42 The translation view on the x-y plane

Therefore, the CST method for the winglet-1 has been presented completely. For the case of winglet 1 with *BPOX* 6, *BPOY* 6 and 4<sup>th</sup> order Bernstein polynomials for twist and trailing edge thickness, the number of design variables is up to 114. Most of the design variables are differentiable to provide surface mesh point sensitivities. However, because the translation mapping is employed in this winglet definition, the planform parameters,  $P_1, P_2$ , length  $P_0P_3$  and  $P_1P_4$ , are not differentiable at the moment since they will change the span of the wing extension which is fundamental to this transformation. In optimisation, the sensitivities of these parameters can be calculated by finite difference.

The second type of winglet (winglet-2) is smooth winglet, which could be defined in a similar way to that of the winglet-1. It could be considered as the winglet-1 with the winglet part removed. Therefore, the control point  $P_2(y,z)$  in the winglet-1 is removed in winglet-2 and replaced by the winglet tip dihedral angle  $\alpha_{dihedral\_winglet}$ . The sweep angle of the leading edge of the transition part  $\alpha_{LE\_tran}$  and the sweep angle of the trailing edge of the transition part  $\alpha_{TE\_tran}$  are replaced by the sweep angle of the leading edge of the winglet  $\alpha_{LE\_winglet}$  and the sweep angle of the trailing edge of the winglet  $\alpha_{TE\_winglet}$ .

The sweep angle of the leading edge of the winglet  $\alpha_{LE\_winglet}$  and the sweep angle of the trailing edge of the winglet  $\alpha_{TE\_winglet}$  are replaced by the sweep angle of the leading edge at the winglet tip  $\alpha_{LE\_tip}$  and the sweep angle of the trailing edge at the winglet tip  $\alpha_{TE\_tip}$ , respectively. The definition and planform parameters are shown in following Figure 4.43 to 4.47. The CST equations are applied in the same way as Equations 4.24 to 4.36.

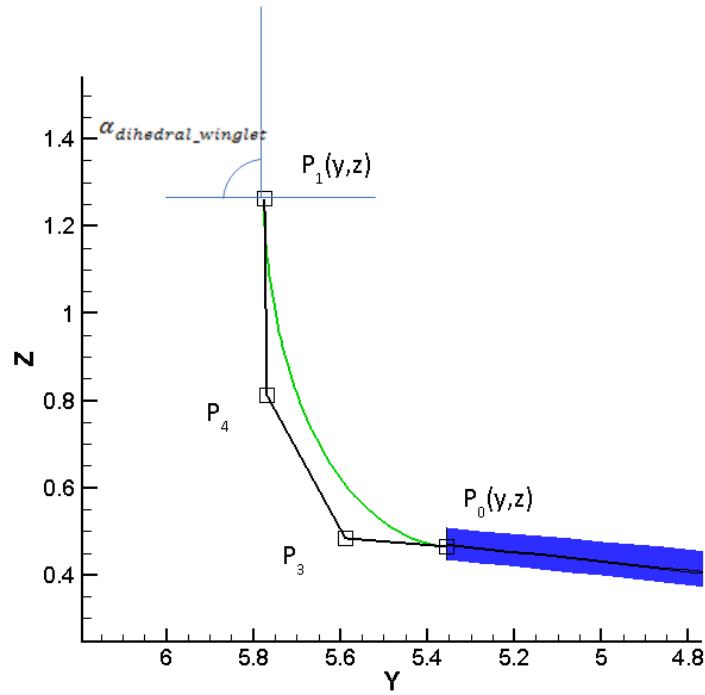


Figure 4.43 The winglet-2 leading edge on the y-z plane and planform parameters

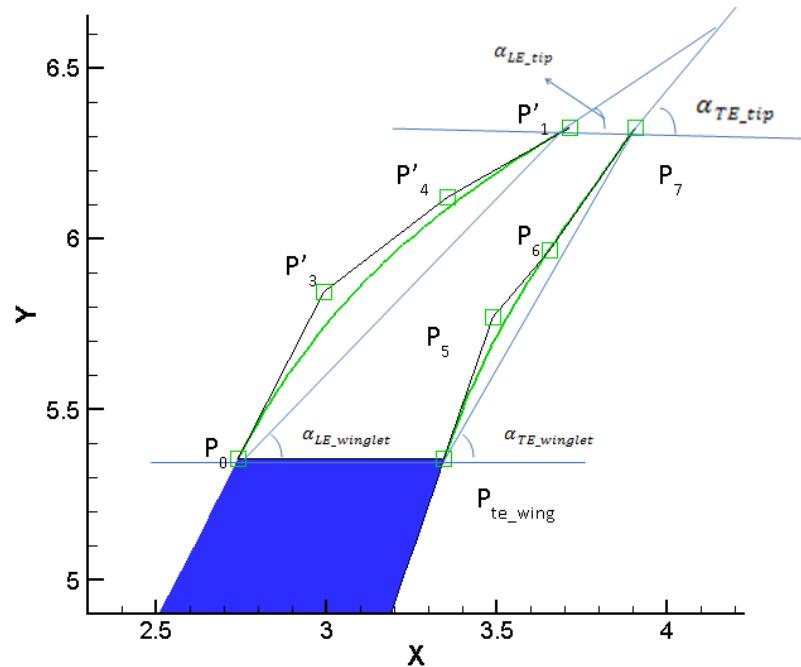


Figure 4.44 The planform parameters of wing extension for winglet-2 on the x-y plane

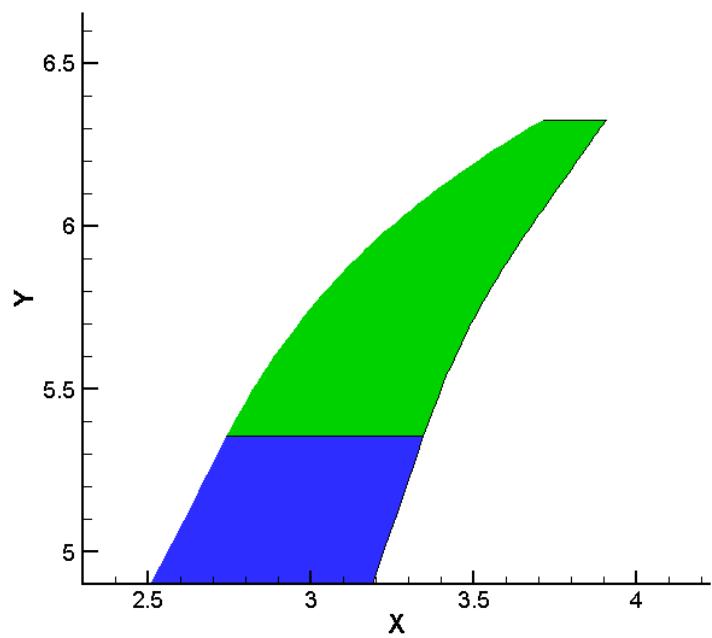


Figure 4.45 The surface of wing extension for winglet-2 on the x-y plane

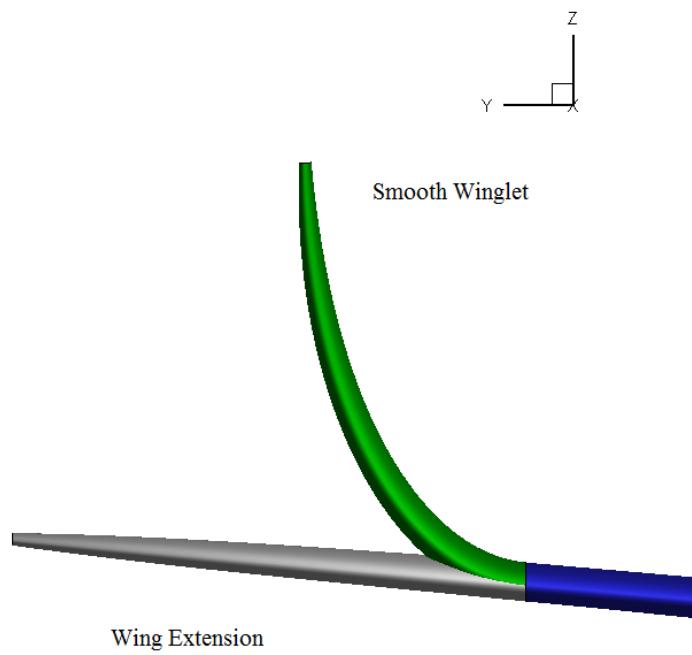


Figure 4.46 The translation of wing extension to winglet-2

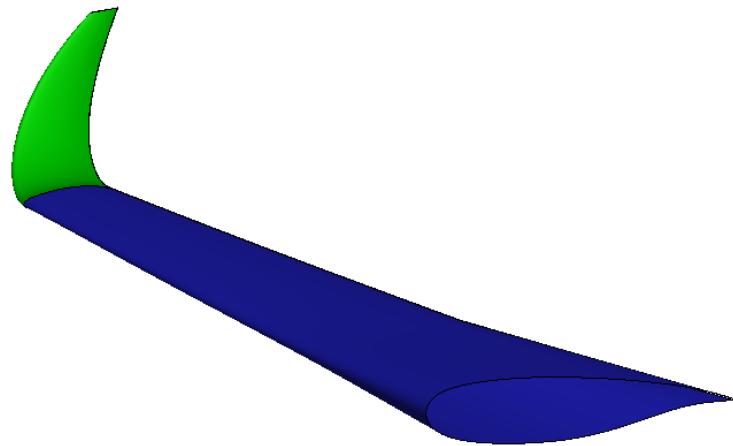
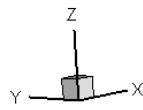


Figure 4.47 The surface of winglet-2

Furthermore, the downward winglet (winglet-3) and smooth winglet (winglet-4) can be generated if the control points  $P_1$ ,  $P_2$  are located below the wing. The Figures 4.46 to 4.47 show the winglet-3 and winglet-4.

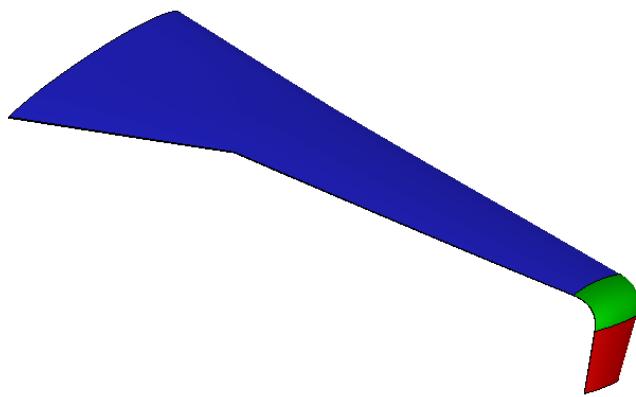
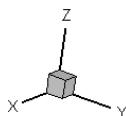


Figure 4.48 The winglet-3

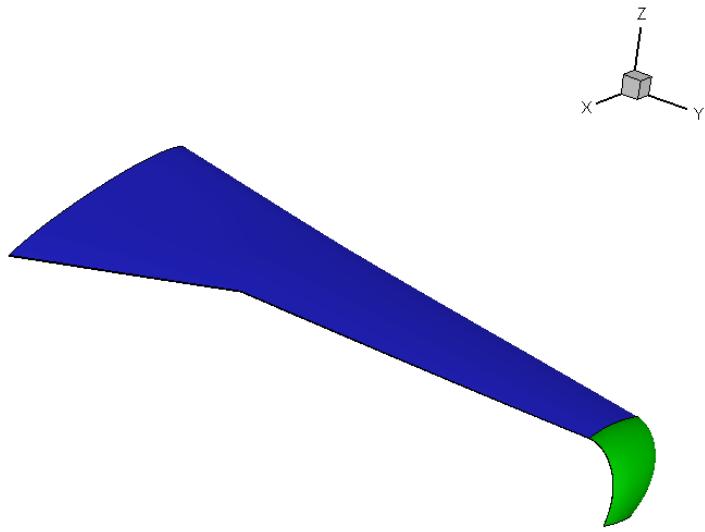


Figure 4.49 The winglet-4

#### 4.3 CST Parameterisation for fuselage (simplified forward, mid and tail cone parts)

The fuselage has a typical ‘body-axis type’ geometry in which one CST equation is employed to define the cross-section profile and another is used to define the distribution of cross-section along the body axis. Table 2.1 has shown the capability of the CST method for representing symmetric cross-section profiles. In the conventional modern civil passenger transport aircraft, the fuselage could be divided into three parts, namely forward part, mid-part and tail cone (these could be notated as nose fuselage, cylinder fuselage and rear fuselage) as in Figure 4.50. Each part will be discussed in the following sections.

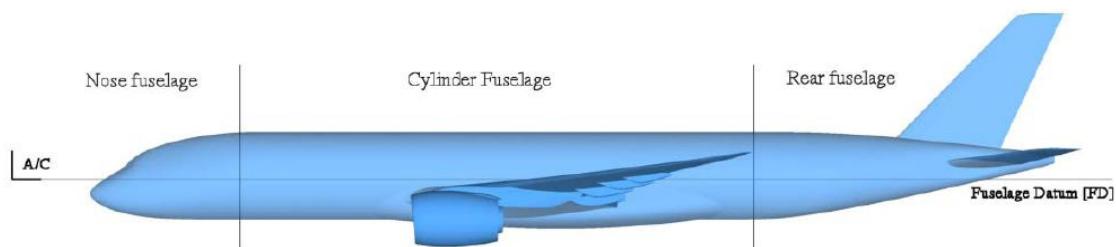


Figure 4.50 The three main parts of the fuselage

#### 4.3.1 Cylindrical fuselage

The mid-part of the fuselage is the cylindrical fuselage with a constant cross-section along the aircraft body axis. In most civil passenger aircraft, the cross-section profile of this component is a smoothly blended cross section. Consequently, a CST function with class function and unit shape function is employed to represent the cross-section. The profile function of the upper lobe is then written in the following form:

$$X(\eta) = \psi \times T_{length} \quad 0 \leq \eta \leq 1 \quad 4.37$$

$$Y(\eta) = \eta \times W_{Fuselage} \quad 0 \leq \eta \leq 1 \quad 4.38$$

$$Z_U(\eta) = 2 \times H_{Fuselage} \times \eta^{0.5} \bullet (1 - \eta)^{0.5} \quad 0 \leq \eta \leq 1 \quad 4.39$$

where  $T_{length}$  is the total length of mid-part fuselage,  $H_{Fuselage}$  is the length from the peak of the upper lobe to the profile centre and  $W_{Fuselage}$  is the width of the section profile. The cross-section is then constantly extruded along the fuselage body axis, and the mid-part fuselage is established. Figure 4.51 shows the CST parametric mid-part fuselage and profile parameters  $H_{Fuselage}$  and  $W_{Fuselage}$ .

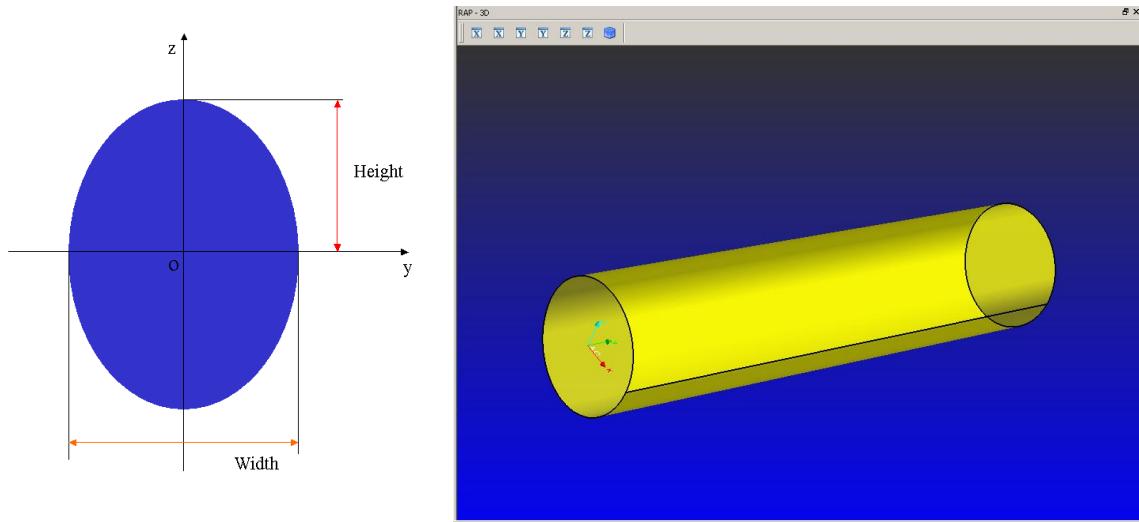


Figure 4.51 The CST parametric mid-part fuselage component

#### **4.3.2 Nose fuselage**

For a realistic aircraft forward part fuselage the geometry is highly complex, due to the presence of the cabin and window shield, which are designed following many constraints. In general, the cabin and window shield are not smooth surfaces and can be decomposed into many sub-components, and the curvature discontinuity appears at the intersection between each sub-component. Therefore, it is extremely difficult to specify a generic definition to represent the forward part fuselage with a real cabin. Figure 4.52 shows the forward part fuselage geometry with cabin from an aircraft model.

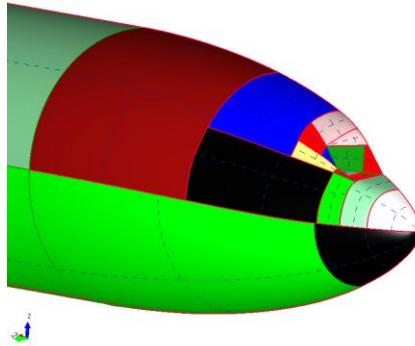


Figure 4.52 The forward part fuselage of an F4 aircraft with cabin

Because of the complexity of a realistic forward part fuselage and the lack of generic definition, the parameterisation for a realistic forward part fuselage is difficult to implement. However, because the aerodynamic optimisation of the nose fuselage is normally carried out in the preliminary design, so a simplified nose fuselage without cabin and window shield can be used as an initial design. The cabin and window shield would be modified manually later based on this initial design. The simplified nose fuselage could be defined as an elliptic cross-section distribution along crown, keel and width line. The CST parametric model for the simplified nose component could be established.

$$X = T_{length} \cdot \psi \quad 4.40$$

$$Y(\eta) = \eta \cdot W_{nose}(\psi) \quad 4.41$$

$$Z_u(\psi, \eta) = 2H_{nose}(\psi) \cdot \eta^{0.5} \cdot (1 - \eta)^{0.5} + Z_{centre}(\psi) \quad 4.42$$

$$Z_u(\psi, \eta) = -2K_{nose}(\psi) \cdot \eta^{0.5} \cdot (1 - \eta)^{0.5} + Z_{centre}(\psi) \quad 4.43$$

where

$T_{length}$  is the total length of forward part fuselage

$Z_{centre}(\psi)$  is centre of cross-section, which is also notated with centre line

$W_{nose}(\psi)$  is width of cross-section, which is also notated as side line

$H_{nose}(\psi)$  is the length from peak of upper lobe of profile to the centre of profile

$K_{nose}(\psi)$  is the length from peak of lower lobe of profile to the centre of profile

Because the nose fuselage of modern civil transport aircraft has an apparently round nose, the CST equations that could be employed to represent it are:

$$H_{nose}(\psi) = \psi^{0.5} \cdot (1 - \psi)^{1.0} \cdot \sum_{r=0}^n A H_i \cdot S_i(\psi) + H_{fuselage} \cdot g(\psi) \quad 4.44$$

$$W_{nose}(\psi) = \psi^{0.5} \cdot (1 - \psi)^{1.0} \cdot \sum_{r=0}^n A W_i \cdot S_i(\psi) + W_{fuselage} \cdot g(\psi) \quad 4.45$$

$$K_{nose}(\psi) = \psi^{0.5} \cdot (1 - \psi)^{1.0} \cdot \sum_{r=0}^n A K_i \cdot S_i(\psi) + K_{fuselage} \cdot g(\psi) \quad 4.46$$

where

$$g(\psi) = -2\psi^3 + 3\psi^2 \quad 4.47$$

$$A H_n = 0 \quad A W_n = 0 \quad A K_n = 0$$

This could maintain the geometrical continuity at the interface between nose fuselage and mid-part fuselage, and also maintain 1<sup>st</sup> order derivative continuity.

Therefore, the CST parametric nose fuselage can be obtained and connected to the parametric mid-part fuselage. Figure 4.53 shows the parametric nose fuselage and illustrates the crown, keel and side lines.

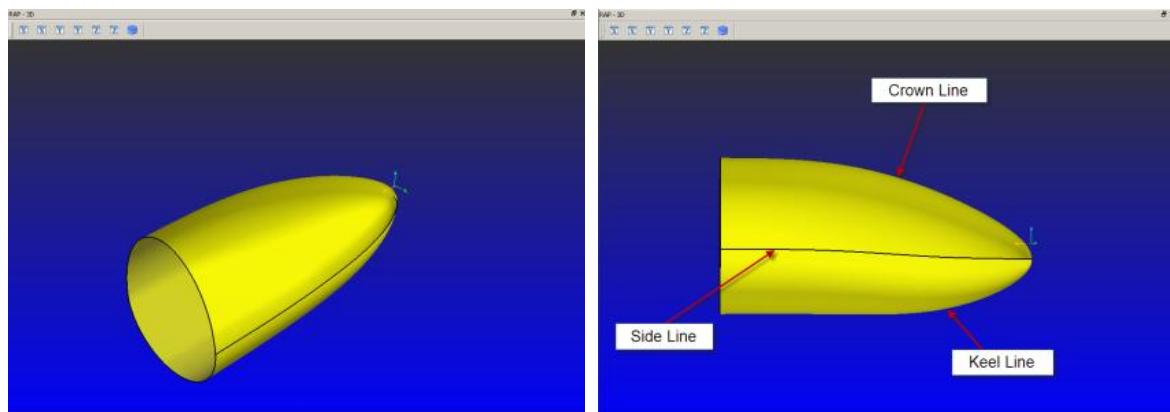


Figure 4.53 The CST parametric forward part fuselage

### 4.3.3 Rear fuselage

The tail cone (rear part fuselage) is defined similarly to the simplified nose fuselage.

Figure 4.54 shows a CAD tail cone model.

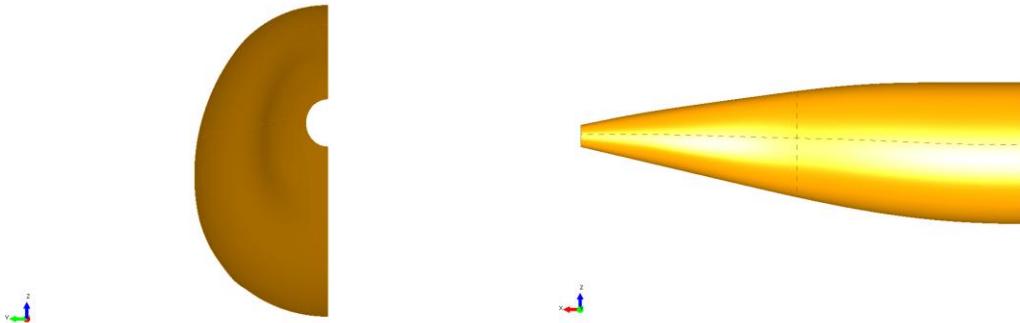


Figure 4.54 A CAD tail cone model

The tail cone is an important aerodynamic component for the fuselage. It contributes importantly to the flow separation in this area and to viscous drag increase. In addition, the HTP work area is also presented on the tail cone. Hence, the cross-section profile of tail cone design is a bit more complex than simple elliptic. It requires the different class parameters  $NC$  to fit the cross-section profile of the tail cone. Some tests have found that only a class function with unit shape function is unable to represent the profile of the tail cone of a realistic aircraft. Consequently, the non-uniform shape function could be employed if it is necessary.

The CST parameterisation for the tail cone is then derived and written as Equations 4.48 to 4.54. Because the tail cone has a symmetric geometry, the CST parameterisation only represents half of the part to ensure its symmetry. Figure 4.55 shows the CST parametric tail cone model.

$$\xi_U(\psi, \eta) = (0.5 \bullet \eta + 0.5)^{NC(\psi)} \times (0.5 - 0.5 \bullet \eta)^{NC(\psi)} \bullet \sum_i^{Ny} \sum_j^{Nx} [Bu_{i,j} \bullet Su_j(\psi) \bullet Sy_i] / 0.5^{2NC(\psi)} \quad 4.48$$

$$\xi_L(\psi, \eta) = (0.5 \bullet \eta + 0.5)^{NC(\psi)} \times (0.5 - 0.5 \bullet \eta)^{NC(\psi)} \bullet \sum_i^{Ny} \sum_j^{Nx} [Bl_{i,j} \bullet Su_j(\psi) \bullet Sy_i] / 0.5^{2NC(\psi)} \quad 4.49$$

where

$$\begin{aligned} Bu_{i,0} &= 1.0 & i &= 1, 2, \dots, Nx \\ Bl_{i,0} &= 1.0 & i &= 1, 2, \dots, Nx \end{aligned} \quad 4.50$$

$$x = T_{length} \bullet \psi \quad 4.51$$

$$y = W_{tail}(\psi) \bullet \eta \quad 4.52$$

$$z_U = \xi_U(\psi, \eta) \bullet H_{tail}(\psi) + Z_{Centre}(\psi) \quad 4.53$$

$$z_L = \xi_L(\psi, \eta) \bullet K_{tail}(\psi) + Z_{Centre}(\psi) \quad 4.54$$

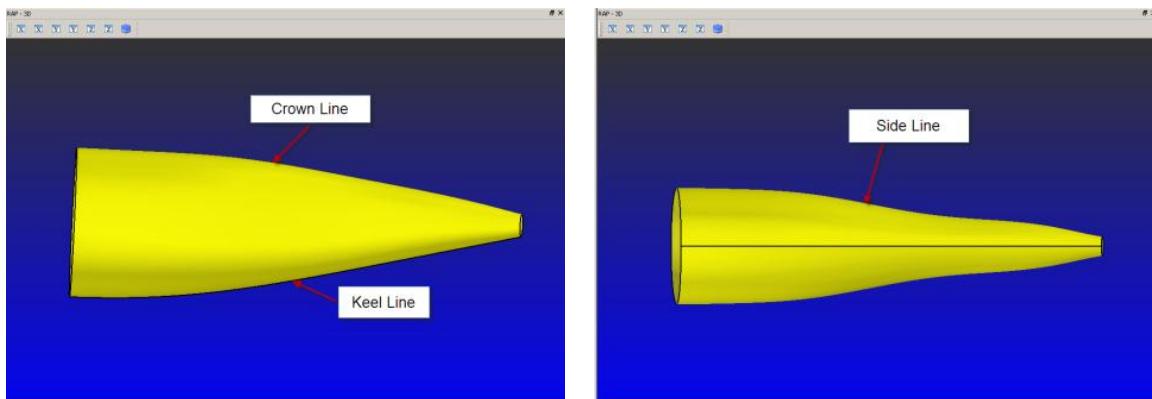


Figure 4.55 The CST parametric tail cone

#### 4.4 CST parameterisation for belly-fairing

In the modern civil passenger transport aircraft, belly-fairing is normally employed at the wing root to reduce interference drag between wing and fuselage, and cover some equipment, for instance the landing gear, air-conditioning, etc., which is exposed outside the fuselage.

The aerodynamic drag would be dramatically reduced if the belly-fairing was well designed, or would be increased if the belly-fairing were designed improperly. The current design methods in industry are still based on manual CAD work. The numerical

optimisation technique for belly-fairing design is interesting. To understand the shape of the belly-fairing, its cross-section profile has been shown in Figure 4.56.

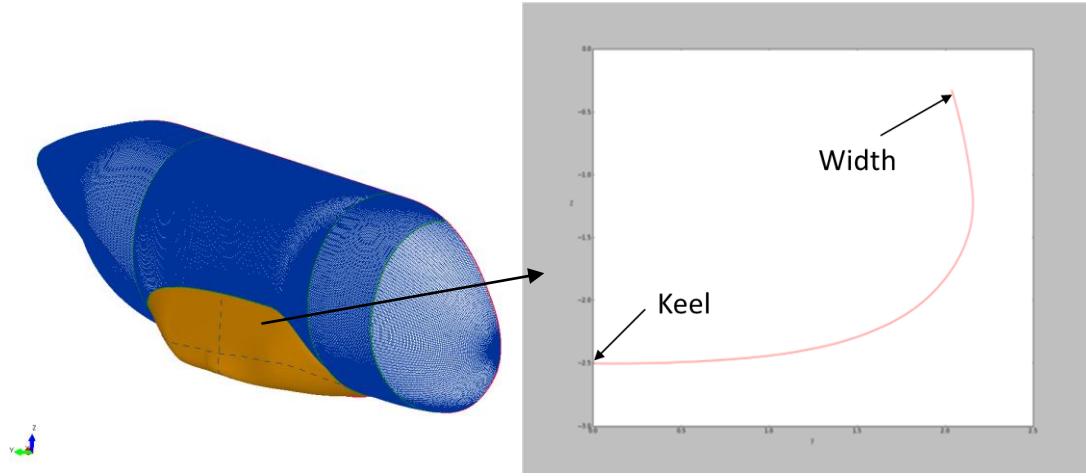


Figure 4.56 Belly fairing model and its profile

This figure illustrates that the cross-section profile of the belly-fairing would be a multi-value function if the analytical function is directly applied to this profile with respect to the x-axis in Cartesian coordinates. Furthermore, it is inconvenient to split the belly fairing to upper and lower parts, especially at its leading and rear areas. Therefore, the best way to manipulate the belly fairing is to transform its coordinates from Cartesian coordinates to cylindrical coordinates.

The belly-fairing is then described by the angle and radius in cylindrical coordinates, and a one-to-one mapping function could be obtained. The origin point of the cylindrical coordinates is set up to be at the centre point of the fuselage. The bounding line of the belly-fairing is the intersection line of belly fairing and fuselage, and it changes as soon as any variant of belly-fairing is applied. Consequently, the belly-fairing must be extended to a larger surface, and two bounding lines which are keel and width lines are employed to control this surface.

In addition, the belly-fairing has a symmetric geometry with respect to the x-z plane. Therefore, the first derivative at the keel line is zero, as  $\frac{dz}{dy} = 0$ . In the cylindrical

coordinate, it is equivalent to require  $\frac{dR}{d\theta} = 0$ . Therefore, in the new CST equation, this condition must be provided naturally.

Eventually, the CST parameterisation for the section profile of the belly fairing could be written as:

$$R(\bar{\theta}) = C_{1.0}^{1.0} \bullet S(\bar{\theta}) + f(\bar{\theta}) \bullet R_{keel} + g(\bar{\theta}) \bullet R_{width} \quad \bar{\theta} = \frac{\theta}{-0.5\pi} \in [0,1] \quad 4.55$$

where  $\bar{\theta}$  is non-dimensionalised angle coordinate,  $\bar{\theta} = 0$  points to width line, and  $\bar{\theta} = 1$  points to keel position.  $S(\bar{\theta})$  is the shape function using Bernstein polynomials:

$$S(\bar{\theta}) = \sum_{i=0}^n [A_i \cdot K_{i,n} \cdot \bar{\theta}^i \cdot (1 - \bar{\theta})^{n-i}] \quad 4.56$$

The class parameters are selected as  $N_1=1.0$  and  $N_2 = 1.0$ , because it is then able to provide the following conditions:

$$C_{1.0}^{1.0} \cdot S(0) = 0 \quad C_{1.0}^{1.0} \cdot S(1) = 0 \quad 4.57$$

$$\frac{dC_{1.0}^{1.0} \cdot S(0)}{d\theta} = A_0 \quad \frac{dC_{1.0}^{1.0} \cdot S(1)}{d\theta} = A_n$$

The details can be found in Appendix C. Therefore,  $A_0$  and  $A_n$  are set to zero, which will guarantee that the CST equation goes through the boundary value at keel and width, and will keep the derivatives to be zero. Furthermore, in order to provide the boundary condition at keel line and width line, the function  $f(\bar{\theta})$  and  $g(\bar{\theta})$  are defined as below.

$$f(\bar{\theta}) = -2\bar{\theta}^3 + 3\bar{\theta}^2 \quad 4.58$$

$$g(\bar{\theta}) = 2\bar{\theta}^3 - 3\bar{\theta}^2 + 1 \quad 4.59$$

These two polynomials have following conditions:

$f(0) = 0$  guarantees that the keel line value will not affect the width line

$f(1) = 1$  guarantees that the profile will go through the keel line

$g(0) = 1$  guarantees that the profile will go through the width line

$g(1) = 0$  guarantees that the width line value will not affect the keel line

$f'(1) = 0$  and  $g'(1) = 0$  guarantee that the derivative of the profile at the keel is zero.

Similar to the CST parametric wing, the each shape function parameter is a distribution function along the body axis. Therefore, the entire CST parametric model for belly fairing can be written as below:

$$R(\bar{\theta}, \psi) = C_{1,0}^{1,0} \bullet \sum_i^{Nx} \sum_j^{Ny} B_{i,j} \bullet Sx_j(\psi) \bullet Sy_i(\bar{\theta}) + f(\bar{\theta}) \bullet R_{keel}(\psi) + g(\bar{\theta}) \bullet R_{width}(\psi) \quad 4.60$$

$$x = \psi \bullet T_{length} \quad 4.61$$

$$y = R(\bar{\theta}, \psi) \bullet \cos(\theta) \quad 4.62$$

$$z = R(\bar{\theta}, \psi) \bullet \sin(\theta) \quad 4.63$$

where  $R_{keel}(\psi)$  is a distribution function of radius at keel position  $\theta = -0.5\pi$ ,  $R_{width}(\psi)$  is a distribution function of radius at width position  $\theta = 0$ , which is normally represented by lower order polynomials or cubic splines. In order to satisfy the boundary condition at keel and width line,  $B_{0,j}$  and  $B_{n,j}$  are set to zero. To ensure that non-physical geometry will not be generated, the following constraints should be added on during the optimisation process:

$$R(\bar{\theta}, 0) \leq R_{Fuselage}(\theta) \quad 4.64$$

$$R(\bar{\theta}, 1) \leq R_{Fuselage}(\theta) \quad 4.65$$

$$\text{Max}(R_{width}(\psi)) < W_{Fuselage} \quad 4.66$$

The entire belly fairing surface represented by the CST method is then shown in Figure 4.57.

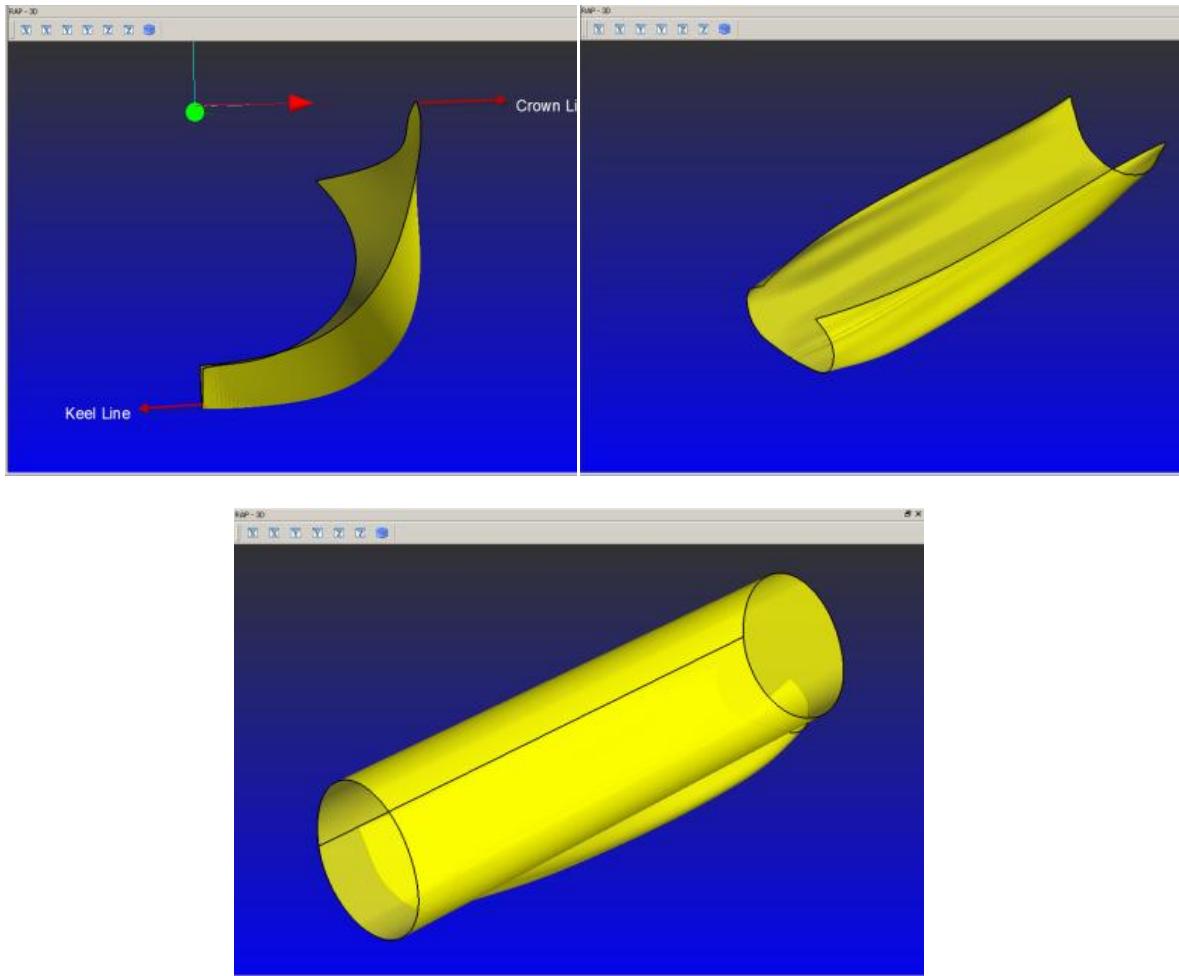


Figure 4.57 The CST belly-fairing parametric model

The accuracy of belly-fairing is also quickly studied by inverse fitting an existing belly-fairing model with total length of about 14 metres. The *BPOX* here implies the Bernstein order for controlling the belly-fairing profile, and the *BPOX* implies the Bernstein order for controlling streamwise (*x* direction). The error distribution contours regarding radius are shown in Figure 4.58 and Figure 4.59.

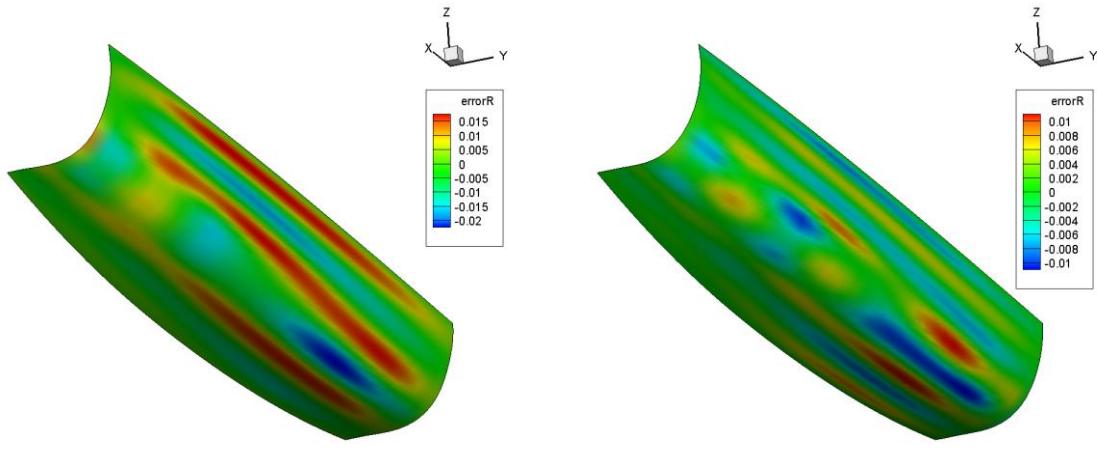


Figure 4.58 The error contour of belly-fairing inverse fitting with *BPOX 6-BPOY 6* (left figure) and *BPOX 8-BPOY 8* (right figure)

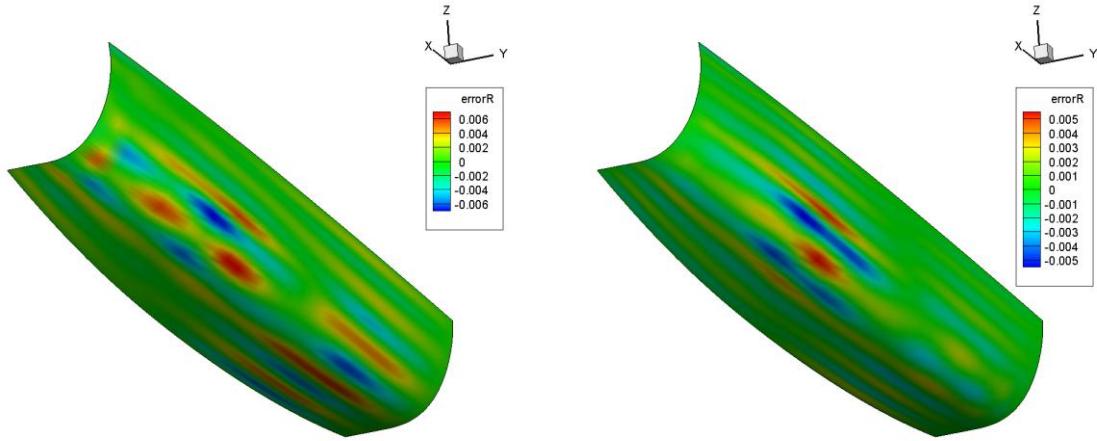


Figure 4.59 The error contour of belly-fairing inverse fitting with *BPOX 10-BPOY 10* (left figure) and *BPOX 12-BPOY 12* (right figure)

The error distribution contour illustrates that the maximum error is always occurring at the curvature change area. The maximum fitting error using BPO 6 in both directions is around 0.02 metres, equivalent to a non-dimensional value of  $1.5 \times 10^{-3}$ . However, the maximum error is dramatically reduced by a half by increasing the BPO to 8 in both directions. When the BPO is increased to 10, the maximum error is reduced to about  $6 \times 10^{-3}$  metre, about  $4 \times 10^{-4}$  in non-dimensional form. However, the BPO 12 in both directions only reduced the maximum error to  $5 \times 10^{-3}$  metre, about  $3.5 \times 10^{-4}$  in non-dimensional form. It shows that the fitting accuracy could be improved by increasing the

Bernstein polynomial order. However, after using 10<sup>th</sup> order Bernstein polynomials, the fitting error could not be reduced significantly. Therefore, the BPO could be selected between 8 and 10 when using the CST method to represent belly-fairing.

#### **4.5 CST parameterisation for the nacelle**

The nacelle could be defined in two ways: one is to define the longitudinal profile for crown line, maximum half-breadth, keel line and then distribute these profiles circumferentially around the longitudinal axis; and the other is to define the cross-section profiles and then distribute the profiles along the longitudinal axis. In this work, the second method is employed to keep the consistency with previous components.

Since Kulfan (2006 to 2010) has demonstrated in detail the CST parameterisation methods for representing the nacelle, it will not be repeated here. However, in Kulfan's paper the inlet of the nacelle is defined using four parameters, namely throat station, throat area, end of inlet station and end of inlet area. However, the CST methods only have leading edge radius, and lack the intuitive control for other parameters, such as throat station, throat area, etc., which are not explicitly involved in CST methods. These parameters are very important for the nacelle design, and are required to be constrained directly. Therefore, the theory of PARSEC parameterisation is studied and employed to represent the longitudinal profile of the nacelle inlet. Figure 4.60 shows the nacelle inlet definition using the PARSEC approach.

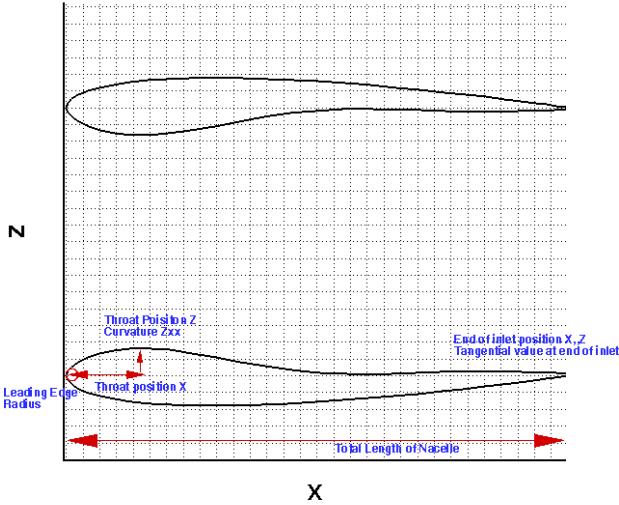


Figure 4.60 The nacelle inlet using PARSEC intuitive parameters

Similar to PARSEC, 12 design variables are introduced to represent the inlet longitudinal profile. These variables are: the leading edge radius on upper lobe and lower lobe ( $R_{le,upper}$ ,  $R_{le,lower}$ ), upper throat position ( $X_{UP}$ ,  $Z_{UP}$ ), upper throat curvature( $Z_{XXUP}$ ), lower throat position( $X_{LO}$ ,  $Z_{LO}$ ), lower throat curvature ( $Z_{XXLO}$ ), upper and lower end of inlet station ( $X_{End, up}$ ,  $Z_{End, up}$ ,  $X_{End, lo}$ ,  $Z_{End, lo}$  ), and tangential value of upper and lower end of inlet ( $T_{End, up}$  and  $T_{End, lo}$ ). A sixth order polynomial, as follows, was used to control the curve:

$$\xi_{PARSEC,inlet} = \sum_{n=1}^6 a_n \cdot \psi^{n-\frac{1}{2}} \quad 4.67$$

In order to determine the coefficients of polynomial, the following equation system is to be solved:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ \psi_{End}^{\frac{1}{2}} & \psi_{End}^{\frac{3}{2}} & \psi_{End}^{\frac{5}{2}} & \psi_{End}^{\frac{7}{2}} & \psi_{End}^{\frac{9}{2}} & \psi_{End}^{\frac{11}{2}} \\ \psi_{UP}^{\frac{1}{2}} & \psi_{UP}^{\frac{3}{2}} & \psi_{UP}^{\frac{5}{2}} & \psi_{UP}^{\frac{7}{2}} & \psi_{UP}^{\frac{9}{2}} & \psi_{UP}^{\frac{11}{2}} \\ \frac{1}{2}\psi_{End}^{-\frac{1}{2}} & \frac{3}{2}\psi_{End}^{\frac{1}{2}} & \frac{5}{2}\psi_{End}^{\frac{3}{2}} & \frac{7}{2}\psi_{End}^{\frac{5}{2}} & \frac{9}{2}\psi_{End}^{\frac{7}{2}} & \frac{11}{2}\psi_{End}^{\frac{9}{2}} \\ \frac{1}{2}\psi_{UP}^{-\frac{1}{2}} & \frac{3}{2}\psi_{UP}^{\frac{1}{2}} & \frac{5}{2}\psi_{UP}^{\frac{3}{2}} & \frac{7}{2}\psi_{UP}^{\frac{5}{2}} & \frac{9}{2}\psi_{UP}^{\frac{7}{2}} & \frac{11}{2}\psi_{UP}^{\frac{9}{2}} \\ -\frac{1}{4}\psi_{up}^{-\frac{3}{2}} & \frac{3}{4}\psi_{up}^{-\frac{1}{2}} & \frac{15}{4}\psi_{up}^{\frac{1}{2}} & \frac{35}{4}\psi_{up}^{\frac{3}{2}} & \frac{53}{4}\psi_{up}^{\frac{5}{2}} & \frac{99}{4}\psi_{up}^{\frac{7}{2}} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{bmatrix} = \begin{bmatrix} \sqrt{2R_{le, upper}} \\ \xi_{TE} \\ \xi_{UP} \\ T_{End,upper} \\ 0 \\ \left[ \frac{\partial^2 \xi}{\partial \psi^2} \right]_{\psi=\psi_{UP}} \end{bmatrix} \quad 4.68$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ \psi_{End}^{\frac{1}{2}} & \psi_{End}^{\frac{3}{2}} & \psi_{End}^{\frac{5}{2}} & \psi_{End}^{\frac{7}{2}} & \psi_{End}^{\frac{9}{2}} & \psi_{End}^{\frac{11}{2}} \\ \psi_{LO}^{\frac{1}{2}} & \psi_{LO}^{\frac{3}{2}} & \psi_{LO}^{\frac{5}{2}} & \psi_{LO}^{\frac{7}{2}} & \psi_{LO}^{\frac{9}{2}} & \psi_{LO}^{\frac{11}{2}} \\ \frac{1}{2}\psi_{End}^{-\frac{1}{2}} & \frac{3}{2}\psi_{End}^{\frac{1}{2}} & \frac{5}{2}\psi_{End}^{\frac{3}{2}} & \frac{7}{2}\psi_{End}^{\frac{5}{2}} & \frac{9}{2}\psi_{End}^{\frac{7}{2}} & \frac{11}{2}\psi_{End}^{\frac{9}{2}} \\ \frac{1}{2}\psi_{LO}^{-\frac{1}{2}} & \frac{3}{2}\psi_{LO}^{\frac{1}{2}} & \frac{5}{2}\psi_{LO}^{\frac{3}{2}} & \frac{7}{2}\psi_{LO}^{\frac{5}{2}} & \frac{9}{2}\psi_{LO}^{\frac{7}{2}} & \frac{11}{2}\psi_{LO}^{\frac{9}{2}} \\ -\frac{1}{4}\psi_{LO}^{-\frac{3}{2}} & \frac{3}{4}\psi_{LO}^{-\frac{1}{2}} & \frac{15}{4}\psi_{LO}^{\frac{1}{2}} & \frac{35}{4}\psi_{LO}^{\frac{3}{2}} & \frac{53}{4}\psi_{LO}^{\frac{5}{2}} & \frac{99}{4}\psi_{LO}^{\frac{7}{2}} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{bmatrix} = \begin{bmatrix} \sqrt{2R_{le, lower}} \\ \xi_{TE} \\ \xi_{UP} \\ T_{End,lower} \\ 0 \\ \left[ \frac{\partial^2 \xi}{\partial \psi^2} \right]_{\psi=\psi_{LO}} \end{bmatrix} \quad 4.69$$

Once the coefficients of equations are solved, the non-dimensional coordinates of the inlet can be established. It is then scaled with respect to the length of nacelle, and translated to the correct position with respect to the leading edge of the outlet cowl of the nacelle. Figure 4.61 shows the CST parametric nacelle model in this work.

$$x = T_{length} \bullet \psi \quad 4.70$$

$$Z_{inlet,upper} = T_{length} \bullet \xi_{up} + Z_{leading,upper} \quad 4.71$$

$$Z_{inlet,lower} = T_{length} \bullet \xi_{lo} + Z_{leading,lower} \quad 4.72$$

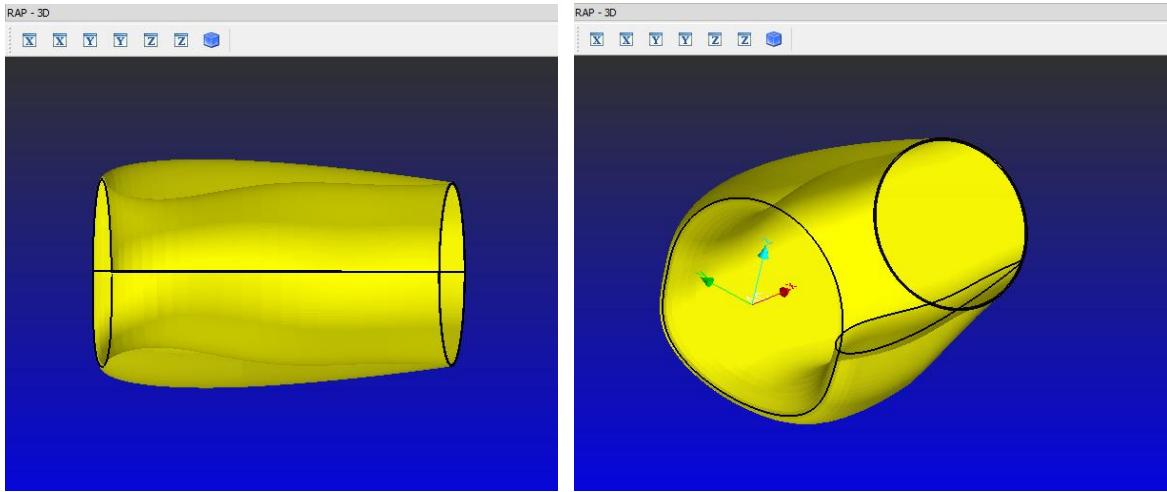


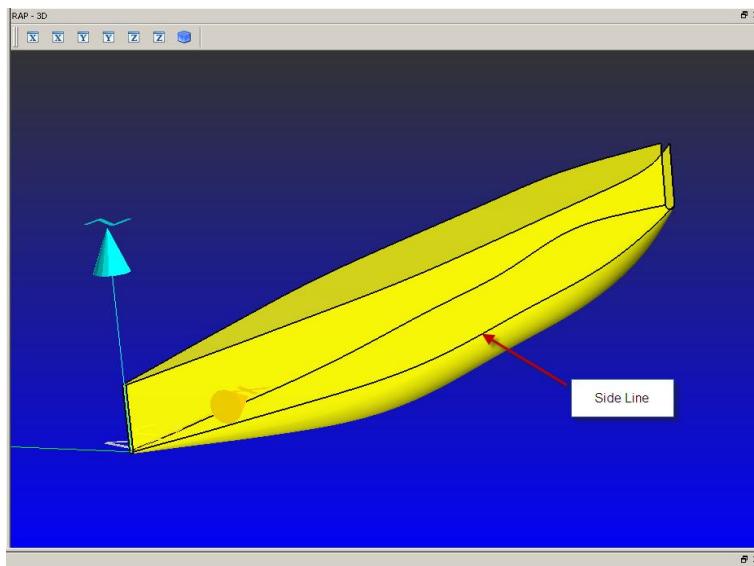
Figure 4.61 The CST parametric nacelle model

#### 4.6 CST parameterisation for flap tracking fairing (FTF) and pylon

The flap track fairing is a pod located at the wing flap lower surface used to cover the flap track and hydraulic equipment and reduce wave drag. The FTF can be decomposed into three parts, namely, a bottom cowl and two straight side walls. The flap track fairing has a symmetric geometry with respect to its own body symmetric plane. Therefore, the bottom cowl could be defined as a type of body cross-section geometry.

The cross-section profile of the bottom cowl of FTF is a half circle represented using the CST class function with unit shape function. The longitudinal distributions of the section profile for keel and side lines of FTF could be represented using a Bernstein polynomial. The mathematical expressions are similar to those of the simplified nose fuselage in Equations 4.40-4.43.

Once the bottom cowl geometry is established, two side walls can be linearly extruded from the side lines. Moreover, although the cross-section of the bottom cowl is a half circle in most FTF cases, a different shape could be applied to the cross-section profile. If needed, the modifications could be applied and could be referenced as a tail cone definition in Equations 4.48-4.54. Figure 4.62 shows the CST parametric FTF.



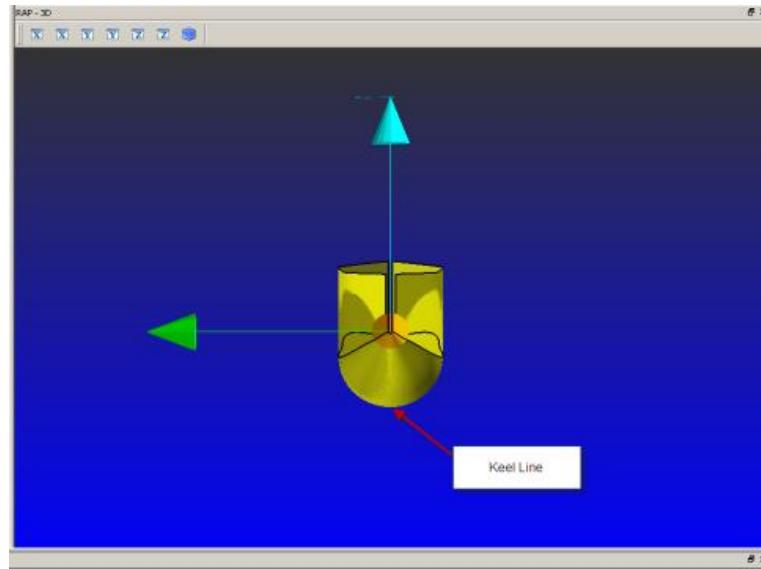


Figure 4.62 The CST parametric model of FTF

The pylon is an important aerodynamic component which is located on the wing to carry the aero-engine for the civil passenger transport aircraft. Pylon design and optimisation is a crucial problem for modern transport aircraft. A well designed pylon could provide significant interference drag reduction between nacelle and wing.

A practical pylon CAD model is shown in Figure 4.63, which illustrates the complex shape of a pylon. The pylon can be decomposed as four basic sub-components, namely, pylon fairing, root fairing and two side walls.

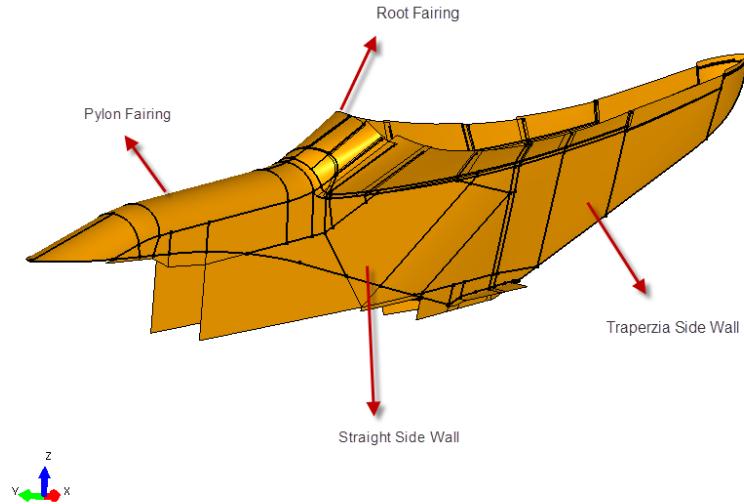


Figure 4.63 A pylon CAD model

The pylon fairing is a part to link the nacelle upper lobe and wing leading edge, which has a significant contribution for reduction of aerodynamic drag. The root fairing is located at the interface between pylon and wing lower surface, and which is designed to reduce the interference drag between pylon and wing. Due to structural requirements, a straight side wall is employed to link the pylon fairing to the nacelle. A trapezium side wall is applied to make the flow converge smoothly at the trailing edge of the pylon behind the straight side wall.

Because the nacelle/wing interference drag contributes mainly to the aerodynamic drag in this area and the pylon root fairing is relatively complex, it is temporarily abandoned to involve the root fairing in parameterisation, and assumes pylon is a symmetric geometry. Therefore, a simplified pylon is shown schematically in the following Figure 4.64.

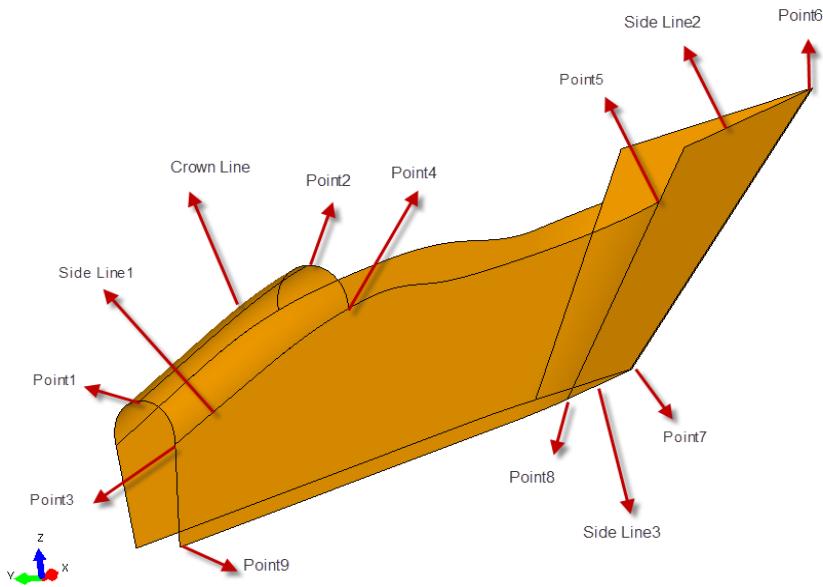


Figure 4.64 A simplified pylon model without root fairing

As the above figure shows, the pylon could be described as having 9 points and 4 lines. Points 1 and 2 are the vertices of the crown line of the pylon fairing. These two points correspond to the pylon relative position between wing and nacelle. Similarly, points 3 and 4 are the vertices of the side line 1 of the pylon fairing. The side line 1 in Figure 4.64 also controls the curve on the upper side of the straight side wall surface. Point 5 is one of the vertices of the straight side wall. Point 6 is the vertex of the pylon on the upper side of the trapezium side wall. The curve between points 5 and 6 is notated as side line 2 which controls the upper side of the trapezium side wall. Point 7 is the ending vertex of the pylon on the lower side of the trapezium side wall. Point 8 is the starting vertex of the lower side of the trapezium side wall. The curve between points 8 and 7 is notated as side line 3 which controls the lower side of the trapezium side wall. Point 9 is the starting vertex of the lower side of the straight side wall. The Point 8 and 9 are then linearly connected to setup the bottom bounding straight line.

Once these points and lines are set up, the three main surfaces could be generated. The CST parameterisation could be employed for the pylon fairing in similar way to the FTF bottom cowl. Similarly, the class parameter of the cross-section profile of the pylon

fairing is a variable. The straight and trapezium side walls could be generated by linking the upper and lower side lines linearly. The CST parametric pylon could be established and is shown below in Figure 4.63.

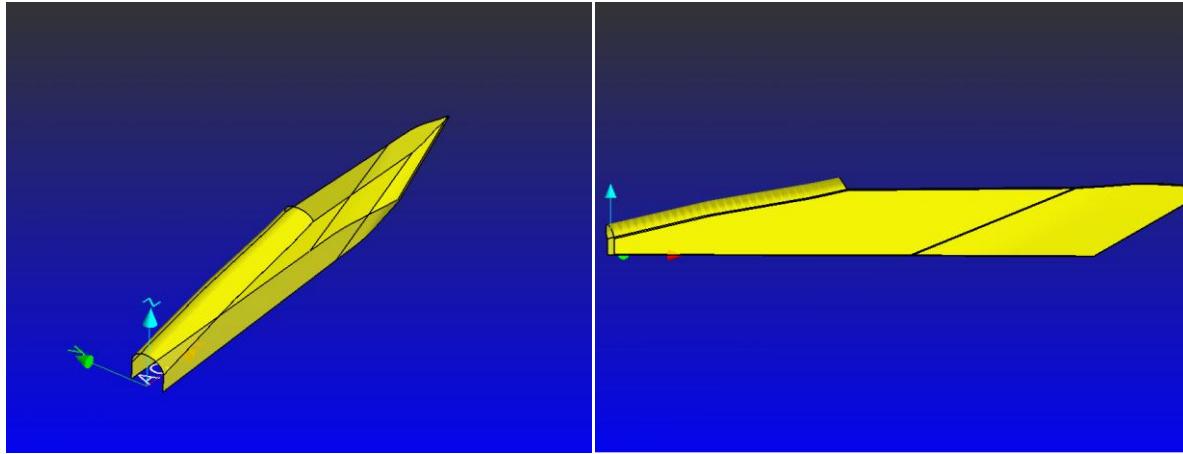


Figure 4.65 CST parametric model of a simplified pylon

For the inverse fitting, the procedure is to extract these points and lines from original surface data and identify the type of cross-section profile of the pylon fairing. Following this, the curve fitting techniques could be applied to represent these side lines. Finally, the side wall surfaces could be generated based these side lines.

Although most types of pylon in civil transport aircraft could be defined by this procedure, it is still a low-fidelity representation for the pylon. For example, in some practical pylons, the side wall could not be represented as a linear link between upper and lower side lines. Furthermore, the real constraints should be added to ensure no ill-conditioned shape appears during optimisation.

#### **4.7 CST parameterisation for three-dimensional shock bump local modification**

The shock control bump and local modification has been proposed by Fulker *et al.* (1993). The basic idea of a shock control bump is to employ the concave part of the bump upstream before the primitive shock to induce a pre-compression. This pre-compression

thus decreases the Mach number before the primitive shock and reduces the strength of the primitive shock.

The design and optimisation of the 2D shock bump has been widely performed by academic researchers (Zhu 2000; Lutz *et al.* 2004). More details could be found in Wong's thesis (2006). The literature has asserted that the shock control bump is able to provide significant shock wave drag reduction for 2D aerofoils. Qin *et al* (2005) have presented the 3D bump is able to effectively reduce wave drag as well. Later on, Qin *et al.* (2008) have successfully extended the shock control bump to a 3D un-swept NLF wing. The results show that the three-dimensional bump is able to reduce wave drag more than two-dimensional case. Therefore, they have applied the control bumps in a three-dimensional blended-wing-body (Wong *et al.* 2007). In their work, the capability and feasibility of a shock control bump for shock wave drag reduction in a three-dimensional practical case has been proven. In recent years, the shock control bump technique is of more and more interest to industry since it is able to be applied to modify existing configurations and provides significant drag reduction with slight modification.

The design parameterisation used before for shock bump control is generally simple. In Wong's work (2006), the bump is split into a two piece-wise curve with respect to the shock crest position. Each is represented by third-order polynomials, see Equation 4.73 and Figure 4.66:

$$\begin{aligned} z_1(x) &= a_1x^3 + b_1x^2 + c_1x + d_1 & 0 < x < x_{crest} \\ z_2(x) &= a_2x^3 + b_2x^2 + c_2x + d_2 & x_{crest} < x < x_{length} \end{aligned} \quad 4.73$$

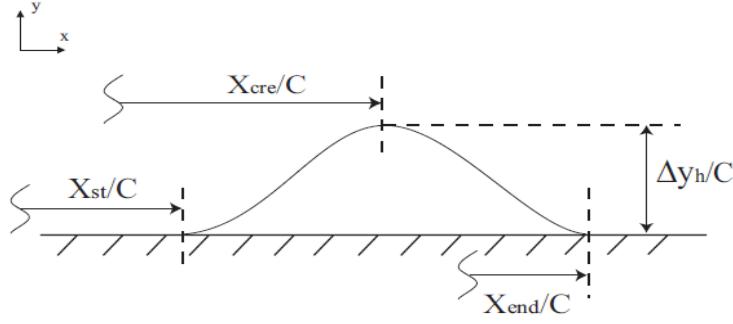


Figure 4.66 Parameterisation for 2D shock control bump using piecewise polynomials  
(Wong 2006)

The boundary condition of this bump is:

$$z_1(0) = 0 \quad 4.74$$

$$z'_1(0) = 0 \quad 4.75$$

$$z_1(x_{crest}) = z_{height} \quad 4.76$$

$$z'_1(x_{crest}) = 0 \quad 4.77$$

$$z_2(x_{crest}) = z_{height} \quad 4.78$$

$$z'_2(x_{crest}) = 0 \quad 4.79$$

$$z_2(x_{length}) = 0 \quad 4.80$$

$$z'_2(x_{length}) = 0 \quad 4.81$$

For this parameterisation, all intuitive parameters, such as bump height  $z_{height}$  and bump crest relative position  $x_{crest}$ , are explicitly presented. However, these bump parameterisation equations only guarantee the first derivative continuity at the start, crest and end position of the bump. For practical industrial manufacture, the second derivative continuity  $C^2$  is required. In order to satisfy  $C^2$  continuity, the order of piecewise polynomials has to be increased to 5. However, higher order polynomials will lead to a high degree-of-freedom and contain more than one peak in the curve; this will cause uncontrollable waviness in the bump. Another parameterisation uses PARSEC bump equations, which have been studied in Chapter 2. The PARSEC bump function has natural  $C^1$  and  $C^2$  continuity at the crest position, and it provides some parameters, such as  $P$ ,  $Q$ ,  $c$ , related to curvature at the start and end position of the bump. However, these

parameters for controlling curvature are not equal to curvature value. The relationship of curvature and parameters is difficult to derive. Therefore, the CST methods have been used to try to represent the shock control bump.

As presented in Chapter 2, the CST methods have two parts: the class function and the shape function. Therefore, if the class parameters  $N_1$  and  $N_2$  are set to 3, and the shape function  $S(\psi) = 1$  is used, as in Equation 4.82, a bump like curve with 1<sup>st</sup> and 2<sup>nd</sup> derivatives of zero at start and end is obtained. The curve and 1<sup>st</sup> and 2<sup>nd</sup> derivative distribution are shown in Figure 4.67 to 4.69.

$$\xi(\psi) = \psi^{3.0} \cdot (1-\psi)^{3.0} \quad 4.82$$

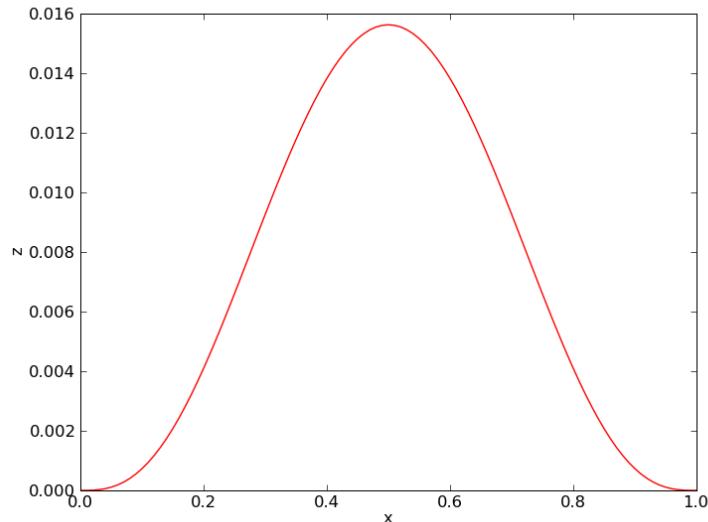


Figure 4.67 The bump curve using the CST methods

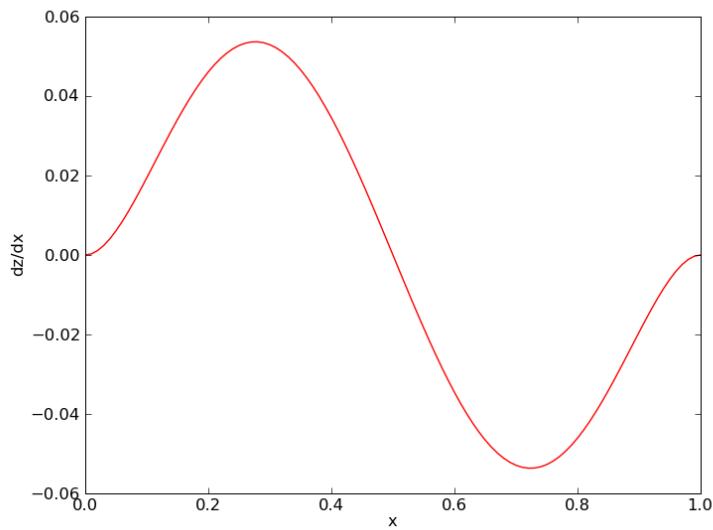


Figure 4.68 The 1<sup>st</sup> derivative distribution of bump curve using the CST method

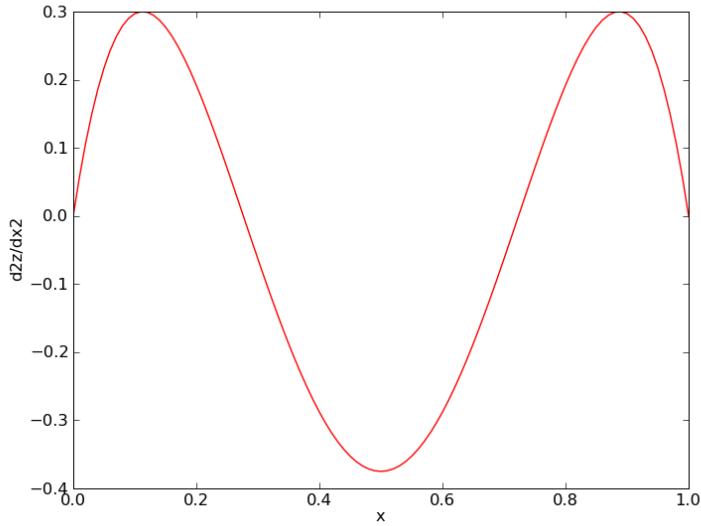


Figure 4.69 The 2<sup>nd</sup> derivative distribution of bump curve using the CST method

The above figures clearly show that the geometry, 1<sup>st</sup> and 2<sup>nd</sup> derivatives are all zero at the start and end position of the bump. In addition, because the class parameters are exponential parameters of the class function, if  $N_1$  and  $N_2$  are set to 3, the bump peak value reduces to 1/64. Therefore, it is better to multiple by 64 in the CST equation to amplify the peak value back to 1. This would be convenient for the user when setting up

their design parameter range. For modification of the CST bump, the shape function is involved. The full description of the CST bump is:

$$\xi(\psi) = 64 \cdot C_{3,0}^{3,0}(\psi) \bullet \sum_i^n A_i \bullet S_i(\psi) \quad 4.83$$

$$x = \psi \cdot x_{length} \quad 4.84$$

$$z = \xi \cdot x_{length} \quad 4.85$$

The two-dimensional bump could be easily extended into a three-dimensional bump using a similar transformation to that for the wing. Finally, the boundary of the 3D bump may not be strictly square. The sweep angle of the leading and trailing edges of the bump may be needed when deployed on the wing to match the flow direction due to the wing sweep. The definition of a three-dimensional bump with sweep angle is shown in the following equations:

$$\xi(\psi, \eta) = 64 \cdot C_{3,0}^{3,0}(\psi) \bullet \sum_i^{Nx} \sum_j^{Ny} [Bu_{i,j} \bullet Sy_j(\eta) \bullet Sx_i(\psi)] \cdot H(\eta) \quad 4.86$$

where

$$H(\eta) = 64 \cdot C_{3,0}^{3,0}(\eta) \bullet \sum_i^n A_i \bullet S_i(\eta) \quad 4.87$$

$$x = \psi \cdot x_{length}(\eta) + \tan(\alpha_{leading}) \cdot \eta \cdot y_{width} \quad 4.88$$

$$y = \eta \cdot y_{width} \quad 4.89$$

$$z = \xi \cdot x_{length}(\eta) \quad 4.90$$

$$x_{length}(\eta) = L_{root} + (\tan(\alpha_{trailing}) - \tan(\alpha_{leading})) \cdot \eta \cdot y_{width} \quad 4.91$$

where  $\alpha_{leading}$  and  $\alpha_{trailing}$  are the leading edge and trailing edge sweep angles of the bump.  $L_{root}$  is the bump length at the start boundary side  $\eta = 0$ . This can provide higher flexibility of a local bump, and generate symmetric or asymmetric bumps in three-dimensional space. The orders of *BPOX* and *BPOY* are recommended to be below 4,

since the Bernstein polynomials are not orthogonal. The performance of the CST bump will be studied in a later optimisation test.

#### 4.8 Calculation of intersection line

The intersection line of two surfaces is an important issue in complex geometric modelling. CFD computation requires watertight geometric modelling for meshing. Furthermore, the aerodynamic flow is very sensitive to the geometry. Therefore, it requires that the surface intersection line must be calculated with high accuracy, robustness and efficiency.

Nowadays, most CAD software packages are able to calculate the surface intersection line. The reason that we still present an intersection line calculation method is based on the following reasons.

First, the CST parameterisation method is a relatively new method which has not been used in any CAD software. Therefore using CAD software or library, an extra process is needed to import the CST parametric model into CAD software and export to CFD meshing after the intersection line has been calculated. Second, if CAD software or library is involved, the challenge for the CAD based parameterisation returns, especially regarding license issue. The third is that the accuracy and robustness of CAD software are out of the user's control. For example, many of the commercial CAD systems employ the polyhedral approximation for the intersection line (Krishnan and Manocha 1997). This method is normally not accurate enough. For example in industrial surface design and CFD processes, extra work between the CAD surface and CFD mesh is normally required to clean up the incorrect intersection line and connectivity and generate watertight surface. These processes are all subject to human intervention and it is desired that they do not happen during an automatic optimisation process. Therefore, an efficient, accurate and robust algorithm is required for calculating the intersection for the CST parametric model.

In general, the methods for calculating the surface intersection line are classified into analytical methods and numerical methods (Li *et al.* 2004; Krishnan and Manocha 1997). Analytical methods are used to find the explicit exact mathematical function to represent the intersections curves (Chandru and Kochar 1987; Sarraga 1983; Heo *et al.* 1999). Analytical methods are exact and robust; however, the exact mathematical function is derived from a mathematic representation of two intersected surfaces. Therefore, it depends on the complexity of intersected surfaces. In general, the surfaces in aircraft are complicated, and so analytical derivation of the intersection is very complicated and may not exist. Therefore, numerical methods are preferred. Patrikalakis (1993) classified numerical methods into three main categories: subdivision methods, lattice evaluations, and marching methods.

Subdivision methods decompose the geometry recursively into much simpler, similar problems which can be solved easily (Hohmeyer 1992; Aziz and Bata 1990). In general, geometries are subdivided continuously until a desired level, such as the flatness sub-piece with linearity of each edge, is attained. Then, the corresponding intersection is calculated from each sub-piece. This method is dependent on the properties of control polytopes of geometries (Lane and Riesenfeld 1980; Lasser 1986). These methods are generally robust; however, the efficiency is dependent on the requirement for accuracy. If high accuracy is required, more steps of the subdivision decomposition will be carried out, which leads to very slow implementation and large data storage.

The lattice evaluation methods take a set of curves from one of the two intersected surfaces. The problem of surface to surface intersection is then degenerated to a lower complexity curve to surface intersection problem (Patrikalakis 1993; Limaiem and Trochu 1995; Rossignac and Requicha 1987). Finally, the discrete points calculated from the curve/surface intersection will be connected to compose intersection curves. This method is simple to implement since it avoids solving a large number of uncertain non-linear equations. However, the discrete step size for decomposing a surface to curves is hard to decide on, and it is difficult to discover the intersection curves of small loops and singular points, especially in the complex case with multiple intersection curves.

The marching methods generally have two steps: finding starting-points on each intersection line and tracing the intersection curve from a starting-point along a certain direction which is determined by local geometry properties (Baja and Hoffmann 1988). When both of the intersected surfaces can be represented mathematically, the intersection line between two surfaces can be described as an algebraic set from a surface representation function where the intersection line is a curve with zero distance between two surfaces. The marching methods have been widely used in calculating the intersection line (Aziz and Bata 1990; Patrikalakis 1993; Wu and Andrade 1999; Krishnan and Manocha 1997; Li *et al.* 2004). This method is high efficient, and the accuracy can be controlled by the user. The robustness depends on two points. First, in the high free-form surfaces, the number of intersection lines of two surfaces could be more than one, and include an open component and loop, as in Figure 4.70. Therefore, it requires a robust method to find starting-points for the intersection lines. Second, it requires a good scheme to perform the marching process. Newton iterations and local geometry differentiation information are normally employed in marching processing.

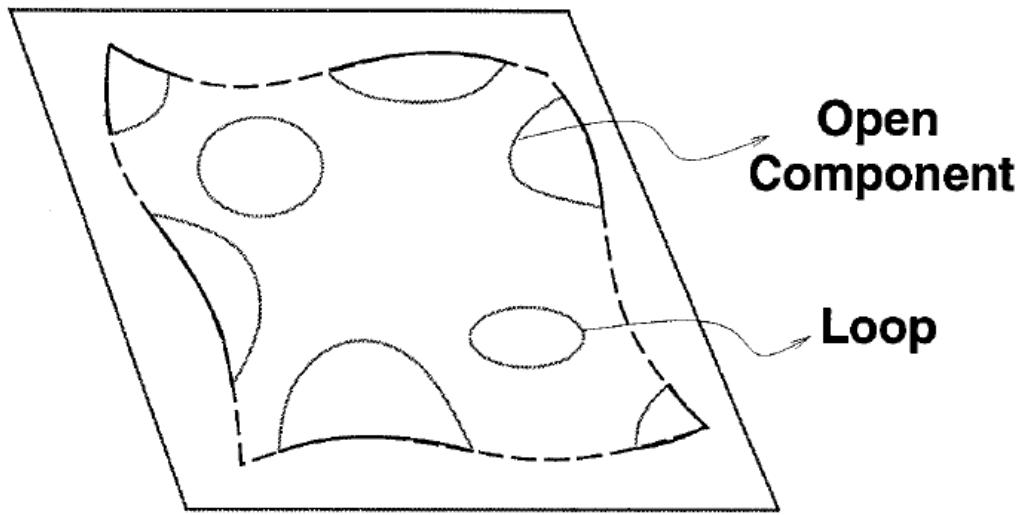


Figure 4.70 The intersection lines of two high free-form surfaces

In calculating the intersection line of the parametric model for the CFD process, the high precision is input first since the CFD mesh, especially in the boundary layer region, is

very small. Therefore, the subdivision method is not selected because of its drawback. The lattice evaluation methods could be useful for certain cases. For example, in the wing/belly-fairing intersection, if the non-dimensional coordinate  $\psi$  in Equation 4.5 is fixed, a line could be obtained along the spanwise direction. The intersection point of this spanwise line and the belly-fairing surface could be easily searched out as an optimisation issue. Then, the next intersection point is obtained by changing the non-dimensional coordinate  $\psi$ , and the process is repeated. This is shown in Figure 4.71.

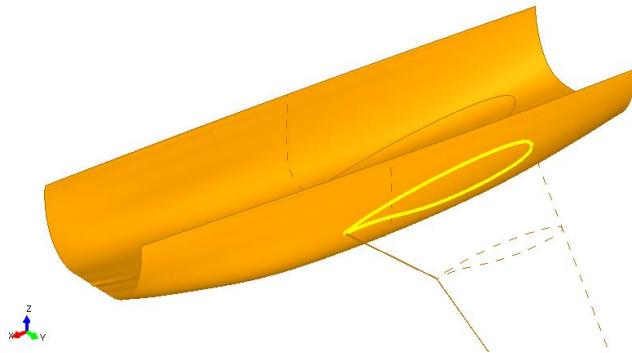


Figure 4.71 The intersection line between wing and belly-fairing

However, in other cases, for example the belly-fairing/fuselage intersection, the loop intersection line occurs and the lattice evaluation is not suitable. Because multiple numbers of intersections between two aerodynamic component surfaces occurs rarely, and the CST parametric geometric model is fully described in algebraic mathematic equations, so it is fully differentiable. Therefore, the marching method is selected to calculate the intersection line between different CST parametric models.

The marching algorithm employed here follows Huang and Zhu's (1997) work. In their work, the Newton-Raphson iteration is employed to calculate the local intersection point. The process for this method is summarised below:

- 1) Calculate the exact starting-point for the intersection line,
- 2) Approximate the next marching point location based on local surface properties,
- 3) Employ the Newton-Raphson iteration convergence to 'exact' local intersect point,

- 4) Detect if the important points, such as turning point and cusp point, exist and if so calculate them,
- 5) Repeat steps 2 to 4 until the intersection point returns to the starting-point or hits the boundary,
- 6) Employ the NURBS curve to fit all discrete intersect points and obtain the mathematic NURBS representation of the intersection line (Piegl and Tiller 1997).

For example, for the CST parametric mid-part fuselage model in vector field  $F(\psi_f, \eta_f)$  and the CST parametric belly-fairing in vector field  $G(\psi_b, \eta_b)$ , the intersection line  $X(t)$  could be described as:

$$X(t) = F(\psi_f, \eta_f) - G(\psi_b, \eta_b) = 0 \quad 4.92$$

In this case, the intersection points of the keel line of the fuselage and belly-fairing are used as starting-points. The starting-points can be calculated exactly because the keel line of the fuselage is a straight line and the keel line of the belly-fairing is a low order polynomial, and both are on the x-z plane. This could be easily solved as a line-line intersection issue. For other cases, for example the wing and belly-fairing, the lattice evaluation method could be employed to calculate the starting-points. Once the starting-point is selected, non-dimensional coordinates on each surface are calculated and notated as  $F_0(\psi_{f,start}, \eta_{f,start})$  and  $G_0(\psi_{b,start}, \eta_{b,start})$ .

Step 2 is to approximate the next marching point position. To approximate this, the direction and step length need to be determined. The direction is defined by the local differential geometry. The step length is crucial because a large step length may lead to an incorrect approximated intersection point and the Newton-Raphson iteration in step 3 may fail to converge, but the small step length could cost more computational time and decrease the efficiency. Huang and Zhu's (1997) suggested the step length can be specified manually or calculated based on local approximated curvature.

At first, the surface units normal  $n_f$  at the fuselage  $F$  ( $\psi_{f,i}, \eta_{f,i}$ ) and  $n_g$  at the belly-fairing  $G$  ( $\psi_{b,i}, \eta_{b,i}$ ) are calculated as:

$$n_f = \frac{\frac{\partial F(\psi_f, \eta_f)}{\partial \psi_f} \Big|_{\substack{\psi_f=\psi_{f,i} \\ \eta_f=\eta_{f,i}}} \times \frac{\partial F(\psi_f, \eta_f)}{\partial \psi_f} \Big|_{\substack{\psi_f=\psi_{f,i} \\ \eta_f=\eta_{f,i}}}}{\left\| \frac{\partial F(\psi_f, \eta_f)}{\partial \psi_f} \Big|_{\substack{\psi_f=\psi_{f,i} \\ \eta_f=\eta_{f,i}}} \times \frac{\partial F(\psi_f, \eta_f)}{\partial \psi_f} \Big|_{\substack{\psi_f=\psi_{f,i} \\ \eta_f=\eta_{f,i}}} \right\|} \quad 4.93$$

$$n_g = \frac{\frac{\partial G(\psi_g, \eta_g)}{\partial \psi_g} \Big|_{\substack{\psi_g=\psi_{g,i} \\ \eta_g=\eta_{g,i}}} \times \frac{\partial G(\psi_g, \eta_g)}{\partial \psi_g} \Big|_{\substack{\psi_g=\psi_{g,i} \\ \eta_g=\eta_{g,i}}}}{\left\| \frac{\partial G(\psi_g, \eta_g)}{\partial \psi_g} \Big|_{\substack{\psi_g=\psi_{g,i} \\ \eta_g=\eta_{g,i}}} \times \frac{\partial G(\psi_g, \eta_g)}{\partial \psi_g} \Big|_{\substack{\psi_g=\psi_{g,i} \\ \eta_g=\eta_{g,i}}} \right\|} \quad 4.94$$

Where, the partial differentiation of geometry of fuselage and belly-fairing could found in appendix D. The approximated marching direction  $T$  is then calculated by the cross product of two surface normals.

$$T = n_f \times n_g \quad 4.95$$

Once the marching step direction is obtained, the step length  $d$  is evaluated based on the approximate local curvature and chord tolerance  $\varepsilon$ . The definition is shown in the following equations and Figure 4.72.

$$d = R \cdot \tan\theta \quad 4.96$$

where

$$\theta = 2\arccos(1 - \varepsilon/R) \quad 4.97$$

$$R = \frac{\|P_i - P_{i-1}\|}{\arccos(T_i \cdot T_{i-1})} \quad 4.98$$

where,  $P_i$  and  $P_{i-1}$  are the current intersection point and previous intersection point, respectively,  $T_i$  and  $T_{i-1}$  are the unit marching direction vectors at the current and previous intersect points. At the first iteration, because  $P_i$  is the starting-point and there is as yet no  $P_{i-1}$ , therefore a specified step length is given

by the user.

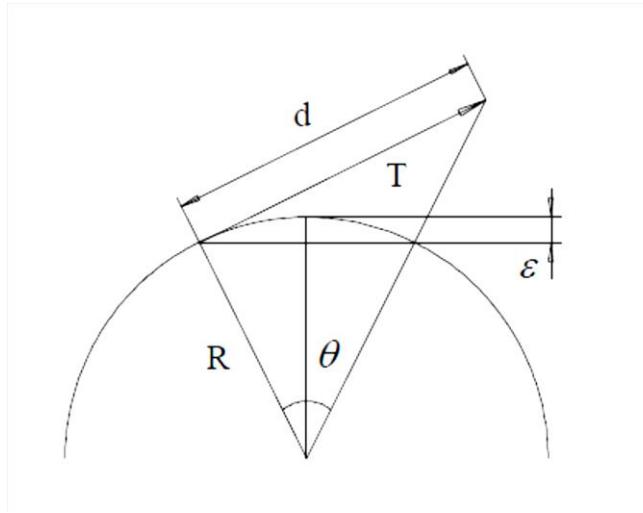


Figure 4.72 The approximation of step length

Finally, the next approximate intersection point location for fuselage  $F$  ( $\psi'_{f,i+1}, \eta'_{f,i+1}$ ) and belly-fairing  $G$  ( $\psi'_{b,i+1}, \eta'_{b,i+1}$ ) is calculated by solving following equation:

$$\begin{bmatrix}
 \frac{\partial F(\psi_f, \eta_f)}{\partial \psi_f} \Big|_{\substack{\psi_f=\psi_{f,i} \\ \eta_f=\eta_{f,i}}} & \frac{\partial F(\psi_f, \eta_f)}{\partial \psi_f} \Big|_{\substack{\psi_f=\psi_{f,i} \\ \eta_f=\eta_{f,i}}} & -\frac{\partial G(\psi_g, \eta_g)}{\partial \psi_g} \Big|_{\substack{\psi_g=\psi_{g,i} \\ \eta_g=\eta_{g,i}}} & -\frac{\partial G(\psi_g, \eta_g)}{\partial \psi_g} \Big|_{\substack{\psi_g=\psi_{g,i} \\ \eta_g=\eta_{g,i}}} \\
 T \cdot \frac{\partial F(\psi_f, \eta_f)}{\partial \psi_f} \Big|_{\substack{\psi_f=\psi_{f,i} \\ \eta_f=\eta_{f,i}}} & T \cdot \frac{\partial F(\psi_f, \eta_f)}{\partial \psi_f} \Big|_{\substack{\psi_f=\psi_{f,i} \\ \eta_f=\eta_{f,i}}} & 0 & 0
 \end{bmatrix} \cdot \begin{bmatrix} \psi'_{f,i+1} - \psi_{f,i} \\ \eta'_{f,i+1} - \eta_{f,i} \\ \psi'_{g,i+1} - \psi_{g,i} \\ \eta'_{g,i+1} - \eta_{g,i} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ d \end{bmatrix} \quad 4.99$$

where, at the starting-point:

$$\psi_{f,i} = \psi_{f,start}$$

$$\eta_{f,i} = \eta_{f,start}$$

$$\psi_{g,i} = \psi_{g,start} \quad 4.100$$

$$\eta_{g,i} = \eta_{g,start}$$

After the approximate marching location is obtained, the Newton-Raphson iteration is employed to solve Equation 4.92 with initial condition  $\psi_{f,i+1}, \eta_{f,i+1}$  and  $\psi_{b,i+1}, \eta_{b,i+1}$  and converging to a local ‘exact’ intersection point. The Newton-Raphson iterative equation for this case is shown below:

$$\begin{aligned} & \left[ \begin{array}{c} \frac{\partial F(\psi_f, \eta_f)}{\partial \psi_f} \Big|_{\substack{\psi_f=\psi_{f,k} \\ \eta_f=\eta_{f,k}}} \quad \frac{\partial F(\psi_f, \eta_f)}{\partial \psi_f} \Big|_{\substack{\psi_f=\psi_{f,k} \\ \eta_f=\eta_{f,k}}} \quad - \frac{\partial G(\psi_g, \eta_g)}{\partial \psi_g} \Big|_{\substack{\psi_g=\psi_{g,k} \\ \eta_g=\eta_{g,k}}} \quad - \frac{\partial G(\psi_g, \eta_g)}{\partial \psi_g} \Big|_{\substack{\psi_g=\psi_{g,k} \\ \eta_g=\eta_{g,k}}} \\ n_2 \cdot \frac{\partial F(\psi_f, \eta_f)}{\partial \psi_f} \Big|_{\substack{\psi_f=\psi_{f,k} \\ \eta_f=\eta_{f,k}}} \quad n_2 \cdot \frac{\partial F(\psi_f, \eta_f)}{\partial \psi_f} \Big|_{\substack{\psi_f=\psi_{f,k} \\ \eta_f=\eta_{f,k}}} \quad 0 \quad 0 \end{array} \right] \begin{bmatrix} \psi_{f,k+1} - \psi_{f,k} \\ \eta_{f,k+1} - \eta_{f,k} \\ \psi_{g,k+1} - \psi_{g,k} \\ \eta_{g,k+1} - \eta_{g,k} \end{bmatrix} \quad 4.101 \\ &= \begin{bmatrix} G(\psi_{g,k}, \eta_{g,k}) - F(\psi_{f,k}, \eta_{f,k}) \\ 0.5n_2 \cdot (G(\psi_{g,k}, \eta_{g,k}) - F(\psi_{f,k}, \eta_{f,k})) \end{bmatrix} \end{aligned}$$

where, the initial condition is defined as:

$$\begin{aligned} \psi_{f,0} &= \psi'_{f,i+1} \\ \eta_{f,0} &= \eta'_{f,i+1} \\ \psi_{g,0} &= \psi'_{g,i+1} \quad 4.102 \\ \eta_{g,0} &= \eta'_{g,i+1} \end{aligned}$$

The Newton-Raphson equation is solved iteratively until  $F(\psi_{f,k+1}, \eta_{f,k+1}) - G(\psi_{g,k+1}, \eta_{g,k+1}) < tolerance$ . Then, the intersection point at  $P_{i+1}$  is employed either  $P_{i+1} = F(\psi_{f,k+1}, \eta_{f,k+1})$  or  $P_{i+1} = G(\psi_{g,k+1}, \eta_{g,k+1})$ .

In order to increase the robustness and local convergence efficient, the Newton-Raphson iteration equation could be solved by fixing a parameter to be constant. For example, if  $\psi_{f,k} = \psi'_{f,i+1}$  is fixed, the Newton-Raphson iteration equation is written as:

$$\begin{aligned} & \left[ \begin{array}{cc|c} \frac{\partial F(\psi_f, \eta_f)}{\partial \psi_f} & \frac{\partial F(\psi_f, \eta_f)}{\partial \psi_f} & -\frac{\partial G(\psi_g, \eta_g)}{\partial \psi_g} \\ \eta_f = \eta_{f,k} & \eta_f = \eta_{f,k} & \eta_g = \eta_{g,k} \\ 1 & 0 & 0 \end{array} \right] \cdot \begin{bmatrix} \psi'_{f,i+1} - \psi_{f,k} \\ \eta_{f,k+1} - \eta_{f,k} \\ \psi'_{g,k+1} - \psi_{g,k} \\ \eta_{g,k+1} - \eta_{g,k} \end{bmatrix} \quad 4.103 \\ & = \begin{bmatrix} G(\psi_{g,k}, \eta_{g,k}) - F(\psi_{f,k}, \eta_{f,k}) \\ \psi'_{f,i+1} - \psi_{f,k} \end{bmatrix} \end{aligned}$$

In the intersection line, the important points such as turning point and cusp point are crucial for geometry topology, see Figure 4.73.

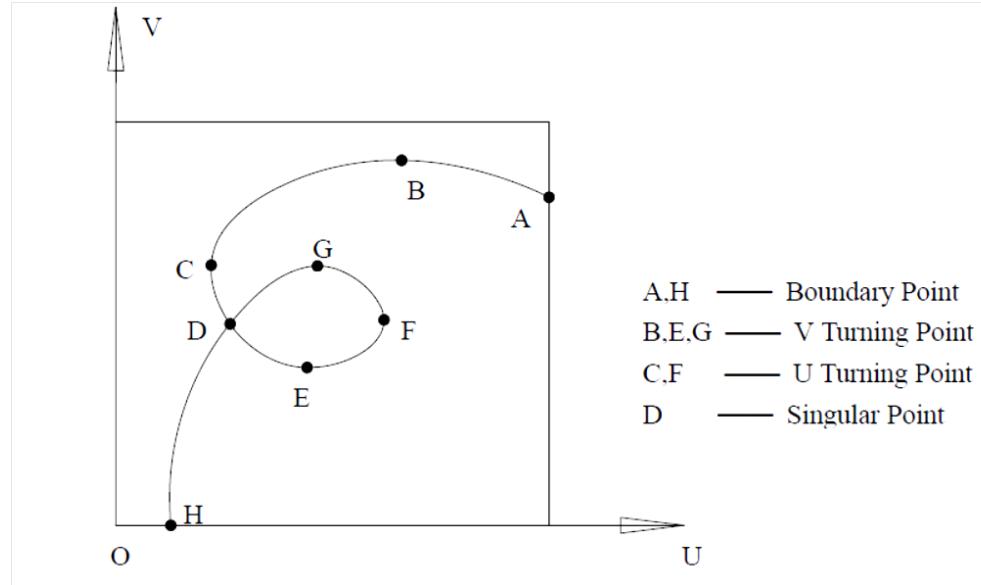


Figure 4.73 The important points on intersection line (Huang and Zhu 1997)

In order to identify if there is an important point between the current intersection point  $P_{i+1}$  and the previous intersection point  $P_i$ , the following conditions are applied:

$$C_i^0 = n_g \left|_{\psi_g = \psi_{g,i}} \cdot \frac{\partial F(\psi_f, \eta_f)}{\partial \psi_f} \right|_{\eta_g = \eta_{g,i}} \quad C_i^1 = n_g \left|_{\psi_g = \psi_{g,i}} \cdot \frac{\partial F(\psi_f, \eta_f)}{\partial \eta_f} \right|_{\psi_f = \psi_{f,i}} \quad 4.104$$

$$C_i^2 = n_f \left|_{\begin{array}{l} \psi_f = \psi_{f,i} \\ \eta_f = \eta_{f,i} \\ \eta_g = \eta_{g,i} \end{array}} \cdot \frac{\partial G(\psi_g, \eta_g)}{\partial \psi_g} \right| \quad C_i^3 = n_f \left|_{\begin{array}{l} \psi_f = \psi_{f,i} \\ \eta_f = \eta_{f,i} \\ \eta_g = \eta_{g,i} \end{array}} \cdot \frac{\partial G(\psi_g, \eta_g)}{\partial \eta_g} \right| \quad 4.105$$

$$C_{i+1}^0 = n_g \left|_{\begin{array}{l} \psi_g = \psi_{g,i+1} \\ \eta_g = \eta_{g,i+1} \\ \eta_f = \eta_{f,i+1} \end{array}} \cdot \frac{\partial F(\psi_f, \eta_f)}{\partial \psi_f} \right| \quad C_{i+1}^1 = n_g \left|_{\begin{array}{l} \psi_g = \psi_{g,i+1} \\ \eta_g = \eta_{g,i+1} \\ \eta_f = \eta_{f,i+1} \end{array}} \cdot \frac{\partial F(\psi_f, \eta_f)}{\partial \eta_f} \right| \quad 4.106$$

$$C_{i+1}^2 = n_f \left|_{\begin{array}{l} \psi_f = \psi_{f,i+1} \\ \eta_f = \eta_{f,i+1} \\ \eta_g = \eta_{g,i+1} \end{array}} \cdot \frac{\partial G(\psi_g, \eta_g)}{\partial \psi_g} \right| \quad C_{i+1}^3 = n_f \left|_{\begin{array}{l} \psi_f = \psi_{f,i+1} \\ \eta_f = \eta_{f,i+1} \\ \eta_g = \eta_{g,i+1} \end{array}} \cdot \frac{\partial G(\psi_g, \eta_g)}{\partial \eta_g} \right| \quad 4.107$$

If  $C_{i+1}^0 C_i^0 < 0$  or  $C_{i+1}^1 C_i^1 < 0$  or  $C_{i+1}^2 C_i^2 < 0$  or  $C_{i+1}^3 C_i^3 < 0$  is detected, there is a turning point or cusp point between  $P_{i+1}$  and  $P_i$ . If  $C_{i+1}^0 C_i^0 < 0$  and  $C_{i+1}^1 C_i^1 < 0$  and  $C_{i+1}^2 C_i^2 < 0$  and  $C_{i+1}^3 C_i^3 < 0$  are all satisfied, there is a cusp point between  $P_{i+1}$  and  $P_i$ . In the regular aircraft component, the cusp conditions do not normally occur. If the turning point is detected between  $P_{i+1}$  and  $P_i$ , the interval should be subdivided gradually to search out the turning point. Finally, an NURBS curve representation is employed to fit the discrete intersection points. Therefore, NURBS is able to redistribute the number of points on intersection lines and maintain the geometry topology. This is significant for providing information for mesh deformation.

The intersection calculation using marching strategy based on the Newton-Raphson iteration and direct CST local differential has been developed. The tests have shown that the error of the equation  $F(\psi_{f,k+1}, \eta_{f,k+1}) - G(\psi_{b,k+1}, \eta_{b,k+1})$  for each intersection point is less than  $10^{-9}$ , which is much lower than the normal CAD toolbox. Figure 4.74 shows the example of the fuselage and belly-fairing intersection line.

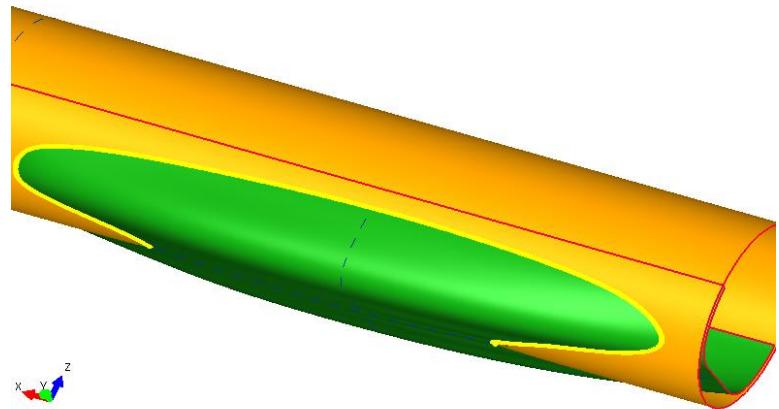


Figure 4.74 The example of intersection line between fuselage and belly-fairing

## Chapter 5 Governing Equation and Numerical Solver

As mentioned at the beginning of this thesis, with the development of computational capability, the CFD technique is playing an increasingly important role in engineering design and it is becoming a standard engineering tool in industry. In numerical optimisation, CFD is employed to calculate the objective and constraints values, such as lift, drag and momentum, for optimisation.

In this thesis, the CFD solver is provided by DLR, so called ‘TAU’, which is an unstructured finite volume Navier-Stokes solver, which – will be only briefly introduced in this chapter. Since TAU has been widely used in industry and research and has been well validated (Gerhold *et al.* 1997; Rudnik *et al.* 2004; Kroll and Fassbender 2005), the comparison of numerical and experimental results will not be repeated here.

For the aerodynamic design of aircraft under subsonic conditions, the fluid is assumed as a continuum in which the fluid quantities vary continuously from one point to another away from the shock wave, and the fluid is compressible where the density is variable rather than constant, and it is assumed to be Newtonian flow where the stress is proportion to local strain rate. This fluid is governed by Navier-Stokes equations. It will be introduced at first in this chapter.

The physical domain is subdivided into finite control volumes by grids. The finite volume method is then applied to grid elements to discrete the governing equations in space, and marches in time. This will generate a system of numerical equations which can be solved with various numerical schemes. The central scheme in spatial discretisation and the implicit scheme in time discretisation will be introduced. One-equation turbulence model equation, Spalart-Allmaras, is employed in this work, and is also presented in this chapter.

## 5.1 Governing equation

The Navier-Stokes equations are generally derived from three conservation laws, which are mass continuity, momentum and energy conservation, applied to an infinitesimally small fluid element. The details of the derivation of these equations have been presented in many literatures (Anderson 2007; White 2008), and therefore they will not be presented in this thesis. The full Navier-Stokes in conservation form is written as below:

$$\frac{\partial \mathbf{W}}{\partial t} + \nabla \cdot [\mathbf{F}^c - \mathbf{F}^v] = \frac{\partial \mathbf{W}}{\partial t} + \mathbf{R}(\mathbf{W}) = 0 \quad 5.1$$

or in integral form

$$\frac{\partial}{\partial t} \iiint_{\Omega} \mathbf{W} d\Omega + \iint_{\partial\Omega} [\mathbf{F}^c - \mathbf{F}^v] \cdot \mathbf{n} \cdot dS = 0 \quad 5.2$$

where  $\mathbf{W}$  is the conserved state vector

$$\mathbf{W} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{bmatrix} \quad 5.3$$

$\rho$  is the fluid density  $u, v, w$  are the velocity component in x,y,z directions in Cartesian coordinates.  $E$  is the total energy per control volume and  $\Omega$  is the integral form for a bounded domain with bound surface  $\partial\Omega$ .  $\mathbf{F}^c$  and  $\mathbf{F}^v$  are the convective and viscous flux tensors and can be written as below with respect to three directions:

$$\mathbf{F}^c = \begin{bmatrix} F_x^c \\ F_y^c \\ F_z^c \end{bmatrix} \quad \mathbf{F}^v = \begin{bmatrix} F_x^v \\ F_y^v \\ F_z^v \end{bmatrix} \quad 5.4$$

where  $F_x^c, F_y^c, F_z^c$  are the convective flux components in x,y,z directions in Cartesian coordinates and  $F_x^v, F_y^v, F_z^v$  are the viscous flux components in x,y,z directions in Cartesian coordinates.

The details of convective and viscous fluxes are given below:

$$F_x^c = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho u v \\ \rho u w \\ \rho H u \end{bmatrix} \quad F_x^v = \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xz} \\ u\tau_{xx} + v\tau_{xy} + w\tau_{xz} - q_x \end{bmatrix} \quad 5.5$$

$$F_y^c = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ \rho Hv \end{bmatrix} \quad F_y^v = \begin{bmatrix} 0 \\ \tau_{yx} \\ \tau_{yy} \\ \tau_{yz} \\ u\tau_{xy} + v\tau_{yy} + w\tau_{yz} - q_y \end{bmatrix} \quad 5.6$$

$$F_z^c = \begin{bmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ \rho Hw \end{bmatrix} \quad F_z^v = \begin{bmatrix} 0 \\ \tau_{zx} \\ \tau_{zy} \\ \tau_{zz} \\ u\tau_{xz} + v\tau_{yz} + w\tau_{zz} - q_z \end{bmatrix} \quad 5.7$$

where  $p$  is the static pressure,  $\tau_{ij}$  is the viscous stress tensor and  $q=(q_x, q_y, q_z)$  is the heat flux vector. In addition, the Euler equations are re-produced from the Navier-Stokes equation by neglecting the viscous terms .

The total energy per control volume  $E$ , for an ideal gas, is defined as:

$$E = \frac{P}{\gamma - 1} + \frac{\rho}{2}(u^2 + v^2 + w^2) \quad 5.8$$

where  $\gamma$  is the constant specific heat ratio and is defined by the relation of  $C_v$  and  $C_p$  which are the specific heats at constant volume and constant pressure, respectively:

$$\gamma = C_p / C_v \quad 5.9$$

For air the constant  $\gamma$  is typically equal to 1.4.

$H$  is the total enthalpy and is expressed as:

$$H = E + \frac{p}{\rho} \quad 5.10$$

The viscous stresses, due to the fluid's molecular viscosity, are defined as:

$$\tau_{xx} = 2\mu \frac{\partial u}{\partial x} + \lambda \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) \quad 5.11$$

$$\tau_{yy} = 2\mu \frac{\partial v}{\partial y} + \lambda \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) \quad 5.12$$

$$\tau_{zz} = 2\mu \frac{\partial w}{\partial z} + \lambda \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) \quad 5.13$$

$$\tau_{xy} = \tau_{yx} = \mu \left( \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \quad 5.14$$

$$\tau_{xz} = \tau_{zx} = \mu \left( \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \right) \quad 5.15$$

$$\tau_{yz} = \tau_{zy} = \mu \left( \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \right) \quad 5.16$$

Furthermore, the molecular viscosity coefficient  $\mu$  depends on the variation of air temperature  $T$  and is characterized by Sutherland's law

$$\mu = \mu_0 \left( \frac{T}{T_0} \right)^{3/2} \frac{T_0 + 110.4K}{T + 110.4K} \quad 5.17$$

where  $T_0$  is the reference temperature in degrees K and the reference viscosity is  $\mu_0 = 1.7894 \times 10^{-5} kg/(m \cdot s)$  for air.

The viscosity  $\mu$  and second coefficient of viscosity  $\lambda$  are related via the bulk viscosity coefficient which is  $\frac{2}{3}\mu + \lambda$ . This can be assumed to be zero so that

$$\lambda = -\frac{2}{3}\mu \quad 5.18$$

The heat flux  $q$  is defined by the thermal conductivity relation (Fourier's law)

$$q_x = -\kappa \frac{\partial T}{\partial x} \quad 5.19$$

$$q_y = -\kappa \frac{\partial T}{\partial y} \quad 5.20$$

$$q_z = -\kappa \frac{\partial T}{\partial z} \quad 5.21$$

where  $\kappa$  is the thermal conductivity coefficient. This can be expressed in terms of the Prandtl number  $Pr$ :

$$\kappa = \mu \frac{C_p}{Pr} \quad 5.22$$

because the ratio  $\frac{C_p}{Pr}$  is approximately constant for most gases. The Prandtl number is taken as 0.72 for air at standard conditions, and is assumed to be constant everywhere.

To close the above equations, an equation of state is introduced. For the ideal gas, which neglects inter-molecular forces, the equation of state is written as:

$$p = \rho RT \quad 5.23$$

where  $R$  is the gas constant with, for air, a value of  $R = 287.05 \text{ J/(kg} \cdot \text{K)}$ .

## 5.2 Reynolds-averaged Navier-Stokes (RANS) simulation and turbulence model

The Navier-Stokes equations embody sufficient physics to represent turbulence down to the smallest scales, at which dissipation to heat takes place. The equations can be solved to capture these turbulent motions, by means of direct numerical simulation (DNS). In this, it is necessary to resolve all scales of the turbulence in space and time. In order to discretise the governing equations sufficiently in space and time, the computational grid and time step must be fine enough. Although it does not employ any turbulence model and provides the most accurate numerical results, the computational cost is too expensive to be feasible for Reynolds numbers of industrial interest. Thus, a method which aims to reduce computational cost has been developed by time-averaging of the governing equations. This results in the Reynolds averaged Navier-Stokes equations (RANS) which is a practical method for solving the engineering problem.

In the time averaging of the Navier-Stokes equations, the flow variables are decomposed into a mean component plus a fluctuating component and the equations are then statistically averaged over time to eliminate the fluctuating component. This averaging process results in a set of equations for the mean component that are of the same form as the original Navier-Stokes equations, but with the appearance of additional terms, the the Reynolds stresses and heat fluxes, in the equations. The Reynolds stresses can then be expressed using various turbulence models. This modelling process has to represent the full effects of the turbulence on the flow, but cannot be fully universal and is thus likely to be a major source of inaccuracy in the ability of the RANS approach to represent complex turbulent flows. In the Tau code, the particular time averaging process used is Favre, or mass-weighted, averaging in order to better represent the compressibility in the RANS equations.

In the Favre averaging process, the conserved variable  $W$  is expressed as the sum of a steady average  $\tilde{W}$  and a fluctuating component  $W''$

$$W = \tilde{W} + W'' \quad 5.24$$

where  $\tilde{W}$  is the vector of mass weighted time-averaged conserved quantities defined as:

$$\tilde{W} = \frac{1}{\bar{\rho}} \lim_{t' \rightarrow \infty} \frac{1}{t'} \int_t^{t+t'} \rho(x, s) W(x, s) ds \quad 5.25$$

where  $\bar{\rho}$  is the conventional statistically averaged density. By applying this mass weighted-averaging to the original Navier-Stokes equations, the Favre averaged Navier-Stokes equations are obtained. These equations are identical to the instantaneous Navier-Stokes equations, with instantaneous variables replaced by the averaged mean components, except for an additional two turbulence correlation terms.

The first correlation term is in the momentum equations and is referred to as the Reynolds stress tensor:

$$\tau_{ij} = -\overline{\rho u_i'' u_j''} \quad 5.26$$

These are added to the molecular stresses in the momentum equations, The second correlation term is a turbulent heat flux which is added to the molecular heat flux  $q_i$ :

$$q_{ti} = \overline{\rho H'' u_i''} \quad 5.27$$

In order to closure this system of equations, a mathematical model has to be developed to approximate these terms. Many RANS models are based on the Boussinesq eddy viscosity assumption (Boussinesq 1877). As a result, the Reynolds stress tensor is modelled as:

$$-\overline{\rho u_i'' u_j''} = \mu_t \left( \frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} - \frac{2}{3} \delta_{ij} \frac{\partial \bar{u}_k}{\partial x_k} \right) - \frac{2}{3} \delta_{ij} \bar{\rho} k \quad 5.28$$

where  $\mu_t$  is the turbulent viscosity, to be obtained from the turbulence model, and  $k$  is the turbulent kinetic energy.

Similarly the turbulent heat flux appearing in the energy equation is approximated as

$$q_{t,i} = -\kappa_t \frac{\partial \bar{T}}{\partial x_i} \quad 5.29$$

with turbulent thermal conductivity  $\kappa_t$  which is normally defined as:

$$\kappa_t = \frac{C_p \mu_t}{Pr_t} \quad 5.30$$

where  $Pr_t$  is the turbulent Prandtl number, usually assumed to be a constant value of 0.9, although in reality it may have a variation through the boundary layer.

Therefore, the purpose of the turbulence model is to model  $\mu_t$  and  $k$ . In one-equation models such as Spalart-Allmaras (Spalart and Allmaras 1992), a transport equation is employed to model eddy-viscosity and the kinetic energy is ignored. In two-equation models, such as the  $k - \omega$  model (Wilcox 1998), two transport equations are used to approximate  $k$  and the second turbulence variable representing the turbulent length scale or dissipation. The turbulent viscosity is then derived from these two transported variables. The Spalart-Allmaras model has been widely used in aerodynamic external flow simulation, having good performance in the case of external aerodynamic flow with small and mild separation. Because the problem in this thesis will be mainly concerned with aerofoil/wing design under cruise conditions, no large separation will be presented. Therefore, the Spalart-Allmaras model is employed in most cases.

### 5.2.1 Spalart-Allmaras Turbulence Model

The Spalart-Allmaras method employs a one-equation model which is a single transport equation to solve the turbulence viscosity  $\mu_T$ . Spalart and Allmaras developed the algebraic model which removed the incompleteness and the turbulence viscosity is based on  $k$ . Thus, the Spalart-Allmaras model is simpler than the two-equation model, and its performance is better than the other one-equation model. In the Spalart-Allmaras model, the turbulence viscosity is calculated by:

$$\mu_T = \rho \tilde{v} f_{v1} \quad 5.31$$

where

$$f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^{-3}} \quad \chi = \frac{\tilde{v}}{v} \quad 5.32$$

Then the governing equation can be gained:

$$\frac{\partial \tilde{v}}{\partial t} + U_j \frac{\partial \tilde{v}}{\partial x_j} = c_{b1} \tilde{S} \tilde{v} - c_{w1} f_w \left( \frac{\tilde{v}}{d} \right)^2 + \frac{1}{\sigma} \frac{\partial}{\partial x_k} \left[ (\nu + \tilde{v}) \frac{\partial \tilde{v}}{\partial x_k} \right] + \frac{c_{b2}}{\sigma} \left( \frac{\partial \tilde{v}}{\partial x_k} \right)^2 \quad 5.33$$

Where  $\nu$  is molecular kinematic viscosity, the other relative formulas are given by:

$$f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}} \quad f_w = g \left[ \frac{1 + c_{w3}^6}{g^6 + c_{w3}^6} \right]^{\frac{1}{6}} \quad g = r + c_{w2} (r^6 - r) \quad 5.34$$

$$r = \frac{\tilde{v}}{\tilde{S} \kappa^2 d^2} \quad \tilde{S} = S + \frac{\tilde{\mu}}{\rho \delta^2 d^2} f_{v2} \quad S = \sqrt{2 \Omega_{ij} \Omega_{ij}} \quad 5.35$$

where  $\Omega_{ij} = \frac{1}{2} \left( \frac{\partial U_i}{\partial x_j} - \frac{\partial U_j}{\partial x_i} \right)$  is the rotation tensor and  $d$  is the minimal distance to the body surface.

Furthermore, the close coefficients are also given:

$$\begin{aligned} C_{b1} &= 0.1335 & C_{b2} &= 0.622 & C_{v1} &= 7.1 \\ \sigma &= 2/3 & C_{w1} &= \frac{C_{b1}}{\kappa^2} + \frac{1+C_{b2}}{\sigma} & C_{w2} &= 0.3 \\ C_{w3} &= 2 & \kappa &= 0.41 \end{aligned} \quad 5.36$$

The Spalart-Allmaras method is designed for aerodynamics flow and especially it has a good performance for flow around an aerofoil and wing. It can predict the separation region more accurately than the Baldwin-Lomax model. However, generally it is only applied to the homogeneous flow without irrotational mean straining.

### 5.3 Finite volume method

The finite volume method is the most versatile discretisation method employed in many CFD solvers. There are two steps to implement this method. The first step is to divide the computation domain into a large number of small control volumes  $\Omega$ . No grid

transformation is required in the finite volume method. Therefore, it could be employed to treat complex geometries.

The next step is to apply the integral form of the governing equation to each control volume  $\Omega$ . As mentioned in section 5.1, the integral form of the governing equation is derived from the conservation law which states that the rate of change of the flow quantity within the control volume must be equal to the flux of each through the boundaries of the control volume . For the momentum and energy equations these fluxes include forces acting on the boundaries. For the control volume  $i$  in Figure 5.1, the finite volume discretisation of the Navier-Stokes equations takes the form:

$$\Omega_i \frac{\partial \mathbf{W}_i}{\partial t} + R_i(\mathbf{W}_i) = 0 \quad 5.37$$

where the vector of conserved variables  $\mathbf{W}$  is assumed to be uniform over the control volume  $i$  with value  $\mathbf{W}_i$ .

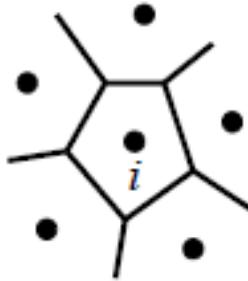


Figure 5.1 The control volume  $i$  in the finite volume method

The residual  $R_i$  is evaluated as the sum of all the fluxes across the boundary of the control volume  $i$ :

$$R_i = \sum_{j=1}^{nfaces} (F_{ij}^c - F_{ij}^v) S_{ij} \quad 5.38$$

where  $nfaces$  is the number of faces surrounding the control volume  $i$  and  $F_{ij}^c$  and  $F_{ij}^v$  are the inviscid and viscous fluxes between cells  $i$  and  $j$ , as shown in Figure 5.2.

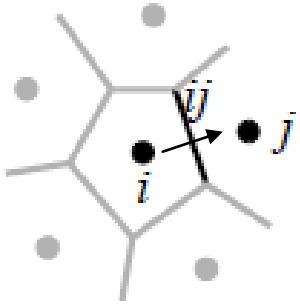


Figure 5.2 The flux between cells  $i$  and  $j$

In this thesis, a second order method is considered. The flux between cells  $i$  and  $j$  can be approximated by:

$$F_{ij}^c = \frac{1}{2}(F_i^c + F_j^c) \cdot n_{ij} \quad 5.39$$

where  $n_{ij}$  is the face normal between cell  $i$  and  $j$ .

However, this simple second order scheme is not numerically stable (Jameson 1995; Dwight 2006). Therefore, it is necessary to add some artificial dissipation, or the flux could be approximated by Riemann solver (van Leer 1979; Roe 1986; Toro 2009). In this work, the artificial dissipation is mainly employed and presented in the next section.

Furthermore, the computational discretisation grid of the TAU solver is generated by the software SOLAR which is developed by the Aircraft Research Association (ARA) (Leatham *et al.* 2000). It is an unstructured mesh and contains grid topology of tetrahedra, triangular prisms, pyramids and hexahedra. The grid generated by SOLAR is called *primary grid* in TAU. TAU uses the cell-vertex finite volume method, where the flow variables are stored on the nodal point of the *primary grid*. The control volume is then generated surrounding the nodal point of the *primary grid*. This results in a secondary grid; which is called *dual grid*, as shown in Figure 5.3.

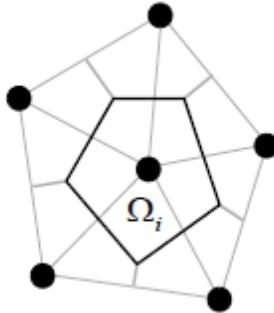


Figure 5.3 The cell-vertex finite volume: black nodal points and grey lines form the primary grid, black lines form the secondary grid

One of the advantages of the finite volume method is the excellent performance for discontinuities especially in flows near or above sonic speed since the flow quantity discontinuity is allowed at the interface. Therefore, the flow discontinuity phenomena, such as shock, could be addressed naturally in the finite volume method. However, for normal 2<sup>nd</sup> or higher order accurate methods in space, the order may be reduced near shocks by introducing limiters to avoid solution oscillations and numerical convergence issues.

#### 5.4 Central convective fluxes

As mentioned above, artificial dissipation is required to achieve a stable numerical scheme. Therefore, Equation 5.39 is rewritten with an artificial dissipation term:

$$F_{ij}^c = \frac{1}{2} (F_i^c + F_j^c) \cdot n_{ij} - \frac{1}{2} \bar{D}_{ij} \quad 5.40$$

where  $\bar{D}_{ij}$  is the artificial dissipation term.

The TAU solver employs the blended 2<sup>nd</sup> and 4<sup>th</sup> order artificial dissipation which is the well-known Jameson-Schmitt-Turkel (JST) scheme (Jameson *et al.* 1981). In this scheme, a 4<sup>th</sup> order dissipation is used across the entire main flow field, and a 2<sup>nd</sup> order dissipation is used near the shock region instead of 4<sup>th</sup> order since 4<sup>th</sup> order is not stable at a discontinuity. The discontinuity is detected by a pressure gradient.

The scalar artificial dissipation is presented here, which is defined as:

$$\bar{D}_{ij} = \lambda_{ij}^c [\bar{\varepsilon}_{(2),ij}(W_j - W_i) - \bar{\varepsilon}_{(4),ij}(L_j(W) - L_i(W))] \quad 5.41$$

where  $\bar{\varepsilon}_{(2),ij}$  and  $\bar{\varepsilon}_{(4),ij}$  control the weights of two dissipations, which are defined as:

$$\begin{aligned} \bar{\varepsilon}_{(2),ij} &= \varepsilon_{(2),ij} s_{(2),ij}^c \bar{\Phi}_{ij} \\ \bar{\varepsilon}_{(4),ij} &= \varepsilon_{(4),ij} s_{(4),ij}^c \bar{\Phi}_{ij} \end{aligned} \quad 5.42$$

where  $\varepsilon_{(2),ij}$  and  $\varepsilon_{(4),ij}$  are the switchers for the shock,  $s_{(2),ij}^c$  and  $s_{(4),ij}^c$  control the level of dissipation independent of the number of surrounding cells and  $\bar{\Phi}_{ij}$  is to control the amount of dissipation to be increased when it crosses the larger faces of cells.

Furthermore,  $s_{(2),ij}^c$  and  $s_{(4),ij}^c$  are presented as below to remove dependence on the number of faces of the control cell:

$$\begin{aligned} s_{(2),ij}^c &= \frac{3(N_i + N_j)}{N_i N_j} \\ s_{(4),ij}^c &= \frac{9[(1 + N_i)N_i + (1 + N_j)N_j]}{(1 + N_i)N_i(1 + N_j)N_j} \end{aligned} \quad 5.43$$

where  $N_i$  is the number of faces of the control cell  $i$ .

$\bar{\Phi}_{ij}$  is defined as:

$$\bar{\Phi}_{ij} = \frac{4\phi_{ij}^{(i)}\phi_{ji}^{(j)}}{\phi_{ij}^{(i)} + \phi_{ji}^{(j)} + 10^{-16}} \quad 5.44$$

$\phi_{ij}^{(i)}$  is defined as

$$\phi_{ij}^{(i)} = \left( \frac{\max(\frac{1}{2}\lambda_i^{\text{sum}} - \lambda_{ij}^c, 0)}{2\lambda_{ij}^c} \right)^{\frac{1}{2}} \quad 5.45$$

where  $\lambda_{ij}^c$  is the maximum eigenvalue between the face of cells  $i$  and  $j$ , which is defined as:

$$\lambda_{ij}^c = \max \left[ \lambda \left( \frac{\partial F_{ij}^c}{\partial W} \right) \right] = \frac{1}{2} |(U_i + U_j) \cdot n_{ij}| + \frac{1}{2} (a_i + a_j) \|n_{ij}\| \quad 5.46$$

where  $\lambda(\frac{\partial F_{ij}^c}{\partial W})$  is the eigenvector of  $\frac{\partial F_{ij}^c}{\partial W}$ , and  $\lambda_i^{\text{sum}}$  is the sum of the maximum convective eigenvalues over all faces of control volume  $i$ . Therefore, the value of  $\bar{\phi}_{ij}$  changes with respect to the cell size and adjusts the amount of dissipation.

The switchers  $\varepsilon_{(2),ij}$  and  $\varepsilon_{(4),ij}$  are defined as:

$$\begin{aligned}\varepsilon_{(2),ij} &= k_{(2)} \max (\Psi_i, \Psi_j) \\ \varepsilon_{(4),ij} &= \max (k_{(4)} - \varepsilon_{(2),ij}, 0)\end{aligned}\quad 5.47$$

where  $k_{(2)}$  and  $k_{(4)}$  are the constant values, which are normally 0.5 and 1/64 respectively.

$\Psi_i$  is to approximate the pressure gradient, which is defined as:

$$\Psi_i = \left| \frac{p_i^{\text{diff}}}{p_i^{\text{sum}}} \right| \quad 5.48$$

where

$$\begin{aligned}p_i^{\text{sum}} &= \sum_{j=1}^{\text{nfaces}} (p_i + p_j) \\ p_i^{\text{diff}} &= \sum_{j=1}^{\text{nfaces}} (p_j - p_i)\end{aligned}\quad 5.49$$

Finally,  $L_i(W)$  is the 3<sup>rd</sup> difference and defined as:

$$L_i(W) = \sum_{j=1}^{\text{nfaces}} (W_j - W_i) \quad 5.50$$

The construction of the central convective flux has been presented. Furthermore, the viscous flux is not as problematic as the convective flux since it is not related to the stability problem. Therefore, artificial dissipation is rarely used in diffusion terms. The details of construction of the viscous flux and further information on construction of flux for turbulence model of TAU can be found in the Dwight's thesis Chapters 2.8 and 2.11 (2006).

## 5.5 Construction of gradient

The gradient of the flow variables is required in order to construct the viscous flux and turbulence sources. The TAU solver offers two ways to obtain the gradient, which are *Green-Gauss* and *least square* methods. The most efficient way is use the *Green-Gauss* method which is based on the Gauss integral theorem:

$$\int_{\Omega_i} \nabla W_i d\Omega = \oint_{\partial\Omega} W_i \cdot n \cdot d\partial\Omega \quad 5.51$$

Therefore, the gradient at the control cell  $i$  in finite volume is given by

$$\Omega_i \nabla W_i = \sum_{j=1}^{n\text{faces}} \frac{1}{2} (W_i + W_j) n_{ij} \quad 5.52$$

The *least square* method was first proposed by Anderson and Bonhaus (1994). Different from the *Green-Gauss* method, the *least square* method employs a Taylor expansion from a local point to its surrounding point rather than the metric terms as face normals or volumes of a local control volume in *Green-Gauss* methods. Consider an arbitrary function  $\phi$  expressed by a Taylor expansion at a local point using its neighboring point:

$$\phi_0 = \phi_1 + \frac{\partial\phi_1}{\partial x} \Delta x + \frac{\partial\phi_1}{\partial y} \Delta y + \frac{\partial\phi_1}{\partial z} \Delta z + O(\Delta x^2) + \text{higher order term} \quad 5.53$$

For a control cell  $i$ , the Taylor expansion could be written for every point surrounding cell  $i$  (see Figure 5.4). Therefore, a system of linear equations can be derived from all neighbouring points.

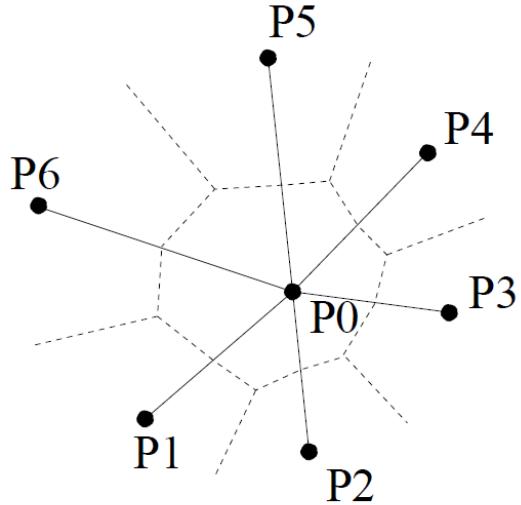


Figure 5.4 Surrounding points used for the least square algorithm

The system of linear equations is

$$\begin{bmatrix} w_1 \Delta x_{01} & w_1 \Delta y_{01} & w_1 \Delta z_{01} \\ w_2 \Delta x_{02} & w_2 \Delta y_{02} & w_2 \Delta z_{02} \\ \vdots & \vdots & \vdots \\ w_N \Delta x_{0N} & w_N \Delta y_{0N} & w_N \Delta z_{0N} \end{bmatrix} \begin{pmatrix} \frac{\partial \phi}{\partial x} \\ \frac{\partial \phi}{\partial y} \\ \frac{\partial \phi}{\partial z} \end{pmatrix} = \begin{pmatrix} w_1(\phi_1 - \phi_0) \\ w_2(\phi_2 - \phi_0) \\ \vdots \\ w_N(\phi_N - \phi_0) \end{pmatrix} \quad 5.54$$

The linear equations are then rewritten as:

$$W[\vec{a}_1 \quad \vec{a}_2 \quad \vec{a}_3] \nabla \phi = Wb \quad 5.55$$

where

$$W = \begin{bmatrix} w_1 & 0 & \dots & 0 \\ 0 & w_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & w_N \end{bmatrix} \quad 5.56$$

W is the matrix with a weighting factor which is related to the geometry and allows the computation at very high cell aspect ratios.

$$w_i = \frac{1}{\sqrt{\Delta x_i^2 + \Delta y_i^2 + \Delta z_i^2}} \quad 5.57$$

In order to solve this linear equation, QR decomposition strategy is employed with a Gram-Schmidt orthogonalization.  $Q$  is an orthogonal matrix  $Q \in R^{n \times m}$  and  $R$  is an upper triangular matrix  $R \in R^{m \times m}$ . The process is presented below:

$$\begin{aligned} Ax &= b \\ QRx &= b \\ Q^T QRx &= Q^T b \\ x &= R^{-1} Q^T b \end{aligned} \tag{5.58}$$

The computation of gradients with the least-square approach has shown much more accurate gradients in comparison to the Green-Gauss theorem. In addition, it achieves better robustness and more accurate solutions than the Green-Gauss approach. This viable algorithm reconstructs linear functions exactly on any type of mixed grids.

## 5.6 Temporal discretisation

In this thesis, all cases are the time-independent steady state case. For the steady state condition, the time rate of the conservative change is zero:

$$\frac{dW_i}{dt} = 0 \tag{5.59}$$

Therefore, the flow residual  $R_i(W_i) = 0$ . However, the practical and efficient way is to employ the corresponding time-dependent problem with fictitious pseudo-time  $t^*$  to drive the solution towards a steady state convergence solution:

$$\Omega_i \frac{dW_i}{dt^*} + R_i(W_i) = 0 \tag{5.60}$$

The temporal discretisation of Equation 5.60 could be:

$$\Omega_i \frac{W_i^{n+1} - W_i^n}{\Delta t} = -R_i(W_i^n) \tag{5.61}$$

where  $\Delta t$  is a time step. This could result in a simple explicit method where the residual is only dependent on the last time step value. The update of the next time step is:

$$W_i^{n+1} = W_i^n - \frac{\Delta t}{\Omega_i} R_i(W_i^n) \tag{5.62}$$

Furthermore, the low-storage  $K$ -step Runge-Kutta scheme could be employed to further discrete the temporal space and it is available in TAU. However, for numerical stability reasons, the time step  $\Delta t$  must be smaller than unity for the one-dimensional convection equation (Le Moigne 2002). For the viscous and turbulence cases, the CFL number has to be even smaller. This results in the explicit method having less efficiency. Therefore, the implicit method is desired.

In the implicit method, Equation 5.60 is discretised with respect to the future time level:

$$\Omega \frac{W^{n+1} - W^n}{\Delta t} = -R(W^{n+1}) \quad 5.63$$

The implicit scheme is more complicated than the explicit scheme since Equation 5.63 is a non-linear algebraic system. In order to solve it, the residual vector needs to be linearised around its time level  $n+1$ :

$$\begin{aligned} R(W^{n+1}) &= R(W^n) + \frac{\partial R(W^n)}{\partial t} \Delta t + O(\Delta t^2) + \text{higher order} \\ &= R(W^n) + \frac{\partial R(W^n)}{\partial W} \frac{\partial W}{\partial t} \Delta t + O(\Delta t^2) \end{aligned} \quad 5.64$$

where the term  $\frac{\partial W}{\partial t} \Delta t$  is:

$$\frac{\partial W}{\partial t} \Delta t = \Delta W^n + O(\Delta t^2) = W^{n+1} - W^n \quad 5.65$$

Applying Equation 5.65 into Equation 5.64 and substituting it into the right hand side of Equation 5.63 results in the establishment of a linear algebraic system:

$$\begin{aligned} \left[ \frac{\Omega}{\Delta t} I + \frac{\partial R(W^n)}{\partial W} \right] (W^{n+1} - W^n) &= -R(W^n) \\ \left[ \frac{\Omega}{\Delta t} I + \frac{\partial R(W^n)}{\partial W} \right] \Delta W^n &= -R(W^n) \\ A(W) \Delta W^n &= -R(W^n) \end{aligned} \quad 5.66$$

Therefore, the aim of the implicit method is to solve this linear algebraic system to obtain  $\Delta W^n$  and update it to the next time level  $W^{n+1} = W^n + \Delta W^n$  until the solution converges.

$A(W)$  is the left hand side (LHS) of the linear system which contains a Jacobian matrix  $\frac{\partial R}{\partial W}(W^n)$ . The right hand side (RHS) contains all elements of the spatial discretisation. In the exact solution, the flow residual will be zero and hence  $\Delta W^n$  will be zero and independent of  $A(W)$ . Therefore, the matrix  $A(W)$  has no effect on the converged solution and can be solved by various methods. In the TAU solver, the LHS is solved by the Lower-Upper Symmetric-Gauss-Seidel (LUSGS) method. The details of LUSGS can be found in Yoon and Jameson (1988) and Dwight's thesis (2006) Chapter 3.2.1. The construction of the Jacobian matrix is also key to the implicit method. Because the structure of the Jacobian matrix is complex and not related to this thesis work, the details of its construction, which includes a Jacobian of a convective, diffusion, artificial dissipation and turbulence model, can be found in Dwight's thesis (2006).

# **Chapter 6 Discrete Adjoint Approach and Numerical Optimisation**

The parameterisation methods and CFD numerical solver have been presented in previous chapters; in this chapter the numerical optimisation methods are presented and studied.

Numerical optimisation methods can be classified as gradient-based and gradient-free. The gradient-free methods obviously do not require any gradient information of the objective function and normally purely depend on the evaluation of the value of the objective function. They are also divided into local and global. The typical local gradient-free methods are the simplex and subplex methods (Nelder and Mead 1965; Rowan 1990). Numerical optimisation builds a simplex model which is a convex hull of  $n+1$  points ( $n$  is the dimension of design space). The algorithm moves the simplex model through design space and automatically shrinks the size of the simplex model. This type of method has been successfully used in aerodynamic optimisation, for example in the work done by Sturdza (2007) and Widhalm and Rozermier (2008). However, this type of method requires evaluating at least  $n$  times the value of the objective function to build up the simplex model. Therefore, it is only efficient for a case with a small number of design variables and is limited to use for 3D large cases.

Global gradient-free methods, for example the response surface method, genetic algorithm and particle swarm optimisation method, have been widely used in aerodynamic optimisation. The genetic algorithm and particle swarm optimisation methods are the typical heuristics methods which have been widely employed in engineering optimisation problems.

The genetic algorithm is one of the evolutionary algorithms which is completely different from the gradient-based methods. This method was first proposed in the 1950s and applied to computer-aided simulation to research natural evolutionary and genetic processes; it was later applied to the function optimisation area. In genetic algorithm, the

optimisation employs the theory of natural selection. Each design variable is represented as a chromosome to identify an individual as a design. The optimisation starts from a large number of individuals (populations) which are randomly distributed in design space. The fitness value of each individual can be calculated based on the objective function at each generation. The reproduction is then carried out with a selection of the parents based on the fitness value. The individual's high fitness value can be selected with high probability using the roulette wheel rule. Afterwards, the selected individuals can be made a pair as the parents. The reproduction of new individuals can be then carried out using crossover and mutation of chromosomes between parents. The all new individuals are formed and replace the old individuals as the new generation. The process of genetic algorithm is running as a cycle, and the process will converge towards an individual with high fitness value after many generations. Although the genetic algorithm does not require any calculation of gradient, the efficiency of optimisation depends on the number of initial populations. For a case with a large number of design variables, more individuals are needed. In aerodynamic optimisation using high-fidelity CFD, each individual indicates one CFD calculation and this could make computation unaffordable when the number of populations is too high. Therefore, this method is mainly used in two-dimensional cases or for optimisation using a low-fidelity and quick CFD tool.

In order to reduce the computational cost, the surrogate model methods, such as response surface model and kriging model, are preferred, which could represent the entire design space. Once the surrogate model is built up, the standard global optimisation technique can be employed to search the optimal solution on the surrogate design space. This technique contains two steps. The first step, called 'design of experiment', is to sample the design space. The Latin hyper cube algorithm is normally selected to determine the location of samples in the design space under the required number of sample points (Forrester *et al.* 2006). The objective function is then evaluated at each sample point. The second step is to build up the surrogate model with respect to the sample points. The construction of the surrogate model is similar to data fitting interpolation, and many techniques could be applied. The most common method is to employ a quadratic surface which is calculated by the least-squares method through all sample points. The more

advanced technique is to use the kriging model which is developed based on the field of spatial statistics and geostatistics (Forrester *et al.* 2008). It is much better than the polynomial-based interpolation and is able to handle the sample data with stochastic processes (Jeong *et al.* 2005).

These methods have been widely used in aerodynamic optimisation. However, the quality of the surrogate model depends on the number of sample points: for a case with hundreds of design variables, a huge number of sample points are required and this could cause large computational expense and difficulties in building up the model. Therefore, the methods have been mainly used in two-dimensional cases or cases with a number of design variables not exceeding 40. There are a number of works using the surrogate model in the literature, for example Madsen *et al.* (2000), Simpson *et al* (2001), Jouhaud *et al.* (2007) and Kanazaki *et al.* (2006). Obayashi *et al.* (2005) have successfully employed the kriging model for a simple 3D wing optimisation; however, the number of design variables is only 35.

The gradient-based methods, such as steepest descent and quasi-Newton, require the sensitivity derivatives of the objective function. In these methods, the optimisers are able to reduce the objective function value by searching the design space along the gradient information. Compared with the other optimisation methods presented above, the gradient-based optimisation method is the only efficient and feasible method for large dimensional optimisation cases, although it only guarantees to find a local optimal solution. However, in aerodynamic optimisation, the calculation of the sensitivity derivatives for aerodynamic coefficients is a challenge. Therefore, the common methods for calculating sensitivities are first briefly reviewed in the first sub-section of this chapter, and the discrete adjoint method is then presented. At the end, the mesh deformation techniques and optimisation framework are introduced.

## 6.1 Common methods to calculate sensitivities

As presented in the introduction, the calculation of gradient  $dI/dD$  for the objective function  $I$  with design variables  $D$  is the key for employing a gradient-based optimisation

method. In Le Moigne's thesis (2002), an excellent review of the calculation of gradient has been provided.

There are four main ways to calculate the gradient, which are the finite-difference method, complex variable method, automatic differentiation method and direct differentiation method. The simplest method to calculate the gradient is the finite-difference method, which is derived from the basic numerical method for calculating the derivatives. The main forms of this method are the forward or backward differencing scheme which is defined as:

$$\frac{\partial I}{\partial D_k} \approx \frac{I(D_k \pm \varepsilon_k) - I(D_k)}{\pm \varepsilon_k} \quad 6.1$$

or central differencing scheme which is defined as:

$$\frac{\partial I}{\partial D_k} \approx \frac{I(D_k + \varepsilon_k) - I(D_k - \varepsilon_k)}{2\varepsilon_k} \quad 6.2$$

where  $\varepsilon_k$  is the perturbation step for the  $k$ -th design variable, '+' is for forward difference and '-' is for backward difference.

The finite-difference method could be easily set up since it only requires the value of the objective function at design variable points and each perturbed design variable location. Therefore, it does not require any additional solver or modified CFD solver. In the early research, it was employed for aerodynamic optimisation. More details can be found in Hicks *et al.* (1974) and Hicks and Vanderplaats (1975).

However, there are two main shortcomings of finite-difference methods. The first is that the method requires calculating the value of the objective function at every perturbed design variable location. This means, for the forward or backward differencing scheme, it needs run a number of design variables (NDV)+1 times flow simulation to obtain the gradient of the objective function. For a central differencing scheme, it requires 2xNDV. With the current computational resource, each flow simulation takes up to a few hours for large 3D cases. Thus, the finite-difference methods are very time consuming. The second main shortcoming is the issue of perturbation step size. As we know already in

mathematics, the finite-difference method requires a small perturbation set up size since a large perturbation step will bring in an error of gradient. However, the small step could cause a round-off error. Kim *et al.* (1999), Hou *et al.* (1995) and Green *et al.* (1993) indicated that the perturbation step size is linked to the convergence criterion of the flow solution. It means the flow solver must converge at each perturbed design location and be lower than the perturbation step size. It would be unaffordable for the high-fidelity flow solver. Moreover, although there are many shortcomings, the finite-difference methods are still employed for aerodynamic optimisation. For example, Destarac and Reneaux (1993) and Reneaux (1984) employed the finite-difference method to evaluate the gradient based on the low-fidelity flow solver. Eyi *et al.* (1996; 1997) used the finite-difference method for calculating the gradient based on the Navier-Stokes flow solver. Cliff *et al.* (2002) employed the finite-difference method for aerodynamic optimisation based on the Euler equation.

Due to the drawbacks of the finite-difference method, it is hard to be employed in optimisation directly. Therefore, it is normally employed to check the accuracy of other methods. Because the exact sensitivities are almost impossible to be obtained for a complex case, the finite-difference method could be considered as the exact solution.

The second method to calculate the gradient is called the complex variable method. Similar to the finite-difference method, it was derived from the Taylor series expansion. The form of the complex variable method is written as:

$$\frac{\partial I}{\partial D_k} \approx \frac{\text{Im}[I(D_k + i\varepsilon_k)]}{\varepsilon_k} \quad 6.3$$

where  $\varepsilon_k$  is the perturbation step and  $\text{Im}[\cdot]$  is the imaginary part of this function.

In this method, the gradient is independent of the unperturbed function value  $I(D_k)$ . A small incremental step is still required to be used for increasing the accuracy; however, the requirement of convergence of the flow solution is not as much as in the finite-difference method. However, it still requires the NDV flow solution to calculate the gradient. In addition, the flow solver must be modified to allow for using complex

variables, which is a very difficult development and the complex variables flow solver could take twice the memory and computing time as the original flow solver. Hence, it could be much slower than the forward or backward finite-difference method. In Nadarajah and Jameson's (2001) work, they employed complex variable methods to check the accuracy of other methods.

Keane and Nair (2005), Mohammadi and Pironneau (2001) and Barthelemy and Hall (1992) presented another way for the automatic differentiation method. The basic idea of automatic differentiation is that a computer program implementing a numerical algorithm can be decomposed into a long sequence of a limited set of elementary arithmetic operations. It then applies the chain rule of differentiation to the original source code to create an extra computer program which calculates the gradients of the outputs. The automatic differentiation methods have two main modes, which are forward mode and reverse mode. The forward mode calculates the differentiation starting from the input to the output. The reverse mode works in the reverse way to the forward mode, and could be faster for a small number of outputs. More details can be found in Keane and Nair (2005) and Gauger *et al.* (2007). It requires some modification of the initial program by inserting an automatic differentiation code. There are various automatic differentiation codes which can be found on-line. The common codes are ADIFOR, TAMC, DAFOR, GRESS, Odysse and PADRE2 (Keane and Nair (2005) p. 197). Furthermore, the automatic differentiation methods are normally used for calculating the derivatives which are required in the direct differentiation methods. This could help to maintain the efficiency of the direct differentiation methods while reducing the cost of calculating complicated derivatives by hand . Detailed information about this can be found in Hou *et al.* (1995), Korivi *et al.* (1994) and Oloso and Taylor (1997).

Because the above methods to calculate the sensitivity derivatives are not normally feasible, other methods, which are called direct differentiation methods, were proposed by Sobiesczanski-Sobieski (1986). The direct differentiation method aims to obtain the sensitivity derivatives by analytically deriving them from differentiation of the governing equation.

The detailed definition of direct differentiation can be found in Le Moigne's thesis (2002), Keane and Nair (2005) and Dwight (2006). For the aerodynamic numerical optimisation, the objective function could be written as:

$$I = I(W(D), X(D), D) \quad 6.4$$

where W is the flow variable, X indicates the grid variable and D is the design variable. Because I, W and X depend on the design variable D, the sensitivity derivatives of the objective function I could be written as:

$$\frac{dI}{dD} = \left( \frac{\partial I}{\partial W} \right) \frac{dW}{dD} + \left( \frac{\partial I}{\partial X} \right) \frac{dX}{dD} + \frac{\partial I}{\partial D} \quad 6.5$$

In Equation 6.4, the calculation of  $\frac{\partial I}{\partial W}$ ,  $\frac{\partial I}{\partial X}$  and  $\frac{\partial I}{\partial D}$  is relatively simple and depends on the class of the objective function.  $\frac{dX}{dD}$  is the grid sensitivity and could be solved in a finite difference or analytical way. The sensitivities of flow field  $\frac{dW}{dD}$  term is difficult to solve.

In Chapter 5, the governing equation and numerical solver have been presented. The flow residual is related to the flow variables, grid and design variables. For a steady state flow, the flow residual is equal to zero:

$$R(W(D), X(D), D) = 0 \quad 6.6$$

The derivatives of the governing equation can be derived with respect to the design variables:

$$\frac{dR}{dD} = \frac{\partial R}{\partial W} \frac{dW}{dD} + \frac{\partial R}{\partial X} \frac{dX}{dD} + \frac{\partial R}{\partial D} = 0 \quad 6.7$$

Equation 6.7 could then be written as:

$$\frac{\partial R}{\partial W} \frac{dW}{dD} = - \frac{\partial R}{\partial X} \frac{dX}{dD} - \frac{\partial R}{\partial D} \quad 6.8$$

Therefore, the  $\frac{dW}{dD}$  term could be solved based on Equation 6.8. Furthermore, the right hand side of Equation 6.8 could be simplified. Depending on the type of the design

variables, the grid sensitivity  $\frac{dX}{dD}$  could be zero when the design variables are flow field parameters, such as Mach number or far-field angle of attack, and the  $\frac{\partial R}{\partial D}$  could be zero when the design variables are the shape parameters. On the left side of Equation 6.8, the  $\frac{\partial R}{\partial W}$  term is the Jacobian matrix of the flow field. It could be taken directly from the implicit flow solver. Practically, the LHS could be solved only once, but the RHS has to be constructed for each design variable. This means that this linear system has to be solved NDV times.

Although the direct differentiation requires an exact Jacobian matrix solution and most flow solvers provide the approximate version, the Jacobian matrix provided by most flow solvers still could be employed for the sensitivities of the direct differentiation method. For the two-dimensional Euler equation, direct inversion of the Jacobian matrix is possible. So Equation 6.8 could then be solved quickly. However, it is hard to converge for the three-dimensional case or viscous flow. The calculation of the Jacobian matrix is complex and requires much memory. Finally, Equation 6.8 requires treating the boundary condition if it solves accurate sensitivity derivatives. Moreover, if the flow governing equation is numerically discretised before it is differentiated, the direct differentiation is a so called discrete approach. In the other way, it is called a continuous approach.

Due to the difficulty of construction of an exact Jacobian matrix, the direct differentiation method is mainly employed in the Euler problem, especially on an unstructured mesh. This method was employed in the early aerodynamic optimisation research. In the references Baysal and Eleshaky (1992), Korivi *et al.* (1994) and Taylor *et al.* (1992), the hand-differentiating method has been introduced. Oloso and Taylor (1997) and Clyde *et al.* (1999) introduced the direct differentiation method using automatic differentiation which has been presented above. Svenningsen *et al.* (1996) and Eleshaky and Baysal (1992) presented some examples for viscous laminar flow.

Because the calculation of sensitivity derivatives using the direct differentiation method still depends on the number of design variables, it is computationally expensive for a

large number of design variables. Therefore, a method with computational cost reduction is desirable, and adjoint methods have been developed.

## 6.2 Discrete adjoint methods

Adjoint methods are similar to the direct differentiation method and apply control theory to the aerodynamic optimisation process. They were first proposed by Pironneau (1973) in Stokes flow. Jameson (1988) successfully developed this methodology to the Euler equation. An excellent presentation of adjoint methods can be found in Jameson (2004) and Le Moigne (2002). As above mentioned, the residual of the governing equation is zero. The basic idea of adjoint methods is to multiply the variation of the governing equation by a Lagrange multiplier  $\lambda$ , and add it to the variation of objective function, Equation 6.8.

### 6.2.1 Discrete adjoint equation

Therefore, a new equation with adjoint operator  $\lambda$  is then obtained as:

$$L = I(W, X, D) + \lambda R(W, X, D) \quad 6.9$$

The derivative of this new equation is:

$$\frac{dL}{dD} = \left( \frac{\partial I}{\partial W} \right) \frac{dW}{dD} + \left( \frac{\partial I}{\partial X} \right) \frac{dX}{dD} + \frac{\partial I}{\partial D} + \lambda^T \left( \frac{\partial R}{\partial W} \frac{dW}{dD} + \frac{\partial R}{\partial X} \frac{dX}{dD} + \frac{\partial R}{\partial D} \right) \quad 6.10$$

It could be rewritten as:

$$\frac{dL}{dD} = \left( \left( \frac{\partial I}{\partial D} \right) + \lambda^T \frac{\partial R}{\partial W} \right) \frac{dW}{dD} + \left( \left( \frac{\partial I}{\partial X} \right) + \lambda^T \frac{\partial R}{\partial X} \right) \frac{dX}{dD} + \frac{\partial I}{\partial D} + \lambda^T \frac{\partial R}{\partial D} \quad 6.11$$

If an appropriate adjoint operator which satisfies the adjoint equation 6.12 could be found, the Equation 6.11 could be simplified.

$$\left( \frac{\partial R}{\partial W} \right)^T \lambda = - \left( \frac{\partial I}{\partial W} \right)^T \quad 6.12$$

Similar to the direct differentiation method presented in the previous subsection, the  $\frac{\partial R}{\partial W}$  term is the Jacobian matrix of the flow field and the calculation of  $\frac{\partial I}{\partial D}$  and  $\frac{dW}{dD}$  for each

design variable could be simplified depending on the type of design variable. The computational cost could be apparently decreased.

$$\frac{dI}{dD} = \frac{dL}{dD} = \left( \left( \frac{\partial I}{\partial X} \right) + \lambda^T \frac{\partial R}{\partial X} \right) \frac{dX}{dD} + \frac{\partial I}{\partial D} + \lambda^T \frac{\partial R}{\partial D} \quad 6.13$$

For the shape design parameters,  $\frac{\partial I}{\partial D}$  and  $\frac{dW}{dD}$  are zero. The function is then written as:

$$\frac{dI}{dD} = \frac{dL}{dD} = \left( \left( \frac{\partial I}{\partial X} \right) + \lambda^T \frac{\partial R}{\partial X} \right) \frac{dX}{dD} \quad 6.14$$

It only requires to solve the adjoint equation once to obtain the sensitivity derivatives for each objective function. Thus, the calculation of the sensitivity derivatives of the objective function could be independent of the number of design variables. Similar to the direct differentiation method, grid sensitivities  $\frac{dx}{dD}$  could be calculated analytically or using finite difference, and which are:

$$\begin{aligned} \left( \frac{\partial I}{\partial X} \right) \frac{dX}{dD_k} &\approx \frac{I(W, X(D_k + \varepsilon), D_k) - I(W, X(D_k), D_k)}{\varepsilon} \\ \left( \frac{\partial R}{\partial X} \right) \frac{dX}{dD_k} &\approx \frac{R(W, X(D_k + \varepsilon), D_k) - R(W, X(D_k), D_k)}{\varepsilon} \end{aligned} \quad 6.15$$

The whole system for sensitivity derivatives could only be solved NCON+1 times, where NCON is number of constraints which includes aerodynamic coefficients. For the aerodynamic optimisation, most constraints could be provided from the flow solver. Thus, the adjoint methods are more efficient than the direct differentiation method.

Furthermore, they are similar to the direct differentiation method. If the flow governing equation is discretised before it is differentiated, the method is the so called discrete adjoint method. In the reversed way, the method is called continuous adjoint method. The adjoint equation 6.12 is shown in discrete form because the flow governing equation is already discretised in flow solver. The continuous adjoint equation is a bit more complex than the discrete adjoint method since it requires more theoretical derivation. The continuous adjoint method for aerodynamic optimisation was first proposed by Pironneau (1973; 1974). However, the first successful application was done by Jameson (1988). For the theoretical analysis, both the discrete adjoint method and the continuous adjoint

method should provide the same results. The development of the discrete adjoint solver is relatively simpler than for the continuous adjoint method since the Jacobian matrix term could be taken as  $\frac{\partial R}{\partial W}$  from the flow solver. However, the discrete adjoint solver requires keeping numerical consistency. For example, if the artificial dissipation and turbulence viscous term are employed in the flow solver, the Jacobian matrix should include these terms. However, the continuous adjoint solver has no restricted consistency requirement.

In Le Moigne's (2002) thesis, a comparison of continuous and discrete adjoint methods was made. He mentioned that Nadarajah and Jameson (2000) and Nadarajah and Jameson. (2001) had compared the effect of both methods on performance and accuracy of the sensitivity derivatives on an unstructured grid. They found that both methods could provide similar results. The discrete method could provide a better result than the continuous method since the discrete method is able to provide the exact numerical gradient. In addition, the continuous method could be improved by increasing the mesh size. The discrete method requires much more memory than the continuous method. Furthermore, Giles and Pierce (1998) presented that the continuous method could suffer a problem for the shock wave case. However, neither of the two methods has apparent advantages compared to the other, which is concluded by Giles and Pierce (2000). Thus, the choice between discrete and continuous methods is more dependent on personal preference and tool availability.

The adjoint methods nowadays have become a mature tool for aerodynamic optimisation. There are some enormous works for aerodynamic optimisation which have already employed the adjoint methods. Due to the difficulty with the viscous term in adjoint methods, especially for the continuous method, continuous adjoint methods were applied to aerodynamic optimisation mainly based on the Euler equation, for example in Jameson (1988), Jameson (2003; 2004), Reuther and Jameson. (1995), Giles and Pierce (1998), Iollo *et al.* (2001) and Brezillon and Gauger (2004). In recent years, the continuous adjoint method for viscous laminar flow and turbulent flow has been developed (Nguyen 2008; Anderson and Venkatakrishnan 1997; Anderson and Bonhaus

1997; Kim *et al.* 2002; Soto *et al.* 2004; Leoviriyakit *et al.* 2003; Widhalm and Ronzheimer 2008).

As the above have mentioned that the discrete adjoint method is simple to be developed compared to the continuous adjoint method, especially when considering viscous flow and shock wave, many researchers prefer to use the discrete adjoint method in their work. Qin *et al* (2004) have successfully performed aerodynamic optimisation for blended wing body configuration based on discrete adjoint solver with full turbulence model. Other works for the discrete adjoint method could be found in Vitturi and Beux (2006), Kim *et al.* (2001), Kim and Sasaki (2001), Nielsen and Anderson (2002), Nielson and Kleb (2006), Thomas *et al.* (2005), Mavriplis (2007), Le Moigne and Qin (2004), Wong *et al.* (2007) and Castonguay and Nadarajah (2007). All these works have shown that both methods could provide good results for aerofoil, wing or wing body aerodynamic optimisation. Because the development of the adjoint solver is not part of this thesis work, the details of construction of the Jacobian matrix for the adjoint solver and construction of RHS of adjoint equation can be found in Dwight (2006) and Le Moigne (2002).

In this thesis, TAU offers a discrete adjoint solver. It is solved by an Incomplete Lower-Upper GMRES algorithm (Saad and Schultz 1986; Jameson and Yoon 1987; Sharov *et al.* 2000; Saad 2003). Therefore, the discrete adjoint solver is employed in this thesis (Brezillon and Dwight 2005; Brezillon *et al.* 2012). The TAU adjoint solver provides the discrete adjoint approach with a full turbulence model (one equation SA model) and laminar viscous flow. The validation of the TAU adjoint solver has been made by Widhalm *et al.* (2010) through comparing the gradient obtained by finite-difference, laminar viscous adjoint and full turbulence adjoint methods. The results showed that the gradient obtained by the full turbulence adjoint solver matches excellently with the finite-difference method. The gradient of lift calculated by laminar viscous adjoint method is matched with the full turbulence adjoint solver; however, the gradient of drag is different. This means that the adjoint solver with the full turbulence model is essential for accuracy of gradient. This point needs to be borne in mind.

### 6.2.2 Discrete adjoint solver with mesh deformation

As presented in subsection 6.2.1, the grid sensitivities  $\frac{dX}{dD}$  could be calculated by finite difference or analytically. Le Moigne (2002) shows the grid sensitivities for algebraic structured mesh deformation. However, the analytical solution of entire grid sensitivities would be hard to obtain for an unstructured mesh deformation algorithm such as spring analogy. Therefore, the finite-difference method is required. However, the mesh deformation could also be time consuming. An additional issue, which is the same with any finite-difference approach, is that it is hard to determine the perturbation step size to obtain accurate sensitivities. Therefore, Nielsen and Park (2005) proposed to employ another adjoint equation to eliminate the grid sensitivities. Therefore, the objective function  $I = I(W(D), X(D), D)$  will be subject to:

$$\begin{aligned} R(W, X, D) &= 0 \\ T(X, D) &= 0 \end{aligned} \tag{6.16}$$

where  $T$  is the residual vector of dependency of mesh coordinates on design parameters (the mesh deformation function).

Then, the two residual functions are added into the objective function with two adjoint operators  $\lambda_{flow}$  and  $\lambda_{mesh}$ :

$$L = I(W, X, D) + \lambda_{flow} R(W, X, D) + \lambda_{mesh} T(X, D) \tag{6.17}$$

The derivative of Equation 6.17 is:

$$\begin{aligned} \frac{dL}{dD} &= \left( \frac{\partial I}{\partial W} \right) \frac{dW}{dD} + \left( \frac{\partial I}{\partial X} \right) \frac{dX}{dD} + \frac{\partial I}{\partial D} + \lambda_{flow}^T \left( \frac{\partial R}{\partial W} \frac{dW}{dD} + \frac{\partial R}{\partial X} \frac{dX}{dD} + \frac{\partial R}{\partial D} \right) \\ &\quad + \lambda_{mesh}^T \left( \frac{\partial T}{\partial X} \frac{dX}{dD} + \frac{\partial T}{\partial D} \right) \end{aligned} \tag{6.18}$$

This is then rewritten as:

$$\begin{aligned} \frac{dL}{dD} &= \left( \frac{\partial I}{\partial D} + \lambda_{flow}^T \frac{\partial R}{\partial D} + \lambda_{mesh}^T \frac{\partial T}{\partial D} \right) + \left( \left( \frac{\partial I}{\partial W} \right) + \lambda_{flow}^T \frac{\partial R}{\partial W} \right) \frac{dW}{dD} \\ &\quad + \left( \left( \frac{\partial I}{\partial X} \right) + \lambda_{flow}^T \frac{\partial R}{\partial X} + \lambda_{mesh}^T \frac{\partial T}{\partial X} \right) \frac{dX}{dD} \end{aligned} \tag{6.19}$$

Therefore, if appropriate  $\lambda_{flow}$  and  $\lambda_{mesh}$  have been found to satisfy the following equations, we have:

$$\left( \frac{\partial R}{\partial W} \right)^T \lambda_{Flow} = - \left( \frac{\partial I}{\partial W} \right)^T \quad 6.20$$

$$\left( \frac{\partial T}{\partial X} \right)^T \lambda_{Mesh}^T = - \left( \frac{\partial I}{\partial X} \right)^T - \left( \frac{\partial R}{\partial X} \right)^T \lambda_{Flow} \quad 6.21$$

After solving Equations 6.20 and 6.21, the derivative of the objective function could be written as:

$$\frac{dI}{dD} = \frac{dL}{dD} = \left( \frac{\partial I}{\partial D} + \lambda_{Flow}^T \frac{\partial R}{\partial D} + \lambda_{Mesh}^T \frac{\partial T}{\partial D} \right) \quad 6.22$$

For the shape design parameters, the terms  $\frac{\partial I}{\partial D}$  and  $\frac{\partial R}{\partial D}$  are zero. Therefore, the derivative is:

$$\frac{dI}{dD} = \frac{dL}{dD} = \lambda_{Mesh}^T \frac{\partial T}{\partial D} \quad 6.23$$

Therefore, the derivatives will be dependent on the mesh adjoint operator solved by Equations 6.21 and 6.22 and  $\frac{\partial T}{\partial D}$ . For the  $T(X, D)$ , the spring analogy or linear elasticity would be considered (Dwight 2009; Illic *et al.* 2012). The simplified form of the deformation is then written as:

$$X_s - K \cdot X_v = 0 \quad 6.24$$

where  $K$  is the stiffness matrix of the mesh deformation system,  $X_s$  and  $X_v$  indicate surface mesh points and volume mesh points respectively.

Therefore, the derivatives in Equations 6.21 and 6.23 are corresponding to:

$$\frac{\partial T}{\partial X} = -K \quad \frac{\partial T}{\partial D} = \frac{\partial X_s}{\partial D} \quad 6.25$$

Therefore, the mesh adjoint  $\lambda_{mesh}$  is related to the sensitivities of the objective function on the surface. The derivative of the objective function then depends on the production of  $\lambda_{mesh}$  and the surface sensitivities with respect to the design variables  $\frac{\partial X_s}{\partial D}, \frac{\partial X_s}{\partial D}$  that have been presented in Chapter 2. This approach has been implemented in the TAU development version (Widhalm *et al.* 2010; Illic *et al.* 2012). Figure 6.1 and 6.2 show the vector of  $\lambda_{mesh}$  over the surface of an RAE 2822 aerofoil solved under Mach 0.73 with the SA turbulence model. Figure 6.3 is the pressure distribution of this aerofoil for comparison.

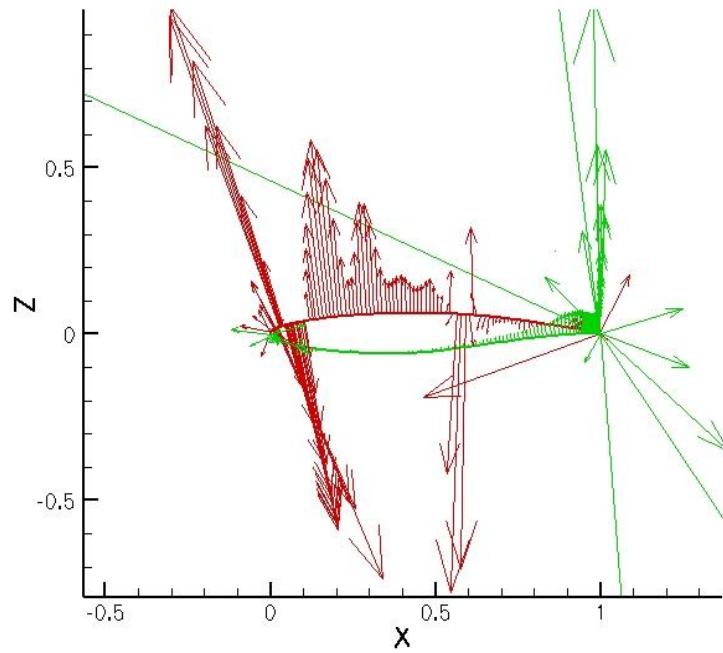


Figure 6.1 The vector of  $\lambda_{\text{mesh}}$  over the surface of RAE 2822 aerofoil for drag

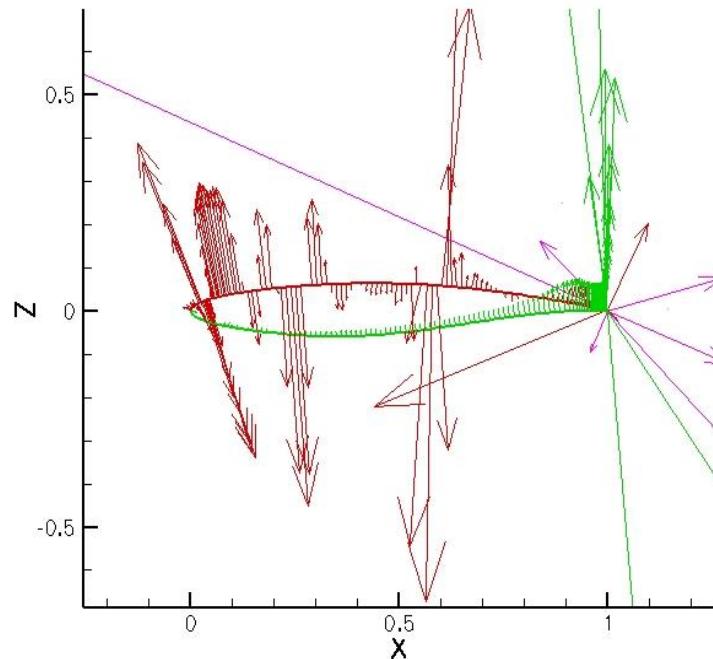


Figure 6.2 The vector of  $\lambda_{\text{mesh}}$  over the surface of RAE 2822 aerofoil for lift

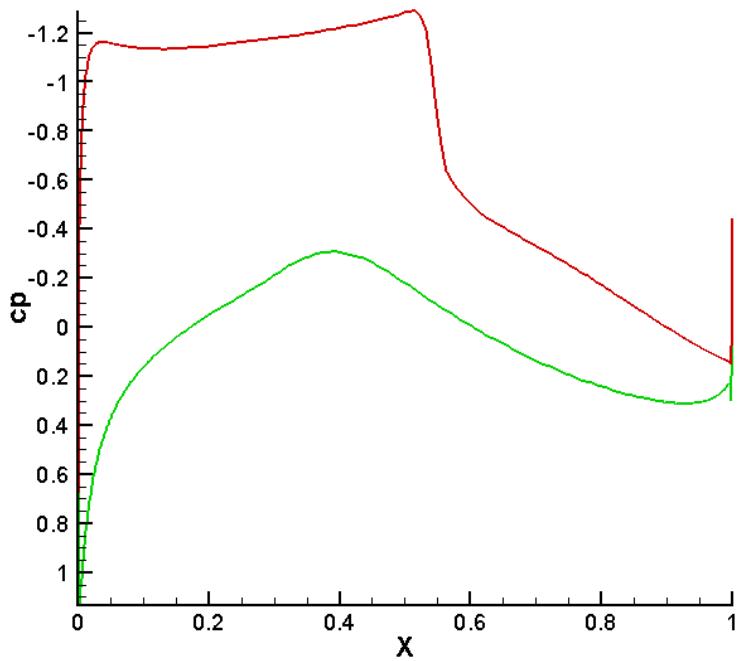


Figure 6.3 The pressure distribution of RAE 2822

Figure 6.1 and 6.2 imply that if the surface points move along the vector direction, the value of the objective function will be increased. In the shock region, it clearly shows the gradient changed in drag and lift. Therefore, this approach could also be employed for guidance for the designer to identify the key region on the surface which is needs to be modified. A quick gradient validation has been performed by comparing the sensitivities obtained by the finite-difference method, adjoint with grid sensitivities using finite difference method, which is denoted as ‘Volgrad’, and the mesh adjoint method, which is denoted as ‘Surfgrad’. The adjoint method with frozen turbulence viscous, that is keeping the turbulent viscosity constant, is also tested. The results are shown in Figure 6.4 and 6.5.

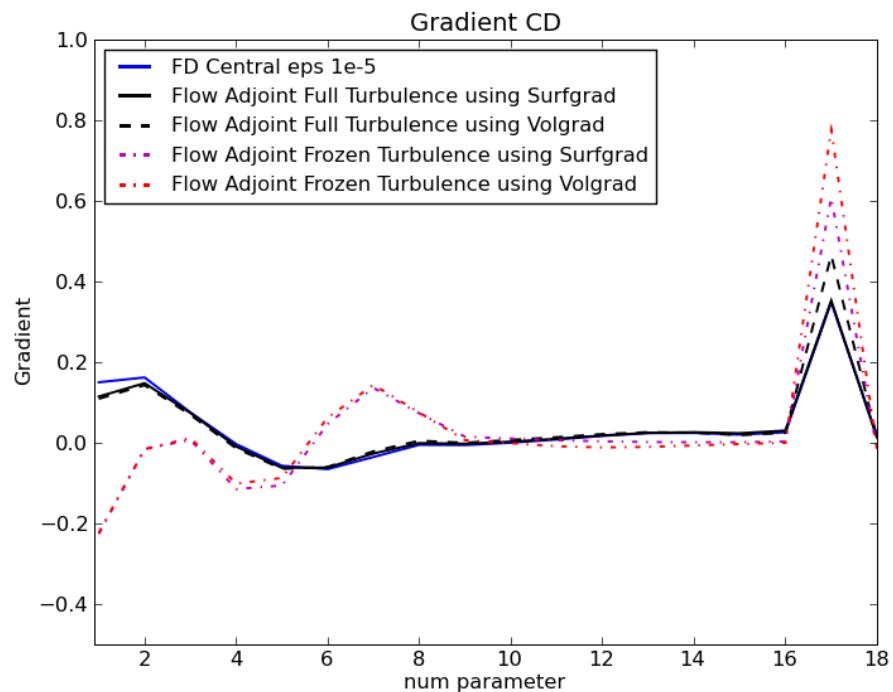


Figure 6.4 Validation of gradient of Cd for RAE 2822 using CST 7<sup>th</sup> order

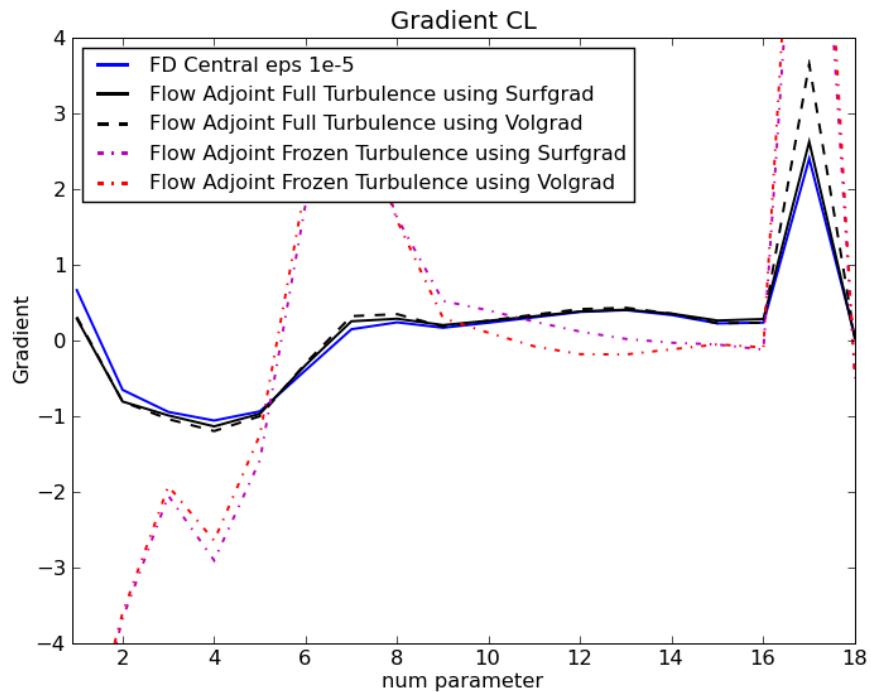


Figure 6.5 Validation of gradient of Cl for RAE 2822 using CST 7<sup>th</sup> order

The comparison results firstly show that the gradients obtained by the adjoint method with full turbulence model are much closer to the finite difference results. However, there are slight errors in the first two parameters. These may be caused by perturbation step size and flow solver convergence in the finite difference method. Hence, the gradients obtained by full turbulence adjoint are considered as accurate. The gradient obtained by the frozen turbulence model is very different from the finite difference case; the signs of the gradient of some parameters are even opposite to finite difference results. This illustrated that the turbulence model could significantly affect the accuracy of gradient of adjoint methods. Therefore, in this thesis, all adjoint solvers must use the full turbulence model.

Secondly, the results illustrate that Volgrad and Surfgrad in general have good agreement. This proves that the adjoint method with mesh deformation adjoint is able to provide corrected gradient without using volume mesh point sensitivities. There is a slight difference at the 17<sup>th</sup> parameter, which is the trailing edge thickness. For this parameter Surfgrad is closer to the finite difference result than Volgrad. This error on Volgrad may because of the perturbation step size of the finite difference method in mesh deformation. This demonstrates that the gradient obtained by Surfgrad would be even more accurate than the Volgrad method, because Surfgrad could avoid the error occurring in the finite difference method from round-off error and step size uncertainty. Therefore, the Surfgrad method is preferred in this thesis when the surface mesh sensitivities could be obtained inexpensively.

### **6.3 Numerical optimisation**

The calculation of gradient of the aerodynamic objective has been reviewed and studied in the above sections. In gradient optimisation, once the gradient has been obtained the search direction in the design space can be given and the design variable can be then updated through numerical optimiser. In this section, the non-linear gradient based optimisation approach is presented.

An unconstrained optimisation could be presented as:

$$\min (F(x)) \quad 6.26$$

where  $x$  is the design variable vector

$$x = [x_1, x_2, \dots, x_{NDV}] \quad 6.27$$

The most gradient optimisation methods use the *line search* strategy (Nocedal and Wright 2006) in which the algorithm finds a direction  $p_k$  and searches along this direction from the current iteration  $x_k$  to the next iteration with lower objective function value. The update of the design variables is:

$$x_{k+1} = x_k + \alpha_k s_k \quad 6.28$$

where  $s_k$  is the search direction in the design space and  $\alpha_k$  is a coefficient which is to control the step size.

The search direction could be evaluated using a zero-order method which is only based on the objective function value. This method is not efficient since it requires many evaluations of the objective function. A more efficient method is based on the gradient of the objective function  $\nabla F(x)$ , which is a first-order method. The steepest descent method is a kind of first order method and has been used in aerodynamic optimisation, in which the search direction is:

$$s_k = -\nabla F(x_k) \quad 6.29$$

In order to further improve the efficiency of optimisation, a second-order method could be used such as quasi-Newton. In this method, the objective function information will be employed to construct a local model function around  $x_k$ . The local model is restricted in some region around  $x_k$ . In the other words, the search direction will be approximated by solving a sub-problem. The typical quasi-Newton method is sequential quadratic programming (SQP). In SQP optimisation, the search direction is approximated by solving a sub-problem with a quadratic model. In addition, SQP optimisation is able to handle non-linear inequality and equality constraints. Therefore, this quadratic model for an optimisation case with inequality and equality constraints could be presented as:

$$\begin{aligned} \min \tilde{F}(s) &= F(x_k) + \nabla F(x_k)^T \cdot s + \frac{1}{2} s^T H_k s \\ \text{subject to } &\nabla c_i^{ie}(x_k) \cdot s + \delta_i c_i^{ie}(x_k) \leq 0 \\ &\nabla c_j^e(x_k) \cdot s + \delta_j c_j^e(x_k) = 0 \end{aligned} \quad 6.30$$

where  $c_i^{ie}$  is the i-th inequality constraint function and  $c_j^e$  is the j-th equality constraint function. The design variables are the components of  $s$  and the optimum is the search direction  $s_k$ .  $H_k$  is the matrix which is a positive definite matrix. This matrix is the identity matrix at the first iteration of the optimisation and is updated during the optimisation. The matrix  $H_k$  is obtained at the end of previous iteration. Once the gradient of the objective function is evaluated, the search direction can be calculated using:

$$s_{k+1} = -H_k \nabla F(x_k) \quad 6.31$$

In addition, it is declared here that the constrained optimisation process should be simplified by defining a pseudo-objective function in which the penalty function is added.

$$\min \Phi(x) = F(x) + r_i \left[ \sum_i (\max [0, c_i^{ie}(x)]) + \sum_j c_j^e(x) \right] \quad 6.32$$

$\alpha_k$  is determined by a one-dimensional search since it has to be able to reduce sufficiently the objective value. This is done by polynomial approximation.

The  $H_k$  matrix is updated using an Broydon-Fletcher-Goldfarb-Shanno (BFGS) update formula. In this method,  $H_{k+1}$  for the next iteration is defined as:

$$H_{k+1} = H_k - \frac{H_k p p^T H_k}{p^T H_k p} + \frac{y y^T}{y^T p} \quad 6.33$$

where

$$\begin{aligned} p &= x_{k+1} - x_k \\ y &= \theta \tilde{y} + (1 - \theta) H_k p \\ \tilde{y} &= \nabla L(x_{k+1}, \lambda_k) - \nabla L(x_k, \lambda_k) \end{aligned} \quad 6.34$$

$L(x, \lambda)$  is the Lagrangian function:

$$L(x, \lambda) = F(x) + \sum_i \lambda_i c_i^{ie}(x) + \sum_j \lambda_j c_j^e(x) \quad 6.35$$

and

$$\theta = \begin{cases} 1 & \text{if } p^T \tilde{y} \geq 0.2 p^T H_k p \\ \frac{0.8 p^T H_k p}{p^T H_k p - p^T \tilde{y}} & \text{if } p^T \tilde{y} < 0.2 p^T H_k p \end{cases} \quad 6.36$$

## 6.4 Mesh deformation

In aerodynamic optimisation, the geometry surface will be modified during optimisation iteration. Therefore, the computational grid needs to be updated with the surface change. Mesh update is key in aerodynamic optimisation, and is required to be robust and fairly cheap.

One option is to regenerate the entire computational grid with a mesh generator. However, this requires that the new generated mesh has the same topology and number of nodes as the previous mesh, because the meshes that have different topologies and number of nodes have different discretisation errors, and this will affect the accuracy level of the CFD solver and bring numerical noise into optimisation space. Otherwise, the mesh must be sufficiently fine. Under this requirement, this method is only feasible for a structured mesh with simple geometry. Since the unstructured method mesh is used in this thesis, the mesh deformation technique will be considered.

For the structured mesh, the most common mesh deformation method is based on an interpolation method. For example, Le Moigne (2002) used the structured mesh deformation where the volume mesh is considered individually with every grid line originating from the internal surface grid already updated and linking the outer far field boundary. Because this method depends on the grid connectivity information, it only can be applied on a structured mesh. Since we are not interested in structured mesh in this thesis, the details can be found in Allen (2002), Le Moigne (2002), Pandya and Baysal (1997) and Burgreen *et al.* (1994).

For the unstructured mesh, the most common method is the spring analogy method. It was first proposed by Batina (1991). In spring analogy method, a spring stiffness  $k_{ij}$  model is applied between two adjacent grid nodes with a spring as:

$$k_{ij} = \frac{1}{|\Delta r_{ij}|} \quad 6.37$$

where  $\Delta r_{ij}$  is the length of the grid node i to grid node j.

A large equilibrium system will be obtained by applying stiffness to the entire mesh domain. The deformation of surface will be applied as boundary movement and propagated to the entire equilibrium system. Jacobian iteration is normally employed to solve this equilibrium until the converged state. The original spring analogy method is not efficient, has convergence difficulty and could cause mesh quality problems when the boundary movement is large. The reason is that the original stiffness model only employed the length, which is able to prevent the adjacent nodes from approaching each other but is unable to prevent any edge-edge, edge-face and face-face intersection.

Therefore, an enhanced spring tension model is proposed by Farhat *et al.* (1998) to prevent cell intersection. Xia and Qin (2005) suggested a simple and robust stiffness tension model, which includes the cell volume surrounding the edge  $ij$ . The stiffness tension model is then defined as:

$$k_{ij} = \frac{1}{|\Delta r_{ij}|} + q \sum_{k \in \mathcal{N}} \frac{1}{V_k^p} \quad 6.38$$

where  $\mathcal{N}$  is the set of cells surrounding the edge  $ij$ ,  $V_k^p$  is the cell volume of cell  $k$  and  $p,q$  are the constants and normally set to  $p=2$  and  $q=1$ .

Since the cell volume is involved in the stiffness model, the negative volumes of cell could be prevented. Liu *et al.* (2006) proposed a method based on the Delaunay graph mapping technique. This method is significantly cheaper, faster and more efficient than spring tension analogy methods. However, it has problem when the large rotation deformation is involved since the Delaunay topology may be changed.

Because the spring tension method lacks robustness for large deformations, Nielsen and Anderson (2002) proposed another method based on modified linear elasticity theory in which the mesh cell is considered as an elastic solid. Dwight (2009) has presented a robustness linear elasticity method. In linear elasticity, the small displacement vector  $u = (\Delta x, \Delta y, \Delta z)$  is subject to body forces and surface tractions. The governing equation is defined as:

$$\nabla \sigma = f \quad 6.39$$

where  $f$  is body force and  $\sigma$  is the stress tensor.

The governing equation is introduced into the entire computational domain. The stress tensor relates to the strain tensor  $\epsilon$ , which is defined as

$$\sigma = \lambda \text{Tr}(\epsilon)I + 2\mu\epsilon \quad 6.40$$

where  $\text{Tr}$  is the trace,  $\epsilon$  is the linear kinematic law and  $\lambda$  and  $\mu$  are the Lamé constants with respect to the elastic material. They are defined in terms of Young's modulus  $E$  and Poisson's ratio  $v$ :

$$\lambda = \frac{vE}{(1+v)(1-2v)} \quad \mu = \frac{E}{2(1+v)} \quad 6.41$$

$E$  is defined as a large positive number for stiffness of the material. Poisson's ratio  $v$  is selected between -1 and 0.5 due to material shrinkage. The linear kinematic law is defined as:

$$\epsilon = \frac{1}{2}(\nabla u + \nabla u^T) \quad 6.42$$

The linear elasticity governing equation is discretised on CFD mesh using finite element methods. Nielsen and Anderson (2002) and Dwight (2009) demonstrated that the linear elasticity method is a very robust method compared with the spring tension method and allows for very large mesh deformation.

However, these methods are computationally expensive, since it is necessary to solve a large linear system. In recent years, a new method based on interpolation is becoming more interesting in research. The radial basis function (RBF) interpolation function is derived to satisfy the translation, rotation invariant and linear displacement in displacement. The RBF method was first employed by Beckert and Wendland (2001) in fluid-structure interaction applications. de Boor *et al.* (2007) have presented and tested the RBF mesh deformation methods for 2D and 3D unstructured mesh. In their test, the RBF mesh deformation has been found to only require the solving of a small system of equations and no grid-connectivity information is needed. In addition, the  $C^2$  continuous surface could be obtained by using thin plate spline RBF function. Therefore, it is a fairly robust and efficient method for unstructured mesh deformation. Jakobsson and Amoignon (2007) have successfully applied the RBF method in Euler adjoint optimisation methods, and have proved that RBF is significantly efficient and robust in aerodynamic optimisation processes.

Furthermore, the RBF deformation is controlled by the base points. It is not necessary for these base points to be surface grid points. In the above presented methods, the deformation information must be located on all CFD surface grid nodes. This is significant for the fluid-structure interaction issue, because the deformation information in fluid-structure interaction is given by the finite element solver, in which the locations of deformation nodes are normally different from CFD surface grid. To interpolate this deformation information from finite elements to every CFD surface node is very difficult and errors may occur. In aerodynamic optimisation, the problem is similar since the deformation may be given by the CAD system or the different coordinates from parameterisation methods. If the above mesh deformation techniques are employed, the deformation information must be interpolated into every CFD surface node, or the deformation information must be generated directly respect to the CFD surface node coordinates in which the coordinate transformation is normally needed. Errors could occur during interpolation and coordinate transformation. However, the RBF method is able to avoid this problem.

The radial basis functions approach is a well-established interpolation method for gridded and scattered data, whereas the most natural context for function approximation is given for scattered data (Beckert and Wendland 2001; Buhmann 2003). In the field of computational fluid dynamics (CFD) it is often used for coupling CFD-grids to finite element structure grids.

The input data in  $d$  dimensions consists of data locations  $x_i$ , merged into the dataset:

$$x = \{x_1, x_2, \dots, x_n\} \in \mathbb{R}^d \quad 6.43$$

and the corresponding function values:

$$f_i = f_i(x_i) \in \mathbb{R} \quad i = 1, 2, \dots, n \quad 6.44$$

The data locations  $x_i$  are called centers or ‘base points’. The goal is to interpolate the function values between the base points by an approximant  $s: \mathbb{R}^d \rightarrow \mathbb{R}$  to satisfy the condition:

$$s|_x = f|_x \quad 6.45$$

In this specific case  $s$  is a linear combination of shifted radially symmetric basis functions  $\phi$ . Radially symmetric means that the value of  $\phi(\cdot)$  depends only on the distance of the argument to the origin, hence it is often written as  $\phi(\|\cdot\|)$ . The distance norm is usually the Euclidean norm:

$$\|\mathbf{x}\| = \sqrt{\sum_{i=1}^3 x_i^2} \quad 6.46$$

$s(\mathbf{x})$  has the general form:

$$s(\mathbf{x}) = \sum_{i=1}^n \alpha_i \phi(\|\mathbf{x} - \mathbf{x}_i\|) \quad 6.47$$

Setting  $s(\mathbf{x}_i)$  equal to  $f_i$  for all  $i \in \{1, \dots, n\}$  leads to the linear system:

$$A\mathbf{y} = \mathbf{b} \quad 6.48$$

with

$$A = \begin{bmatrix} \phi_{11} & \dots & \phi_{1n} \\ \vdots & \ddots & \vdots \\ \phi_{n1} & \dots & \phi_{nn} \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} f_1 \\ \vdots \\ f_n \end{bmatrix} \quad 6.49$$

where

$$\phi_{ij} = \phi(\|\mathbf{x}_i - \mathbf{x}_j\|) \quad 1 \leq i, j \leq n \quad 6.50$$

A unique interpolation is usually (for most  $f$ ) guaranteed if the base points are all distinct and there are at least two of them (Baxter 1992). A few examples for the radial basis functions are given in Table 6.1 (Rendall and Allen 2009).

Table 6.1 Common radial basis function

Radial Basis Function	$\phi(r)$
Volume Spline	$\phi(r) = r$
Gaussian	$\phi(r) = e^{-ar}$
Thin plate spline	$\phi(r) = r \ln r$
Wendlands C0	$\phi(r) = (1-r)^2$
Wendlands C2	$\phi(r) = (1-r)^4(4r+1)$
Euclid's Hat	$\phi(r) = \pi[\left(\frac{1}{12}r^3\right) - a^2r + (\frac{4}{3}a^3)]$

Rendall and Allen (2009) pointed out that the radial basis functions can be classified as global, local and compact. The global functions have zero or non zero value at radial point and the value of function grows moving away from radial point; examples are volume spline, thin plate spline. The local functions have non-zero value at radial point and decay moving away from radial point; examples are Wendlands functions and Gaussian. The compact function shares the properties of local functions, however, its value can reach to zero at certain distance from radial point in terms of additional parameters; example is Euclid's Hat. Global function can handle large deformation, however, it could affects the far field boundary surface. The local function is normally able to provide smooth shape.

An important attribute of this interpolation method is the possibility to expand the approach of Equation 6.47 by adding a polynomial to the definition without losing the uniqueness of the coefficients. For function values  $f_i$ , which show a polynomial character, the appended polynomial improves the interpolation quality and handle the rotation deformation. The only restriction is that the polynomial must have a degree  $m \geq 1$  and is non-zero at all base points. This leads to:

$$s(\mathbf{x}) = \sum_{i=1}^n \alpha_i \phi(\|\mathbf{x} - \mathbf{x}_i\|) + p(\mathbf{x}) \quad 6.51$$

The coefficients can be computed by solving:

$$s(\mathbf{x}_i) = \sum_{i=1}^n \alpha_i \phi(\|\mathbf{x} - \mathbf{x}_i\|) + p(\mathbf{x}) = f_i \quad 6.52$$

$$\sum_{i=1}^n \alpha_i q(\mathbf{x}_i) = 0 \quad \forall q: \deg(q) \leq \deg(p) \quad 6.53$$

The extra equation takes up the extra degrees of freedom given by the polynomial coefficients to allow a unique interpolation. The uniqueness can be guaranteed if  $\phi$  is 'conditionally positive definite'. More details on the theory of this topic can be found in Buhmann (2003).

Again, the requirements on  $X$  are not very strong. For a linear polynomial,  $X$  must only contain four base points, which do not lie on a plane. Furthermore, if the function values  $f_i$  at the base points were generated by a linear function, they would be reproduced exactly by the linear polynomial (Beckert and Wendland 2001).

In the following, the dimension is set to  $d = 3$  in this document. Since  $x = (x_x, x_y, x_z)$ , the polynomial is linear and can be written as:

$$p(x) = \beta_0 + \beta_1 x_x + \beta_2 x_y + \beta_3 x_z \quad 6.54$$

So Equation 6.51 can be abstracted to matrix notations:

$$Hy = b \quad 6.55$$

with

$$H = \begin{bmatrix} A & P \\ P^T & 0 \end{bmatrix} \quad 6.56$$

$$A = \begin{bmatrix} \phi_{11} & \dots & \phi_{1n} \\ \vdots & \ddots & \vdots \\ \phi_{n1} & \dots & \phi_{nn} \end{bmatrix} \quad 6.57$$

$$P = \begin{bmatrix} 1 & x_1 & y_1 & z_1 \\ \vdots & \ddots & & \vdots \\ 1 & x_n & y_n & z_n \end{bmatrix} \quad 6.58$$

$$y = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_n \\ \beta_1 \\ \vdots \\ \beta_4 \end{bmatrix} \quad 6.59$$

$$b = \begin{bmatrix} f \end{bmatrix} = \begin{bmatrix} f_1 \\ \vdots \\ f_n \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad 6.60$$

Given that the interpolation matrix  $A$  is invertible, we can solve for  $y$  and  $b$  to obtain:

$$\alpha = A^{-1}f - A^{-1}PA_pP^TA^{-1}f \quad 6.61$$

and

$$\beta = A_pP^TA^{-1}f \quad 6.62$$

where

$$A_P = (P^T A^{-1} P)^{-1} \quad 6.63$$

Only one-dimensional function values have been presented so far, while in this case the function values are three-dimensional. Therefore each coordinate of the mesh nodes has to be interpolated separately. Now we apply this type of approximation to the displacements  $\Delta\mathbf{x}_x$ ,  $\Delta\mathbf{x}_y$  and  $\Delta\mathbf{x}_z$  in each coordinate direction:

$$\Delta\mathbf{x}_x = s_x(\mathbf{x}) = \sum_{i=1}^n \alpha_i^x \phi(\|\mathbf{x} - \mathbf{x}_i\|) + \beta_1^x + \beta_2^x x + \beta_3^x y + \beta_4^x z \quad 6.64$$

$$\Delta\mathbf{x}_y = s_y(\mathbf{x}) = \sum_{i=1}^n \alpha_i^y \phi(\|\mathbf{x} - \mathbf{x}_i\|) + \beta_1^y + \beta_2^y x + \beta_3^y y + \beta_4^y z \quad 6.65$$

$$\Delta\mathbf{x}_z = s_z(\mathbf{x}) = \sum_{i=1}^n \alpha_i^z \phi(\|\mathbf{x} - \mathbf{x}_i\|) + \beta_1^z + \beta_2^z x + \beta_3^z y + \beta_4^z z \quad 6.66$$

In addition, in the basic RBF, the deformation of different boundaries could influence each other since the base point could control the entire domain. Therefore, an extra algorithm is employed to prevent this, which is called *group-weighting* and *deformation blending*.

A *group-weighting* approach is used to allow the independent movement of different model parts/boundaries in the grid. Otherwise the deformations of different boundaries could influence each other and unintentional surface deformation would be the result. Separating the interpolation by a group protects the shape of the different bodies. Therefore, the interpolation matrix  $H^g$  of each group  $g$  has to be computed and applied to the grid nodes separately. Finally, the deformation result for each grid point is calculated by a weighted average of each group-deformation result.

The deformation-blending approach supports the protection of boundary layer cells and the usage of radial basis functions  $\phi(\|\mathbf{x}\|)$  with limits  $\phi(\|\mathbf{x}\|) \rightarrow \infty$  for  $\|\mathbf{x}\| \rightarrow \infty$ . These radial basis functions, which increase with increasing distance from the base point of a deforming body, need to be restricted further away from the surface of this body. Otherwise local deformations would influence the whole mesh. Additionally, the added

polynomial of the interpolation approach would deform the whole volume mesh. Consequently, this approach, which is implemented to recover linear deformations exactly, cannot be used without the blending of deformation values.

Hence, the notations are expanded by an elevated group index  $g$  for  $n_g$  groups. As input data there are  $n_s^g$  base points  $\mathbf{x}_{s,i}^g \in R^3$  for each group  $g$  merged into the datasets:

$$\mathbf{x}_s^g = \left\{ \mathbf{x}_{s,1}^g, \mathbf{x}_{s,2}^g, \dots, \mathbf{x}_{s,n_s^g}^g \right\} \text{ with } g = 1, \dots, n_g \quad 6.67$$

The function values that are going to be interpolated are the deformation vectors:

$$\Delta \mathbf{x}_{s,i}^g = \Delta \mathbf{x}(\mathbf{x}_{s,i}^g) = \begin{pmatrix} \Delta \mathbf{x}_{s,i}^{g,x} \\ \Delta \mathbf{x}_{s,i}^{g,y} \\ \Delta \mathbf{x}_{s,i}^{g,z} \end{pmatrix} \in R^3 \text{ with } i = 1, \dots, n_s^g, g = 1, \dots, n_g \quad 6.68$$

the interpolation matrix  $H^g$  in equation 6.55 has to be computed only once for each boundary group instead of computing it for each dimension separately, since the matrix depends only on the base points  $\mathbf{x}_{s,i}^g$  and the chosen radial basis function  $\phi$ . So the interpolation matrices  $H^g$  for each group can be stated as:

$$H^g = H(\mathbf{x}_s^g, \phi) \quad 6.69$$

For each dimension  $k \in \{x, y, z\}$ , the interpolation coefficients  $\alpha^{g,k} = [\alpha_1^{g,k}, \alpha_2^{g,k}, \dots, \alpha_{n_s^g}^{g,k}]^T$  and  $\beta^{g,k} = [\beta_1^{g,k}, \beta_2^{g,k}, \dots, \beta_{n_s^g}^{g,k}]^T$  can be calculated by inverting Equation 6.69:

$$\begin{pmatrix} \alpha^{g,k} \\ \beta^{g,k} \end{pmatrix} = (H^g)^{-1}(\mathbf{b}^{g,k}) \quad 6.70$$

where

$$\mathbf{b}^{g,k} = \begin{bmatrix} f_1^{g,k} \\ \vdots \\ f_n^{g,k} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad 6.71$$

The actual interpolation algorithm calculates the deformations of the grid nodes:

$$\mathbf{d}\mathbf{x}_{v,i} = (dx_{v,i}^x, dx_{v,i}^y, dx_{v,i}^z)^T \quad 6.72$$

For the volume mesh grid node  $\mathbf{x}_{v,i}$ , the governing equation for the displacement is:

$$dx_{v,i}^{g,k} = \sum_{j=1}^{n_x^g} \alpha_j^{g,k} \phi(\|x_{v,i} - x_{s,j}^g\|) + \beta_1^{g,k} + \beta_2^{g,k} x_{v,i} + \beta_3^{g,k} y_{v,i} + \beta_4^{g,k} z_{v,i} \quad 6.73$$

Two new functions have been introduced: the blending function blend and the weighting function weight. The weighting function averages the individual group deformations. Because its limit for  $d$  goes to zero, it needs a small value cut-off for numerical reasons:

$$f_{blend}(d_i^g, g) = \begin{cases} 0 & d_i^g > RZW^g \\ 1 & d_i^g < RFW^g \\ \frac{RZW^g - d_i^g}{RZW^g - RFW^g} & \text{else} \end{cases} \quad 6.74$$

$$f_{weight}(d_i^g, g) = \frac{1}{\sqrt{\max\{d_i^g, 10^{-10}\}}} \quad 6.75$$

where  $d_i^g$  is the distance from grid node  $x_{v,i}$  to the nearest surface of group  $g$ ,  $RZW^g$  is Radius Zero Weight and  $RFW^g$  is Radius Full Weight, which are controlling the deformation of the grid nodes. Therefore, the final displacement value for  $x_{v,i}$  is defined:

$$dx_{v,i} = \frac{\sum_{g=1}^{n_g} f_{blend}(d_i^g, g) \cdot f_{weight}(d_i^g, g) \cdot dx_{v,i}^{g,k}}{\sum_{g=1}^{n_g} f_{weight}(d_i^g, g)} \quad 6.76$$

The blending function is sketched in Figure 6.6. If a grid node is close to a boundary of group  $g$  with a distance less than  $RFW^g$ , it will move approximately like the boundary. This functionality can be used to conserve the sensitive boundary layer part of a grid. Further away from the boundary with a distance  $d_i^g \geq RZW^g$ , the deformation is zero.

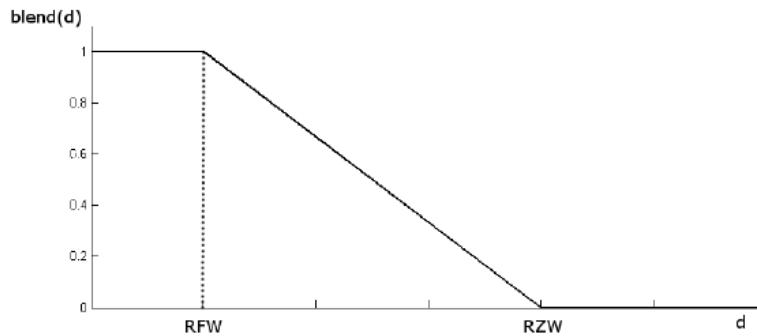


Figure 6.6 Blending function for grid node deformation computation, including the parameter radius full weight (RFW) and radius zero weight (RZW)

After presenting the RBF methodology, the other benefit of using RBF mesh deformation could be found. The calculation of the inverse of the interpolation matrix  $H^{g^{-1}}$  is the most time consuming process in RBF methods. However, the interpolation matrix  $H^g$  only depends on the location of base points. If the base points are not changed during optimisation iteration, the interpolation matrix needs only to be calculated once at the start of optimisation. This could significantly save on computational time for mesh deformation and accelerate the optimisation processing time.

In this thesis work, the TAU solver package provides the RBF mesh deformation. The details of the algorithm of RBF mesh deformation and validation in TAU can be found in Kroll *et al.* (2008). Therefore, further validation will not be provided in this thesis. Furthermore, the performance of TAU deformation has been improved by an algorithm to reduce the number of initial base points. The details can be found in Rendall and Allen (2009).

## 6.5 Optimisation framework

The optimisation framework using Surfgrad adjoint strategy is schematically presented in Figure 6.7.

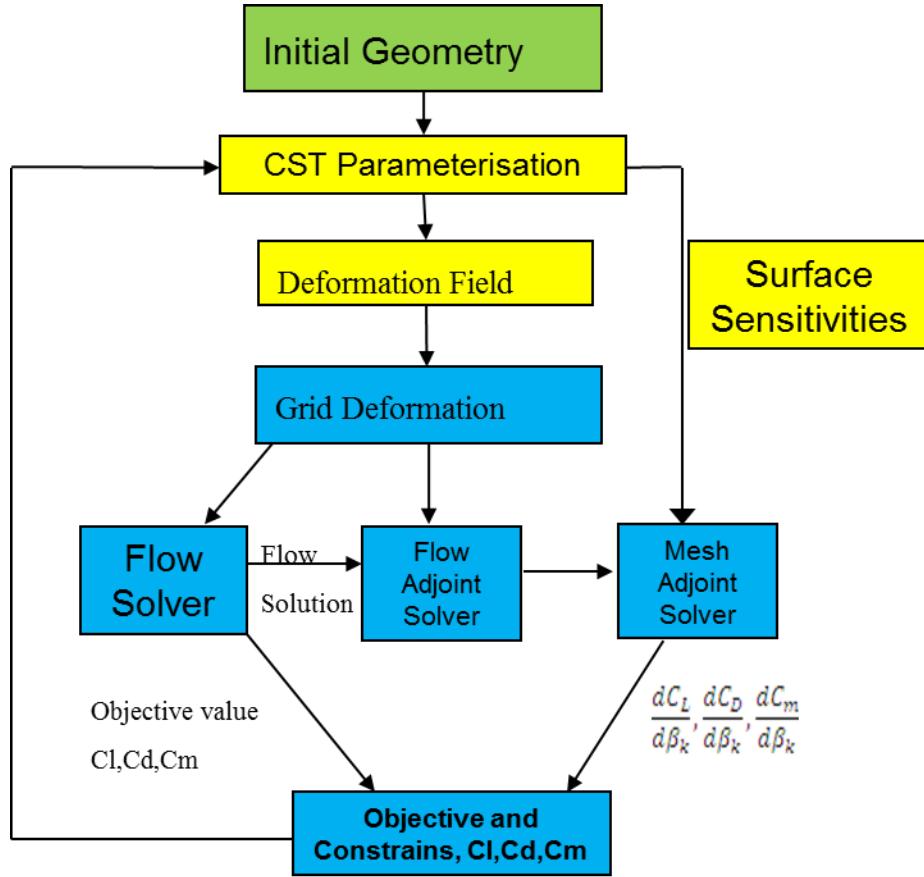


Figure 6.7 Adjoint optimisation framework of Surfgard

This optimisation process is built up by linking different modules using Python programming. The process starts at the initial geometry, and it is then parameterised using CST parameterisation methods to obtain initial design variables. Considering the mesh deformation could generate a slight error due to the mesh deformation algorithm, in order to avoid error accumulation during optimisation the mesh deformation will always use the mesh of the initial geometry. As presented in the mesh deformation subsection, the RBF interpolation matrix can then be kept and used throughout the entire optimisation.

After case is initialised, the optimisation could be run iteratively until its convergence or it hits the stop criterion. In each optimisation iteration step, the process will start at the design variables which are given by the optimiser (at first iteration, these are the initial design variables). The new geometry will be generated and subtracted from the initial

geometry to obtain the deformation base control point, which is the deformation field in Figure 6.7, for RBF mesh deformation. Once the deformation field is obtained, the second module will be called on to carry out RBF mesh deformation, based on the mesh of the initial geometry, to obtain the new mesh.

The next module will then employ the new mesh and run flow solver to simulate the flow field and calculate the objective values. Once the flow solver has finished, the objective values will be sent to the optimiser and the flow residual will be sent to the next module for running the flow adjoint solver due to the requirement of a discrete adjoint solver. In Surfgrad strategy, the solution of the flow adjoint solver will be sent to last module to run mesh adjoint. As presented in Chapter 2, the CST method is differentiable. The analytical surface mesh sensitivities can be obtained and used with the solution of the mesh adjoint solver to calculate the gradient. The gradient information is then sent to the optimiser, which will check the optimisation convergence based on the objective value and gradient. If the optimisation has not yet converged, the optimiser will update the design variables using the algorithm presented in sub-section 6.3 and start a new iteration. In addition, if Volgrad strategy is used, the module for running mesh adjoint solver will be replaced by a module to run NDV times mesh deformation and calculate volume mesh sensitivities using finite difference.

In addition, this optimisation framework is able to record the results at each design iteration and build up a database for future use. Moreover, the failures of the CFD solver and unrealistic geometry are difficult to avoid. When these situations occur, the optimisation framework will give the optimiser a large value to prevent it continually searching in that region.

# Chapter 7 Optimisation in Two-Dimensions

The geometric parameterisation, numerical solver, adjoint approach and mesh deformation have been presented in previous chapters. Therefore, the test of optimisation methods can be carried out. The optimisation test cases in two-dimensions were performed at first to examine the optimisation process and compare the different parameterisation methods. The aerofoil with bump optimisation was then tested.

In the first part, three parameterisation methods presented in Chapter 3, including CST, iCST and RCST, have been employed to perform drag reduction optimisation. In the second part, the performance of three parameterisation methods for bump optimisation, including standard polynomial, CST and PARSEC, have been investigated.

## 7.1 Two-dimensional aerofoil optimisation

In two-dimensional optimisation, the RAE 2822 aerofoil is selected as the test case since it has been widely employed in the aerodynamic research area. The flow condition of this test case is set to  $Mach = 0.75$  and  $Re = 6.5 \times 10^6$ . The optimisation is the constrained drag reduction. Therefore, the optimisation is described as:

$$\begin{aligned} & \min \quad Cd \\ & \text{subject to} \quad Cl \geq 0.75 \\ & \quad V \geq V_0 \end{aligned} \tag{7.1}$$

where  $V$  is the torsion box volume which is the area of the aerofoil between 20% and 60% chord. The initial angle of attack is set to  $\alpha = 2.0^\circ$ , which corresponds to  $Cl=0.7$ . Therefore, this optimisation is to minimise the drag and increase the lift while maintaining the torsion box volume to no less than that of the initial aerofoil. The angle of attach will be employed as an extra design variable to adjust the lift coefficient.

Three parameterisation methods, which are CST, iCST and RCST, have been tested. Two CST methods, which are 7<sup>th</sup> order and 10<sup>th</sup> order, are employed to investigate the effect of the CST polynomial order on optimisation results. Because the geometric fitting results

of the RCST methods presented in Chapter 3 show that it is possible to represent the aerofoil in high accuracy with very low order, the 6<sup>th</sup> order RCST is employed to carry out this test. In order to reduce the complexity of optimisation, the weight rational parameters are fixed during optimisation.

The design variables are obtained from the results showed in Chapter 3. The bound value of each design variable has to be set up. As introduced in Chapters 2 and 3 for all three parameterisation methods, the first design parameter on each surface corresponds to the leading edge radius. Therefore, the lower bound of the first design variable on the upper surface must be positive, and the upper bound of the first design variable on the lower surface must be negative to avoid illusory geometry. In the CST and RCST methods, for the upper surface, the lower bounds of design variables are set to about 20% of their initial value and the upper bounds are set to a relatively high value. Similarly, for the lower surface, the upper bounds are restricted to about 20% and the lower bounds are set to a relatively low value. Furthermore, for the i-th design parameter on the upper surface, the lower bound of this parameter must be greater than the upper bound of the i-th design parameter on the lower surface. The purpose of this extra setting is to avoid upper and lower geometry intersection.

Since iCST is an intuitive method, the bounds of design variables can be set up easily. The range of the leading edge radius can be 80% and 160% of initial value. The x locations of control points  $x_1, x_2$  and crest  $x_{crest}$  are allowed to move up and down by 10% of chord length. The height of control point  $z_1, z_2$  is allowed to move 10% of the initial value and the crest point  $z_{crest}$  is allowed to move 2% of the initial value towards to the centre line of the aerofoil and 40% in the opposite direction. The rest of the design variables are allowed to move by 10% of their initial value, positively and negatively.

The torsion box volume is calculated by numerical integration, and the gradient of the torsion box volume is provided using the finite difference method with a very small step  $\varepsilon = 1 \times 10^{-6}$  at low computational cost.

Furthermore, the number of grids of RAE 2822 is about 90,000 and  $y^+ = 1$  with 1.1 growth rate on the surface normal direction, as shown in Figure 7.1.

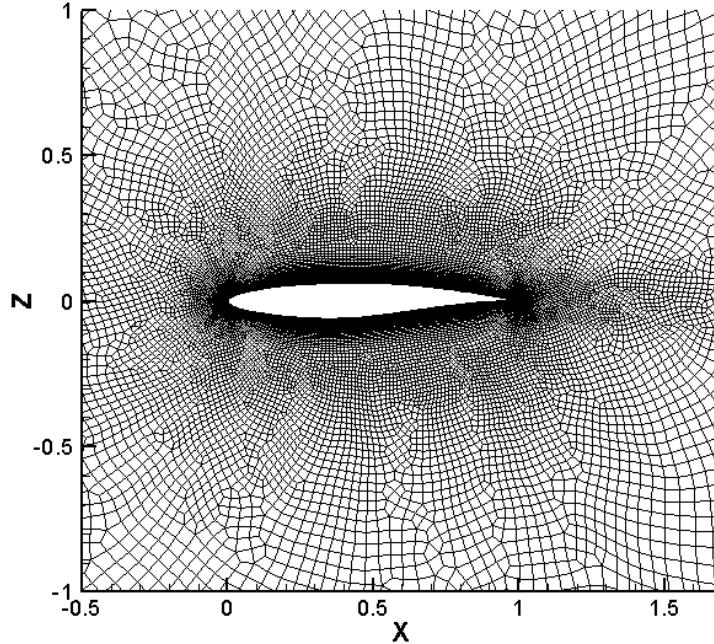


Figure 7.1 Mesh of RAE 2822

The influence of flow solver convergence has been discussed in Le Moigne (2002) and Widhalm *et al.* (2010). The convergence level of the flow solver significantly affects the accuracy of the gradient of adjoint solver. The flow solver is required to converge as much as possible. From their studies, the  $10^{-8}$  order convergence will provide a reasonable high accuracy of gradient. Therefore, in this work, all flow results are converged at least to  $10^{-8}$ . The convergence of the adjoint solver obviously affects the accuracy of gradient. Hence, the adjoint solver must be converged to at least  $10^{-6}$ . Figure 7.2 shows the optimisation convergence history of Cd and Cl of the 7<sup>th</sup> order CST. Figure 7.3 shows the optimisation convergence history of Cd and Cl of the 10<sup>th</sup> order CST.

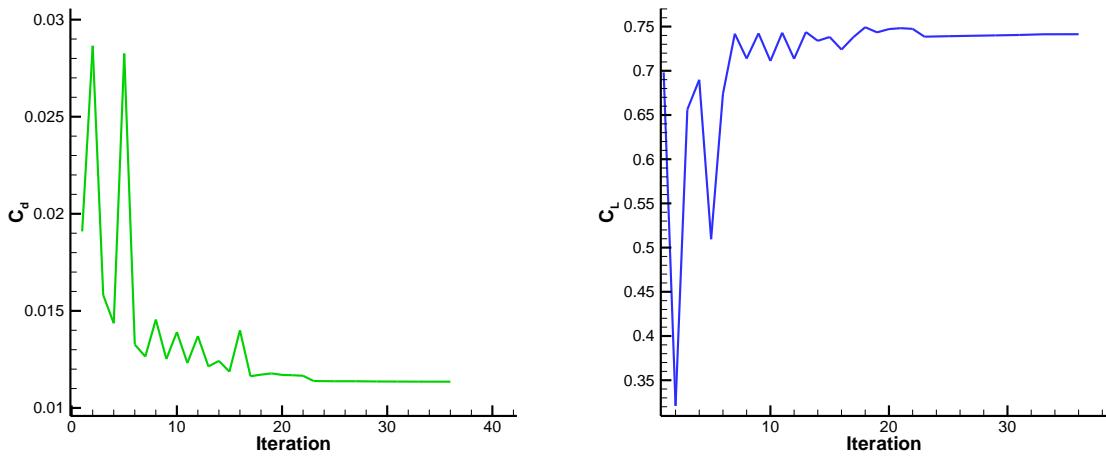


Figure 7.2 Cd (left) and Cl (right) optimisation history of 7<sup>th</sup> order CST

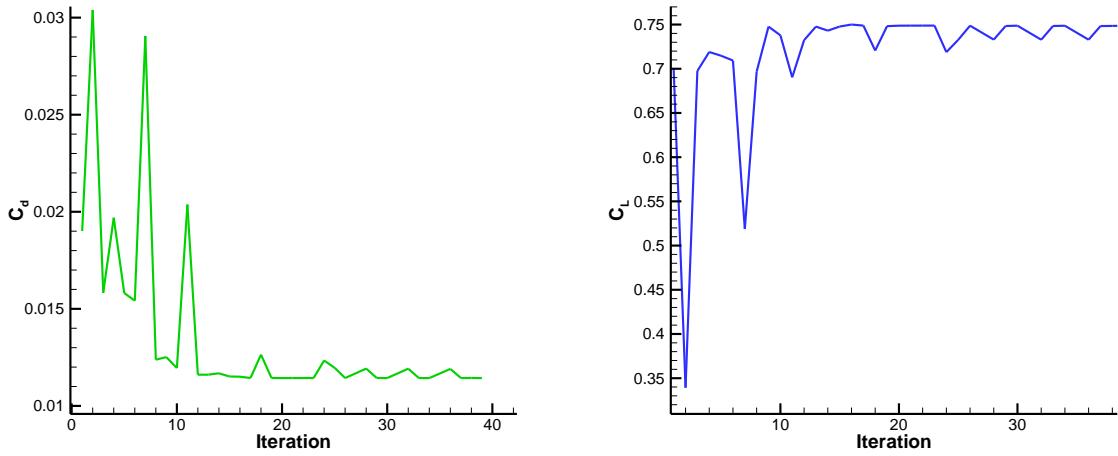


Figure 7.3 Cd (left) and Cl (right) optimisation history of 10<sup>th</sup> order CST

At the start of optimisation, a large spike is occurred which has an unexpected drag increase. During optimisation, there are also some large spikes before iteration 10. This is because the optimiser lacks information to build up a sequential model at the first iteration. The other reason is that the lift coefficient in this case is 0.7, and it is not satisfied constraint. Consequently, the optimiser perturbs the design variables with a large step in order to reach the lift constraint, and this large step will generate unexpected aerofoil with large drag and low lift. Once the spike is occurred, the SQP optimiser will be restarted with a smaller step. Therefore, the objective value will be back to normal

after spike. After about 5 iterations, this is mitigated. The convergence history shows that the 7<sup>th</sup> order CST reaches convergence after 20 iterations. The 10<sup>th</sup> order CST shows that the value changes periodically after 20 iterations and the minimum value will not be reduced. This may be because the parameterisation has more local control which requires long optimisation iterations so that all design variables converge. Therefore, the lower order of the CST methods could be more efficient in optimisation convergence. Table 7.1 shows the values of aerodynamic coefficients and volume constraints of the initial aerofoil and optimal results.

Table 7.1 Aerodynamic coefficients and constraint values

	Cl	Cd	Cl/Cd	Volume
Initial	0.6982	0.01909	36.57	0.0453
7th order CST	0.7414	0.01135	65.30	0.0453
10th order CST	0.7487	0.01143	65.48	0.0453

Both optimum aerofoils satisfy the volume geometry constraints. However, both cases do not exceed 0.75 lift coefficient constraints. The optimum aerofoil of 10<sup>th</sup> order CST for Cl is 0.7487 which is higher than the 7<sup>th</sup> order CST. Even if its Cd is 1 drag count more than the 7<sup>th</sup> order CST, the 10<sup>th</sup> order CST still obtains a higher lift-to-drag ratio. However, both cases achieve a 40% drag reduction which is considered as a significant improvement. Figure 7.4 and 7.5 show the Cp contour of the flow field of initial and optimum aerofoils obtained by 7<sup>th</sup> order CST and 10<sup>th</sup> order CST methods, respectively.

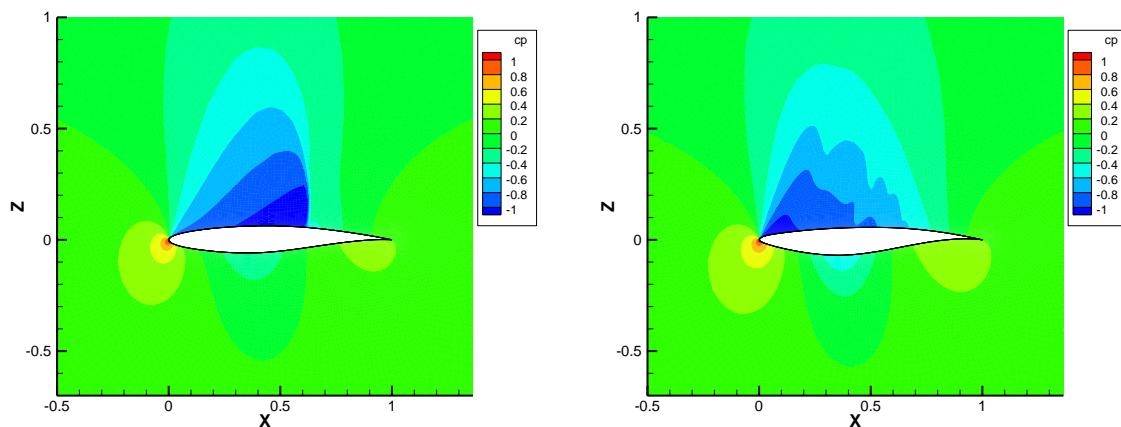


Figure 7.4 The contour of pressure coefficients of initial aerofoil (left) and optimum aerofoil (right) obtained by 7<sup>th</sup> order CST

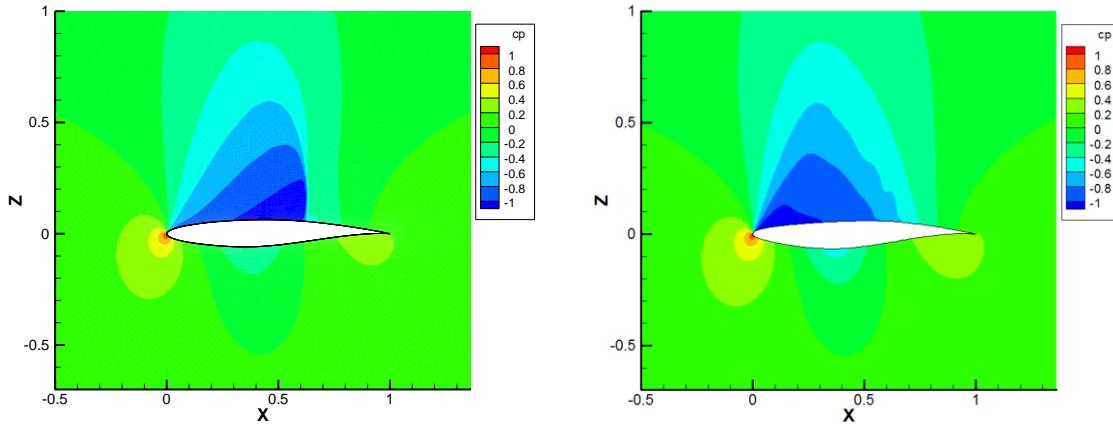


Figure 7.5 The contour of pressure coefficients of initial aerofoil (left) and optimum aerofoil (right) obtained by 10<sup>th</sup> order CST

The Cp contour figures illustrate that both methods are able to eliminate the shock. This is also proven in Figure 7.6 which shows the Cp distribution of both cases. In addition, the Cp contour shows there is a little unsatisfactory Cp distribution in the optimal aerofoil obtained by 7<sup>th</sup> order CST method. This phenomenon should be investigated by checking the Cp distribution.

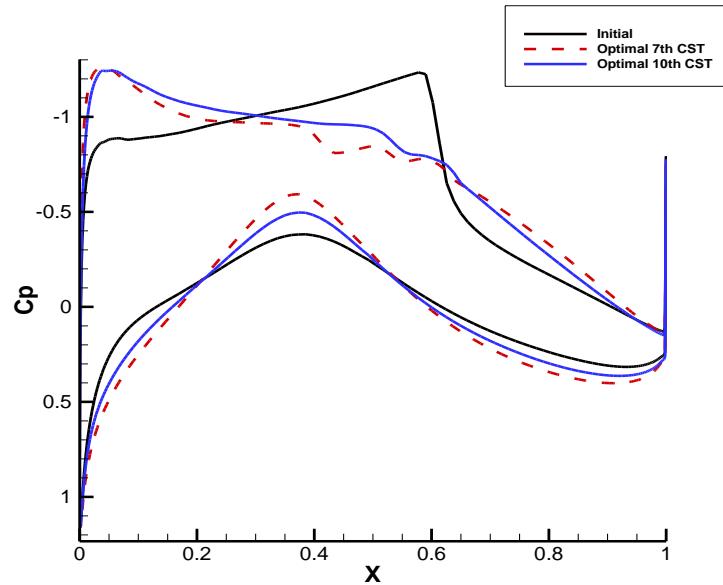


Figure 7.6 Cp distributions of initial aerofoil and optimal aerofoil obtained by 7<sup>th</sup> order CST and 10<sup>th</sup> order CST

Figure 7.6 clearly shows that there is unsMOOTH Cp distribution between 40% and 55% chord length of the optimal aerofoil obtained by the 7<sup>th</sup> order of CST. The result of the 10<sup>th</sup> order of the CST is much smoother. The reason for this could firstly be the accuracy of gradient since a small error in gradient could make the optimisation approach a real optimal point but not be able to reach it. Especially, the shock region is extremely sensitive to surface smoothness, so the accuracy of gradient in this region is important.

The second reason is local control of the 7<sup>th</sup> order CST is less than the 10<sup>th</sup> order CST. This region is just before the shock and requires good surface control to eliminate shock, and the geometric volume constraint is applied in the region between 20% and 60% chord length. Therefore, the parameterisation methods have to have a large flexibility in this region to achieve shock free and smooth conditions. Obviously, the flexibility of the 10<sup>th</sup> order CST is higher than the 7<sup>th</sup> order. Therefore, the 10<sup>th</sup> order CST obtained a better aerofoil than the 7<sup>th</sup> order in two-dimensions. A comparison of the aerofoils will be shown at the end of this study. Figure 7.7 shows the optimisation history of Cd and Cl when using the iCST method.

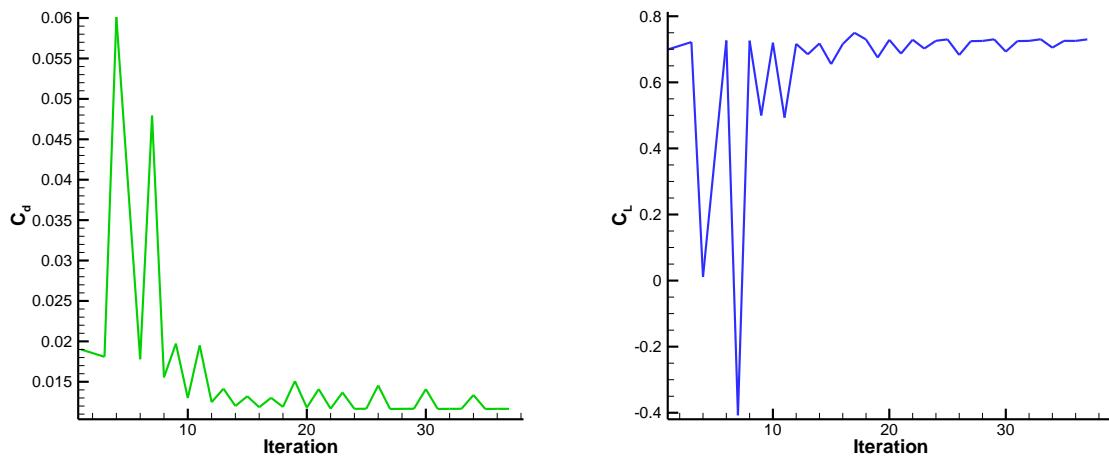


Figure 7.7 Cd (left) and Cl (right) optimisation history of iCST

Similarly with the 10<sup>th</sup> order CST method, the optimisation shows periodic variation after 20 iterations. The minimum objective value is then reduced slightly. The optimisation is

considered as converged. Table 7.2 shows the aerodynamic coefficients of initial and optimal aerofoils.

Table 7.2 Aerodynamic coefficients and constraint values

	C <sub>l</sub>	C <sub>d</sub>	C <sub>l</sub> /C <sub>d</sub>	Volume
Initial	0.6982	0.01909	36.57	0.0453
iCST	0.7300	0.01166	62.63	0.0453

The aerodynamic coefficients show that although iCST effectively reduced the drag, however the lift is not increased by as much as with the CST methods. The drag is also higher than the 10<sup>th</sup> order CST method by about 2 drag counts. The geometric volume constraint is still well satisfied. Figure 7.8 shows the Cp contour of flow field and Figure 7.9 shows the Cp distribution.

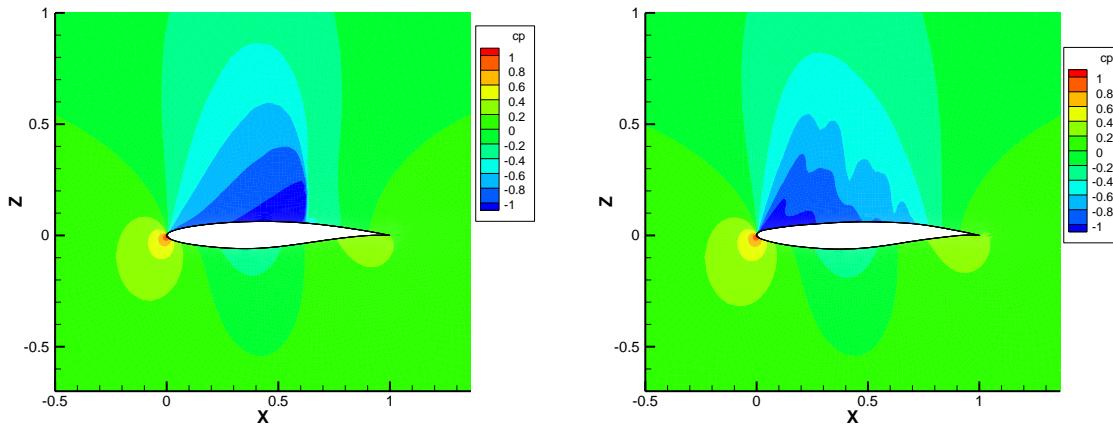


Figure 7.8 The contour of pressure coefficients of initial aerofoil (left) and optimum aerofoil (right) obtained by iCST

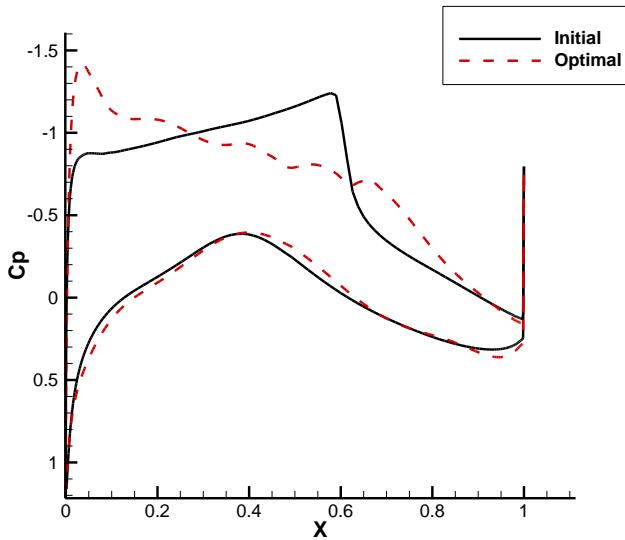


Figure 7.9 Cp distributions of initial and optimal aerofoils obtained by iCST method

The Cp contour and distributions clearly show that the shock is eliminated at the optimal aerofoil. This demonstrates that the iCST method is able to deal with drag reduction optimisation. In addition, at the leading edge of the upper surface, a high pressure suction peak is obtained with the iCST method, which did not appear in the CST test. This shows the capability of local control of the iCST method. However, the Cp distribution is not very smooth. The reason is because the iCST method uses second order curvature control over the surface. Because the aerodynamic flow is very sensitive to the curvature, the design parameters of second order curvature have to be converged very well with the requirement of high level of accuracy of gradient. This is a demanding requirement to be achieved. Therefore, care should be taken in the setting of the curvature parameters.

Finally, the RCST method has been tested. Figure 7.10 shows the optimisation history of the RCST method.

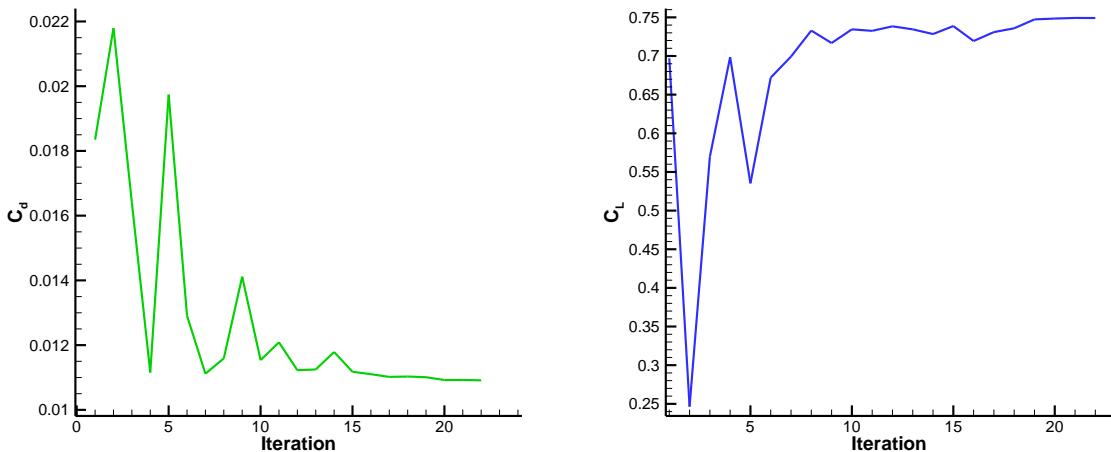


Figure 7.10  $C_d$  (left) and  $C_l$  (right) optimisation history of RCST

The optimisation history shows that RCST converges smoothly without periodical change. Both  $C_d$  and  $C_l$  are converged after 20 iterations. This is due to the lower order polynomial. Table 7.3 shows the aerodynamic coefficient and constraint values of the RCST method.

Table 7.3 Aerodynamic coefficients and constraint values

	$C_l$	$C_d$	$C_l/C_d$	Volume
Initial	0.6982	0.01909	36.57	0.0453
RCST	0.7491	0.01091	68.65	0.0453

Table 7.3 shows RCST achieves the more drag reduction than the other three test cases; it is reduced by about 42%. The lift increase is almost equal to the required 0.75. Therefore, RCST with lower order polynomials is able to search out a better design point than the CST method using higher order polynomials. This achieves a balance between a lower number of design variables and a higher flexibility of design space. Figure 7.11 shows the  $C_p$  contour of flow field and Figure 7.12 shows the  $C_p$  distributions.

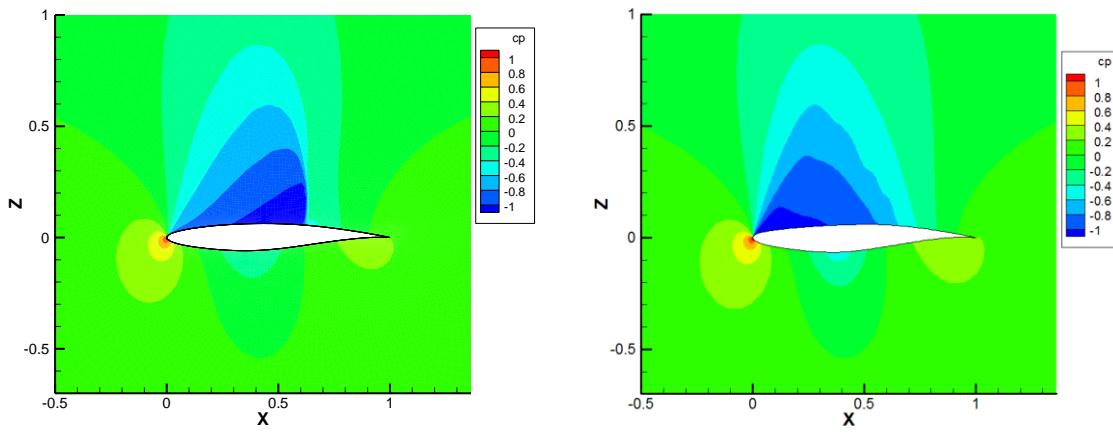


Figure 7.11 The contour of pressure coefficients of initial aerofoil (left) and optimum aerofoil (right) obtained by RCST

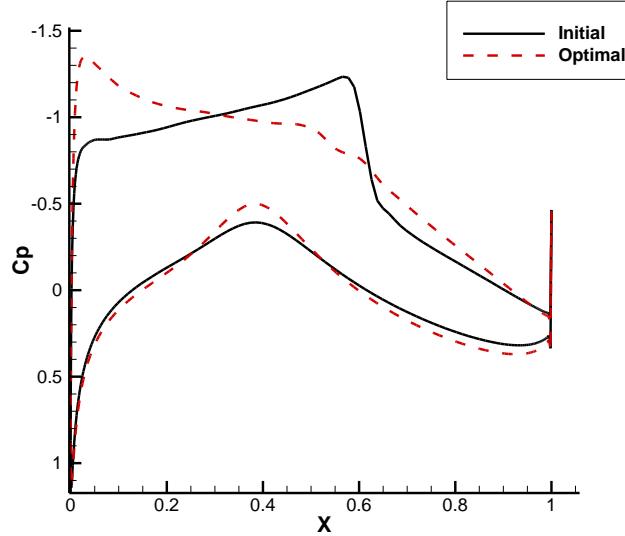


Figure 7.12 Cp distributions of initial and optimal aerofoils obtained by RCST method

This shows that the RCST method obtained the smoothest Cp distribution of all test cases. Because the rational equation is employed, RCST has better local control than the CST method of the same order. Therefore, even if the lower order polynomial is used, RCST is still able to well control the surface to achieve the convergence condition in optimisation. Finally, the aerofoil shapes obtained by the above four different approaches are compared in Figure 7.13.

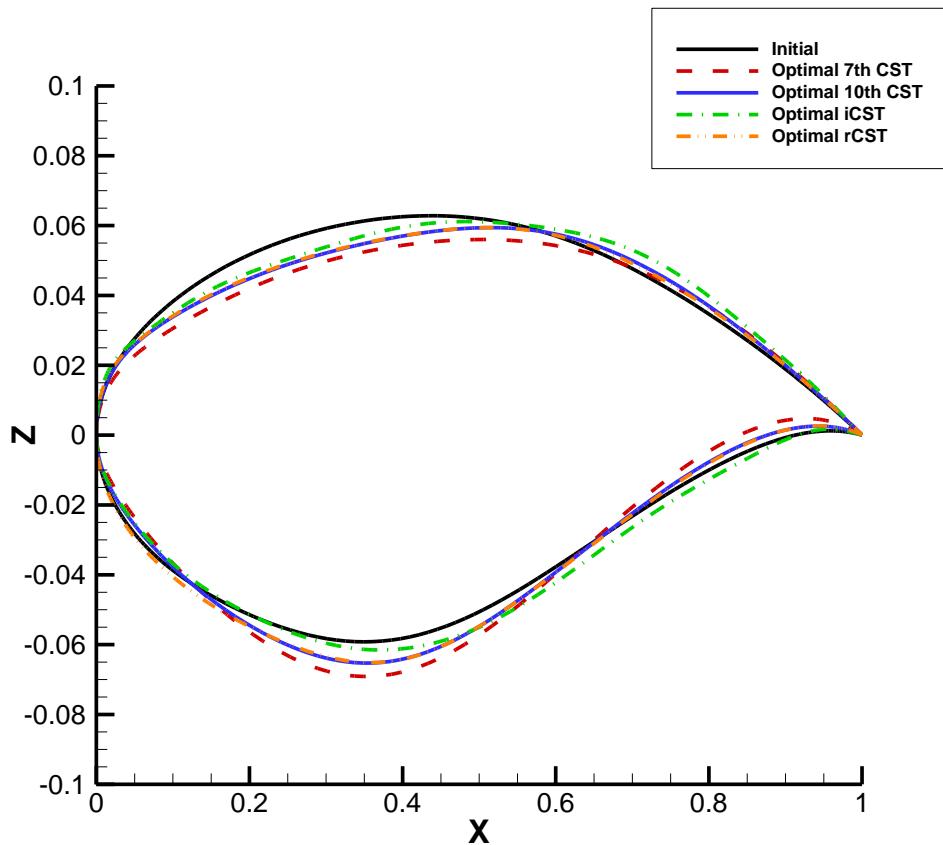


Figure 7.13 Comparison of the initial aerofoil and optimal aerofoils obtained by various parameterisation methods

From Figure 7.13, the 7<sup>th</sup> order CST provides the largest variety on the aerofoil. This is because it attempts to flatten the entire upper surface to eliminate the shock wave. At the same time, the geometric volume constraint has to be satisfied. The 7<sup>th</sup> order CST then pushes the lower surface to the outer bound to maintain volume constraint. Therefore, the 7<sup>th</sup> order CST provides the largest modification.

The 10<sup>th</sup> order CST and RCST achieve almost the same optimal aerofoils. As shown in the optimisation history, RCST has better convergence performance on optimisation by employing a 6<sup>th</sup> order polynomial and is able to provide very smooth geometry. Therefore, RCST shows excellent performance on both geometry representation and aerodynamic drag optimisation.

iCST shows a smaller change on the aerofoil. This is because the design parameters of iCST are limited and are more restricted than other methods. For example, the crest x location is not moved down as far as with the other methods. However, iCST already effectively eliminates the shock wave. The intuitive parameters are helpful to understand the design space. Although the curvature parameter of the iCST method affects the optimisation convergence, it is still a useful parameterisation method for drag minimisation optimisation.

## 7.2 Shock bump optimisation in the two-dimensional aerofoil

Shock control bump optimisation is performed to investigate the performance of three different parameterisation methods. In this case, the RAE 2822 is still employed for this test since Wong (2006) has successfully performed shock bump optimisation on it. The flow condition is  $\text{Mach} = 0.73$ ,  $\text{Re} = 6.5 \times 10^6$ ,  $\text{Cl} = 0.75$ .

In the aerofoil drag minimisation test case, it is noticed that even if the hard constraint has been employed for lift coefficient, it is still possible that the optimal solution of optimisation does not satisfy the required lift. In addition, the angle of attack has to be employed as an extra design parameter to adjust the flow field to match the target lift. However, the author here considers that it might not be very efficient to employ angle of attack as a design parameter and put lift into the constraint condition in optimiser. TAU solver offers a function, the so-called target-lift-iteration, which is to adjust the angle of attack during the CFD solver iteration to match the required target lift. If this function is employed, the target lift could be automatically matched within flow solver rather than in optimisation. Theoretically, it could be able to reduce one constraint in optimisation and improve the efficiency of optimisation.

If this strategy is used, the objective function must be modified since the optimisation is working at target lift  $\text{Cl}_{\text{target}}$  rather than at the pre-existing lift  $\text{Cl}$ . The modified objective function is defined as:

$$I = Cd - \frac{\left(\frac{\partial Cd}{\partial \alpha}\right)}{\left(\frac{\partial Cl}{\partial \alpha}\right)} (Cl - Cl_{target}) \quad 7.2$$

where  $\frac{\partial Cd}{\partial \alpha}$  is the derivative of drag respect to angle of attack, and  $\frac{\partial Cl}{\partial \alpha}$  is the derivative of lift respect to angle of attack.

In a well converged flow simulation, the resulting  $Cl$  will be equal or very close to  $Cl_{target}$ . Therefore, the second term of the objective function could be neglected. The derivative of the objective function is then:

$$\frac{dI}{dD} = \frac{dCd}{dD} - \frac{\left(\frac{\partial Cd}{\partial \alpha}\right)}{\left(\frac{\partial Cl}{\partial \alpha}\right)} \frac{dCl}{dD} \quad 7.3$$

Therefore, the gradient of the lift is still important to the optimisation. This objective function has been successfully employed in multi-point optimisation (Reuther *et al.* 1999; Kim *et al.* 2001; Dwight 2006; Illic 2012). The detail of the derivation can be found in Appendix E.

The length of shock bump is limited to be a maximum of 20% of chord length of aerofoil. The starting location of the bump is bounded between 35% and 55%. The bump crest position is bounded between 10% and 90% with respect to bump length. The height is limited to be a maximum 0.5% of chord length. For the PARSEC bump, the parameters P and Q are set to constants 1 and 0 to keep the curvature of bump ramp to zero. The parameter c is variable between -1 and 1.

The parameters of the length and position of the CST bump on the aerofoil are set up with the same standard cubic polynomial and PARSEC bump. However, there is no intuitive parameter in CST bump to control the height and position. Therefore, the 2<sup>nd</sup> order polynomial is used with 3 shape parameters. Each shape parameter is bound between 0 and 0.007. Figure 7.14 to 7.16 show the optimisation history of drag from three parameterisation methods.

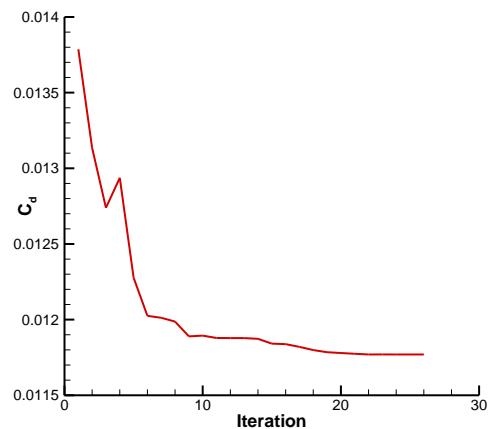


Figure 7.14 Optimisation of drag using CST bump

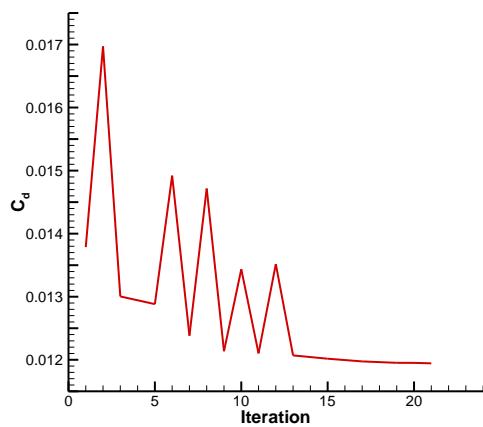


Figure 7.15 Optimisation of drag using PARSEC bump

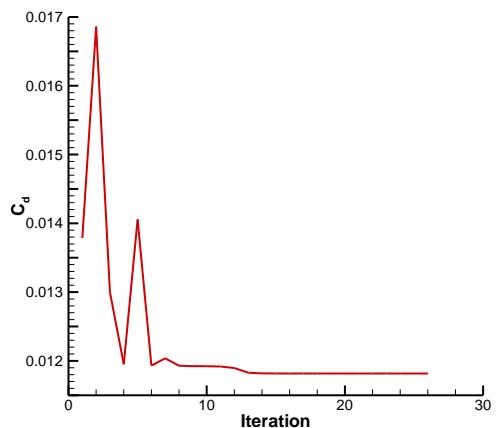


Figure 7.16 Optimisation of drag using standard cubic bump

The optimisation history shows that the CST bump and cubic bump achieve convergence after 10 iterations. Although a small oscillation occurs at the start of optimisation in PARSEC bump, it still reaches optimal results after 15 iterations. This shows that there is not much difference in their performance with respect to optimisation efficiency. Figure 7.17 to 7.19 show the Cp contour of flow over of the initial aerofoil without bump and the aerofoil with optimal bump obtained by three methods.

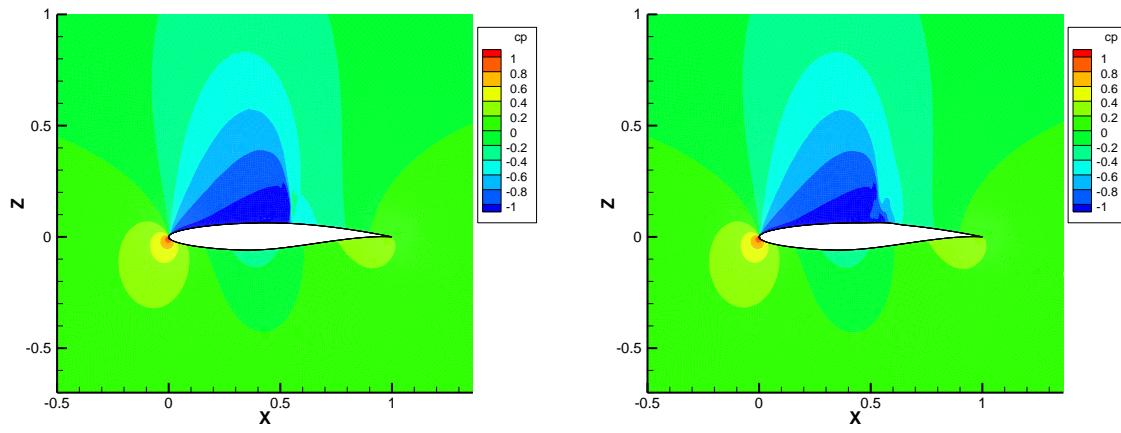


Figure 7.17 Contour of pressure coefficient of aerofoil without bump (left) and aerofoil with optimal CST bump (right)

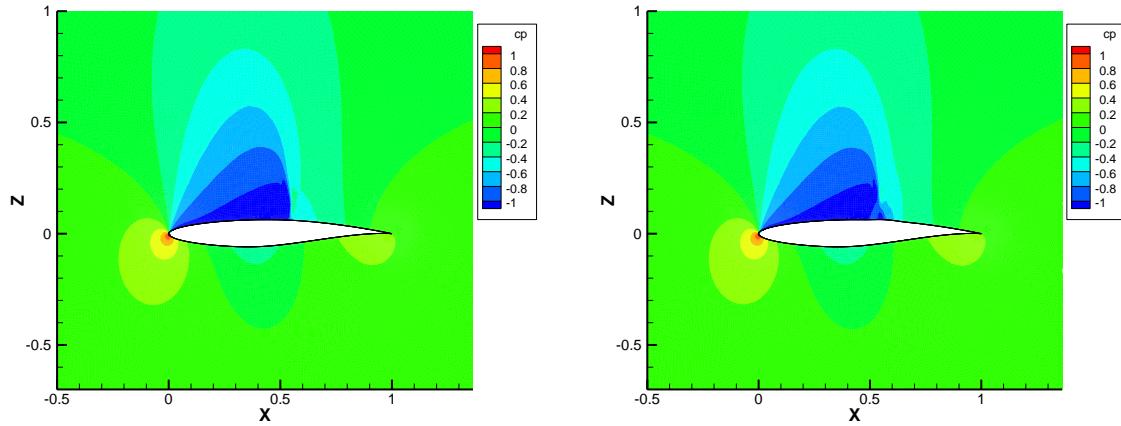


Figure 7.18 Contour of pressure coefficient of aerofoil without bump (left) and aerofoil with optimal PARSEC bump (right)

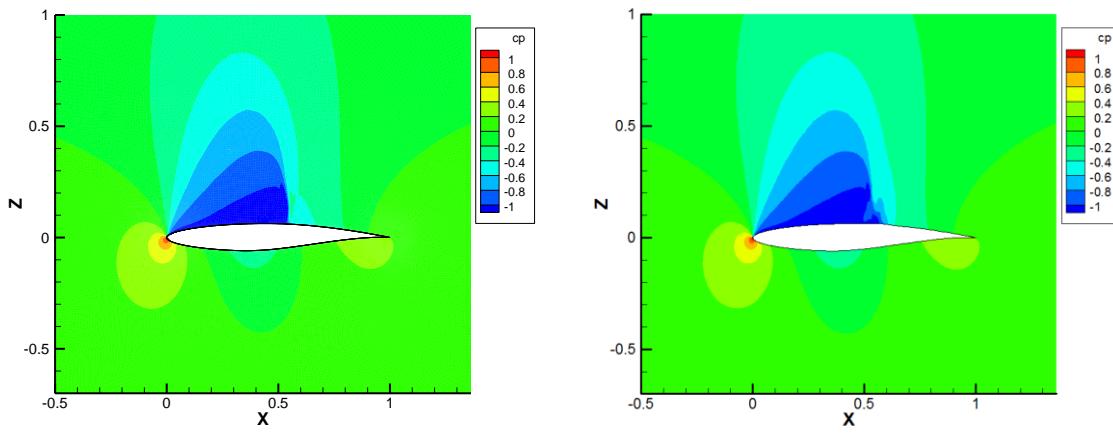


Figure 7.19 Contour of pressure coefficient of aerofoil without bump (left) and aerofoil with optimal standard cubic bump (right)

The Cp contour plot figures show that all three methods are able to weaken the normal shock on the initial aerofoil. This is proven in the Cp distribution plot in Figure 7.20.

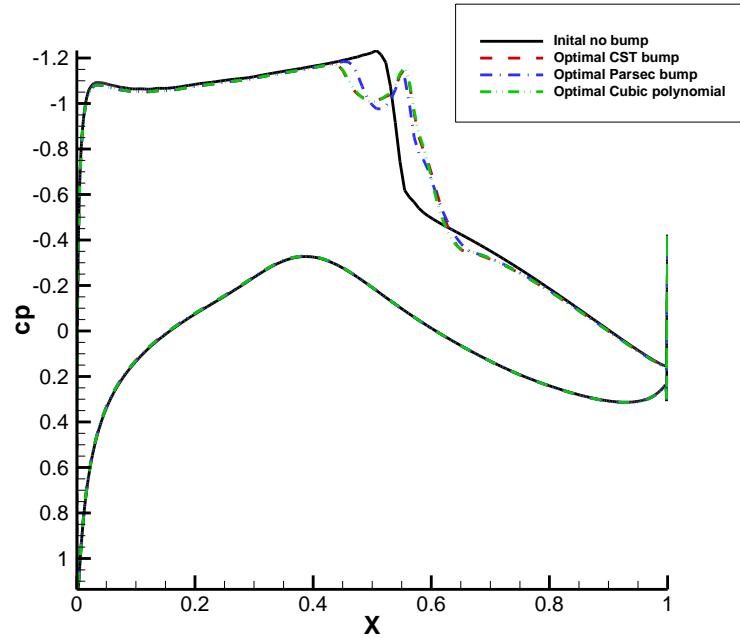


Figure 7.20 Comparison of Cp distribution

The Cp distribution figure shows that all three methods significantly reduce the magnitude of the maximum pressure peak before shock. The PARSEC bump reduces the maximum negative suction pressure slightly more than the other two bumps. However,

the CST bump and cubic bump start reducing pressure earlier than the PARSEC bump. In addition, all methods effectively push the shock downstream. The CST and cubic bumps push the shock slightly more than the PARSEC bump. Further information could be checked from the aerodynamic coefficients in Table 7.4.

Table 7.4 Aerodynamic coefficients

	$C_d_{total}$	$C_d_{pressure}$	$C_d_{viscous}$	$C_l$	L/D
Initial	0.013786	0.008141	0.005645	0.75	54.40
CST	0.011774	0.006087	0.005687	0.75	63.70
Parsec	0.011974	0.006275	0.005699	0.75	62.64
Cubic	0.011816	0.006135	0.005681	0.75	63.47

Table 7.4 shows the total drag is reduced by about 20 drag counts in all three cases. The drag reduction is mainly from the pressure drag which indicates the wave drag reduction. The viscous drag is only increased within 1 drag count. The CST bump achieves the lowest drag of the three cases; however it is only 1 drag count lower than the PARSEC and cubic bumps. Figure 7.21 shows the bump shape on the aerofoil.

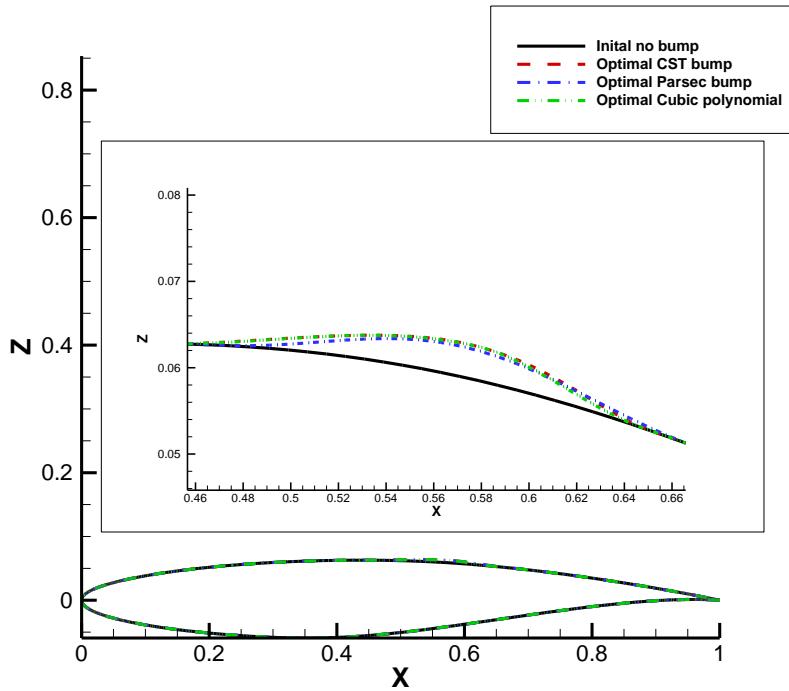


Figure 7.21 Comparison of bump shape

All optimal bumps achieve the maximum limit length 20% chord. The maximum heights of three bumps are all close to 0.4% chord. The bump location of the CST and cubic bumps are slightly more upstream than the PARSEC bump. In general, there is no significant difference between the three bumps. They all converge to a similar design shape.

Hence, it can be concluded that the different bump parameterisation methods have no significant effects on optimisation results. The CST and cubic methods are slightly better than the PARSEC methods in terms of the optimisation convergence. However, the CST method could maintain the curvature continuity at the transition area between bump and aerofoil. This could be useful for practical engineering design in terms of manufacturing issues.

## Chapter 8 Optimisation in Three-Dimensions

After the two-dimensional optimisation has been carried out, the three-dimensional optimisation is then performed in order to examine performance of parameterisation methods on optimisation.

The first test case is to perform a wing drag optimisation case. Because the iCST method is proposed for a two-dimensional aerofoil design, it cannot generate a three-dimensional surface directly. It needs to be allocated various design stations along spanwise, and an extra interpolation algorithm is then required to generate the three-dimensional surface. This methodology will increase the number of design variables and bring other issues, such as the calculation of the surface sensitivities with respect to the control aerofoil and interpolation algorithm. Therefore, the iCST method is not employed in the three-dimensional study. CST is employed in wing drag optimisation. The CST methods with various different order polynomials are also examined. This test case is carried out using flow adjoint and mesh adjoint ‘Surfgrad’ strategy. The wing drag optimisation using the RCST parameterisation method is then carried out to investigate its performance.

The second test case is winglet aerodynamic optimisation in which a winglet will be retrofitted onto an existing wing and then optimised. Two types of winglet, winglet 1 and winglet 2, are employed and parameterised by the CST methods. The planform parameters are also involved in optimisation. Due to a certain issue which is explained later, this optimisation only used adjoint ‘Volgrad’ strategy with mesh sensitivities calculated by finite difference. At the end, the shock control bump optimisation using the CST method is tested to examine the performance of this new bump model in the 3D case.

In addition, the current mesh adjoint in Tau is only supported in its sequential mode. This means it cannot be applied to the complex case with a large number of grids due to the

memory limit. The ‘Volgrad’ is an option. However, the mesh deformation could be time consuming for large number of grids. For example, to deform a mesh with 3 million nodes, the current RBF mesh deformation employing a pre-existing inverted interpolation matrix still requires 5-6 minutes. For a case with 100 parameters, it requires 100 times more mesh deformation which requires  $5 \times 100$  minutes. Even if this could be done in parallel on HPC, it still could be time consuming due to HPC queuing. So unfortunately, three-dimensional entire aircraft optimisation with complex intersection is not carried out at present.

## 8.1 Wing optimisation using CST methods

The wing drag optimisation is studied at first. The F6 wing is selected to carry out this test since it has been parameterised in Chapter 4 using CST with different orders. In order to test a case with a stronger initial shock wave, the Mach number is increased and the flow condition of this test case is set to:  $\text{Mach} = 0.80$ ,  $C_l = 0.6$  and  $\text{Re} = 5 \times 10^6$ . The computational mesh is the same mesh as employed in Chapter 4. In the practical case, the pitching momentum is hard to decide on since increased pitching momentum will affect the longitudinal stability; however, decreased pitching momentum will increase trim drag in conventional aircraft. Therefore, the pitching momentum constraint is not considered in this work.

### 8.1.1 Influence of different order of the CST methods on wing optimisation

The first study is to investigate the influence of the order of the CST methods. Due to the computational expense, only 4 CST parametric models are selected, namely *BPOX 6-BPXY 6*, *BPOX 6-BPOY 8*, *BPO 6-BPO 10* and *BPO 10-BPOY 10*. In Chapter 7 the two-dimensional test cases show streamwise higher order polynomials could provide more local control on the surface; however, the optimisation convergence could be worse. Therefore, the first three selected orders are aimed to test the influence of the polynomial order on spanwise. Then, CST with *BPOX 10-BPOY 10* is employed to test the influence of streamwise higher order polynomials in the three-dimensional case.

The planform parameters, including wing span, crank position, leading edge sweep angle, chord length distribution and leading height distribution, are fixed in three-dimensional wing optimisation. This is because these parameters are normally defined at an early conceptual design stage in practical engineering processes. They are not allowed to be changed in the detailed aerodynamic design stage. Therefore, the surface design parameters, parameters of twist distribution and trailing edge thickness distribution are allowed to be changed.

In order to avoid the non-linear behaviour in optimisation, the twist angle at wing root should be fixed since it also indicates the angle of attack (Smith *et al.* 2013). If it is employed as a design variable, it could potentially affect optimisation efficiency. Therefore, the first design variable of twist is frozen.

As presented in Chapter 4, a piecewise Bernstein polynomial is employed in twist and trailing edge thickness distribution functions. In order to avoid a non-smooth surface, the orders of polynomials are chosen to be as low as possible with relatively high flexibility. Therefore, a 3<sup>rd</sup> order polynomial is used to represent twist distribution in the inboard wing, and a 4<sup>th</sup> order polynomial is used to represent twist distribution in outboard wing. In trailing edge thickness distribution, a 4<sup>th</sup> order polynomial is used in the inboard wing and a 3<sup>rd</sup> order polynomial is used in outboard wing. The smooth condition, which has been presented in Chapter 4 with Equation 4.16, has to be applied at piecewise break points. The number of design variables of twist and trailing thickness are 7 and 8, respectively. Therefore, the total number of design variables of CST with *BPOX 6-BPXY 6*, *BPOX 6-BPOY 8*, *BPO 6-BPO 10* and *BPO 10-BPOY 10* are 113, 141, 169 and 257, respectively.

The bounds of each surface parameter are limited to move 50% from their initial value positively and negatively for allow large flexibility. In this case, the distribution function is employed to represent the tangential value of twist rather than twist angle value. Therefore, the bounds of each twist design variable are set up to [-0.1,0.1] which corresponds to [-5.7°, 5.7°]. The trailing edge thickness is a non-dimensional thickness

with respect to the local chord length. Hence, the bounds of each trailing edge thickness parameter are set up to [0.003, 0.005].

The target lift iteration is still employed in this case. The optimisation is then described as:

$$\begin{aligned}
 \min \quad & I = Cd - \frac{\left(\frac{\partial Cd}{\partial \alpha}\right)}{\left(\frac{\partial Cl}{\partial \alpha}\right)} (Cl - Cl_{target}) \\
 \text{subject to} \quad & V_i \geq V_{i,\text{initial}} \quad i = 1, 2, \dots, 21 \\
 & \text{twist smoothness condition} = 0 \\
 & \text{trailing edge thickness smoothness condition} = 0
 \end{aligned} \tag{8.1}$$

where 21 aerofoils are cut and extracted averagely along spanwise, which are at 0%, 5%, 10% up to wingtip, to control geometric thickness constraint. The torsion box volume calculated between 20% and 60% chord length is still employed and  $V_i$  indicates the torsion box volume of the  $i$ -th section. The torsion box volume of the initial geometry is shown in Table 8.1.

Table 8.1 Torsion box volume of initial geometry

$V_{1,\text{initial}}$	$2.577 \times 10^{-1}$	$V_{12,\text{initial}}$	$5.837 \times 10^{-2}$
$V_{2,\text{initial}}$	$2.339 \times 10^{-1}$	$V_{13,\text{initial}}$	$5.166 \times 10^{-2}$
$V_{3,\text{initial}}$	$2.103 \times 10^{-1}$	$V_{14,\text{initial}}$	$4.615 \times 10^{-2}$
$V_{4,\text{initial}}$	$1.875 \times 10^{-1}$	$V_{15,\text{initial}}$	$4.135 \times 10^{-2}$
$V_{5,\text{initial}}$	$1.656 \times 10^{-1}$	$V_{16,\text{initial}}$	$3.682 \times 10^{-2}$
$V_{6,\text{initial}}$	$1.451 \times 10^{-1}$	$V_{17,\text{initial}}$	$3.230 \times 10^{-2}$
$V_{7,\text{initial}}$	$1.261 \times 10^{-1}$	$V_{18,\text{initial}}$	$2.775 \times 10^{-2}$
$V_{8,\text{initial}}$	$1.088 \times 10^{-1}$	$V_{19,\text{initial}}$	$2.336 \times 10^{-2}$
$V_{9,\text{initial}}$	$9.309 \times 10^{-2}$	$V_{20,\text{initial}}$	$1.949 \times 10^{-2}$
$V_{10,\text{initial}}$	$7.911 \times 10^{-2}$	$V_{21,\text{initial}}$	$1.660 \times 10^{-2}$
$V_{11,\text{initial}}$	$6.590 \times 10^{-2}$		

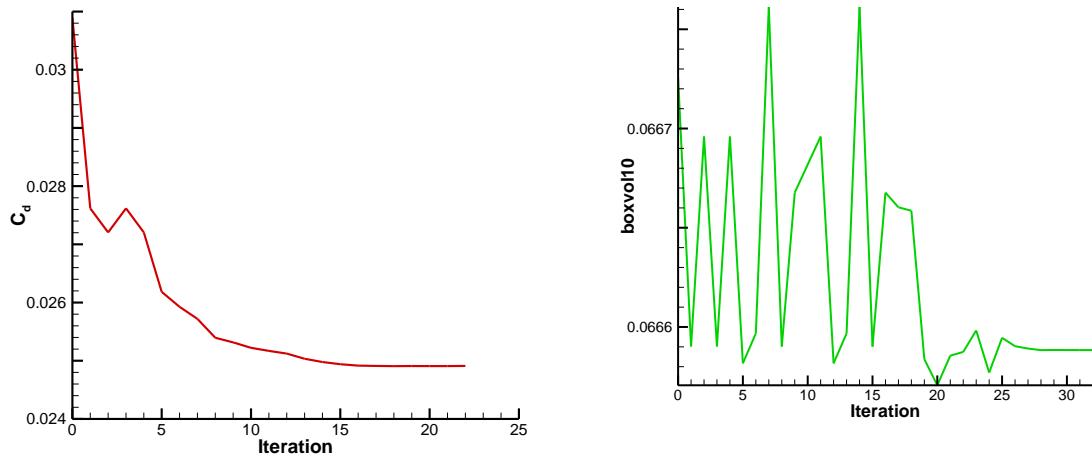


Figure 8.1 Optimisation history of drag (left) and the 10<sup>th</sup> torsion box volume (right) using CST with *BPOX 6-BPOY 6*

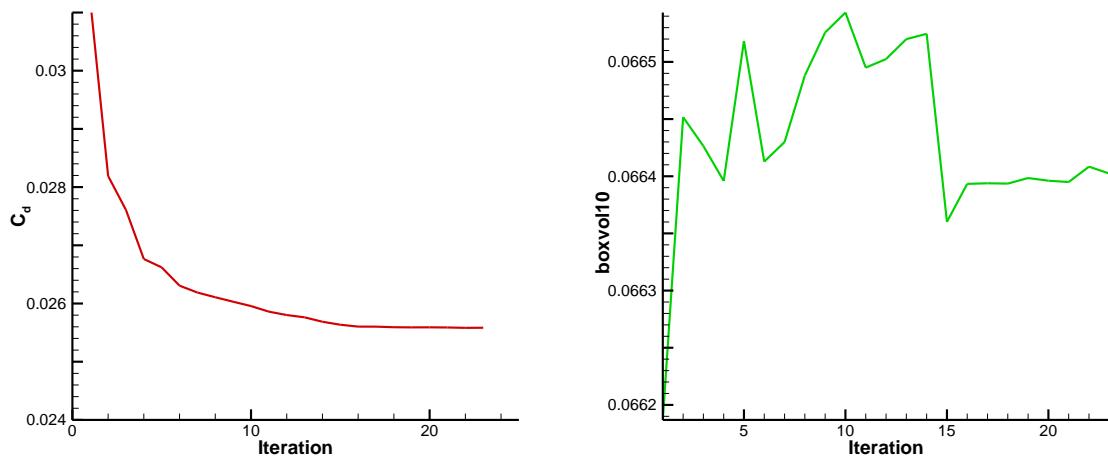


Figure 8.2 Optimisation history of drag (left) and the 10<sup>th</sup> torsion box volume (right) using CST with *BPOX 6-BPOY 8*

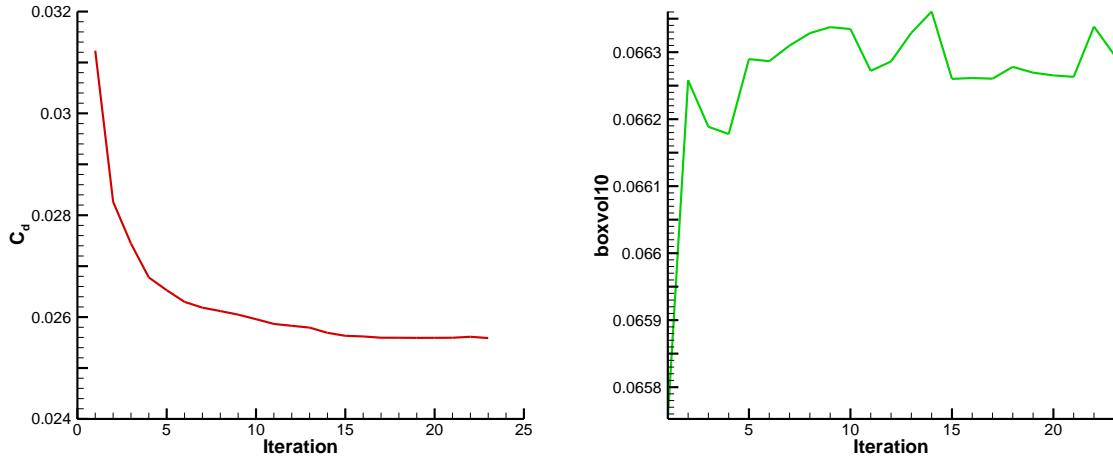


Figure 8.3 Optimisation history of drag (left) and the 10<sup>th</sup> torsion box volume (right) using CST with *BPOX 6-BPOY 10*

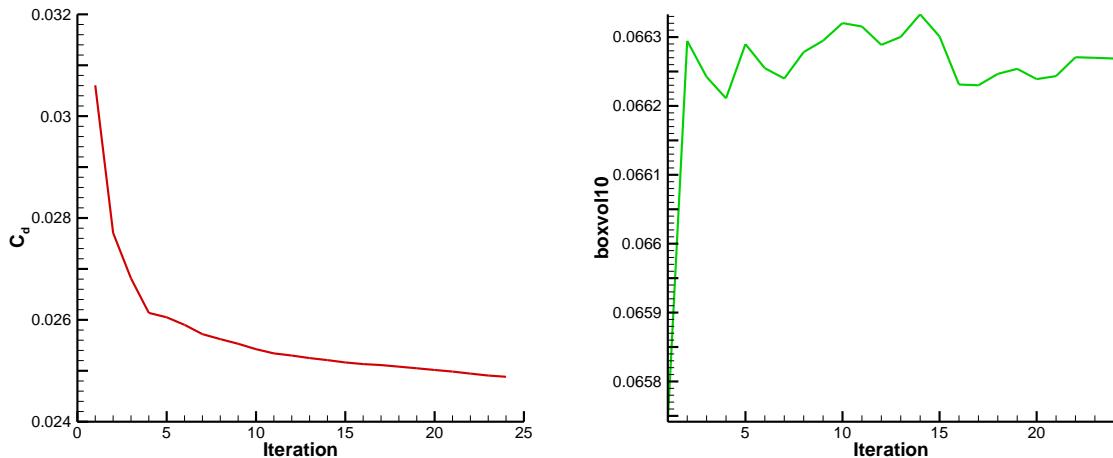


Figure 8.4 Optimisation history of drag (left) and the 10<sup>th</sup> torsion box volume (right) using CST with *BPOX 10-BPOY 10*

Figure 8.1 to 8.4 show the optimisation history of drag and the 10<sup>th</sup> torsion box volume of four test cases. The figures of drag history show that the optimisation reaches convergence successfully. The test case using 6<sup>th</sup> order streamwise polynomial converges slightly faster than 10<sup>th</sup> order which achieves the convergence after about 15 iterations. The cast with *BPOX 10* and *BPOY 10* reaches convergence after about 20 iterations.

The history of the 10<sup>th</sup> torsion box volume shows all optimal solutions obtained by four methods satisfy the required constraint, although their initial values are slightly different due to different initial parametric models. Compared with the 2D test case, there is no oscillation at the beginning of optimisation. The efficiency of three-dimensional optimisation is high. The higher streamwise order polynomial will require slightly more iterations to reach convergence. Figure 8.5 to 8.9 show the Cp contour on the wing surface of initial and optimal solutions. Although the initial geometries are slightly different, there is not much difference on the flow field. Hence, only the initial flow field obtained by *BPOX 6* and *BPOY 6* is shown.

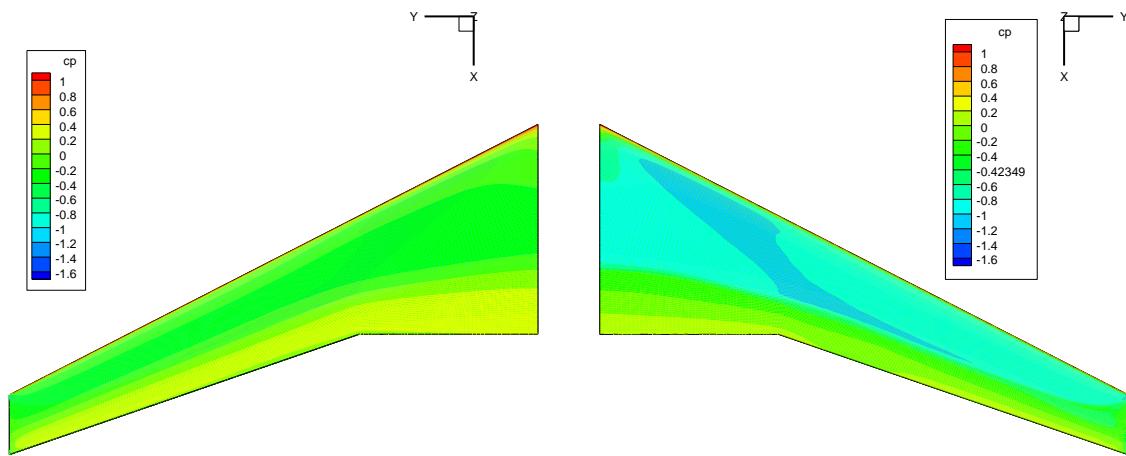


Figure 8.5 The Cp contour plot of initial wing surface, lower (left) and upper (right)

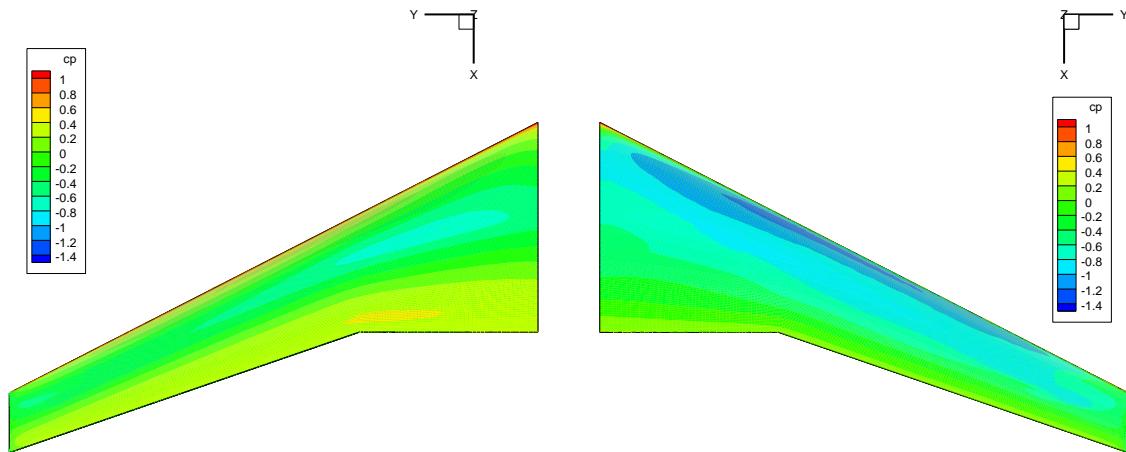


Figure 8.6 The Cp contour plot of optimal wing surface, lower (left) and upper (right), obtained by CST with *BPOX 6* and *BPOY 6*

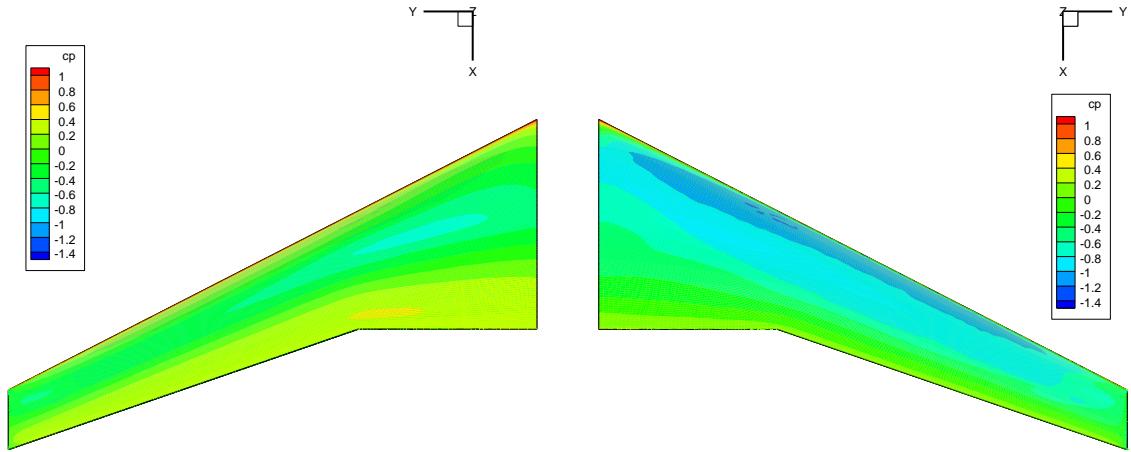


Figure 8.7 The Cp contour plot of optimal wing surface, lower (left) and upper (right), obtained by CST with *BPOX* 6 and *BPOY* 8

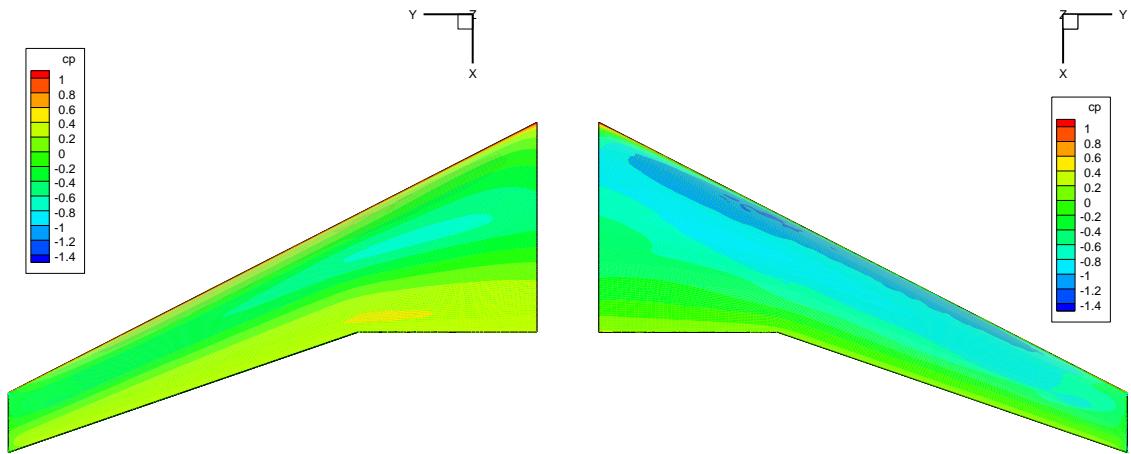


Figure 8.8 The Cp contour plot of optimal wing surface, lower (left) and upper (right), obtained by CST with *BPOX* 6 and *BPOY* 10

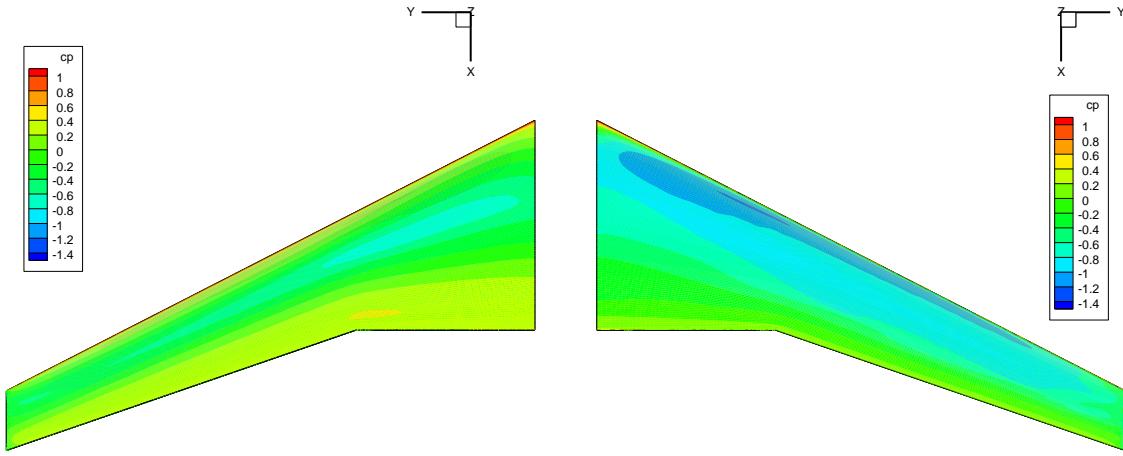


Figure 8.9 The Cp contour plot of optimal wing surface, lower (left) and upper (right), obtained by CST with *BPOX* 10 and *BPOY* 10

From the Cp contour of the initial geometry, it can be noticed that a strong shock wave is presented along the entire upper surface. In the optimal geometries, all test cases are able to manage to weaken this strong shock wave. Higher pressure suction in the leading edge area is achieved in optimal geometries. On the lower surface, a lower pressure region is generated in optimal geometries. This is clearly shown in the Cp distribution plots in Figure 8.10 to 8.15.

In Figure 8.8, there is a slight non-smoothness at the edge of the leading edge suction area on the upper surface. This could be due to the slight waviness on geometry caused by the higher order polynomials on spanwise. Apart from this, there is, in general, no apparent difference that could be noticed by comparing the Cp contour figures. Six sections at the spanwise location, 10%, 30%, 50%, 70%, 80% and 90%, are extracted from the wing. The Cp distributions along chordwise and the aerofoil shape of these sections are plotted in the following Figure 8.10 to 8.15.

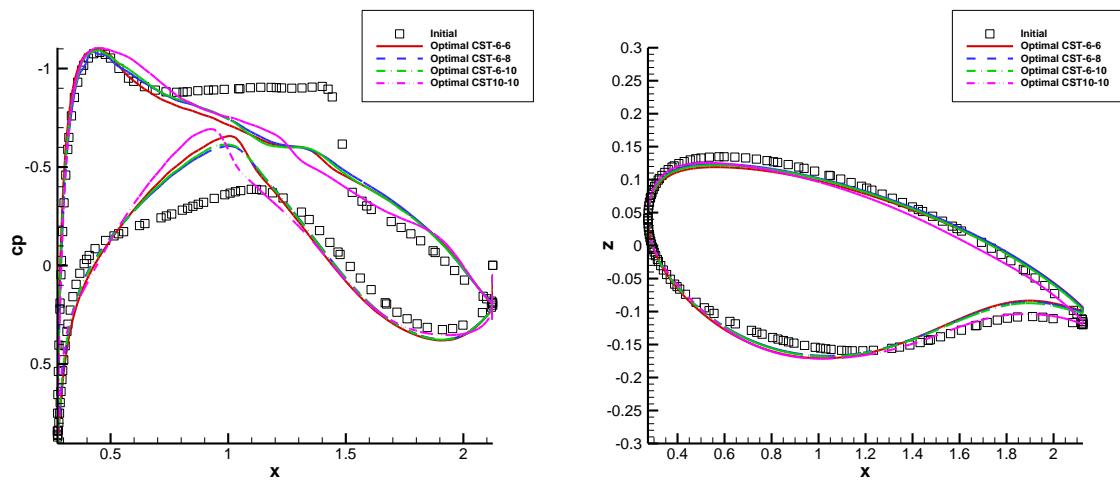


Figure 8.10 Cp distribution (left) and aerofoil shapes (right) at 10% span of wing

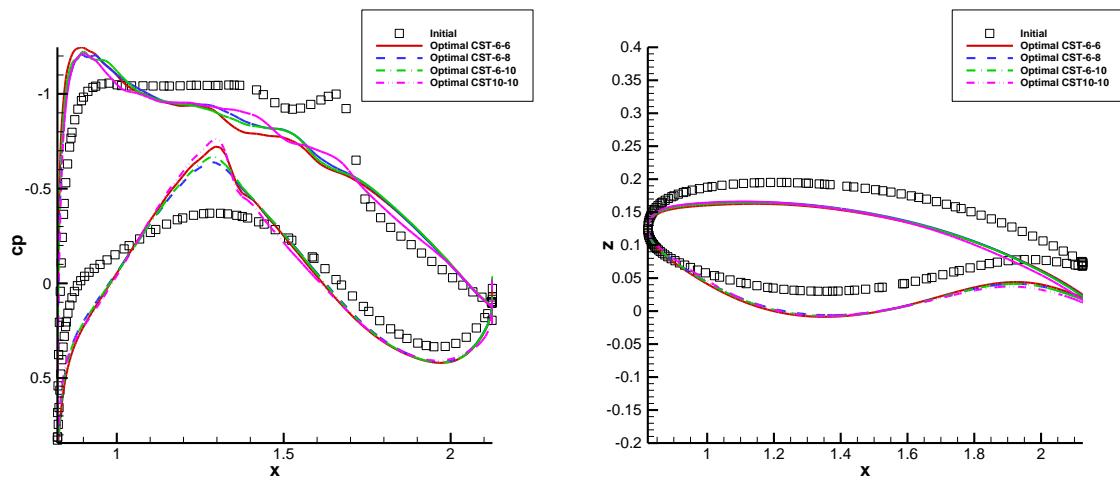


Figure 8.11 Cp distribution (left) and aerofoil shapes (right) at 30% span of wing

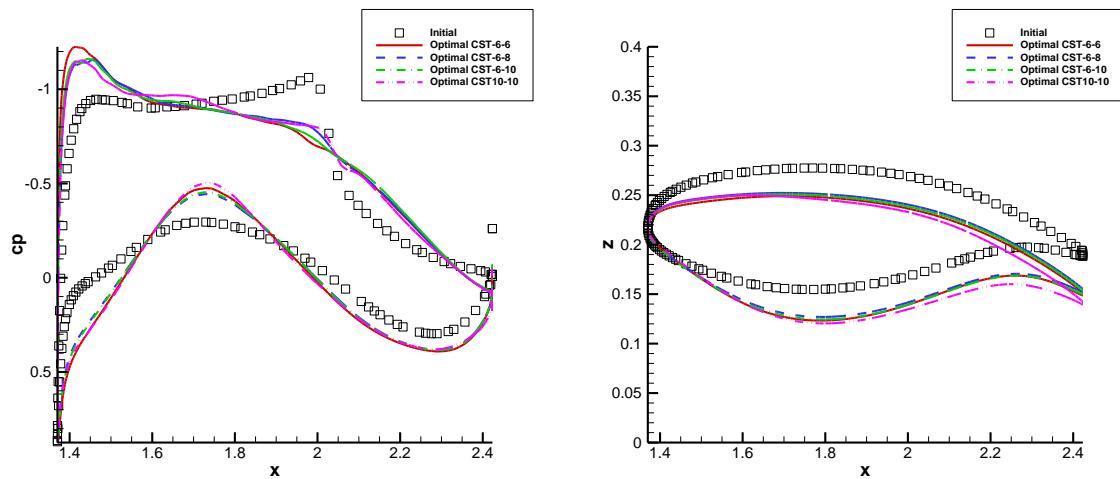


Figure 8.12 Cp distribution (left) and aerofoil shapes (right) at 50% span of wing

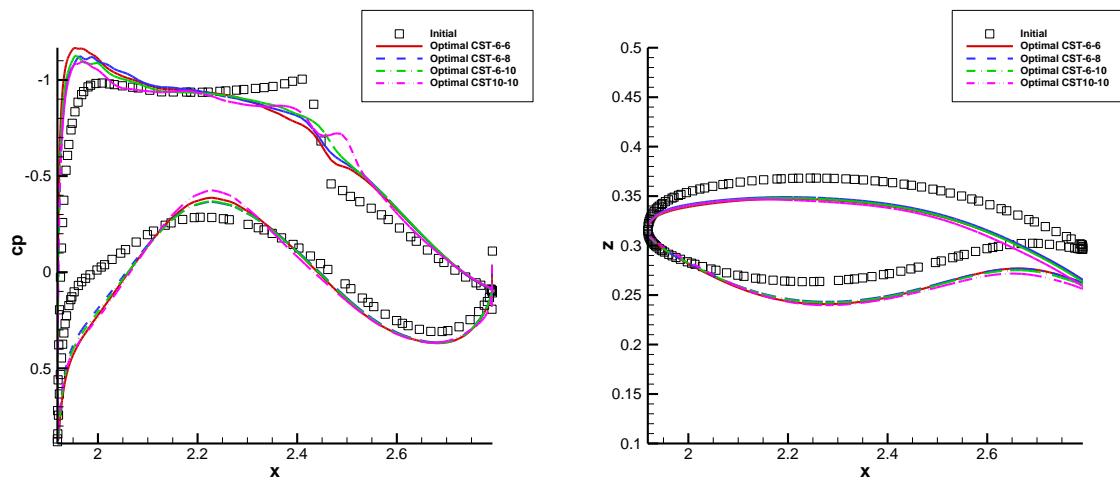


Figure 8.13 Cp distribution (left) and aerofoil shapes (right) at 70% span of wing

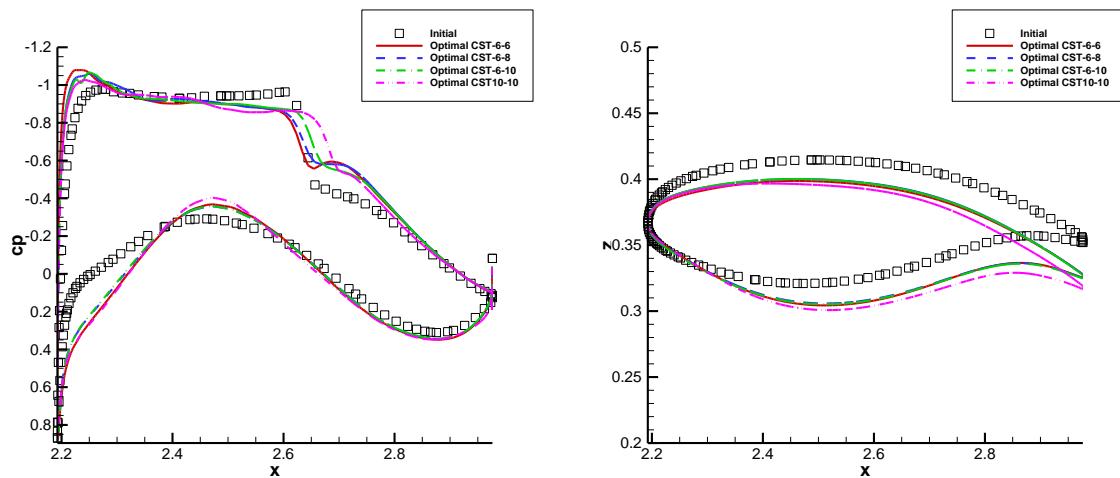


Figure 8.14 Cp distribution (left) and aerofoil shapes (right) at 80% span of wing

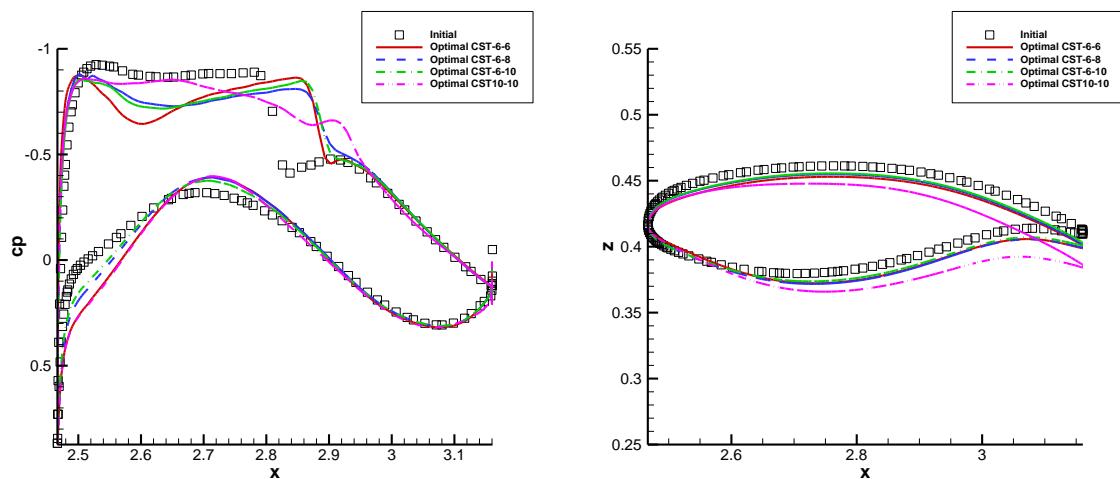


Figure 8.15 Cp distribution (left) and aerofoil shapes (right) at 90% span of wing

The  $C_p$  distribution plots in Figure 8.10 and 8.12 show the aerofoil at the inboard and near crank areas. The strong shock wave is clearly presented in the initial geometry. In the optimal geometries, this shock wave has been almost removed. A higher leading edge pressure suction peak appears at locations 30% and 50% of wingspan. The pressure on the lower side shows the pressure at about 30% chordwise on optimal geometries, which is lower than the initial geometry. This is because the optimisation modifies the upper surface to be flatter than the initial geometry. In order to maintain the torsion box volume, the lower surface is pushed to the outside direction of the initial geometry, which will give more flow acceleration and reduce the pressure. This can be seen by comparison of aerofoil shapes. In addition, comparing the optimal aerofoil shapes obtained by four parameterisation methods, there is almost no difference between the CST using the same order on streamwise. The CST with *BPOX* 10 and *BPOY* 10 obtained a slightly higher pitching up twist angle.

Figure 8.13 and 8.14 show the sections and  $C_p$  distributions at 70% and 80% of wingspan. The results are similar to the previous three locations. Although the reduction of the shock wave is not as much as in the previous three cases, they are all effectively weakened. The higher pressure suction is still presented, but it is reducing with increase of the span. On the lower surface side, the pressure of optimal geometries is lower than the initial geometry, but the change is getting smaller. This can be understood from aerofoil comparison figures. The curvature of the geometry on the lower surface side has not changed too much. The twist angle is larger than the initial geometry and the CST with *BPOX* and *BPOY* 10 still provides larger twist.

The most noticeable difference is achieved at location 90% closer to the wingtip. The shock wave still remains in the optimal results obtained by the CST with 6<sup>th</sup> order in chordwise direction. The shock wave has been moved downward to the trailing edge. The strength of shock wave is slightly weakened. Therefore, the drag is still reduced. However, the CST with *BPOX* 10 and *BPOY* 10 effectively weakens this shock wave and provides a smoother pressure gradient in the leading edge area. In addition, all trailing

edge thickness parameters are the final value at the lower bound. This means the trailing edge will be always become thin.

Figure 8.16 shows the twist angle distribution of initial and optimal wings. Table 8.2 shows the aerodynamic coefficients of initial and optimal wings. In order to further understand the optimal design, the drag is decomposed into induced drag and wave and viscous drag using the Nearfield/Farfield balance drag decomposition tool called FFD70, based on Destarac (2003). The spurious drag is not shown, since it is very small and not important. Because the initial geometries are slightly different due to the different order CST, the aerodynamic geometry is slightly different. Therefore, we will focus more on the proportion of coefficient variation.

Table 8.2 Aerodynamic coefficients (drag units in drag count)

	Cl	Cd_total	Cd_wave	Cd_induced	Cd_vis	Cl/Cd	Cmx
CST with BPOX 6 and BPOY 6							
Initial	0.6	308.9	33.32	133.53	134.97	19.42	1.29
Optimal	0.6	249	7.02	122.99	113.77	24.1	1.37
		-19.39%	-78.93%	-7.89%	-15.71%	24.12%	<b>6.60%</b>
CST with BPOX 6 and BPOY 8							
Initial	0.6	312.17	33.01	133.27	134.63	19.22	1.288
Optimal	0.6	255.82	6.14	123.4	115.02	23.45	1.365
		-18.05%	-81.40%	-7.41%	-14.57%	22.03%	<b>6.01%</b>
CST with BPOX 6 and BPOY 10							
Initial	0.6	312.31	32.3	133.36	134.76	19.21	1.288
Optimal	0.6	255.86	5.77	123.1	114.98	23.45	1.368
		-18.07%	-82.14%	-7.69%	-14.68%	22.06%	<b>6.16%</b>
CST with BPOX 10 and BPOY 10							
Initial	0.6	306.06	31.22	133.45	134.6	19.6	1.29
Optimal	0.6	248.81	5.09	122.93	114.26	24.11	1.381
		-18.71%	-83.49%	-7.88%	-15.11%	23.01%	<b>7.08%</b>

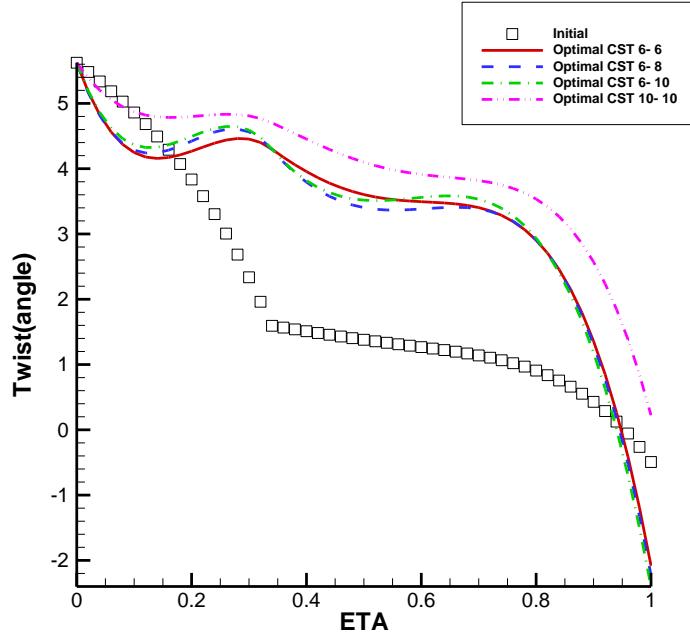


Figure 8.16 Twist distribution

Table 8.2 shows that all parameterisation methods are able to reduce the total drag by more than 18%. CST with *BPOX* 6 and *BPOY* 6 achieves the most drag reduction which is 19.39%. The viscous form drag is reduced 15.71% which is more than other cases. However, the wave drag in this case is reduced by 78.93%. The other methods with higher polynomial order in spanwise could provide more than 80% wave drag reduction. Therefore, the higher order polynomials on spanwise are helpful to improve the wave drag. However, considering the convergence speed and geometry smoothness, CST with *BPOX* 6 and *BPOY* 6 is already able to provide a decent improvement.

The twist distribution shows that the twist angle discontinuity is smoothed in optimal geometries. The twist angle of optimal geometries at location between root and 20% inboard wing is getting smaller. In the CST with 6<sup>th</sup> order in chordwise, the twist angle at location between 95% wingspan and wingtip is getting smaller as well. This indicates that the lift is reduced at inboard wing and increased at outer board wing. The rolling momentum coefficients  $C_{mx}$  are increased in all optimal geometries by about 6%, and induced drag is decreased by about 8%. This indicates that the lift distribution along

spanwise is changed by optimisation. Hence, the lift coefficient distribution along spanwise is plotted in Figure 8.17.

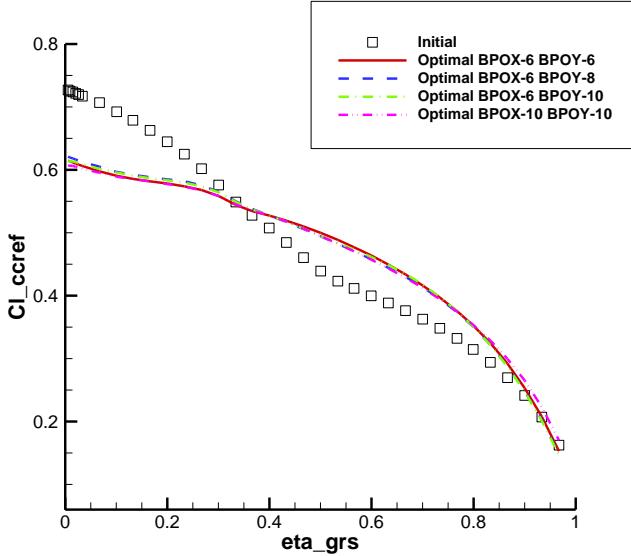


Figure 8.17 The spanwise lift distribution

The lift distribution illustrates that the initial lift distribution is close to a triangular distribution, and the lift distribution of all optimal geometries approaches an elliptical distribution. Due to the wave drag presented in this case, the lift distribution of optimal geometries is not a perfect elliptical distribution (Le Moigne 2002; Qin *et al* 2004, Le Moigne and Qin 2006). This phenomenon was found and discussed by Qin *et al* (2005). The centre of lift position is located at 40.10% of wingspan in the initial geometry. Consequently, in the optimal geometries, the centre of lift position is moved to 42.72% of wingspan. This implies a heavier structure is required to support this aerodynamic loading (Jupp 2001). Therefore, an extra constraint is needed to make a more practical optimisation.

### 8.1.2 Wing optimisation with rolling momentum constraint

As presented in the last test case, the CST methods with lower order polynomials can efficiently reduce the drag in optimisation. However, the optimisation results in shifting the centre of lift to the outer board to reduce induced drag. Therefore, the lift distribution has to be constrained to achieve a practical solution.

The current TAU adjoint solver only supports the objective values, such as  $C_l$ ,  $C_d$  and  $C_m$ , which are with respect to entire surface integration. Therefore, it is hard to directly use lift distribution as a constraint. However, as mentioned above, the rolling momentum  $C_{mx}$  implies the lift bending momentum. Therefore, the optimisation with  $C_{mx}$  constraint is tested. The  $C_{mx}$  is applied into the equality constraint.

The optimisation is then described as:

$$\begin{aligned} \min \quad & I = C_d - \frac{\left(\frac{\partial C_d}{\partial \alpha}\right)}{\left(\frac{\partial C_l}{\partial \alpha}\right)} (C_l - C_{l\text{target}}) \\ \text{subject to} \quad & V_i \geq V_{i,\text{initial}} \quad i = 1, 2, \dots, 21 \\ & C_{mx^*} \geq C_{mx\text{initial}} \\ & \text{twist smoothness condition} = 0 \\ & \text{trailing edge thickness smoothness condition} = 0 \end{aligned} \quad 8.2$$

Because the target lift iteration strategy is employed,  $C_{mx}$  is replaced by  $C_{mx^*}$  where:

$$C_{mx^*} = C_{mx} - \frac{\left(\frac{\partial C_{mx}}{\partial \alpha}\right)}{\left(\frac{\partial C_l}{\partial \alpha}\right)} (C_l - C_{l\text{target}}) \quad 8.3$$

In the last test, although the CST with high order polynomials on spanwise could get slightly better wave drag reduction, the CST with lower order polynomials has better convergence rate and is able to provide decent optimal solution. Therefore, in this test, CST with *BPOX 6* and *BPOY 6* is employed. All the settings of bounds are kept the same as for the last test. Figure 8.18 and 8.19 show the optimisation history of this test.

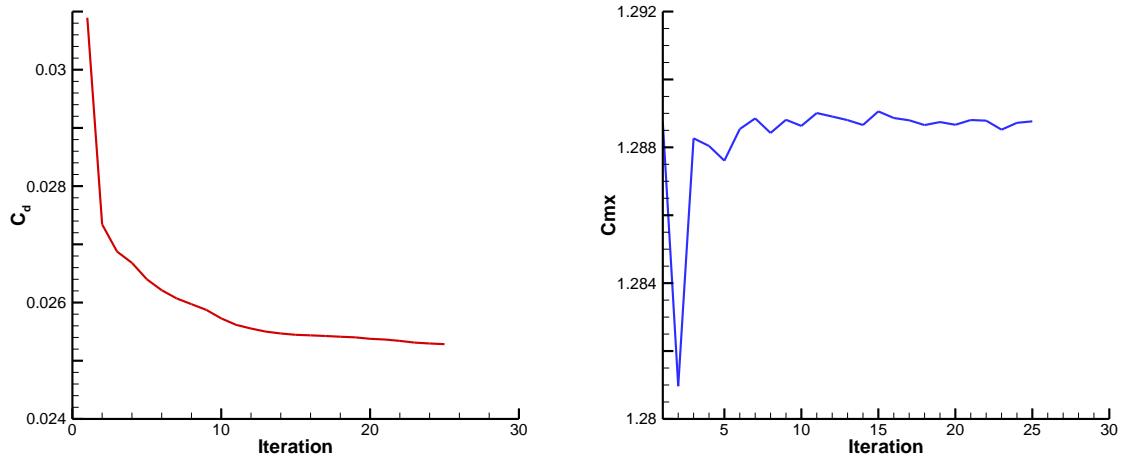


Figure 8.18 Optimisation history of drag (left) and  $C_{mx}$  (right) using CST with *BPOX 6-BPOY 6*

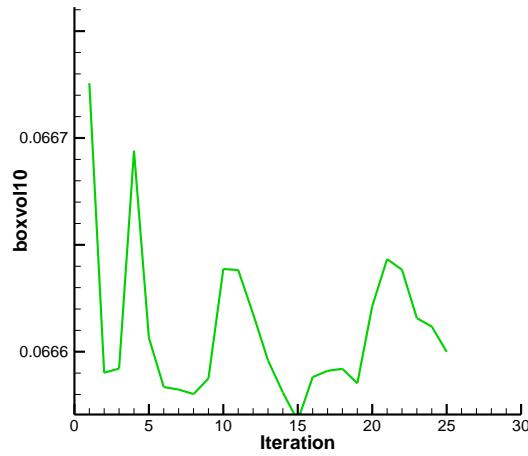


Figure 8.19 Optimisation history of the 10<sup>th</sup> torsion box volume using CST with *BPOX 6-BPOY 6*

Figure 8.18 shows the optimisation history of drag and  $C_{mx}$ . Similar to the previous test case, the optimisation achieves convergence very fast in about 13 iterations. The  $C_{mx}$  constraint is maintained during optimisation. Figure 8.19 shows the history of torsion box volume and illustrates that geometric constraints are also satisfied in the optimal solution. Figure 8.20 shows the  $C_p$  contour on the surface of optimal geometry.

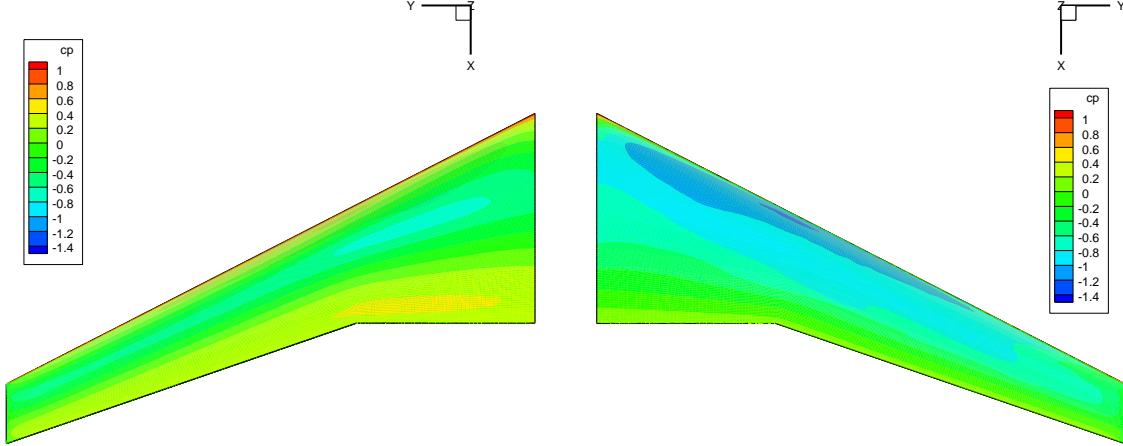


Figure 8.20 The Cp contour plot of optimal wing surface, lower (left) and upper (right), obtained by CST with *BPOX* 6 and *BPOY* 6 in optimisation with Cmx constraint

The Cp contour plot shows that the optimal solution is very similar to the previous test case. The shock wave is significantly reduced with a higher pressure suction peak presented at the inboard wing, and a lower pressure area appears on the lower surface. As with the previous test case, six sections at the spanwise locations 10%, 30%, 50%, 70%, 80% and 90% are extracted and the Cp distributions and aerofoil shape are compared with the results of optimisation without Cmx constraint, as shown in the following figures.

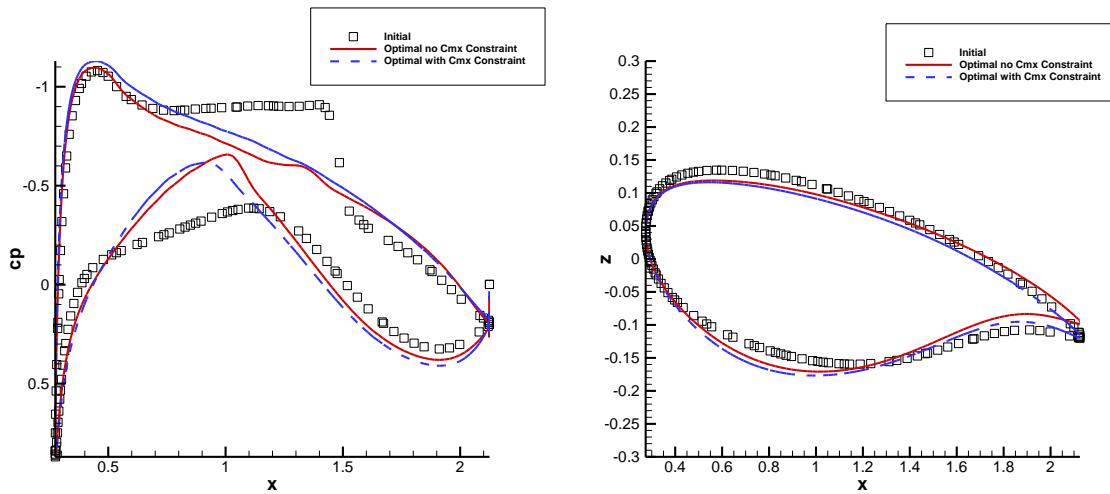


Figure 8.21 Cp distribution (left) and aerofoil shapes (right) at 10% of wingspan

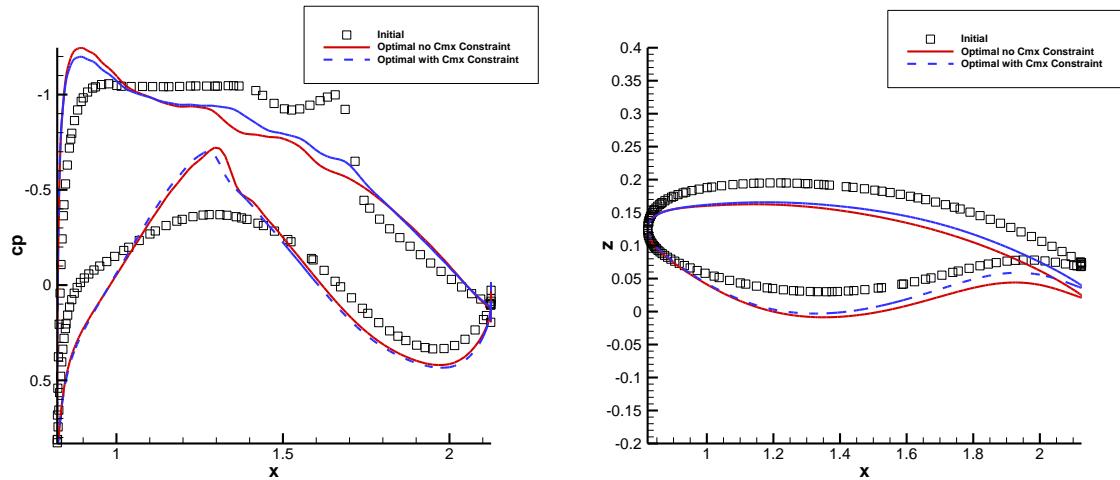


Figure 8.22 Cp distribution (left) and aerofoil shapes (right) at 30% of wingspan

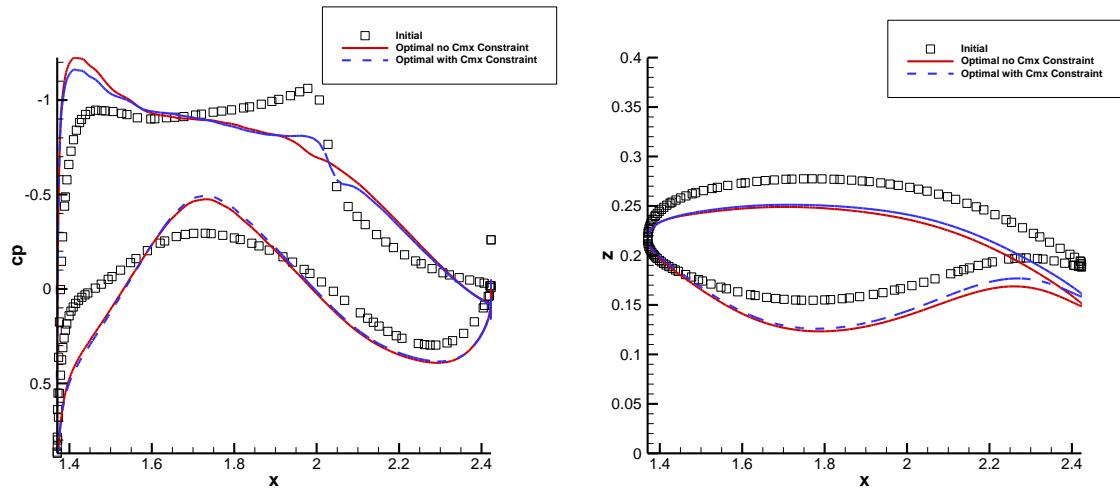


Figure 8.23 Cp distribution (left) and aerofoil shapes (right) at 50% of wingspan

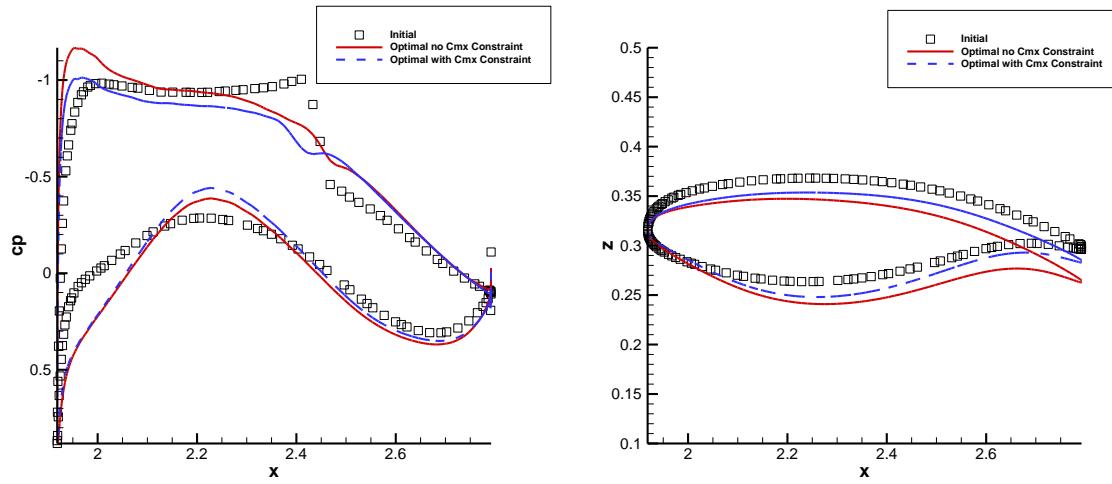


Figure 8.24 Cp distribution (left) and aerofoil shapes (right) at 70% of wingspan

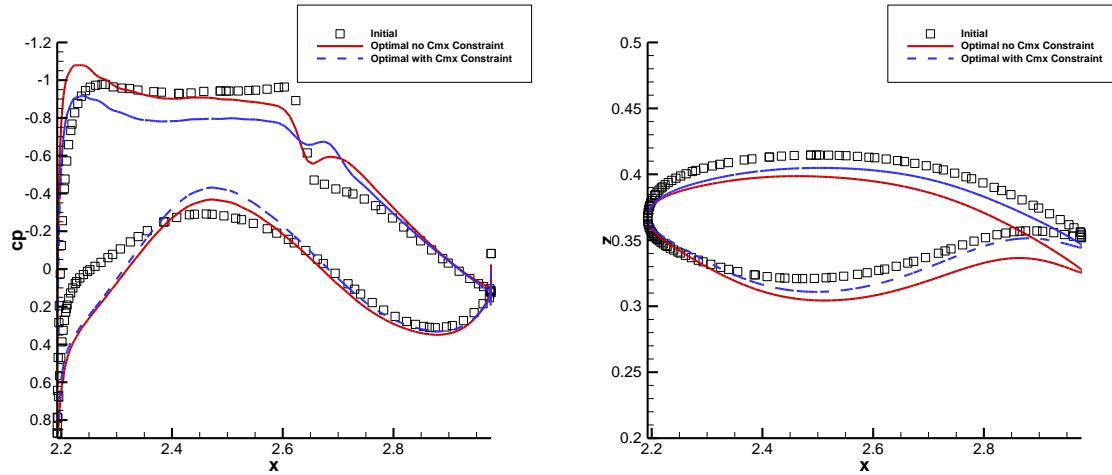


Figure 8.25 Cp distribution (left) and aerofoil shapes (right) at 80% of wingspan

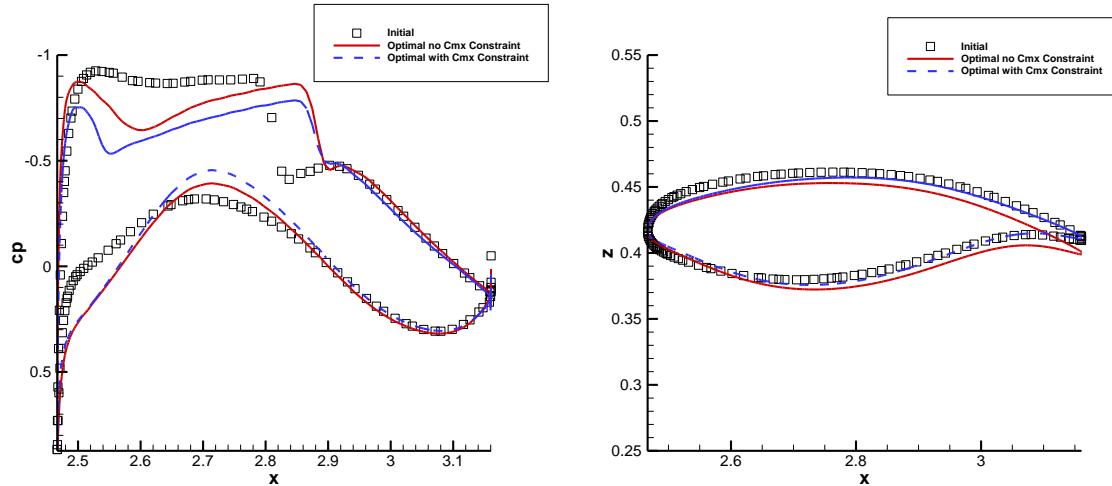


Figure 8.26 Cp distribution (left) and aerofoil shapes (right) at 90% of wingspan

The Cp distribution shows that the results of optimisation with Cmx constraint are very similar to the results without Cmx constraint. The structures of the Cp distribution are very similar to each other. The shock wave is almost removed at 10% and 30% of wingspan, and weakened at 50%, 70% and 80% of wing span. At 90% of wingspan, the shock is moved downwards to the trailing edge.

At 10% of wingspan, the optimal Cp distribution on the upper surface of optimisation with Cmx constraint has a higher pressure suction peak than in the previous case, and at 70%, 80% and 90% the optimal Cp distribution on the upper surface of optimisation with Cmx constraint has a lower pressure suction. The twist distribution is shown in Figure 8.27. The lift distribution is plotted in Figure 8.28 and aerodynamic coefficients are shown in Table 8.3.

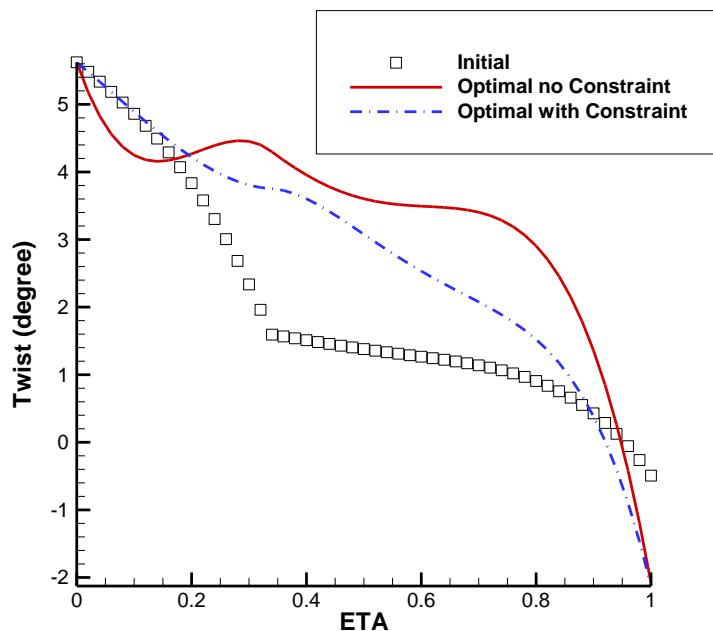


Figure 8.27 Twist distribution

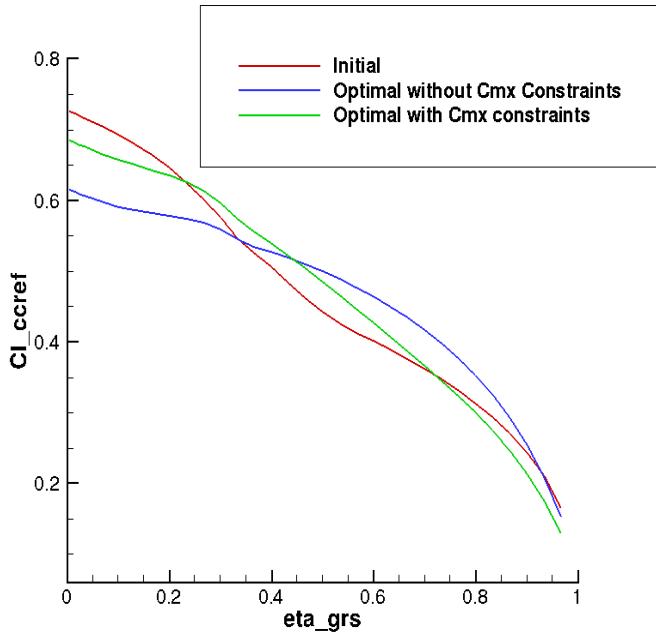


Figure 8.28 The lift distribution along span

Table 8.3 Aerodynamic coefficients of optimal results (drag units in drag count)

	Cl	Cd_total	Cd_wave	Cd_induced	Cd_vis	Cl/Cd	Cmx
Initial	0.6	308.9	33.32	133.53	134.97	19.42	1.290
Optimal	0.6	255	6.39	128.6	114.84	23.53	1.289
		-17.45%	-80.82%	-3.73%	-14.91%	21.14%	-0.01%

The twist distribution shows that the twist angle of the optimal geometry with 20% of wingspan is almost equal to that of the initial geometry. Between 20% and 90% of wingspan, the twist angle is larger than in the initial geometry but smaller than the optimisation without Cmx constraints. After 90% of wingspan, the twist angle is smaller than for the initial geometry.

Figure 8.28 shows that the optimal solution has a different lift distribution from the initial geometry. However, it is much closer to the initial geometry compared with the previous result of optimisation without Cmx constraint. The centre of lift of the new results is

40.11% and in the initial geometry is 40.10%. Therefore, in this optimisation, the centre of lift has been constrained successfully.

Furthermore, the aerodynamic coefficients show that the optimisation with  $C_{mx}$  constraint reduces the total drag by 17.45%. This is smaller than the previous optimisation which was 19.39%. This is because the induced drag is not reduced as much as in the previous case. However, when using  $C_{mx}$  constraint, the wave drag is reduced by 80.82% which is more than in the previous case. This means that the  $C_{mx}$  constraint makes the optimisation more focused on wave drag. The optimal geometry could be considered to be more practical than in the previous case.

## 8.2 Wing optimisation using RCST methods

The RCST method for the three-dimensional wing has been proposed in Chapter 4. In Chapter 7, the performance of the RCST method on 2D aerofoil optimisation has been examined. Hence, the performance of RCST method on the 3D wing is investigated in this section.

The test case of CST has shown the importance of  $C_{mx}$  constraint. Therefore, the objective function of optimisation in this case is same as with the previous test case in Equation 8.2, and the  $C_{mx}$  constraint is employed.

The order of RCST is set to *BPOX* 6 and *BPOY* 6 since it is proven in Chapter 4 that this order is able to represent original the F6 wing model with high accuracy. Similar to the test of RCST on 2D, the weight parameters are fixed during optimisation to avoid a non-linear effect. All the bounds of design parameters are kept the same as for the previous test case. Figure 8.29 and 8.30 show the optimisation history of this test.

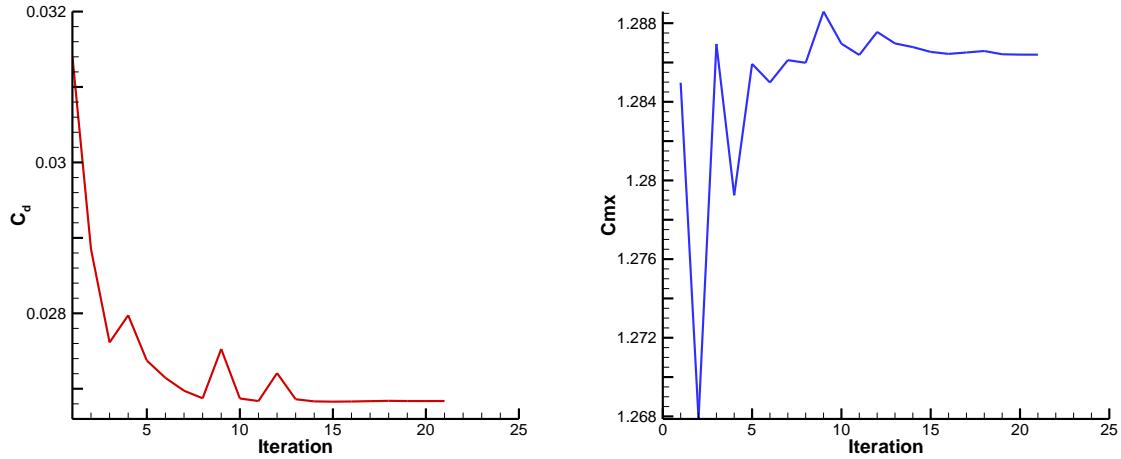


Figure 8.29 Optimisation history of drag (left) and  $C_{mx}$  (right) using RCST with *BPOX 6 -BPOY 6*

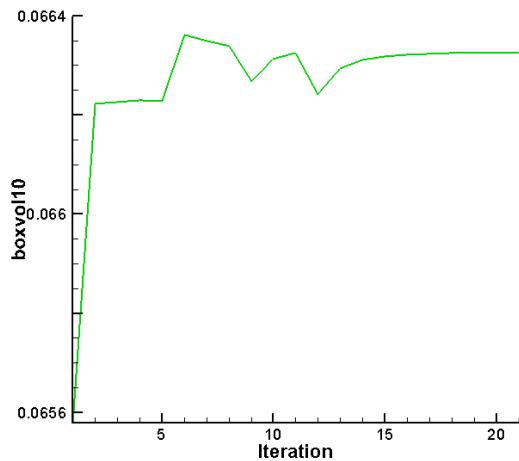


Figure 8.30 Optimisation history of the 10<sup>th</sup> torsion box volume using RCST with *BPOX 6- BPOY 6*

Figure 8.29 shows the optimisation history of drag and  $C_{mx}$ . As with CST cases, the optimisation achieves convergence in about 13 iterations. The  $C_{mx}$  constraint is satisfied during optimisation. Figure 8.30 shows the history of torsion box volume and proves that geometric constraints are also satisfied in the optimal solution. However, the  $C_d$  is not reduced as much as in the CST method. Figure 8.31 and 8.32 show the  $C_p$  contour on the surface of initial and optimal geometries.

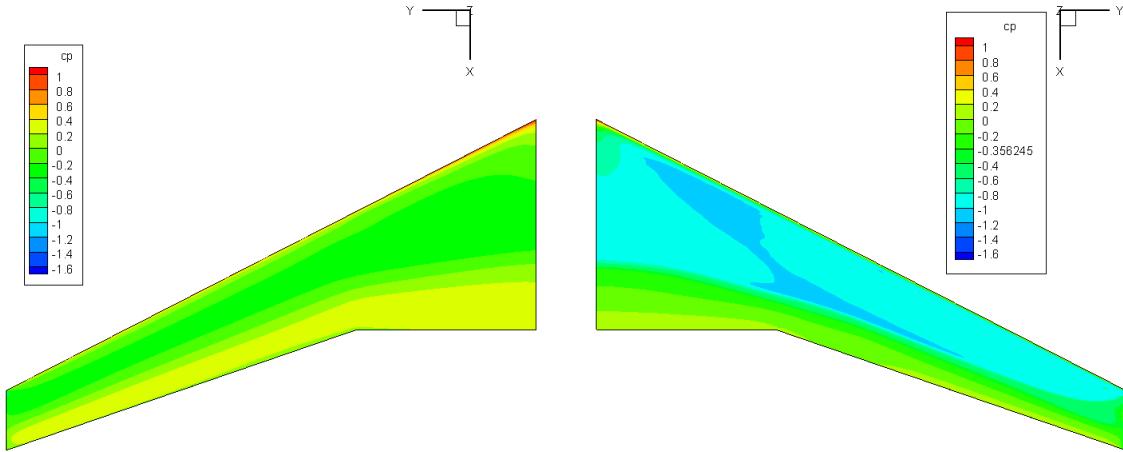


Figure 8.31 The Cp contour plot of Initial wing surface, lower (left) and upper (right), which is represented by RCST with *BPOX* 6 and *BPOY* 6

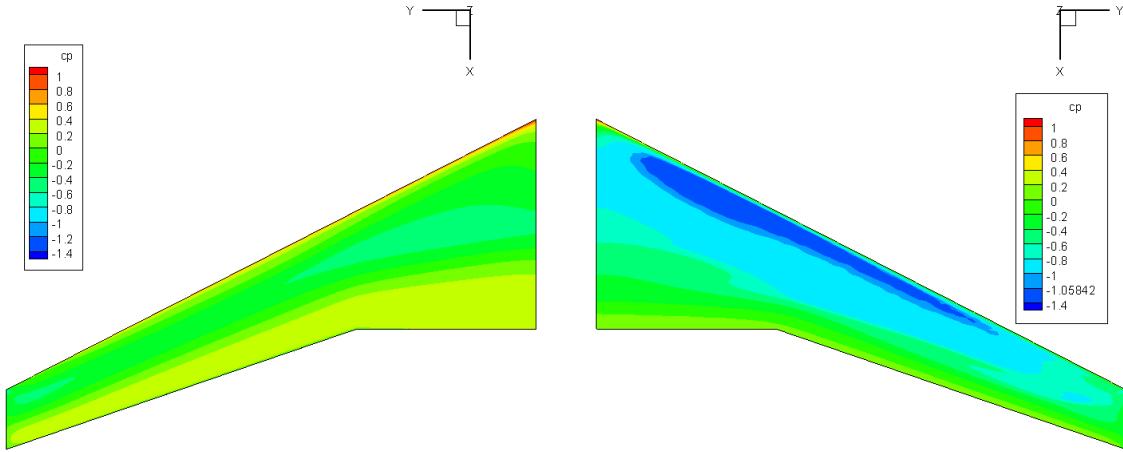


Figure 8.32 The Cp contour plot of optimal wing surface, lower (left) and upper (right), obtained by RCST with *BPOX* 6 and *BPOY* 6 in optimisation with Cmx constraint

The Cp on initial geometry represented by RCST is generally similar to the initial geometry of the original CST method with the same order. On the upper surface, the lambda shock on the initial geometry of the RCST method is slightly weaker than on the initial geometry of the original CST method. The Cp contour on the optimal geometry shows a similar result to the CST test case. However, the leading edge suction area is larger than the result of the original CST method, and the shock wave on the upper surface is not weakened significantly. In addition, no low pressure area is generated in the optimal geometry of the RCST method, which is different from the test of the original

CST method. Six sections at the spanwise locations 10%, 30%, 50%, 70%, 80% and 90% are also extracted to compare the  $C_p$  distributions and aerofoil shape of initial and optimal solutions.

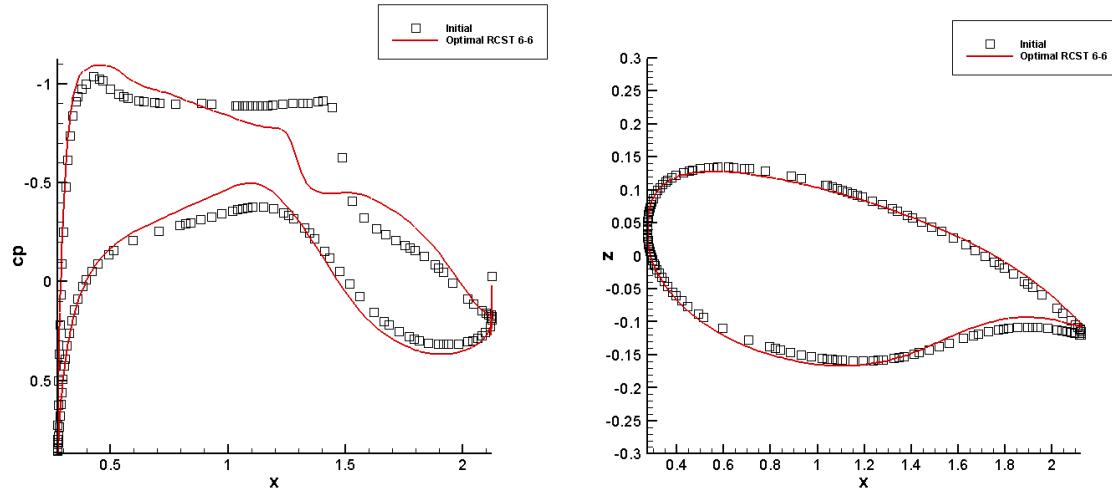


Figure 8.33  $C_p$  distribution (left) and aerofoil shapes (right) at 10% of wingspan

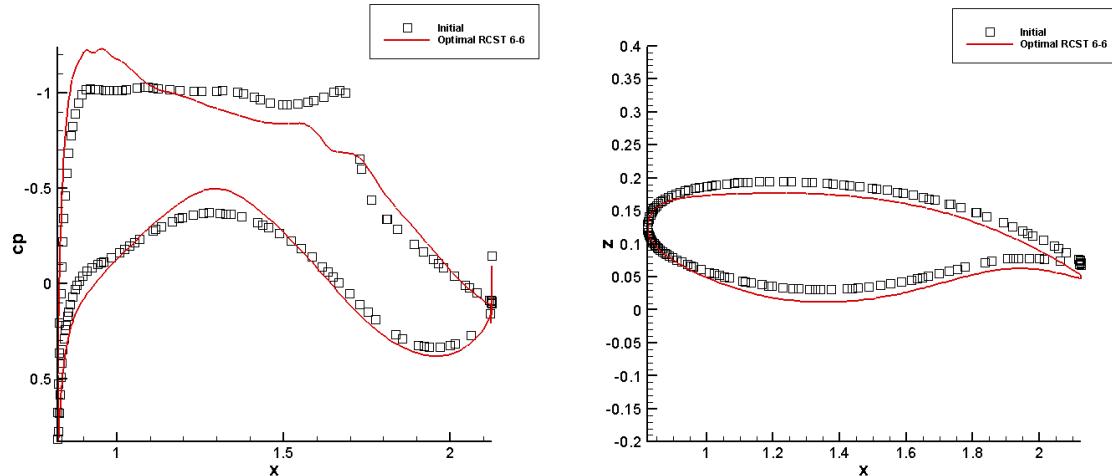


Figure 8.34  $C_p$  distribution (left) and aerofoil shapes (right) at 30% of wingspan

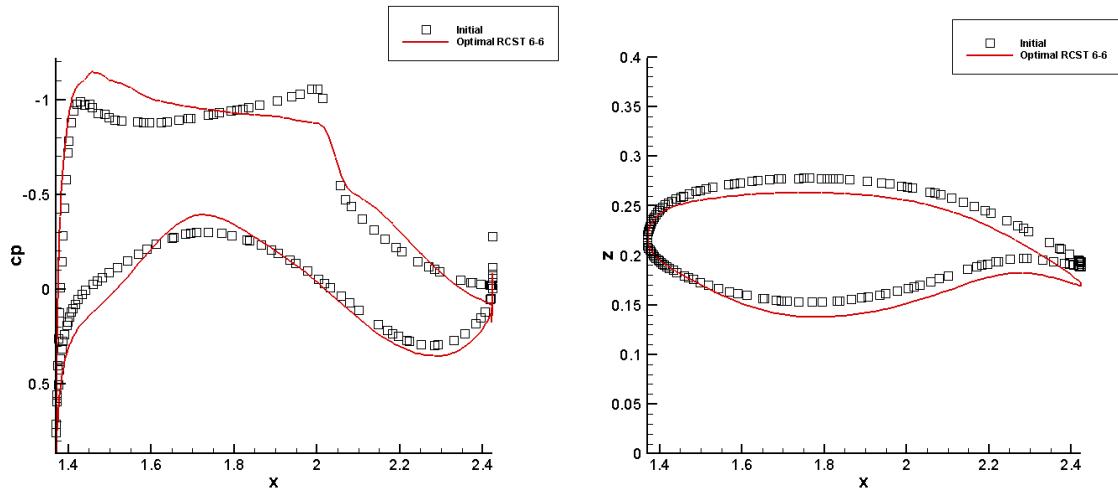


Figure 8.35 Cp distribution (left) and aerofoil shapes (right) at 50% of wingspan

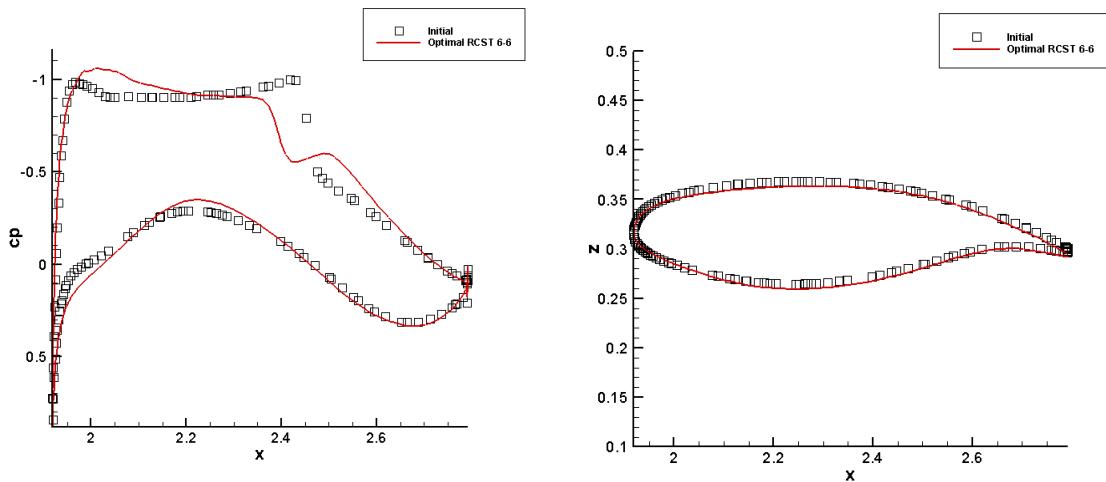


Figure 8.36 Cp distribution (left) and aerofoil shapes (right) at 70% of wingspan

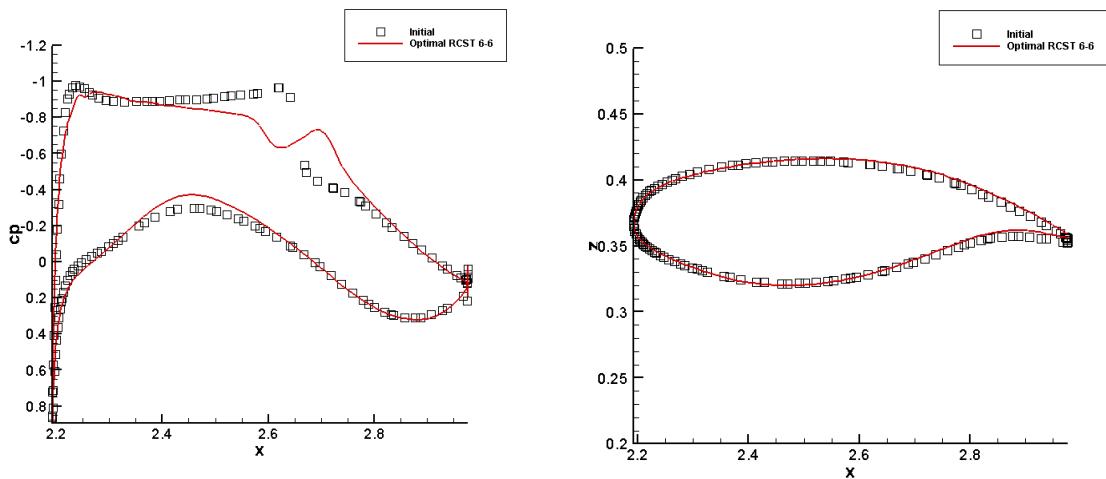


Figure 8.37 Cp distribution (left) and aerofoil shapes (right) at 80% of wingspan

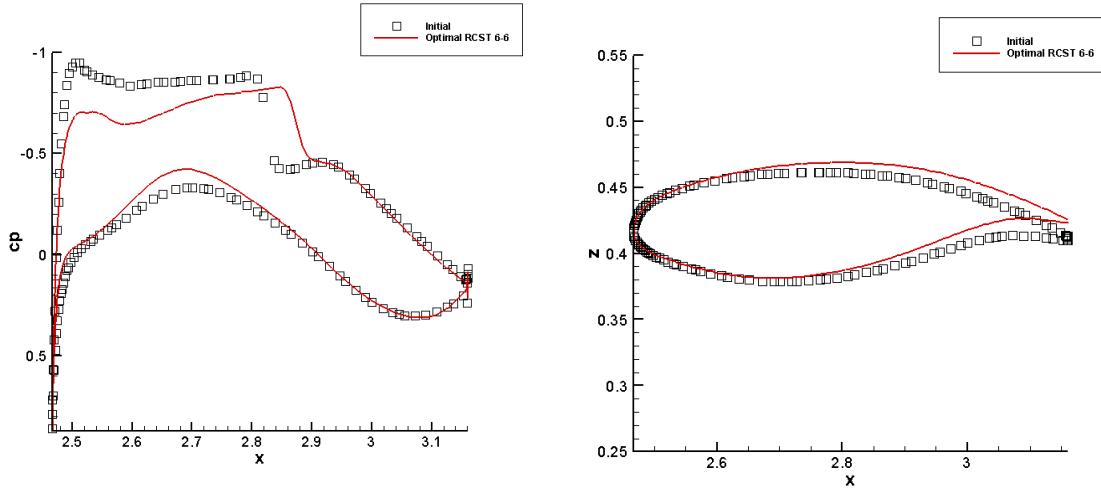


Figure 8.38 Cp distribution (left) and aerofoil shapes (right) at 90% of wingspan

As regards the above figures, firstly the high pressure suction appears at 10%, 30%, 50% and 70% of the wingspan. Secondly, they illustrate that the Cp distribution of the optimal geometry on the lower side of the wing is modified slightly compared with the Cp distribution on the original geometry.

Finally, the Cp distributions on the upper surface show that the shock wave is almost removed only at 30% of wingspan, and significantly weakened at 10%, 50% and 80% of wingspan. At 70% of wingspan, although the shock wave has been weakened, its location has moved forward to the leading edge direction, which is not good for drag reduction. At 90% of wingspan, the shock is weakened and moved downward to the trailing edge.

The twist distribution is plotted in Figure 8.39 and compared with original CST method. The lift distributions of optimal results of RCST and CSR are plotted in Figure 8.40 and aerodynamic coefficients are shown in Table 8.4.

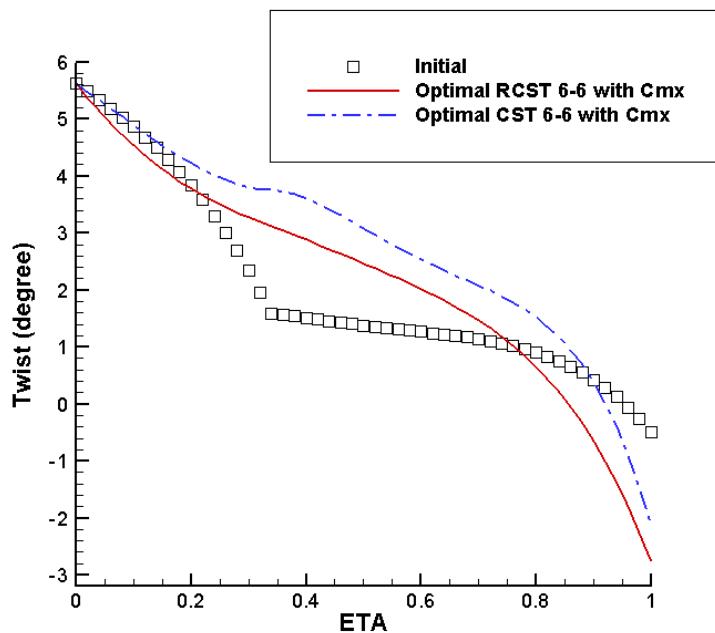


Figure 8.39 Twist distribution

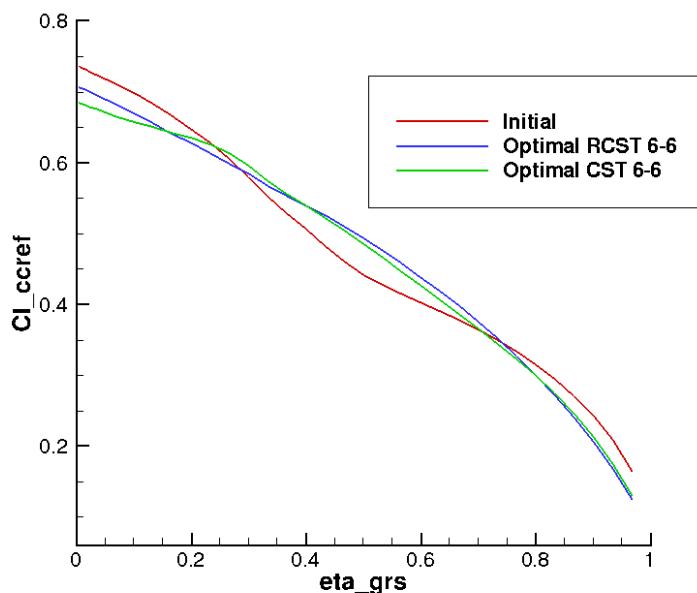


Figure 8.40 The lift distribution along span

Table 8.4 Aerodynamic coefficients of optimal results using RCST (drag units in drag count)

	Cl	Cd_total	Cd_wave	Cd_induced	Cd_vis	Cl/Cd	Cmx
Initial	0.6	312.71	30.84	133.69	136.35	19.19	1.287
Optimal RCST	0.6	268.39	10.32	128.89	119.53	22.36	1.286
	-	<b>-14.17%</b>	<b>-66.54%</b>	-3.59%	-12.34%	16.51%	-0.05%

The twist distribution shows that the twist angle distribution of optimal geometry of the RCST method is similar to that with optimal geometry of the CST method. However, the RCST method obtains about 1 more degree of pitch down the twist. The twist distribution curve is smoother than in the original CST.

The results of lift distribution obtained by RCST and CST are very similar to each other. The centre of lift of the new results is 40.05%, and of the initial geometry is 40.02%. Therefore, in this optimisation, the centre of lift has been constrained successfully in the RCST method. The aerodynamic coefficients show that the optimisation using RCST only with Cmx constraint reduces the total drag by 14.17%, compared with 17.45% in optimisation using the CST method. The wave drag is reduced by 66.54% in RCST, which is much less than the CST method. The induced drag is reduced to 128.89 drag counts, which is similar to the optimal result of CST method. The viscous drag is only reduced by 12.34%, which is also less than the 14.91% of the CST method.

The results show that RCST in 3D wing optimisation does not reduce the wave and viscous drag as much as in the original CST method. The reason is that the weighting coefficients take more local control to improve the inverse fitting accuracy due to the complexity of 3D wing geometry. However, although the fitting accuracy is improved, this could reduce the surface control capability of the original design parameters. Therefore, if the weight coefficients are fixed during optimisation, the capability of RCST for drag reduction is not as good as the original CST method. However, using the weight coefficients in optimisation is not recommended since this will significantly increase the number of design variables and bring non-linear effects into optimisation.

Therefore, in the 3D case, RCST is suitable for the case which requires only slight modification of initial geometry.

### 8.3 Winglet optimisation

As presented in Chapter 4, the winglet is an important aerodynamic component for future aircraft. Also, industry is interested in retrofitting the winglet to existing wings. Therefore, winglet optimisation is performed in this thesis.

The F6 wing is still employed in this case. Four types of winglet, namely winglet-2, winglet-2, winglet-3 and winglet-4, are employed to retrofit to the F6 wing. The definitions of four winglets are consistent with Chapter 4. The winglet-1 and winglet-2 are upward winglets, and the winglet-3 and winglet-4 are downward winglets. The flow condition is kept the same as in previous cases.

However, there are two limitations to using mesh adjoint in this case. The first is that the CST winglet parameterisation method defines the parametric model along the wing extension. The wing extension is respect to the arc-length of the winglet leading edge which is decided by winglet planform parameters. However, the arc-length of the complex curve is only calculated numerically and hard to be differential analytically. Therefore, it is hard to directly calculate analytical surface sensitivities. The second reason is due to the limitation of mesh adjoint solver presented in the introduction of this chapter. Therefore, this test is carried out by ‘Volgrad’ strategy. However, using ‘Volgrad’ requires performing NDV times of mesh deformation, which is time consuming. Hence, only the winglet is optimised in this case; the wing shape is fixed.

The initial winglet is considered as a direct extension of the wingtip with respect to the winglet planform parameters. Therefore, the initial surface parameters of the winglet are same as for the F6 wingtip parameters. In order to reduce the number of design variables, the winglet is parameterised by CST with 6<sup>th</sup> order in chordwise and 6<sup>th</sup> order in spanwise. The twist angle and trailing thickness of the initial winglet and the F6 wingtip are

constants. The twist and trailing edge thickness distribution is controlled by 3<sup>rd</sup> order Bernstein polynomials. The twist is limited between -5.7° and 5.7° for high flexibility, and the non-dimensional trailing edge thickness is limited between 0.003 and 0.005. The other initial design parameters of four winglet planforms and their bounds are listed in Tables Table 8.5 to Table 8.8.

Table 8.5 Planform parameters of winglet-1

	Lower bound	Initial value	Upper bound
$P_0P_3/P_0P_1$	0.2	0.4	0.6
$P_1P_4/P_0P_1$	0.2	0.4	0.6
$\alpha_{Le\_trans}$	29	35	40
$\alpha_{Le\_winglet}$	29	40	50
$P'_0P'_3/P_0P_1$	0.2	0.4	0.6
$P'_1P'_4/P_0P_1$	0.2	0.4	0.6
$\alpha_{Te\_winglet}$	19	30	35
$\alpha_{Te\_trans}$	19	25	30
$P_{TE\_wing}P_5/P_{TE\_wing}P_7$	0.2	0.4	0.6
$P_6P_5/P_{TE\_wing}P_7$	0.2	0.4	0.6
$P1_y$	0.1	0.3	0.42
$P1_z$	0.1	0.25	0.5
$P2_y$	0.05	0.12	0.32
$P2_z$	0.1	0.734	1.5

Table 8.6 Planform parameters of winglet-2

	Lower bound	Initial value	Upper bound
$P_0P_3/P_0P_1$	0.1	0.26	0.3
$P_1P_4/P_0P_1$	0.2	0.5	0.6
$\alpha_{Le\_winglet}$	29	45	50
$\alpha_{Le\_tip}$	29	60	80
$P'_0P'_3/P_0P_1$	0.2	0.4	0.6
$P'_1P'_4/P_0P_1$	0.2	0.3	0.6
$\alpha_{Te\_tip}$	19	35	40
$\alpha_{Te\_winglet}$	19	30	35
$P_{TE\_wing}P_5/P_{TE\_wing}P_7$	0.2	0.4	0.6
$P_6P_5/P_{TE\_wing}P_7$	0.2	0.4	0.6
$\alpha_{dihedral\_winglet}$	-89	89	89
$P1_y$	0.1	0.42	0.42
$P1_z$	0.1	0.8	1

Table 8.7 Planform parameters of winglet-3

	Lower bound	Initial value	Upper bound
$P_0P_3/P_0P_1$	0.2	0.4	0.6
$P_1P_4/P_0P_1$	0.2	0.4	0.6
$\alpha_{Le\_trans\_}$	29	35	40
$\alpha_{Le\_winglet}$	29	40	50
$P'_0P'_3/P_0P_1$	0.2	0.4	0.6
$P'_1P'_4/P_0P_1$	0.2	0.4	0.6
$\alpha_{Te\_winglet}$	19	30	35
$\alpha_{Te\_trans}$	19	25	30
$P_{TE\_wing}P_5/P_{TE\_wing}P_7$	0.2	0.4	0.6
$P_6P_5/P_{TE\_wing}P_7$	0.2	0.4	0.6
$P1_y$	0.1	0.3	0.42
$P1_z$	-0.3	-0.425	-0.01
$P2_y$	-0.38	0.12	0.32
$P2_z$	-0.4	-0.367	-0.01

Table 8.8 Planform parameters of winglet-4

	Lower bound	Initial value	Upper bound
$P_0P_3/P_0P_1$	0.1	0.26	0.3
$P_1P_4/P_0P_1$	0.2	0.5	0.6
$\alpha_{Le\_winglet}$	29	45	50
$\alpha_{Le\_tip}$	29	60	80
$P'_0P'_3/P_0P_1$	0.2	0.4	0.6
$P'_1P'_4/P_0P_1$	0.2	0.3	0.6
$\alpha_{Te\_tip}$	19	35	40
$\alpha_{Te\_winglet}$	19	30	35
$P_{TE\_wing}P_5/P_{TE\_wing}P_7$	0.2	0.4	0.6
$P_6P_5/P_{TE\_wing}P_7$	0.2	0.4	0.6
$\alpha_{dihedral\_winglet}$	-89	-89	89
$P1_y$	0.1	0.42	0.42
$P1_z$	-0.5	-0.4	-0.01

The total number of design variables is 114 for winglet-1 and winglet-3 and 113 for winglet-2 and winglet-4. The optimisation is to minimise the total drag under the constrained lift. Therefore, the optimisation setup is same as Equation 8.1, and described as Equation 8.4.

$$\begin{aligned}
 \min \quad & I = Cd - \frac{\left(\frac{\partial Cd}{\partial \alpha}\right)}{\left(\frac{\partial Cl}{\partial \alpha}\right)} (Cl - Cl_{target}) \\
 \text{subject to} \quad & V_i \geq V_{i,\text{initial}} \quad i = 1, 2, \dots, 21 \\
 & \text{twist smoothness condition} = 0 \\
 & \text{trailing edge thickness smoothness condition} = 0
 \end{aligned} \tag{8.4}$$

For winglet-1 and winglet-3, 12 sections along the winglet on the wing extension are extracted to be constrained with torsion box volume. The total span of winglet-1 and winglet-3, which are  $P1_y + P2_y$ , is constrained between 0.30 and 0.42 which is 7.8% of the wing span length. The height of winglet-1, which is  $P1_z + P2_z$ , is constrained

between 0.6 and 1.0. The height of winglet-3, which is  $P1_z + P2_z$  is constrained between -0.5 and -0.4.

For winglet-2 and winglet-4, 6 sections are extracted since they are only composed of one part. No extra constraints are employed to limit the total span since the total span has been limited in the bound of design parameters.

The optimisation is then carried out. The computational meshes in this case are more than 4 million. Considering the computational cost, the optimisation is stopped at around 15 iterations. The optimisation history is shown in Figure 8.41 and 8.42.

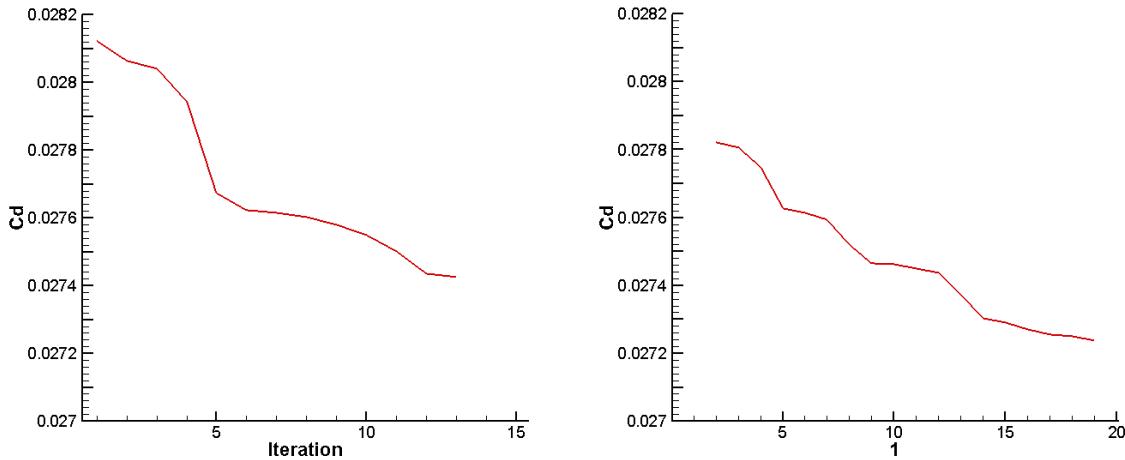


Figure 8.41 Optimisation history of drag of winglet-1 (left) and winglet-2 (right)

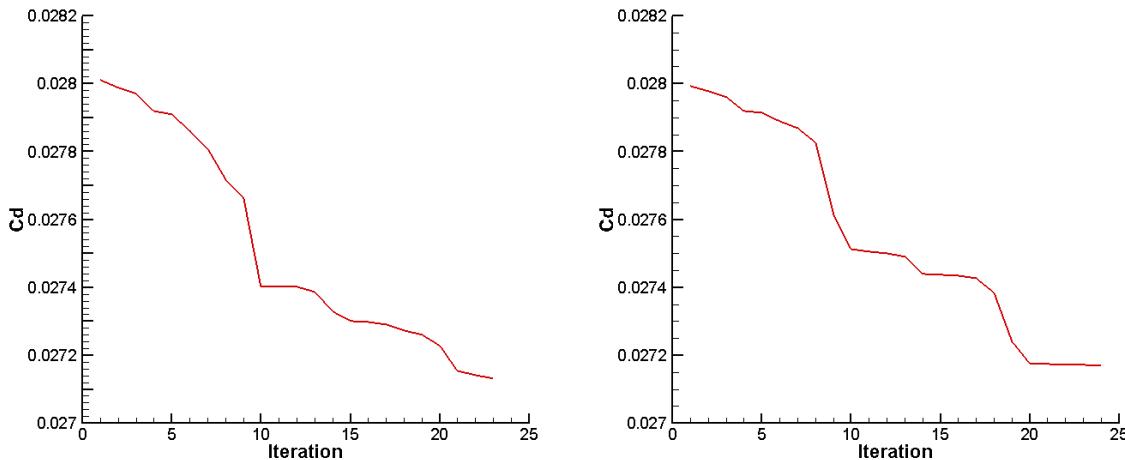


Figure 8.42 Optimisation history of drag of winglet-3 (left) and winglet-4 (right)

The optimisation history shows that the drag has been significantly reduced. The winglet-3 reaches convergence after 10 iterations. The other cases provide about 10 drag counts reduction. The geometric volume constraints are all satisfied during optimisation. The aerodynamic coefficients of the initial and optimised winglet-1 are shown in Table 8.9, and those for the initial and optimised winglet-2 are shown in Table 8.10.

Table 8.9 Aerodynamic coefficients of winglet-1 (drag unit in drag count)

	Cl	Cd_tot	Cd_w	Cd_i	Cd_v	Cl/Cd	Cmx
Wing no winglet	0.6	308.95	33.32	133.53	134.97	19.42	1.288
Initial winglet	0.6	281.22	24.63	107.31	140.6	21.33	1.403
Optimised winglet	0.6	274.27	22.39	105.79	137.9	21.88	1.418
Initial/ Optimised		-2.47%	-9.09%	-1.42%	-1.92%	2.59%	1.07%
Wing/ Optimised winglet		-11.21%	-32.80%	-20.77%	2.15%	12.68%	10.09%

Table 8.10 Aerodynamic coefficients of winglet-2 (drag unit in drag count)

	Cl	Cd_tot	Cd_w	Cd_i	Cd_v	Cl/Cd	Cmx
Wing no winglet	0.6	308.95	33.32	133.53	134.97	19.42	1.288
Initial winglet	0.6	278.6	24.58	108.89	137.2	21.53	1.398
Optimised winglet	0.6	272.91	22.83	107.32	135.7	21.99	1.41
Initial/ Optimised	-	-2.04%	-7.12%	-1.44%	-1.11%	2.14%	1.14%
Wing/ Optimised winglet	-	-11.65%	-31.48%	-19.63%	0.56%	13.24%	9.78%

The aerodynamic coefficients show that retrofitting the winglet on the wing could significantly reduce the drag. The winglet-1 reduced the total drag by 27 drag counts in which wave drag and induced drag are reduced by about 8 drag counts and drag 26 counts, respectively, and the viscous drag is increased by about 6 drag counts. In the wing with winglet-2, total drag is reduced by about 30 drag counts, which includes 8 drag counts for wave drag, 25 drag counts induced drag reduction and 3 drag counts viscous drag increase. It is noticed here that the winglet-2 could effectively reduce the wave drag and induced drag by amounts similar to the winglet-1. However, the viscous penalty of the winglet-2 is smaller than for the winglet-1.

The optimised results show that the optimisation further reduces the total drag by about 7 drag counts on both winglet-1 and winglet-2. The contribution is mainly from the wave drag reduction. In the optimised winglet, both winglet-1 and winglet-2 reach the maximum span of 0.42; however, their height is reduced by about 0.1 to compromise the viscous drag. This could be noticed from the Cp contour and skin friction Cf line on the surface in Figure 8.43 to 8.46.

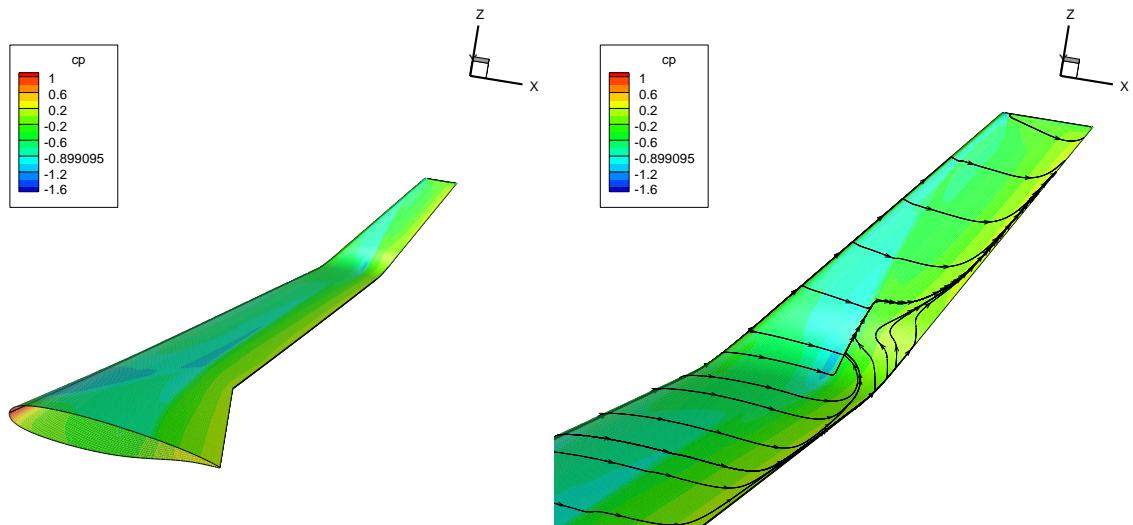


Figure 8.43 The Cp contour and zoomed in view with Cf lines of initial design of winglet-1

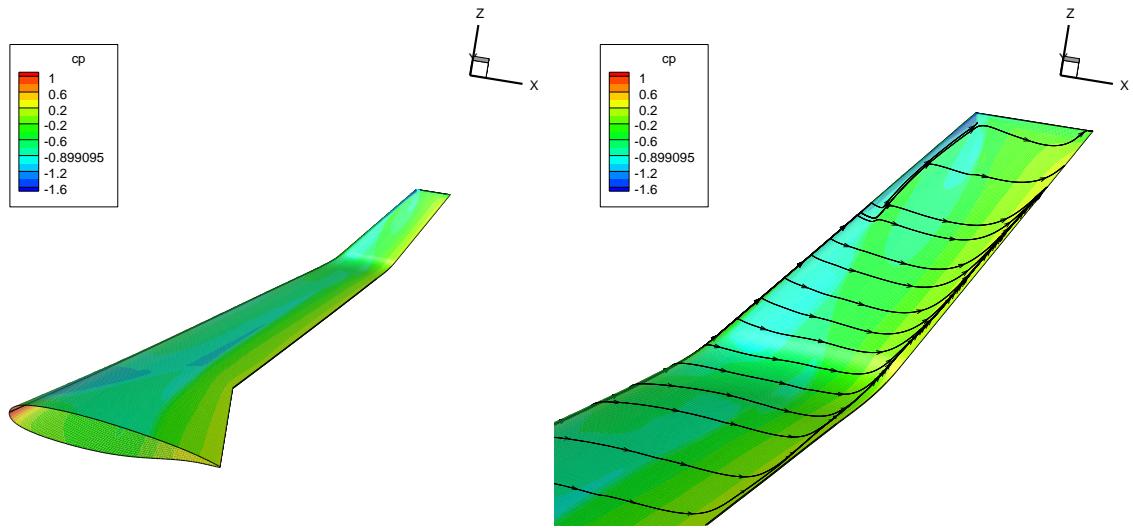


Figure 8.44 The Cp contour and zoomed in view with Cf lines of optimised design of winglet-1

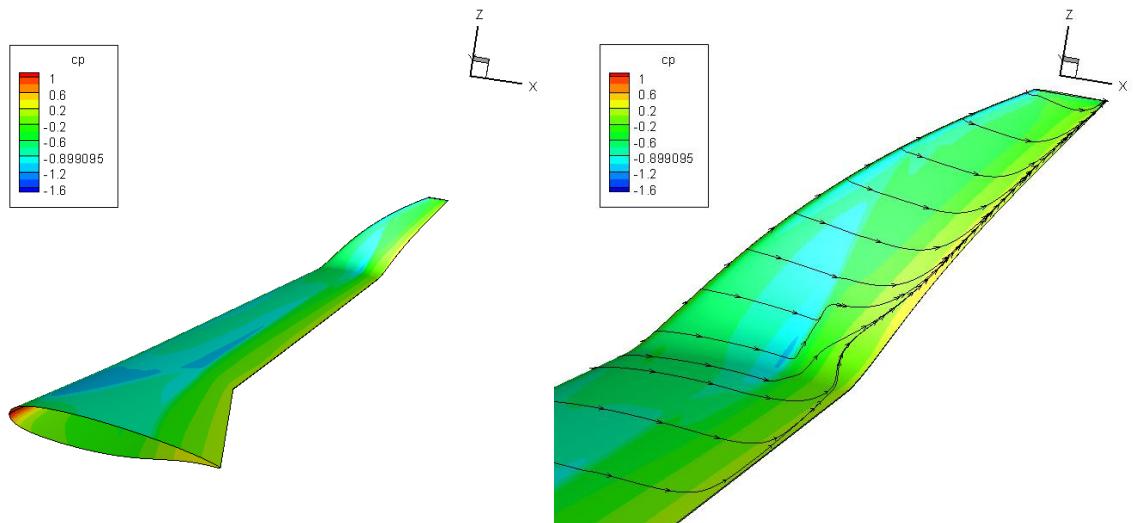


Figure 8.45 The Cp contour and zoomed in view with Cf lines of initial design of winglet-2

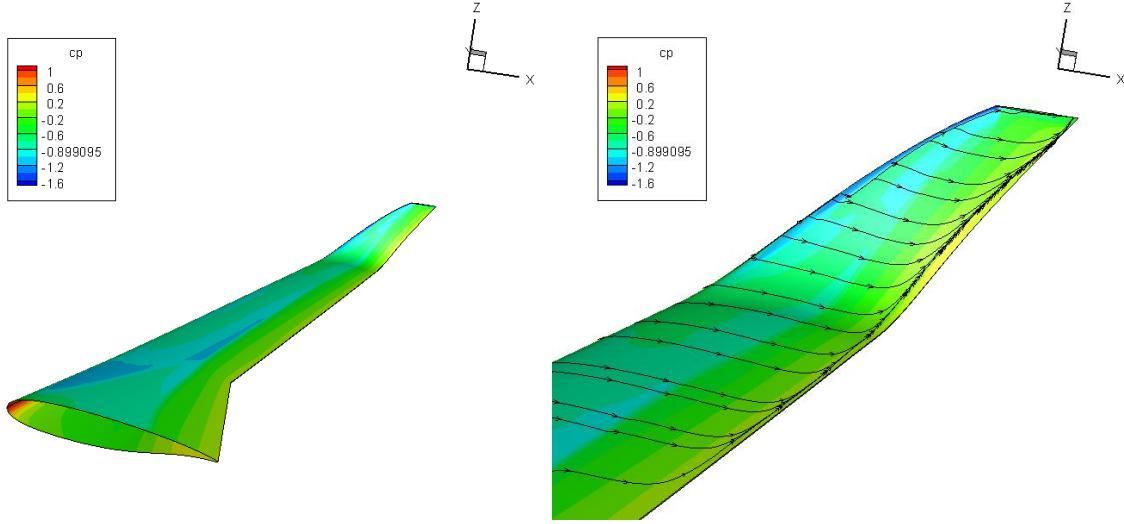


Figure 8.46 The Cp contour and zoomed in view with Cf lines of optimised design of winglet-2

The Cp contour figures show that both of the initial winglets have a shock in the junction area between wing and winglet. This shock also causes reattachment flow behind it. In the optimised winglet, this shock wave is weakened and flow reattachment is removed. However, a large pressure suction has appeared in the optimised geometry near the winglet tip. This causes a weak shock in the leading edge. The Cl distribution and Cd wave drag distribution along the span are plotted in Figure 8.47 and Figure 8.48.

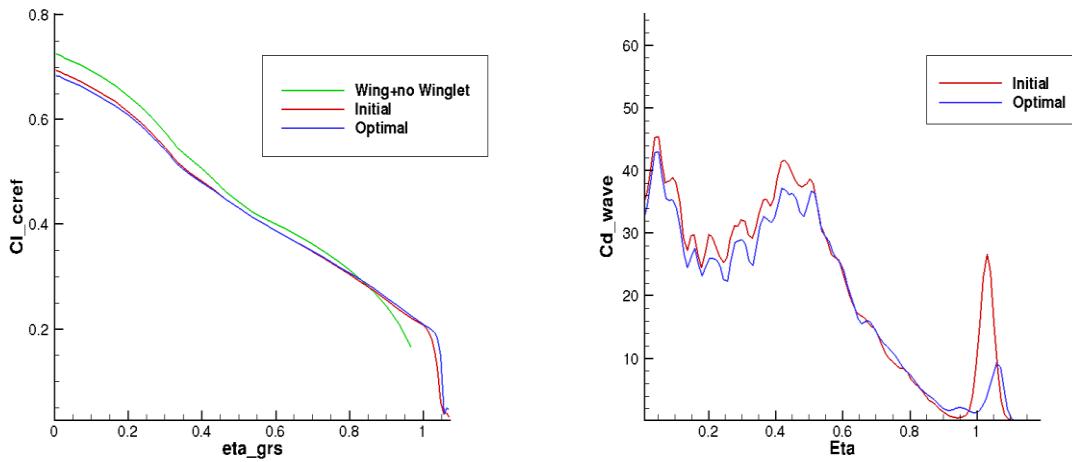


Figure 8.47 Cl (left) and Cd wave drag (right) distribution along span of winglet-1

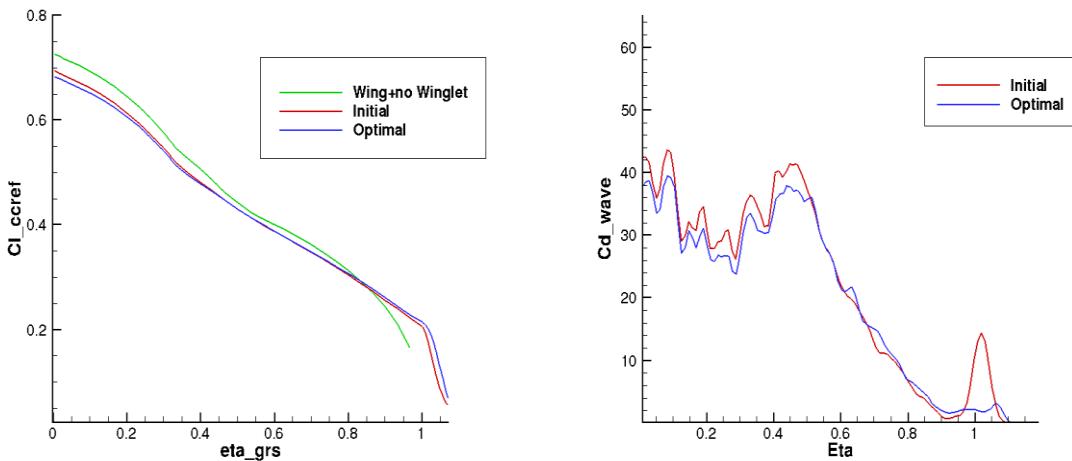


Figure 8.48 Cl (left) and Cd wave drag (right) distribution along span of winglet-2

The Cl figures illustrate that the lift is slightly increased at the optimised winglet. Therefore, the induced drag is further reduced a little. The Cd wave drag distribution shows that the optimisation successfully reduced the wave drag at the junction area in the initial geometry. However,  $Cm_x$  is increased by about 10% with respect to the clear wing. The wing root bending is the key factor in winglet design. This requires multi-disciplinary constraints in future work.

The aerodynamic coefficients of the wing without winglet, initial and optimised winglet-3 are shown in Table 8.11, and initial and optimised winglet-4 are shown in Table 8.12.

Table 8.11 Aerodynamic coefficients of winglet-3 (drag unit in drag count)

	Cl	Cd_tot	Cd_w	Cd_i	Cd_v	Cl/Cd	Cmx
Wing no winglet	0.6	308.95	33.32	133.53	134.97	19.42	1.288
Initial winglet	0.6	280.1	25.65	113.26	135.2	21.41	1.357
Optimised winglet	0.6	271.56	21.61	113.85	129.22	22.09	1.360
Initial/Optimised	-	-3.05%	-15.75%	0.52%	-4.42%	3.20%	0.23%
Wing/Optimised winglet	-	-12.10%	-35.14%	-14.74%	-4.26%	13.77%	<b>5.60%</b>

Table 8.12 Aerodynamic coefficients of winglet-4 (drag unit in drag count)

	Cl	Cd_tot	Cd_w	Cd_i	Cd_v	Cl/Cd	Cmx
Wing no winglet	0.6	308.95	33.32	133.53	134.97	19.42	1.288
Initial winglet	0.6	279.92	24.98	114.12	133.01	21.43	1.355
Optimised winglet	0.6	271.75	21.47	110.87	131.15	22.08	1.372
Initial/Optimised	-	-2.92%	-14.05%	-2.85%	-1.40%	3.03%	1.25%
Wing/Optimised winglet	-	-12.04%	-35.56%	-16.97%	-2.83%	13.69%	<b>6.51%</b>

In the initial design, both winglet-3 and winglet-4 are able to reduce the total drag about 28 drag counts, which is slightly less than winglet-1 and winglet-2. The wave drag is reduced by a level similar to the upward winglet. The downward winglet reduced the induced drag by less than the upward winglet. However, the downward winglet could

maintain the viscous drag compared to the clear wing. Similarly, the winglet-4 has less viscous drag than the winglet-3. In addition,  $C_{mx}$  of the initial downward winglets are much less than that of the upward winglets. This is because the direction of aerodynamic force on the downward winglets may point to outer of the wing and then generate a slightly negative bending momentum. This could be a benefit for overall aircraft design.

The optimisation results show that the total drag is further reduced by about 8 drag counts. The total drag of the optimal solutions obtained by both of the downward winglets is about 271 drag counts, which is smaller than the upward winglets. For the winglet-3, the main drag reduction is the contribution from wave drag and viscous drag. The induced drag is not changed since the height of the winglet-3 has only increased from -0.4 to -0.45. The contribution to drag reduction of the winglet-4 is from wave drag and induced drag, and viscous drag is only reduced by about 2 drag counts. For the winglet-4, because the height is increased from -0.4 to -0.5, the induced drag is reduced by about 3.3 drag counts but the viscous drag is only reduced by 2 drag counts.  $C_{mx}$  is increased by 5.6% and 6.51% with respect to the clear wing in winglet-3 and winglet-4, respectively. This is much less than for the upward winglet. The  $C_p$  distribution on the surface is plotted in following figures.

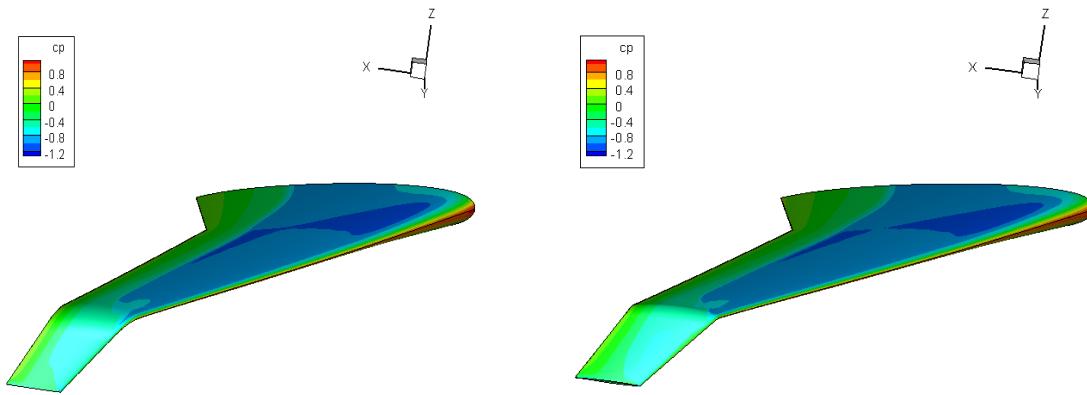


Figure 8.49 Cp Contour of initial and optimised winglet-3

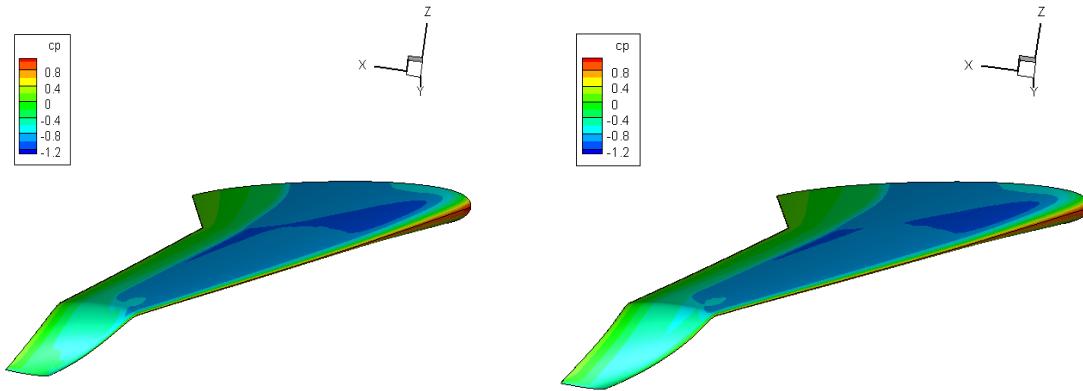


Figure 8.50 Cp Contour of initial and optimised winglet-4

The Cp contour plots show that both of the optimal downward winglets are larger than their initial design. This shows that the optimiser tries to increase the area of winglet to carry more loads to reduce the induced drag and angle of attack. As a result the wave drag is reduced on the wing.

Moreover, another phenomenon is observed from the Cp distribution figures: there is no shock wave on the downward winglet. This property is much better than for the upward winglet.

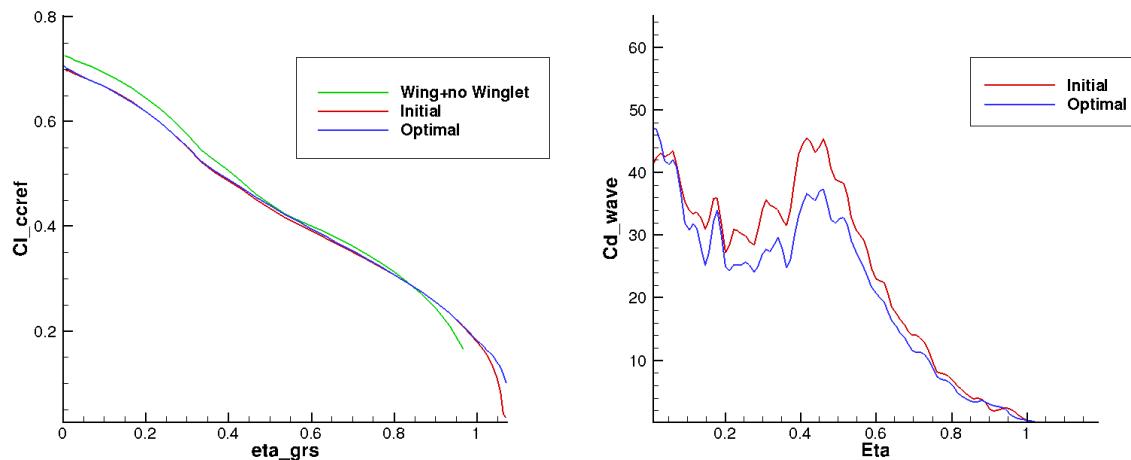


Figure 8.51 Cl (left) and Cd wave drag (right) distribution along span of winglet-3

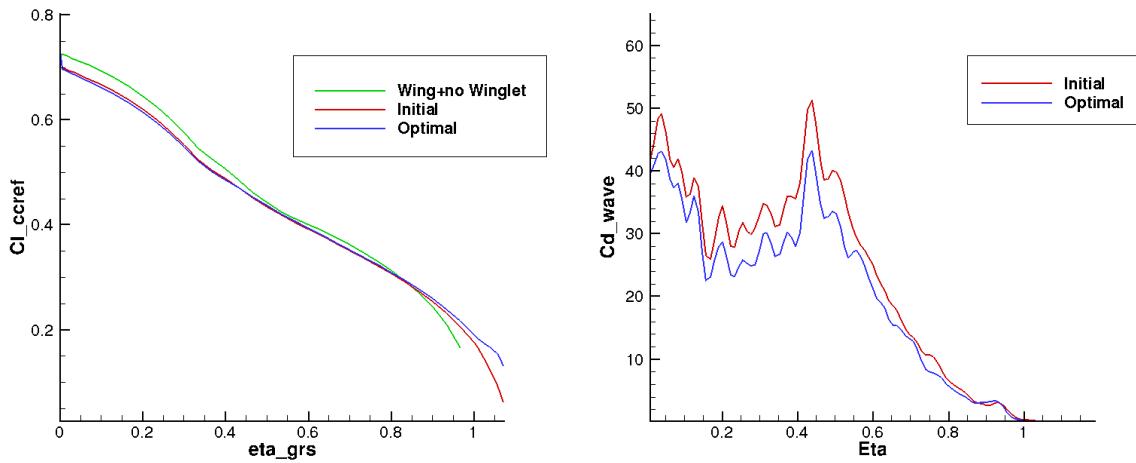


Figure 8.52  $Cl$  (left) and  $Cd$  wave drag (right) distribution along span of winglet-4

The above Figure 8.51 and 8.52 show the  $Cl$  and  $Cd$  wave drag distribution along the spanwise direction. The  $Cl$  distribution figures prove that the local lift on the optimised downward winglet has been increased. The downward winglet has more smoothness lift distribution at the junction of wing and winglet than the upward winglet. The  $Cd$  wave drag distribution of optimal downward winglets show that no wave drag is detected at both of the downward winglets, and there is slight wave drag reduction on the wing due to the variation of angle of attack.

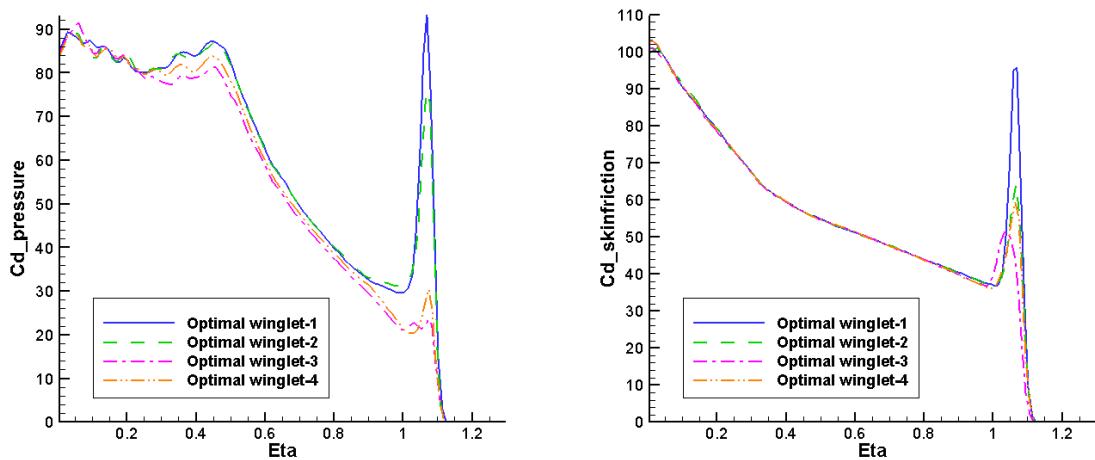


Figure 8.53  $Cd$  pressure drag (left) and friction drag (right) of optimal results of four types of winglet

Figure 8.53 compares the pressure drag and skin friction drag distribution along wingspan of the optimal results of four types of winglet. It clearly shows that both of the upward winglets have huge amount of pressure drag at the junction area of wing and winglet, and both downward winglets obtained a fairly small pressure drag. The skin friction is mainly related to the height of winglet. Therefore, the optimal winglet-1 is the highest, 0.85, and obtained the highest skin friction drag. The optimal winglet-2 and winglet-4 have heights 0.65 and 0.5, hence their skin friction drag is medium level. The optimal winglet-3 has 0.45 height which causes the lowest skin friction drag.

This research shows that the CST parameterisation is able to handle winglet optimisation successfully. It could be used in future winglet optimisation research. The good properties of the downward winglet have been investigated. First, the downward winglet has less wave drag and pressure drag than the upward winglet due to the open junction on the upper surface. Second, the downward winglet is able to achieve similar level of total drag reduction as the upward winglet but with less  $C_{mx}$ , which is significant for overall aircraft design. Therefore, the downward winglet is worth further study in the future.

#### 8.4 Shock bump optimisation on the wing

In Chapter 7, three types of shock control bumps, which are CST, PARSEC and cubic polynomials, have been tested on 2D aerofoil shock reduction optimisation. Although the test shows that the type of bump does not have a significant effect on shock reduction, the CST bump could theoretically provide C2 continuity at the boundary between aerofoil and bump. In Chapter 4, the CST for a three-dimensional bump has been described. Therefore, the CST bump on a 3D wing is tested. The F6 wing is still selected as the baseline wing. The flow condition is kept same as with previous cases.

As demonstrated in Chapter 6, the mesh adjoint is able to directly provide the sensitivities of objective to surface points for design guidance. Hence, the surface sensitivity of the F6

wing is plotted to assist to deploy the bumps. Based on the work of Wong *et al* (2007), the bumps could be deployed without gap between bumps' edge. Therefore, 12 bumps are deployed one by one in the area with negative sensitivities from the wing root along wingspan. The sensitivity of Cd to surface point z direction and bump position is plotted in Figure 8.54.

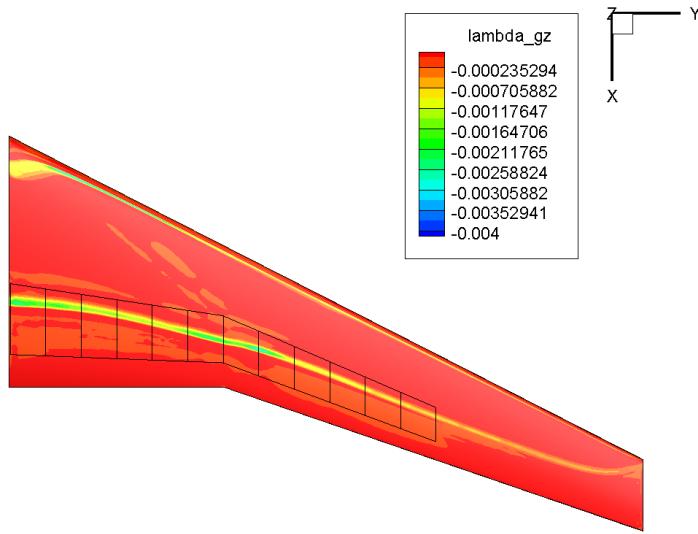


Figure 8.54 Sensitivities of Cd to surface point Z direction and boundaries of bumps

The first 6 bumps are at the inboard wing with indexes 1 to 6. Bump 1 starts at  $x=1.25$  and  $y=0.01$  and has length 0.6 and width 0.3. The leading edge sweep angle of bump 1 is  $8.60^\circ$  and the trailing edge sweep angle is  $2.24^\circ$ . Bumps 2 to 6 consistently follow bump 1 with the same width, leading edge and trailing sweep angle. The lengths of bumps 2 to 6 are continually reduced with respect to the leading edge and trailing edge of the bumps. Bumps 7 to 12 are at the outboard wing with same width, 0.3. The leading edge sweep angle of bumps 7 to 12 is  $23.35^\circ$  and their trailing edge sweep angle is  $20.37^\circ$ .

Each bump has 3<sup>rd</sup> order on chordwise, 3<sup>rd</sup> order on spanwise, and 3<sup>rd</sup> order to control height distribution polynomials. Wong (2006) found the length and width of bumps are not sensitive in optimisation. Hence, the length and width are fixed during optimisation.

The total number of design variables is  $12 \times ((4 \times 4) + 4) = 240$ . The flow condition is the same as for the previous test case with  $\text{Mach} = 0.80$ ,  $\text{Cl} = 0.6$  and  $\text{Re} = 5 \times 10^6$ . The target lift iteration is still employed, so the objective function is shown in Equation 8.5, and no extra constraint is applied.

$$\min \quad I = Cd - \frac{\left(\frac{\partial Cd}{\partial \alpha}\right)}{\left(\frac{\partial Cl}{\partial \alpha}\right)} (Cl - Cl_{target}) \quad 8.5$$

The initial bump parameters are set to zero, the parameters of height distribution are 1.0. The bound of each bump parameters is between 0.0 and 0.02, the bound of height distribution parameters is between 0.0 and 2.0.

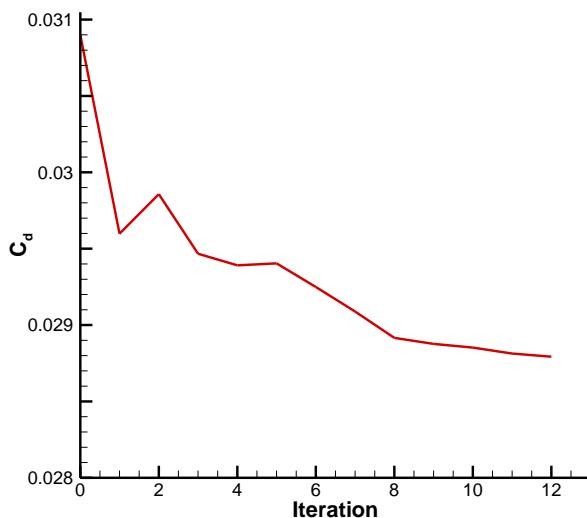


Figure 8.55 Optimisation history of Cd

Figure 8.55 shows the optimisation history of  $Cd$ . The optimisation is stopped at 12 iterations. The overall drag has been significantly reduced during optimisation. Optimised 12 bump on wing is shown in Figure 8.56. The surface  $C_p$  contour plot with range between -1.2 and -0.4 and are shown in Figure 8.57. The  $C_p$  contour plots and skin friction lines are shown in Figure 8.58.

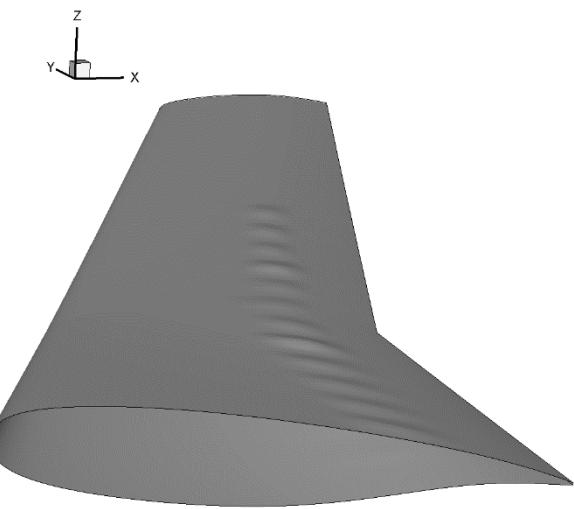


Figure 8.56 Optimised 12 bumps on wing

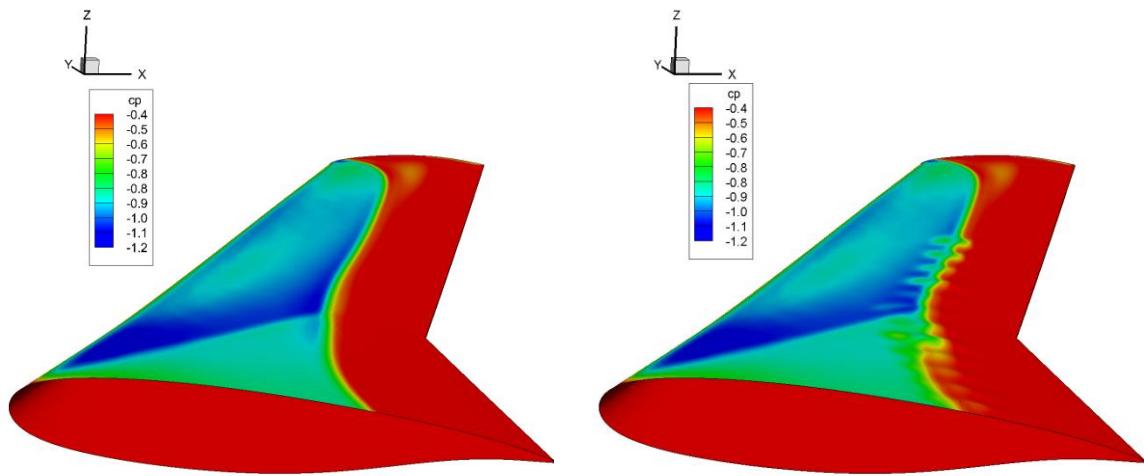


Figure 8.57 Surface Cp contour in range -1.2 to -0.4 of initial wing without bump(left)  
and optimised bumps (right)

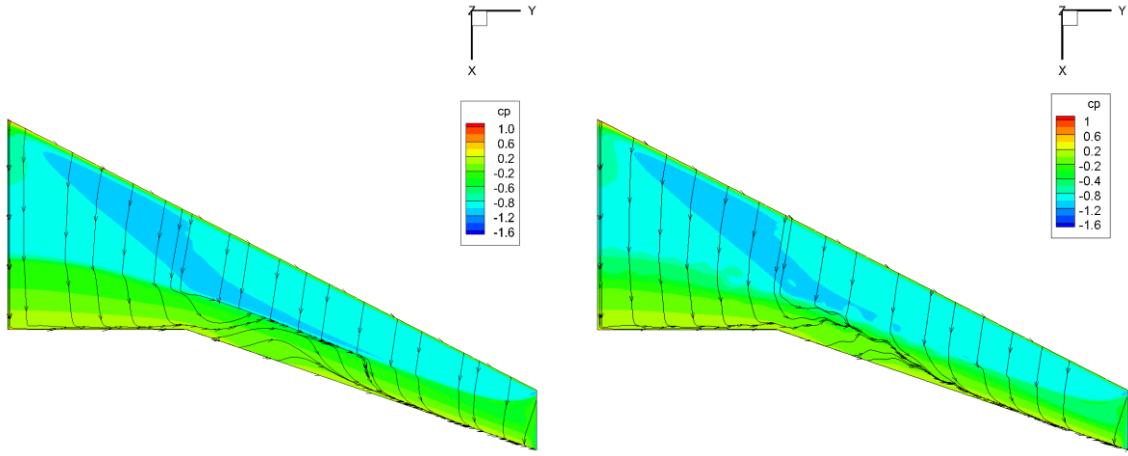


Figure 8.58 Cp Contour and Cf lines plot on upper surface of initial wing without bump (left) and optimised bumps (right)

Figure 8.56 clearly shows 12 bumps have been growth up after optimisation. The Cp contour plot in Figure 8.57 shows that the strong shock of lambda shock structure has been weakened by the bumps. The skin friction Cf lines in Figure 8.58 show that there is a large shock induced flow separation at the outboard wing. After a bump is deployed, this flow separation is reduced and the detachment line is pushed downward to the wing trailing edge. More information is shown by Cp distribution and aerofoil section cut from the wing at  $Y = 0.16, 1.36, 2.56$  and  $3.46$ , which are the middle points of bumps 1, 5, 9 and 12 (see Figure 8.59 to 8.62).

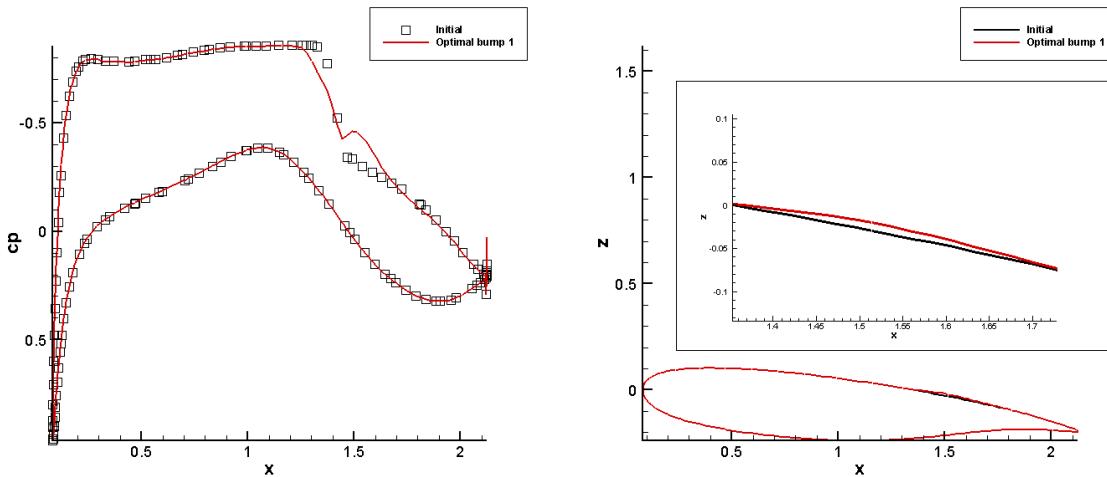


Figure 8.59 Cp distribution and aerofoil section cut at middle of bump 1

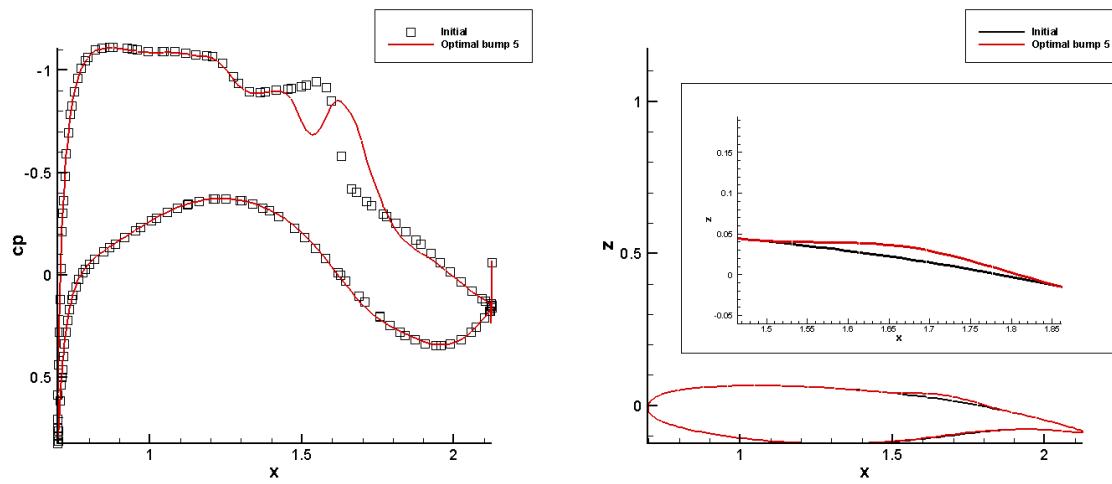


Figure 8.60 Cp distribution and aerofoil section cut at middle of bump 5

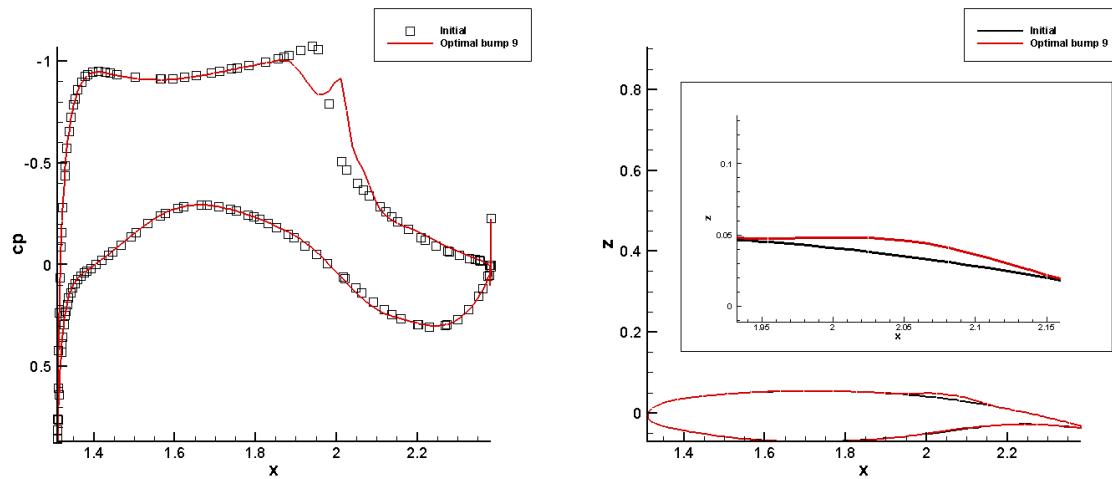


Figure 8.61 Cp distribution and aerofoil section cut at middle of bump 9

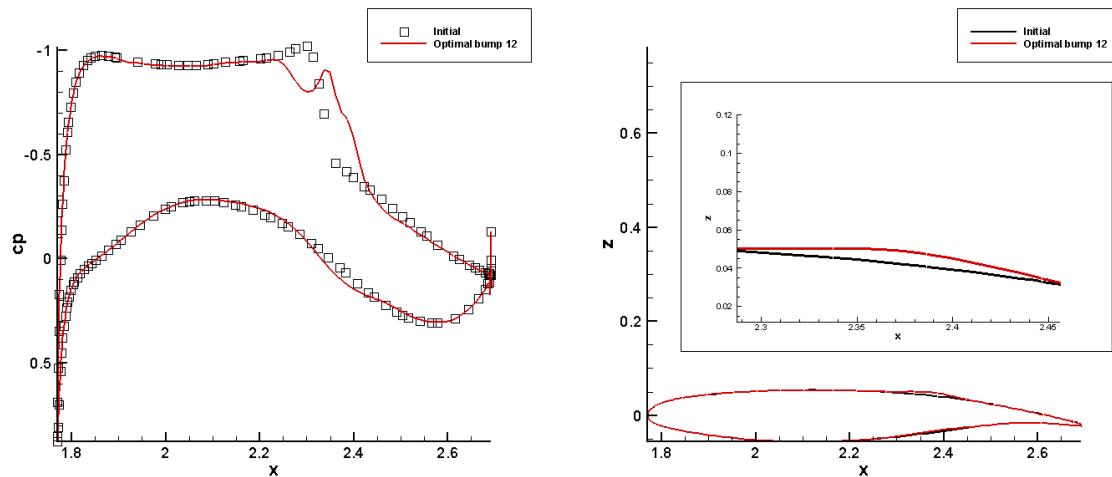


Figure 8.62 Cp distribution and aerofoil section cut at middle of bump 12

The section cut figures show that the height of a bump increases along spanwise. With increasing bump height, the shock strength decreases, in particular at bumps 9 and 12. This because the surface sensitivities obtained by mesh adjoint at the outboard wing are more than at the inboard wing. The aerodynamic coefficients and Cd wave drag distribution along spanwise are shown in Table 8.13 and Figure 8.63.

Table 8.13 Aerodynamic coefficients (drag units in drag count)

	Cl	Cd_total	Cd_wave	Cd_induced	Cd_pressure	Cd_friction	Cl/Cd
Initial	0.6	308.95	33.32	133.53	75.23	59.74	19.42
Optimal	0.6	287.83	20.18	132.31	68.19	60.00	20.85
	-	-6.84%	-39.44%	-0.91%	-9.36%	0.44%	7.34%

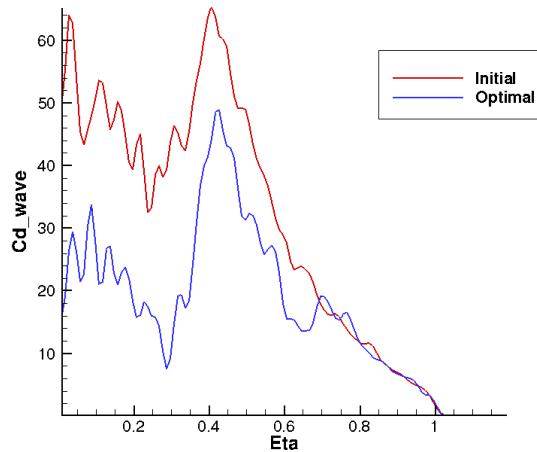


Figure 8.63 Cd wave along span of initial wing and optimal bumps

Figure 8.63 shows that the CST bump has effectively reduced the wave drag in the deployed area. Although the height of the bump is not large as at the inboard wing, the wave drag is significantly reduced at the inboard wing. The total drag is reduced by 6.84% which includes about 40% wave drag reduction and 9.36% pressure drag reduction. The penalty of friction drag is only increased by 0.26 drag counts. Therefore, this test proves that CST bump parameterisation is efficient for shock control bump optimisation, and could be an option for future research.



# Chapter 9 Conclusion and Future Work

## 9.1 Summary

There were two main objectives in this thesis: the first was to develop parameterisation methods for aircraft components, and the second was to apply parameterisation to aerodynamic optimisation with an adjoint approach.

In the first part of the thesis, the current common geometric parameterisation methods were presented. The CST and PARSEC methods were selected as starting-points after a literature review. Two parameterisation methods were proposed, which were the iCST and RCST methods. The iCST method aimed to combine the high flexibility from the CST method with the full intuitiveness from the PARSEC method. This method introduced eight extra intuitive parameters at two x-positions between the crest and the leading edge and between the crest and the trailing edge on each surface of the aerofoils in relation to the original PARSEC-12 method. Therefore, 28 design variables were used to parameterise the entire aerofoil. The power form polynomials used in PARSEC have been replaced by the 10<sup>th</sup> order CST equation for curve representation. Its performance was examined by the accuracy of inverse fitting for a range of aerofoils in comparison with the 12<sup>th</sup> order CST and PARSEC-12 methods. The RCST was to introduce rational Bernstein polynomials into the original CST methods to provide more flexibility and local control with lower order polynomials.

The comparative study of CST, PARSEC, iCST and RCST was done by comparing their inverse fitting accuracy over various aerofoils. The results showed that the 12<sup>th</sup> order CST and the iCST method with the same number of design variables are able to represent supercritical and wind turbine aerofoils with a high level of fitting accuracy, while the representation of complex NLF aerofoils is also acceptable but shows slightly higher error after the half chord position. With an insufficient number of control parameters, the standard PARSEC-12 failed to fit most of the aerofoils tested here, and was particularly

problematic for NLF and wind turbine aerofoils. The RCST showed excellent accuracy in representing all aerofoils with only 6<sup>th</sup> order polynomials. The iCST method can be viewed as an extension of the PARSEC method with full intuitive parameters. However, it can also be viewed as a transformation of the originally non-intuitive parameters in the CST method to a full set of intuitive parameters through a transformation matrix. This provided opportunities for aerodynamic designers to understand the relationship between the aerodynamic properties and the geometric features and to guide the exploration of the design space by selecting proper design variables and setting proper bounds/constraints in the optimisation process. To look at it in another way, although RCST did not provide any intuitive parameter, it increased the flexibility of the original CST method with reduction of the number of design variables. These were also favourite properties of parameterisation which are pursued in geometric parameterisation development.

Afterwards, the CST parameterisation method has been successfully developed to parameterise an entire passenger transport aircraft for aerodynamic optimisation. The eleven main components of a civil passenger aircraft are used for this research, including wing, HTP, VTP, winglet, forward part fuselage, mid-part fuselage, tail cone, belly-fairing, nacelle, flap track fairing and pylon.

The wing, HTP, VTP and winglet were defined as wing type geometry. The tests of inverse fitting of original CST and RCST parameterisation for wing have been carried out to examine the capability of representation of wing type geometry of the original CST parameterisation methods. Using the higher order Bernstein polynomial can obviously improve the accuracy of inverse fitting. A CFD simulation has been run to investigate the difference between original and approximated models on aerodynamics. It showed that the pressure distributions on the lower side of surface will not be affected on approximated models. However, the pressure distributions on the upper surface were more sensitive and difficult to match, especially in the leading edge before 20% chord area. This drawback has been improved by RCST; the results of the RCST showed significant improvement in capturing the pressure suction peak in the leading edge area.

The CST method for representing the wing tip device is developed by topology transformation of wing extension with respect to winglet planform parameters. To keep the geometric continuity between wing and winglet, a few constraints were introduced.

The fuselage and nacelle were defined as the ‘body cross-section’ type geometry. Because of the complexity of the realistic nose fuselage with cabin and window shield, a simplified nose fuselage without cabin window shield was used in this work. The simplified nose fuselage, cylindrical fuselage and tail cone can then be easily established. In some case, the profile shape of tail cone was more complicated, so the shape function was involved to assist to represent the cross-section profile. The extra constraints should be introduced to ensure the geometric continuity between three parts. The representation of nacelle inlet has been slightly modified, in which the theory of PARSEC parameterisation methods was used to provide intuitive parameters. The CST for belly fairing was developed by applying coordinate transformation of Cartesian coordinate to Cylindrical coordinate. The flap tracks fairing and simplified pylon model were treated in a similar way to the fuselage. Furthermore, CST was also developed for bump design to obtain  $C^2$  continuity between bump and initial geometry.

Finally, an intersection line calculation algorithm based on the New-Raphson method has been presented. The purpose was to calculate the intersection line directly from the CST parametric model without using an additional CAD package and to improve the intersection line accuracy. The results showed this intersection line algorithm is able to calculate the intersection line between different CST models with very high accuracy.

In the second part of this thesis, the optimisation methods for aerodynamic design have been reviewed. The adjoint approach was selected to carry out the optimisation study. The optimisation framework was developed based on python programming. The radial basis function was employed to perform mesh deformation.

The two-dimensional optimisation was performed at first, which was mainly to test the drag reduction using CST with 7<sup>th</sup> order and 10<sup>th</sup> order, and iCST and RCST with 6<sup>th</sup>

order methods. The optimisation results showed that all four methods were able to successfully reduce the drag and eliminate the shock wave. The original CST with lower order polynomials has better convergence than the CST with higher polynomial order. However, the original CST with lower order may lack local control and experiences difficulty in the shock region. The RCST showed the best optimal aerofoil with shock-free performance. At the same time, the RCST achieved the fastest convergence of all methods, due to its lower number of design variables. The iCST method was also able to effectively eliminate the shock wave. Because of the specific intuitive constraint and curvature parameters, the optimal aerofoil presented a some non-smoothness in the pressure distribution; however, it presented the highest pressure suction peak at the leading edge, which demonstrates that iCST is capable of better local control. It can be concluded that all of the methods were successful parameterisation methods for handling drag reduction. The selection of one of the three geometry parameterisation methods could depend on the specific requirement.

The two-dimensional shock control bump optimisation has been carried out to test the performance of CST, PARSEC and cubic polynomial bump parameterisation methods. The results showed that all three methods are able to significantly reduce the total drag. There was no obvious difference between the three methods in terms of their optimal solution and optimisation convergence. However, the CST bump was able to offer better  $C^2$  continuity. Therefore, the CST bump could be employed in future shock control bump studies.

Three-dimensional optimisation of wing drag reduction has been implemented. The first case was to compare the influence of the order of the CST methods. The optimisation demonstrated that higher order CST will require a little more optimisation iterations to reach convergence; however, there was no significant difference. The optimal results showed that all CST with different orders achieve very similar geometries with successful drag reduction. The spanwise high order could slightly improve the shock wave drag reduction. However, the lower order CST guaranteed the smoothness of wing surface. Furthermore, the overall drag reduction optimisation would cause the shift of the centre

of lift to reduce induced drag which was not desirable in practical engineering design. Therefore, the second optimisation was then carried out employing  $C_{mx}$  constraint to mitigate the shift of the centre of lift. The results showed that the optimisation of  $C_{mx}$  constraint could successfully reduce drag while maintaining the position of the centre of lift. Although the lift distribution may have changed slightly, the optimal result is still more practical for engineering design application than the previous test. Therefore, it suggested that the  $C_{mx}$  constraint should be employed in the future for wing optimisation. Subsequently, RCST parameterisation method has been tested, and the results showed that it was able to reduce the drag. However, the drag reduction of RCST was not as good as for the original CST, because of the complexity of the 3D wing. The weight coefficients could improve geometric fitting accuracy, but they reduced the surface sensitivities of control parameters and allow geometry only to be modified near the initial design.

The second test case of three-dimensional optimisation was to optimise the winglet. Four types of winglet, namely, winglet-1, winglet-2, winglet-3 and winglet-4, were employed to retrofit on the F6 wing. The optimisation of upward winglets showed successful drag reduction. In the initial design, both retrofitted upward winglets achieved the same level of total drag reduction with respect to the clean wing. The winglet-2 had less viscous drag than the winglet-1. The optimisations for both upward winglets were then successfully performed to reduce the shock wave drag presented in the transition area between the wing and the initial winglet. The total drag reduction obtained by two types of retrofitted initial downward winglets was slightly less than upward winglets, due to the smaller induced drag reduction. However, it had less viscous drag and  $C_{mx}$  bending momentum, both of which are essential for aircraft design. Optimisation of the downward winglet gave a higher drag reduction. The total drag of both of the optimal downward winglets was slightly lower than for the optimal upward winglets. Comparison showed that the optimal winglet-3 had the lowest pressure drag and skin friction drag of the four optimal winglets. Although the  $C_{mx}$  of optimal downward winglets was slightly increased too, they were still much less than for the upward winglet.

Finally, 12 shock control bumps represented by CST methods were deployed onto the F6 wing surface and optimised. The optimisation has successfully driven the bumps to weaken the shock strength in their deployed area and effectively reduced the wave drag. Therefore, the CST bump is also successful in the 3D case and could be used in future work.

## 9.2 Future work

Two parameterisation methods, iCST and RCST, have been developed and tested in this thesis. The optimisation results show that iCST is able to effectively reduce the shock wave drag. However, the curvature parameters may generate some non-smoothness in the pressure distribution. Therefore, in future work, it would be worth trying the lower order iCST method with removal of some curvature parameters. Furthermore, the iCST method is so far only presented for two-dimensional aerofoils: in the future, it could be further developed for the three-dimensional aerodynamic components.

The optimisation has been successfully performed in this thesis: the wing and winglet have been successfully optimised by the adjoint approach. However, the winglet optimisation in this thesis only considered the aerodynamic factor. The improvement of aerodynamic-induced drag will normally bring more loading in the winglet area, where the structure penalty must be considered in practical engineering. Therefore, future work on winglet optimisation study should involve multi-disciplinary constraints. Moreover, all the optimisation work in this thesis is done only under single flow conditions. The optimal results may not be robust with respect to variation of design conditions. Therefore, if computational resource allows, the optimisation should be carried out under multi-point design conditions in future work.

Unfortunately, the entire 3D aircraft optimisation was not implemented due to the limitation of the TAU adjoint solver, although all 3D parametric models are ready. This could be overcome in the near future with the development of the TAU mesh adjoint solver. Even if this is ready for large cases in the future, there is still one issue remaining, which is the sensitivity of the surface mesh points. Although the CST method is able to

provide analytical surface sensitivities for each component, it would be difficult when they are assembled into an entire aircraft after the intersection line is presented. The intersection line would change the geometry topology and make the actual sensitivities different from the analytical solution. Therefore, the other way to calculate surface sensitivities, such as that using design velocity proposed by Armstrong *et al.* (2009), could be employed and coupled with current parameterisation methods for entire aircraft optimisation with all its components parameterised.

## References

- ACARE. (2002). *Strategic Research Agenda*. <http://www.acare4europe.com/>
- Allen, C.B. (2002). Aeroelastic computations using algebraic grid motion. *Aeronautical Journal*, 106 (1064): 559-570.
- Anderson, G., Aftosmis, M. and Nemec, M. (2012). Parametric Deformation of Discrete Geometry for Aerodynamic Shape Design. *50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, AIAA 2012-0965
- Anderson, J.D. (2007). *Fundamentals of aerodynamics*. New York, London: McGraw Hill Higher Education.
- Anderson, W.K. and Bonhaus, D.L. (1994). An Implicit Upwind Algorithm for Computing Turbulent Flows on Unstructured Grids. *Computers & Fluids*, 23(1): 1-21.
- Anderson, W.K. and Bonhaus, D.L. (1997). *Aerodynamic design on unstructured grids for turbulent flows*. Hampton, Va. Springfield, Va., NASA Technical Report.
- Anderson, W.K., Karman, S.L. and Burdyshaw, C. (2008). *Geometry Parameterisation Using Control Grids*. AIAA 2008-6028.
- Anderson, W.K., Karman, S.L. and Burdyshaw, C. (2009). Geometry Parameterisation Method for Multidisciplinary Applications. *AIAA Journal*, 47(6). Pp. 1568-1578
- Anderson, W.K. and Venkatakrishnan, V. (1997). *Aerodynamic design optimisation of unstructured grid with a continuous adjoint formulation*. Washington, D.C: AIAA-1997-643.
- Andreoli, M., Janka, A. and Desideri, J.A. (2003). *Free-form-deformation parameterisation for multilevel 3D shape optimisation in aerodynamics*. INRIA Report No 5019.
- Antunes, A.P., Azevedo, J.a.L.F. and Silva, R.G.D. (2009). A Framework for Aerodynamic Optimisation Based On Genetic Algorithms. *AIAA 2009-1094*

- Armstrong, C.G., Robinson, T.T. and Chua, H.S. (2009). Parametric effectiveness of CAD models and strategies for its improvement. *8th World Congress on Structural and Multidisciplinary Optimisation*. Lisbon, Portugal.
- Ashill, P.R., Fulker, J.L. and Shires, A. (1993). *A novel technique for controlling shock strength of laminar-flow aerofoil sections*. 1st European Forum on Laminar Flow Technology (pp. 175-183).
- Aziz, N. and Bata, R. (1990). Bezier Surface/Surface intersection. *IEEE Computer Graphics and Applications*, 10(1): 50-58.
- Bajaj, C. and Hoffmann, C. (1988). Tracing surface intersections. *Computer Aided Geometric Design*, 5(4): 285-307.
- Baldwin, B. and Lomax, H. (1978). Thin-layer approximation and algebraic model for separated turbulent flows. *16th Aerospace Sciences Meeting*, AIAA 1978-257
- Barr, A.H. (1984). Global and Local Deformation of Solid Primitives. *Computer Graphics*, 18(3): 21-30.
- Barthelemy, J.F. and Hall, L. (1992). *Automatic differentiation as a tool in engineering design*. Washington, D.C, AIAA-1992-4743.
- Batina, J.T. (1991). Unsteady Euler Algorithm with Unstructured Dynamic Mesh for Complex-Aircraft Aerodynamic Analysis. *AIAA Journal*, 29(3): 327-333.
- Baxter, B. (1992). *The interpolation theory of radial basis functions*. PhD thesis, University of Cambridge.
- Baysal, O. and Eleshaky, M. (1992). Aerodynamic design optimisation using sensitivity analysis and computational fluid dynamics. *AIAA Journal*, 30(3): 718-725.
- Beckert, A. and Wendland, H. (2001). Multivariate interpolation for fluid-structure-interaction problems using radial basis functions. *Aerospace Science and Technology*, 5(2): 125-134.
- Bentamy, A., Guibault, F. and Trepanier, J.-Y. (2005). *Aerodynamic Optimisation of a Realistic Aircraft Wing*. AIAA, 2005-332.

- Bogue, D. and Crist, N. (2008). CST Transonic Optimisation Using Tranair++. *46th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA 2008-321.
- Boor, C.D. (1972). On Calculating with B-Splines. *Journal of Approximation Theory*, 6: 50-62.
- Boor, C.D. (1977). Package for Calculating with B-Splines. *Siam Journal on Numerical Analysis*, 14(3): 441-472.
- Boussinesq, J. (1877). Theorie de l'Ecoulement Tourbillonant. *Comptes-Rendus de l'Academie les Sciences*, 23: 46-50.
- Brezillon, J. and Dwight, R.P. (2005). Discrete Adjoint of the Navier-Stokes Equations for Aerodynamic Shape Optimisation. *Evolutionary and Deterministic Methods for Design* (EUROGEN).
- Brezillon, J. and Gauger, N.R. (2004). 2D and 3D aerodynamic shape optimisation using the adjoint approach. *Aerospace Science and Technology*, 8(8): 715-727.
- Brezillon, J., Ronzheimer, A., Haar, D., Abu-zurayk, M. and Lummer, M. (2012). *Development and application of multi-disciplinary optimisation capabilities based on high-fidelity methods*. AIAA 2012-1757.
- Brodersen, O., Eisfeld, B., Raddatz, J. and Frohnappel, P. (2008). DLR results from the third AIAA computational fluid dynamics drag prediction workshop. *Journal of Aircraft*, 45(3): 823-836.
- Brodersen, P. (2002). Drag prediction of engine-airframe interference effects using unstructured Navier-Stokes calculations. *Journal of Aircraft*, 39(9): 927-935.
- Bugeda, G. and Onate, E. (1999). Optimum aerodynamic shape design for fluid flow problems including mesh adaptivity. *International Journal for Numerical Methods in Fluids*, 30(2): 161-178.
- Buhmann, M.D. (2003). *Radial basis functions: theory and implementations*. Cambridge: Cambridge University Press.
- Burgreen, G.W. and Baysal, O. (1994). Aerodynamic Shape Optimisation Using Preconditioned Conjugate-Gradient Methods. *AIAA Journal*, 32(11): 2145-2152.

Burgreen, G.W., Baysal, O. and Eleshaky, M.E. (1994). Improving the Efficiency of Aerodynamic Shape Optimisation. *AIAA Journal* 32(1): 69-76.

Campbell, R.L. (1992). *An approach to constrained aerodynamic design with application to aerofoils*. NASA.Technial Report

Castonguay, P. and Nadarajah, S. (2007). *Effect of Shape Parameterisation on Aerodynamic Shape Optimisation*. AIAA-2007-59.

Ceze, M., Hayashi, M. and Volpe, E. (2009). *A Study of the CST Parameterisation Characteristics*. AIAA 2009-3767.

Chandru, V. and Kochhar, B. (1987). Geometric modeling: algorithms and new trends. Analytic techniques for geometric intersection problem. *SIAM*: 305-318.

Cliff, S.E., Thomas, S.D., Baker, T. J., Jameson, A. and Hicks, R. M. (2002). *Aerodynamic shape optimisation using unstructured grid methods*. AIAA 2002-5550.

Clyde, G., Gene, H. and Perry, N. (1999). Simultaneous aerodynamic analysis and design optimisation (SAADO) for a 3-D rigid wing. *14th Computational Fluid Dynamics Conference*, AIAA 1999-3296.

Cosentino, G.B. and Holst, T.L. (1986). Numerical Optimisation Design of Advanced Transonic Wing Configurations. *Journal of Aircraft*, 23(3): 192-199.

de Boer, A., van der Schoot, M.S. and Bijl, H. (2007). Mesh deformation based on radial basis function interpolation. *Computers & Structures*, 85(11-14): 784-795.

de Mattos, B.S., Macedo, A.P. and da Silva Filho, D.H. (2003). Considerations about Winglet Design. *21st Applied Aerodynamics Conference*. Orlando.

Désidéri, J.a., Antipolis, I.S. and Lausanne, É.P.F.D. (2004). Multilevel Shape Parameterisation for Aerodynamic Optimisation—Application to Drag and Noise Reduction of Transonic/Supersonic Business Jet. *European Congress on Computational Methods in Applied Sciences and Engineering*.

Destarac, D. (2003). Far-Field/Near-Field Drag Balance and Application of Drag Extraction in CFD. *VKI Lecture Series. CFD-based Aircraft Drag Prediction and Reduction*, National Institute of Aerospace, Hampton (VA), November 3-7, 2003.

- Destarac, D. and Reneaux, J. (1993). Numerical Optimisation Applied to Transport Aircraft Aerodynamics. *Recherche Aerospaciale*, (2): 39-55.
- Driver, J. and Zingg, D.W. (2007). Numerical aerodynamic optimisation incorporating laminar-turbulent transition prediction. *AIAA Journal*, 45(8): 1810-1818.
- Duvigneau, R., Chaigne, B. and Désidéri, J.a., (2006). *Multi-Level Parameterisation for Shape Optimisation in Aerodynamics and Electromagnetics using a Particle Swarm Optimisation Algorithm*. INRIA Report.
- Dwight, R.P. (2009). Robust Mesh Deformation using the Linear Elasticity Equations. *Computational Fluid Dynamics 2006*. H. Deconinck and E. Dick, eds. Springer Berlin Heidelberg: 401-406.
- Dwight, R.P. (2006). *Efficiency improvements of RANS-based analysis and optimisation using implicit and adjoint methods on unstructured grids*. PhD thesis, School of Mathematics, University of Manchester.
- Elliott, J. and Peraire, J. (1996). *Practical 3D aerodynamic design and optimisation using unstructured meshes*. AIAA-1996-4170: 1479-1485.
- Epstein, B., Peigin, S. and Tsach, S. (2006). A new efficient technology of aerodynamic design based on CFD driven optimisation. *Aerospace Science and Technology*, 10(2): 100-110.
- Eyi, S. and Lee, K.D. (1997). Inverse aerofoil design using the Navier Stokes equations. *Engineering Optimisation*, 28(4): 245-262.
- Eyi, S., Lee, K.D. Rogers, S. E. Rogers and Kwak, D. (1996). *High-lift design optimisation using the Navier-Stokes equations*. Washington, D.C: Journal of Aircraft, V33 No.3.pp 499-504
- Falcão, L., Gomes, A. and Suleiman, A. (2010). Multidisciplinary Design Optimisation of a Morphing Wingtip. *2nd International Conference on Engineering Optimisation*. Lisbon, Portugal.
- Farhat, C., Degand, C., Koobus, B. and Lesoinne, M. (1998). Torsional springs for two-dimensional dynamic unstructured fluid meshes. *Computer Methods in Applied Mechanics and Engineering*, 163(1-4): 231-245.

- Farin, G. (1983). Algorithms for Rational Bezier Curves. *Computer-Aided Design*, 15(2): 73-77.
- Farouki, R.T. (2000). Legendre-Bernstein basis transformations. *Journal of Computational and Applied Mathematics*, 119(1-2): 145-160.
- Felker, F.F. (2002). *Aircraft with elliptical winglets*, US Patents.6484968
- Forrest, A.R. (1968). *Curves and surfaces for computer-aided design*. [S.l.], Cambridge University. ISBN 0901224022
- Forrester, A.I.J., Bressloff, N.W. and Keane, A. J. (2006). Optimisation using surrogate models and partially converged computational fluid dynamics simulations. *Proceedings Royal Society of London A*, 462(2071): 2177-2204
- Forrester, A.I.J., Søbester, A. and Keane, A.J. (2008). *Engineering Design via Surrogate Modelling: A Practical Guide*.
- Fudge, D.M. and Zingg, D.W. (2005). *A CAD-Free and a CAD-Based Geometry Control System for Aerodynamic Shape Optimisation*. AIAA 2005-451.
- Fuhrmann, H. (2005). *Design Optimisation of a Class of Low Reynolds, High Mach Number Aerofoils for use in the Martian Atmosphere*. AIAA 2005-4603.
- Fulker, J.L., Ashill, P.R. and Simmons, M.J. (1993). *Study of simulated active control of shock waves on an aerofoil TR 93025*. Technical report. DERA.
- Gardner, B.A. and Selig, M.S. (2003). *Aerofoil Design Using a Genetic Algorithm and an Inverse Method*. AIAA 2003-0043.
- Gauger, N.R., Walther, A. Moldenhuer, C. and Widhalm, M. (2007). Automatic differentiation of an entire design chain for aerodynamic shape optimisation. *New Results in Numerical and Experimental Fluid Mechanics*, Vol 96: 454-461.
- Gerhold, T., Friedrich, O. and Evans, J. D. (1997). *Calculation of complex three-dimensional configurations employing the DLR-T-code*. AIAA 1997-167.
- Gerontakos, P. and Lee, T. (2006). Effects of winglet dihedral on a tip vortex. *Journal of Aircraft*, 43(1): 117-124.

- Giammichele, N. and Trépanier, J.-Y. (2007). *Aerofoil Generation and Optimisation using Multiresolution B-spline Control with Geometrical Constraints*. AIAA 2007-1916.
- Giles, M.B. and Pierce, N.A. (1998). On the properties of solutions of the adjoint Euler equations. *Numerical Methods for Fluid Dynamics VI*. ICFD (1998) pp.1–16.
- Giles, M.B. and Pierce, N.A. (2000). An introduction to the adjoint approach to design. *Flow Turbulence and Combustion*, 65(3-4): 393-415.
- Grasso, F. (2012). Design and Optimisation of Tidal Turbine Aerofoil. *Journal of Aircraft*, 49(2): 636-643.
- Gratzer, L.B. (1994). *Blended winglet*, Patents US 5348253 .
- Green, L., Newman, P. and Haigler, K. (1993). *Sensitivity derivatives for advanced CFD algorithm and viscous modeling parameters via automatic differentiation*. AIAA 1993-3321.
- Guerrero, J.E., Maestro, D. and Bottare, A. (2012). Biomimetic spiroid winglets for lift and drag control. *Comptes Rendus Mécanique*, 340(1-2): 67-80.
- Hacioglu, A. and Ozkol, I. (2003). Transonic aerofoil design and optimisation by using vibrational genetic algorithm. *Aircraft Engineering and Aerospace Technology*, 75(4): 350-357.
- Haderlie, J. and Crossley, W.A. (2009). A Parametric Approach to Supercritical Aerofoil Design Optimisation. *9th AIAA Aviation Technology, Integration, and Operations Conference (ATIO)*, AIAA 2009-6950
- Hager, J.O., Eyi, S. and Lee, K.D. (1994). 2-Point Transonic Aerofoil Design using Optimisation for Improved Off-Design Performance. *Journal of Aircraft*, 31(5): 1143-1147.
- Hantrais-Gervois, J.-L., Grenon, R., Mann, A. and Buescher, A. (2009). Downward pointing winglet design and assessment within the M-DAW research project. *Aeronautical Journal*, 113(1142): 221-232.
- Harris, C.D., (1990). *NASA supercritical aerofoils : a matrix of family-related aerofoils*. Washington, D.C.: NASA.

He, B., Roehl, P., Irani, R., Thamboo, S. and Srivatsa, S. (1998). CAD and CAE integration with application to the forging shape optimisation of turbine disks. *39th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference and Exhibition*, AIAA 1998-2032

Heo, H., Kim, M. and Elber, G. (1999). The intersection of two ruled surfaces. *Computer-Aided Design* 31(1): 33-50.

Hicken, J.E. (2009). *Efficient algorithms for future aircraft design: Contributions to aerodynamic shape optimisation*. PhD thesis, University of Toronto

Hicks, R.M. and Henne, P.A. (1978). Wing Design by Numerical Optimisation. *Journal of Aircraft*, 15(7): 407-413.

Hicks, R.M., Murman, E.M. and Vanderplaats, G. N. (1974). *An assessment of aerofoil design by numerical optimisation*. NASA Technical Report 19740020369

Hicks, R.M., Vanderplaats, G.N. (1975). *Applications of numerical optimisation to the design of low-speed aerofoils*. NASA. Technical Report 750524

Hohmeyer, M. (1992). *Robust, Efficient surface intersection for solid modeling*. PhD thesis, Berkeley, University of California.

Hou, G., Maroju, V., Taylor, A., Korivj, and Newman, P. (1995). *Transonic turbulent aerofoil design optimisation with automatic differentiation in incremental iterative forms*. AIAA.1995-1692

Huang, L. and Zhu, X. (1997). *A Practical Algorithm for Surface/Surface Intersection*. BUAA. Report. <http://www.cs.berkeley.edu/~hling/research/paper/intersection.htm>

Ilic, C., Widhalm, M. and Brezillon, J. (2012). Efficient Polar Optimisation of Transport Aircraft in Transonic Rans Flow using Adjoint Gradient Based Approach. *European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2012)*.

Iollo, A., Ferlauto, M. and Zannetti, L. (2001). An aerodynamic optimisation method based on the inverse problem adjoint equations. *Journal of Computational Physics*, 173(1): 87-115.

Jakobsson, S. and Amoignon, O. (2007). Mesh deformation using radial basis functions for gradient-based aerodynamic shape optimisation. *Computers & Fluids*, 36(6): 1119-1136.

Jameson, A. (1995). Analysis and Design of Numerical Schemes for Gas-Dynamics .1. Artificial Diffusion, Upwind Biasing, Limiters and Their Effect on Accuracy and Multigrid Convergence. *International Journal of Computational Fluid Dynamics*, 4(3-4): 171-218.

Jameson, A. (2004). *Efficient Aerodynamic Shape Optimisation*. AIAA-2004-4369.

Jameson, A., (1988). *Aerodynamic design via control theory*. Journal of Scientific Computing Vol.3 No.3

Jameson, A. and Kim, S. (2003). Reduction of the adjoint gradient formula for aerodynamic shape optimisation problems. *AIAA Journal*, 41(11): 2114-2129.

Jameson, A., Pierce, N.A. Martinelli, L. (1997). *Optimum Aerodynamics Design Using the Navier-Stokes Equations*. AIAA Paper 97-0101.

Jameson, A., Schmidt, W. and Turkel, E. (1981). Numerical solution of the Euler equations by finite volume methods using Runge Kutta time stepping schemes. *14th Fluid and Plasma Dynamics Conference*, AIAA 1981-1259

Jameson, A. and Yoon, S. (1987). Lower-Upper Implicit Schemes with Multiple Grids for the Euler Equations. *AIAA Journal*, 25(7): 929-935.

Jeong, S., Chiba, K. and Obayashi, S. (2005). *Data Mining for Aerodynamic Design Space*. AIAA 2005-5079.

Jeong, S., Obayashi, S. and Yamamoto, K. (2005). Aerodynamic optimisation design with Kriging model. *Transactions of the Japan Society for Aeronautical and Space Sciences*, 48(161): 161-168.

Jouhaud, J.C., Sagaut, P., Montagnac, M. and Laurenceau, J. (2007). A surrogate-model based multidisciplinary shape optimisation method with application to a 2D subsonic aerofoil. *Computers & Fluids*, 36(3): 520-529.

Jupp, J. (2001). Wing aerodynamics and the science of compromise. *Aeronautical Journal*, 105(1053): 633-641.

Kanazaki, M., Tanaka, K., Jeong, S. and Yamamoto, K. (2006). Multi-Objective Aerodynamic Optimisation of Elements' Setting for High-lift Aerofoil Using Kriging Model. *44th AIAA Aerospace Sciences Meeting and Exhibition*, AIAA 2006-1471.

Keane, A.J. and Nair, P.B. (2005). *Computational approaches for aerospace design: the pursuit of excellence*. Chichester: John Wiley. ISBN 0470855401

Khurana, M.S., Winarto, H. and Sinha, A.K. (2008). *Aerofoil Geometry Parameterisation through Shape Optimizer and Computational Fluid Dynamics*. AIAA 2008-295.

Khurana, M.S., Winarto, H., and Sinha, A.K. (2008). Application of Swarm Approach and Artificial Neural Networks for Aerofoil Shape Optimisation. *AIAA 2008-5954*.

Khurana, M.S., Winarto, H. and Sinha, A.K. (2009). *Aerofoil Optimisation by Swarm Algorithm with Mutation and Artificial Neural Networks*. AIAA 2009-1278.

Kim, C.S., Kim, C. and Rho, O.H. (2001a). Sensitivity analysis for the Navier-Stokes equations with two-equation turbulence models. *AIAA Journal*, 39(5): 838-845.

Kim, C.S., Kim, C. and Rho, O.H. (2001b). *Aerodynamic sensitivity analysis for turbulent flows on chimera overlaid grids*. AIAA 2001-0267.

Kim, H.J., Kim, C. and Rho, O.H. (1999). *Aerodynamic sensitivity analysis for Navier-Stokes equations*. AIAA.1999-402

Kim, H.J. and Nakahashi, K. (2005). Unstructured Adjoint Method for Navier-Stokes Equations. *JSME International Journal*, 48(2) (Series B).

Kim, H.J., Sasaki, D., Obayashi, S. and Nakahashi, K. (2001). Aerodynamic optimisation of supersonic transport wing using unstructured adjoint method. *AIAA Journal*, 39(6): 1011-1020.

. Kim, S., Alonso, J.J. and Jameson, A. (1999). *A gradient accuracy study for the adjoint-based Navier-Stokes design method*. AIAA. 1999-299

Kim, S., Alonso, J.J. and Jameson, A. (2002). Design Optimisation of High-Lift Configurations Using a Viscous Continuous Adjoint Method. *AIAA 2002-0844*.

Korivi, V., Sherman, L. and Taylor, A. (1994). *First- and second-order aerodynamic sensitivity derivatives via automatic differentiation with incremental iterative methods*. AIAA-1994-4262.

Korivi, V., Taylor, A. and Newman, P. (1994). Aerodynamic optimisation studies using a 3-D supersonic Euler code with efficient calculation of sensitivity derivatives. *Proceedings of the 5th AIAA/NASA/USAF/ISSMO Symposium on Multidisciplinary Analysis and Optimisation*, AIAA.1994-4270

Krishnan, S. and Manocha, D. (1997). An efficient surface intersection algorithm based on lower-dimensional formulation. *ACM Transactions on Graphics* 16(1): 74-106.

Kroll, N. and Fassbender, J.K. (2005). *MEGAFLOW - Numerical Flow Simulation for Aircraft Design*, Springer Berlin Heidelberg. ISBN, 978-3-540-24383-0

Kroll, N., Heinrich, R., Krueger, W. and Nagel, B. (2008). Fluid-Structure Coupling for Aerodynamic Analysis and Design: A DLR Perspective (Invited). *46th AIAA Aerospace Sciences Meeting and Exhibition*, AIAA 2008-561

Kulfan, B. (2009). New Supersonic Wing Far-Field Composite-Element Wave-Drag Optimisation Method. *Journal of Aircraft*, 46(5): 1740-1758.

Kulfan, B.M. (2007a). Recent Extensions and Applications of the “CST” Universal Parametric Geometry Representation Method. *AIAA 2007-7709*.

Kulfan, B.M. (2007b). A Universal Parametric Geometry Representation Method – “CST”. *AIAA 2007-62*.

Kulfan, B.M. (2008). Universal parametric geometry representation method. *Journal of Aircraft*, Vol 45, No.1. 142-159.

Kulfan, B.M. (2010). Recent extensions and applications of the 'CST' universal parametric geometry representation method. *Aeronautical Journal*, 114(1153): 157-176.

Kulfan, B.M. and Bussoletti, J.E. (2006). "Fundamental" Parametric Geometry Representations for Aircraft Component Shapes. *AIAA 2006-6948*.

Ladson, C.L., Brooks, C.W., Hill, A.S. and Sproles, D. (1996). *Computer program to obtain ordinates for NACA aerofoils*. NASA Technical Memorandum 4741.

- Lambert, P.A. (1995). Optimisation de formes en aerodynamique. Application a la conception des nacelles de moteurs civils. Ph.D. thesis, Ecole Centrale Paris, France,.
- Lamousin, H.J. and Waggenspack, W.N. (1994). NURBS-Based Free-Form Deformation. *IEEE Computer Graphics and Applications*, 14(6): 59-65.
- Lane, J.M. and Riesenfeld, R.F. (1980). A theoretical development for the computer generation of piecewise polynomial surfaces. *IEEE Trans Pattern Analysis and Machine Intelligence*, 2(1): 35-46.
- Lane, K. and Marshall, D. (2010). Inverse Aerofoil Design Utilizing CST Parameterisation. *48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, AIAA 2010-1228
- Lasser, D. (1986). Intersection of parametric surfaces in the Bernstein-Bézier representation. *Computer-Aided Design*, 18(4): 186-192.
- Le Moigne, A. (2002). *A discrete Navier-Stokes adjoint method for aerodynamic optimisation of Blended Wing-Body configurations*. Ph.D. thesis, .
- Le Moigne, A. and Qin, N. (2004). Variable-fidelity aerodynamic optimisation for turbulent flows using a discrete adjoint formulation. *AIAA Journal*, 42(7): 1281-1292.
- Le Moigne, A. and Qin, N. (2006). Aerofoil profile and sweep optimisation for a blended wing-body aircraft using a discrete adjoint method. *The Aeronautical Journal*, Vol 110 No. 1111: 589-604.
- Leatham, M., Stokes, S. and Shaw, J.A. (2000). Automatic mesh generation for rapid-response Navier-Stokes calculations. AIAA.2000-2247
- Leoviriyakit, K., Kim, S. and Jameson, A. (2003). Viscous Aerodynamic shape optimisation of wings including planform variables. *AIAA 2003-3498*.
- Lepine, J., Guibault, F., Trepanier, J.Y. and Pepin, F. (2001). Optimized nonuniform rational B-spline geometrical representation for aerodynamic design of wings. *AIAA Journal*, 39(11): 2033-2041.
- Li, W. and Krist, S. (2005). Spline-Based Aerofoil Curvature Smoothing and Its Applications. *Journal of Aircraft*, 42(4).

- Li, X.Y., Jiang, H., Chen, S. and Wang, X.C. (2004). An efficient surface-surface intersection algorithm based on geometry characteristics. *Computers & Graphics-UK*, 28(4): 527-537.
- Limaiem, A. and Trochu, F. (1995). Geometric algorithms for the intersection of curves and surfaces. *Computers and Graphics*, 19(3): 391-403.
- Liu, X.Q., Qin, N. and Xia, H. (2006). Fast dynamic grid deformation based on Delaunay graph mapping. *Journal of Computational Physics*, 211(2): 405-423.
- Lutz, T., Kutzbach, M. and Wagner, S. (2004). *Investigations on shock control bumps for infinite swept wings*. AIAA-2004-2702.
- Madsen, J.I., Shyy, W. and Haftka, R.T. (2000). Response surface techniques for diffuser shape optimisation. *AIAA Journal*, 38(9): 1512-1518.
- Mann, A. (2007). *Wing tip device*. WO Patent 2,007,031,732.
- Mann, A. and Elsholz, I. (2005). The M-DAW Project - Investigations in Novel Wing Tip Device Design. *43rd AIAA Aerospace Sciences Meeting and Exhibition*, AIAA 2005-461
- Mavriplis, D.J. (2007). Discrete adjoint-based approach for optimisation problems on three-dimensional unstructured meshes. *AIAA Journal*, 45(4): 740-750.
- Meheut, M., Grenon, R., Carrier, G., Defos, M. and Duffau, M. (2009). Aerodynamic Design of Transonic Flying Wing Configurations. in *KATnet II: Conference on Key Aerodynamic Technologies*. Bremen, Germany.
- Minnella, G., Rodriguez, Y. and Ugas, J. (2010). *Aerodynamic Shape Design Optimisation of Winglets*. E. S. D. Project. EML 4905
- Mohammadi, B. and Pironneau, O. (2001). *Applied shape optimisation for fluids*. Oxford Univerisity Press.
- Mohammed, M., Franck, P. and Min-suk, K. (2012). NREL wind turbine aerodynamics validation and noise predictions using a Lattice Boltzmann Method. *18th AIAA/CEAS Aeroacoustics Conference (33rd AIAA Aeroacoustics Conference)*, AIAA 2012-2290

- Mousavi, A., Castonguay, P. and Nadarajah, S.K. (2007). Survey of Shape Parameterisation Techniques and Its Effect on Three-dimensional Aerodynamic Shape Optimisation. *AIAA Computational Fluid Dynamics Conference*, AIAA 2007-3837.
- Nadarajah, S.K. and Jameson, A. (2000). *A comparison of the continuous and discrete adjoint approach to automatic aerodynamic optimisation*. AIAA.2000-667
- Nadarajah, S.K., Jameson, A. (2001). *Studies of the continuous and discrete adjoint approaches to viscous automatic aerodynamic shape optimisation*. AIAA.2001-2530
- Nakayama, H., Kim, H-J. Matsushima, K., Nakahashi, K. and Takenaka, K. (2006). *Aerodynamic Optimisation of Multi-Element Aerofoil*. AIAA 2006-1051.
- Nelder, J.A. and Mead, R. (1965). A Simplex-Method for Function Minimization. *Computer Journal*, 7(4): 308-313.
- Nelson, A., Nemec, M. and Aftosmis, M.J. (2005). *Aerodynamic Optimisation of Rocket Control Surfaces Using Cartesian Methods and CAD Geometry*. AIAA, 2005-4836.
- Nemec, M., Aftosmis, M.J. and Pulliam, T. (2004). *CAD-Based Aerodynamic Design of Complex Configurations Using a Cartesian Method*. AIAA 2004-113.
- Nemec, M. and Zingg, D.W. (2002). Multi-Point and Multi-Objective Aerodynamic Shape Optimisation. AIAA 2002-5548
- Nguyen, N.T. (2008). Adjoint optimisation of one-dimensional hyperbolic equations with constrained periodic boundary conditions. *Computer Methods in Applied Mechanics and Engineering*, 197(51-52): 4683-4691.
- Nielsen, E.J. and Anderson, W.K. (2002). Recent improvements in aerodynamic design optimisation on unstructured meshes. *AIAA Journal*, 40(6): 1155-1163.
- Nielsen, E.J. and Kleb, W.L. (2006). Efficient construction of discrete adjoint operators on unstructured grids using complex variables. *AIAA Journal*, 44(4): 827-836.
- Nielsen, E.J. and Park, M. (2005). *Using An Adjoint Approach to Eliminate Mesh Sensitivities in Computational Design*. AIAA 2005-491.
- Nocedal, J. and Wright, S.J. (2006). *Numerical optimisation*. New York, London: Springer.

- Obayashi, S., Jeong, S. and Chiba, K. (2005). *Multi-objective design exploration for aerodynamic configurations*. AIAA paper 2005-4666
- Obayashi, S. and Oyama, A. (1996). Three-Dimensional Aerodynamic Optimisation with Genetic Algorithm. *Proceedings of the Third ECCOMAS CFD Conference*, Wiley, Chichester, England, U.K., 1996, pp. 420-424.
- Oloso, A. and Taylor, A. (1997). *Aerodynamic shape-sensitivity analysis and design optimisation of the IBM-SP2*. Washington, D.C, AIAA-1997-2273.
- Oyama, A., Obayashi, S. Nakahashi, K. and Nakamura, T. (2002). Aerodynamic optimisation of transonic wing design based on evolutionary algorithm. *Third International Conference on Nonlinear Problems in Aviation and Aerospace*, Vol 1 and 2, Proceedings: 537-546, 778.
- Padulo, M., Maginot, J. Guenov, M. and Holden, C. (2009). Aerofoil Design under Uncertainty with Robust Geometric Parameterisation. *AIAA 2009-2270*.
- Pagaldipti, N. and Chattopadhyay, A. (1996). A discrete semianalytical procedure for aerodynamic sensitivity analysis including grid sensitivity. *Computers & Mathematics with Applications*, 32(3): 61-71.
- Painchaud-Ouellet, S., Tribes, C. Trepanier, J. and Pelletierx, D. (2006). Aerofoil Shape Optimisation Using a Nonuniform Rational B-Splines Parameterisation Under Thickness Constraint. *AIAA Journal*, 44(10).pp.2170-2178
- Pandya, M.J. and Baysal, O. (1997). Gradient-based aerodynamic shape optimisation using alternating direction implicit method. *Journal of Aircraft*, 34(3): 346-352.
- Patrikalakis, N.M. (1993). Surface-to-Surface Intersections. *IEEE Computer Graphics and Applications*, 13(1): 89-95.
- Paul, S. and Ruxandra, B. (2012). Two-dimensional aerofoil shape optimisation for aerofoils at low speeds. *AIAA Modeling and Simulation Technologies Conference*, AIAA 2012-4790

Pehlivanoglu, M.Y.V. (2009). Representation Method Effects on Vibrational Genetic Algorithm in 2-D Aerofoil Design. *Journal of Aeronautics and Space Technologies*, 4(2): 7-13.

Peigin, S. and Epstein, B. (2004). Robust optimisation of 2D aerofoils driven by full Navier-Stokes computations. *Computers & Fluids*, 33(9): 1175-1200.

Penner, J.E. (1999). *Aviation and the global atmosphere : a special report of IPCC Working Groups I and III in collaboration with the Scientific Assessment Panel to the Montreal Protocol on Substances that Deplete the Ozone Layer*. Cambridge: Cambridge University Press.

Pfeiffer, N. (2004). Numerical Winglet Optimisation. *42nd AIAA Aerospace Sciences Meeting and Exhibition*, AIAA 2004-213

Piegl, L.A. and Tiller, W. (1997). The NURBS book. Berlin; London: Springer.

Pironneau, O. (1973). On optimum profiles in Stokes flow. *Journal of Fluid Mechanics*, 59(01): 117-128.

Pironneau, O. (1973). Optimum Profiles in Stokes Flow. *Journal of Fluid Mechanics*, 59(Jun5): 117-128.

Pironneau, O. (1974). Optimum Design in Fluid Mechanics. *Journal of Fluid Mechanics*, 64(Jun3): 97-110.

Powell, S. and Søbester, A. (2010). Application-Specific Class Functions for the Kulfan Transformation of Aerofoils. *13th AIAA/ISSMO Multidisciplinary Analysis Optimisation Conference*, 2010-9269

Qin, N., Wong, W.S. and Le Moigne, A. (2008). Three-dimensional contour bumps for transonic wing drag reduction. *Journal of Aerospace Engineering*, 222: 619-629.

Qin, N., Wong, W.S. and Sellars, N. Validation and optimisation of 3D bumps for transonic wing drag reduction, 2005, CEAS/KATnet Conference on Key Aerodynamic Technologies, 20-22 June 2005, Bremen, Germany.

Qin, N., Vavalle, A. and Le Moigne, A. Spanwise lift distribution for blended wing body aircraft, Journal of Aircraft, Vol.42, No.2, 2005, pp356-365.

- Qin, N., Vavalle, A., Le Moigne, A., Laban, M., Hackett, K., and Weinerfelt, P. (2004) Aerodynamic considerations of blended wing body aircraft, *Progress in Aerospace Sciences*, Volume 40, Issue 6, Pages 321-343,
- Qin, Z., Yu, F., Shi, Z.W. and Wang, Y. (2006). Adaptive inertia weight particle swarm optimisation. *Artificial Intelligence and Soft Computing - Icaisc 2006, Proceedings* 4029: 450-459.
- Rajendran, S. (2012). *Design of Parametric Winglets and Wing tip devices: A Conceptual Design Approach.*, Thesis, Linköping University,
- Rendall, T.C.S. and Allen, C.B. (2009). Efficient mesh motion using radial basis functions with data reduction algorithms. *Journal of Computational Physics*, 228(17): 6231-6249.
- Reneaux, J. (1984). Numerical optimisation method for aerofoil design. *La Recherche Aerospatiale (English Edition)*, 5: 1-19.
- Reneaux, J. (2004). Overview on Drag Reduction Technologies for Civil Transport Aircraft. *European Congress on Computational Methods in Applied Sciences and Engineering, ECCOMAS*. Jyväskylä
- Reuther, J., Jameson, A. (1995). *Supersonic wing and wing-body shape optimisation using an adjoint formulation*. NASA Research Center, Report RIACS tr-95-14
- Reuther, J., Jameson, A., Farmer, J., Martinelli, L. and Saunders, D. (1996). *Aerodynamic shape optimisation of complex aircraft configurations via an adjoint formulation*. AIAA.1996-94
- Reuther, J.J., Jameson, A. Alonso, J.J. Rimlinger, M.J. and Saunders, D.(1999). Constrained multipoint aerodynamic shape optimisation using an adjoint formulation and parallel computers, part 1and 2. *Journal of Aircraft*, 36(1): 50-74.
- Robinson, G.M. and Keane, A.J. (2001). Concise Orthogonal Representation of Supercritical Aerofoils. *Journal of Aircraft*, 38(3): 580-583.
- Roe, P.L. (1986) Characteristic-Based Schemes for the Euler Equations. Annual Review of Fluid Mechanics, 18, 337-365.

- Ronzheimer, A. (2005). Shape Parametrization Using Freeform Deformation. *MEGAFLOW - Numerical Flow Simulation for Aircraft Design*, 89: 211-222.
- Ronzheimer, A. (2006). Prospects of Geometry Parameterisation based on Freeform Deformation in MDO. *ERCOFTAC 2006: Design Optimisation: Methods & Applications*, pp. 1-10. *ERCOFTAC 2006 International Conference Las Palmas de Gran Canaria, Spain*. ( ISBN 84-85650-12-3.).
- Ronzheimer, A., Natterer, F.J. and Brezillon, J. (2010). *Aircraft Wing Optimisation Using High Fidelity Closely Coupled CFD and CSM Methods*. AIAA 2010-9078.
- Rossignac, J.R. and Requicha, A.A.G. (1987). Piecewise-circular curves for geometric modeling. *IBM J. Res. Dev*, 31(3): 296-313.
- Rowan, T. (1990). *Functional stability analysis of numerical algorithms*. PhD thesis, University of Texas..
- Rudnik, R., Heinrich, R. Eisfeld, B. and Schwarz, T. (2004). DLR Contributions to Code Validation Activities within the European High Lift Project EUROLIFT. *New Results in Numerical and Experimental Fluid Mechanics IV*. C. Breitsamter, B., Laschka, H.-J. Heinemann and R. Hilbig, Springer Berlin Heidelberg. 87: 42-49.
- Rusak, Z. (1993). Transonic Flow around the Leading-Edge of a Thin Aerofoil with a Parabolic Nose. *Journal of Fluid Mechanics*, 248: 1-26.
- Rusak, Z. (1994). Subsonic flow around the leading edge of a thin aerofoil with a parabolic nose. *Euro. Jnl of Applied Mathematics*, 5: 283-311.
- Saad, Y. (2003). *Iterative methods for sparse linear systems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2003.
- Saad, Y. and Schultz, M. (1986). GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *Journal on scientific and statistical computing*, 7(3): 856-869.
- Samareh, J. (2004). *Aerodynamic Shape Optimisation Based on Free-form Deformation*. AIAA 2004-4630.

- Samareh, J.A. (2001). Survey of shape parameterisation techniques for high-fidelity multidisciplinary shape optimisation. *AIAA Journal*, 39(5): 877-884.
- Samareh, J.A. (2005). Geometry and Grid/Mesh Generation Issues for CFD and CSM Shape Optimisation. *Optimisation and Engineering*, 6: 21-32.
- Sarraga, R.F. (1983). Algebraic methods for intersection. *Comput. Vis. Graph. Image Process*, 22: 222-238.
- Sasaki, D. and Obayashi, S. (2003). *Low-Boom Design Optimisation for SST Canard-Wing-Fuselage Configuration*. AIAA-2003-3432.
- Sasaki, D., Obayashi, S. and Nakahashi, K. (2002). Navier-Stokes optimisation of supersonic wings with four objectives using evolutionary algorithm. *Journal of Aircraft*, 39(4): 621-629.
- Sclafani, A.J., Vassberg, J.C., Harrison, N.A., Rumsey, C.L., Rivers, S.M. and Morrison, J.H. (2008). CFL3D/OVERFLOW results for DLR-F6 wing/body and drag prediction workshop wing. *Journal of Aircraft*, 45(3): 762-780.
- Sederberg, T.W. and Parry, S.R. (1986). Free-Form Deformation of Solid Geometric Models. *Computer Graphics*, 20(4): 151-160.
- Shahrokhi, A. and Jahangirian, a. (2007). Aerofoil shape parameterisation for optimum Navier-Stokes design with genetic algorithm. *Aerospace Science and Technology*, 11(6): 443-450.
- Sharov, D., Luo, H. Baum, J. and Lohner, R. (2000). *Implementation of unstructured grid GMRES+ LU-SGS method on shared-memory, cache-based parallel computers*. AIAA 2000-927
- Silisteau, P. and Botez, R. (2012). Two-dimensional aerofoil shape optimisation for aerofoils at low speeds. *AIAA Modeling and Simulation Technologies Conference*, AIAA 2012-4790
- Simpson, T. W., Mauery, T. M., Korte, J. J., and Mistree, F., “Kriging Models for Global Approximation in Simulation-Based Multidisciplinary Design Optimization,” *AIAA Journal*, Vol. 39, No. 12, 2001, pp. 2233–2241. doi:10.2514/2.1234

- Smith, S., Nemec, M. and Krist, S. (2013). Integrated Nacelle-Wing Shape Optimisation for an Ultra-High Bypass Fanjet Installation on a Single-Aisle Transport Configuration. *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, AIAA 2013-0543
- Sobieczky, H. (1998). Parametric Aerofoils and Wings. *Notes on Numerical Fluid Mechanics* edited by K. Fujii and G. S. Dulikravich Vol. 68, Vieweg Verlag, pp. 71-88.
- Sobieczky, H. and Seebass, A.R. (1984). Supercritical Aerofoil and Wing Design. *Annual Review of Fluid Mechanics*, 16: 337-363.
- Sobiesczanski-Sobieski, J. 1986: The case for aerodynamic sensitivity analysis. Sensitivity analysis in engineering. *NASA CP-2457*, pp. 77–96
- Song, W. and Keane, A.J. (2004). A Study of Shape Parameterisation Methods for Aerofoil Optimisation. *10th AIAA/ISSMO Multidisciplinary Analysis and Optimisation Conference*, AIAA 2004-4482.
- Soto, O., Lohner, R. and Yang, C. (2004). An adjoint-based design methodology for CFD problems. *International Journal of Numerical Methods for Heat & Fluid Flow*, 14(5-6): 734-759.
- Spalart, P. and Allmaras, S. (1992). A one-equation turbulence model for aerodynamic flows. *30th Aerospace Sciences Meeting and Exhibition*, AIAA 1992-439
- Sripawadkul, V. and Padulo, M. (2010). A Comparison of Aerofoil Shape Parameterisation Techniques for Early Design Optimisation. *AIAA-2010-9050*.
- Straathof, M.H. and van Tooren, M.J.L. (2011). Extension to the Class-Shape-Transformation Method Based on B-Splines. *AIAA Journal*, 49(4): 780-790.
- Sturdza, P. (2007). Extensive supersonic natural laminar flow on the Aerion business jet. *AIAA 2007- 685*.
- Sung, C.-H., Kwon, J-H. (2001). *Aerodynamic design optimisation using the Navier-Stokes and adjoint equations*. AIAA.2001-266

Svenningsen, K.H., Madsen, J.I., Hassing, N. H. and Pauker, W. H. (1996). Optimisation of flow geometries applying quasianalytical sensitivity analysis. *Applied Mathematical Modelling*, 20(3): 214-224.

Szollos, A., Smid, M. and Hajek, J. (2009). Aerodynamic optimisation via multi-objective micro-genetic algorithm with range adaptation, knowledge-based reinitialization, crowding and epsilon-dominance. *Advances in Engineering Software*, 40(6): 419-430.

Tai, T.C. Huson, G.G and Gregorek, G.M. (1988). *Transonic characteristics of a humped aerofoil*. Journal of Aircraft, Vol 25, No. 8 pp. 673-674

Takenaka, K., Hatanaka, K., Yamazaki, W. and Nakahashi, K. (2008). Multidisciplinary Design Exploration for a Winglet. *Journal of Aircraft*, 45(5): 1601-1611.

Tang, Z. and Désidéri, J.A. (2002). *Towards Self-Adaptive Parameterisation of Bézier Curves for Aerofoil Aerodynamic Design*. INRIA Report No. 4572.

Tangler, J.L. and Somers, D.M. (1995). *NREL Aerofoil Families for HAWTs*. NREL Technical Report.

Taylor, A.C., Hou, G.W., and Korivi, V.M. (1992). Methodology for Calculating Aerodynamic Sensitivity Derivatives. *AIAA Journal*, 30(10): 2411-2419.

Thomas, J.P., Hall, K.C. and Dowell, E.H. (2005). Discrete adjoint approach for modeling unsteady aerodynamic design sensitivities. *AIAA Journal*, 43(9): 1931-1936.

Tiller, W. (1983). Rational B-Splines for Curve and Surface Representation. *IEEE Computer Graphics and Applications*, 3(6): 61-69

Timoleon, K., Gunter, F. and Savill, A.M. (2012). Multi-objective Shape Optimisation for Horizontal-Axis Wind Turbine Blades. *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference 20th AIAA/ASME/AHS Adaptive Structures Conference 14th AIAA 2012*-1353

Toro, E.F. (2009). *Riemann solvers and numerical methods for fluid dynamics: a practical introduction*. Dordrecht, London: Springer.

Townsend, J.C., Samareh, J.A. Weston, R.P. and Zorumski, W.E. (1998). *Integration of a CAD System Into an MDO Framework*. NASA TM-207672.

Truong, A.H., Oldfield, C.A. and Zingg, D.W. (2008). Mesh movement for a discrete-adjoint Newton-Krylov algorithm for aerodynamic optimisation. *AIAA Journal*, 46(7): 1695-1704.

van Leer, B. 1979. Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method. *Journal of Computational Physics*, 32, 101-136.

Vavalle, A. and Qin, N. (2007). Iterative response surface based optimisation scheme for transonic aerofoil design. *Journal of Aircraft*, 44(2): 365-376.

Venter, G. and Sobieszczański-Sobieski, J. (2004). Multidisciplinary optimisation of a transport aircraft wing using particle swarm optimisation. *Structural and Multidisciplinary Optimisation*, 26(1-2): 121-131.

Versprille, K.J. (1975). *Computer-aided design applications of the rational b-spline approximation form*. Ph.D. thesis. Syracuse University.

Viken, J.K. (1986). *Boundary-layer stability and aerofoil design*. NASA Technical Report.

Viken, J.K., Viken, S.A. Pfenninger, W. and Morgan, H.L. (1987). *Design of the low-speed NLF(1)-0414F and the high-speed HSNLF(1)-0213 aerofoils with high-lift systems*. NASA Technical Report.

Vitturi, M.D. and Beux, F. (2006). A discrete gradient-based approach for aerodynamic shape optimisation in turbulent viscous flow. *Finite Elements in Analysis and Design*, 43(1): 68-80.

Wang, X., Shen, W.Z., Zhu, W.J., Sorensen, J.N. and Jin, C. (2009). Shape optimisation of wind turbine blades. *Wind Energy*, 12(8): 781-803.

Watt, A. and Watt, M. (1992). *Advanced Animation and Rendering Techniques*. New York: Addison-Wesley. ISBN, 0201544121,

Weierman, J. and Jacob, J. (2010). Winglet Design and Optimisation for UAVs. *28th AIAA Applied Aerodynamics Conference*, AIAA 2010-4224.

- White, F.M. (2008). *Fluid mechanics*. New York; London: McGraw-Hill.
- Widhalm, M., Brezillon, J. Ilic, C. and Leicht, T. (2010). Investigation on Adjoint Based Gradient Computations for Realistic 3d Aero-Optimisation. *13th AIAA/ISSMO Multidisciplinary Analysis Optimisation Conference*, AIAA 2010-9129.
- Widhalm, M. and Ronzheimer, A. (2008). Comparison between Gradient-free and Adjoint Based Aerodynamic Optimisation of a Flying Wing Transport Aircraft in the Preliminary Design. *AIAA 2007-4060*.
- Wilcox, D.C. (1998). *Turbulence Modeling for CFD* - 2nd ed. DCW Industries,
- Winnemoller, T. and van Dam, C.P. (2007). Design and numerical optimisation of thick aerofoils including blunt trailing edges. *Journal of Aircraft*, 44(1): 232-240.
- Wong, W.S. (2006). *Mechanisms and Optimisations of 3D Shock Control Bumps*. PhD thesis, Department of Mechanical Engineering, The University of Sheffield. .
- Wong, W.S., Le Moigne, A. and Qin, N. (2007). Parallel adjoint-based optimisation of a blended wing body aircraft with shock control bumps. *Aeronautical Journal*, 111(1117): 165-174.
- Wu, H.-Y., Yang, S. and Liu, F. (2003). *Comparison of Three Geometric Representations of Aerofoils for Aerodynamic Optimisation*. AIAA 2003-4095.
- Wu, S.T. and Andrade, L.N. (1999). Marching along a regular surface surface intersection with circular steps. *Computer Aided Geometric Design*, 16(4): 249-268.
- Xia, H. and Qin, N. (2005). Detached-eddy simulation for synthetic jets with moving boundaries. *Modern Physics Letters B*, 19(28n29): 1429-1434.
- Yamazaki, W., Mouton, S. and Carrier, G. (2008). *Efficient Design Optimisation by Physics-Based Direct Manipulation Free-Form Deformation*. AIAA 2008-5953.
- Yamazaki, W., Mouton, S. and Carrier, G. (2010). Geometry Parameterisation and Computational Mesh Deformation by Physics-Based Direct Manipulation Approaches. *AIAA Journal*, 48(8): 1817-1832.
- Yoon, S. and Jameson, A. (1988). Lower-Upper Symmetric-Gauss-Seidel Method for the Euler and Navier-Stokes Equations. *AIAA Journal*, 26(9): 1025-1026.

- Zhu, F. and Qin, N. (2013). Intuitive Class/Shape Function Parameterisation for Aerofoils. *AIAA Journal*, 52(1): 17-25.
- Zhu, Y.P. (2000). *Computational Study of Shock Control at Transonic Speed*. PhD thesis, Cranfield University.
- Zingg, D.W. and Billing, L. (2007). Toward Practical Aerodynamic Design Through Numerical Optimisation. *AIAA 2007-3950. t. A. C. F. D. Conference*.
- Zingg, D.W. and Elias, S. (2006). *On Aerodynamic Optimisation Under a Range of Operating Conditions*. AIAA 2006-1053.
- Zuo, Y., Su, W. and Gao, Z. (2006). *An Aerodynamic Optimisation Method Based on Dynamic Boundary Control*. AIAA 2006-7107.

## Publication List

Zhu, F. and Qin, N. et al, *Comparison of Three Geometric Parameterisation methods and Their Effect on Aerodynamic Optimisation*, Eurogen 2011, 14th-16th Sep 2011, Capua, Italy

Zhu, F. and Qin, N. *Investigation of CST Geometric Parameterisation Methods for an Entire Transport Aircraft*, Applied Aerodynamics Group Conference, Bristol, 16-19 July, 2012

Zhu, F. and Qin, N. (2013). *Intuitive Class/Shape Function Parameterisation for Aerofoils*. AIAA Journal, 52(1): 17-25.

# Appendices

## Appendix A: Derivatives of Bezier curve

The Bezier curve is described as:

$$R(u) = \sum_{i=0}^n P_i B_{i,n}(u) \quad A.1$$

$$0 \leq u \leq 1$$

with Bernstein polynomials

$$B_{i,n}(u) = K_{i,n} \cdot u^i (1-u)^{n-i} \quad A.2$$

Then, the derivative of the Bezier curve function is:

$$\frac{d}{du} R(u) = \frac{d}{du} \sum_{i=0}^n P_i B_{i,n}(u) = \sum_{i=0}^n P_i \frac{d}{du} B_{i,n}(u) \quad A.3$$

The derivative of the Bernstein polynomials is:

$$\begin{aligned} \frac{d}{du} B_{i,n}(u) &= \frac{d}{du} K_{i,n} \cdot u^i (1-u)^{n-i} \\ &= K_{i,n} \cdot i \cdot u^{i-1} (1-u)^{n-i} - K_{i,n} \cdot (n-i) \cdot u^i (1-u)^{n-i-1} \\ &= \frac{n(n-1)!}{(i-1)! (n-i)!} \cdot u^{i-1} (1-u)^{n-i} - \frac{n(n-1)!}{(i-1)! (n-i-1)!} \cdot u^i (1-u)^{n-i-1} \\ &= n B_{i-1,n-1}(u) - n B_{i,n-1}(u) = n(B_{i-1,n-1}(u) - B_{i,n-1}(u)) \end{aligned} \quad A.4$$

where

$$B_{-1,n-1}(u) \equiv B_{n,n-1}(u) \equiv 0 \quad A.5$$

Therefore, the derivative of the Bezier Curve function is:

$$\begin{aligned} \frac{d}{du} R(u) &= \frac{d}{du} \sum_{i=0}^n P_i B_{i,n}(u) = \sum_{i=0}^n P_i \frac{d}{du} B_{i,n}(u) \\ &= \sum_{i=0}^n n(B_{i-1,n-1}(u) - B_{i,n-1}(u)) \cdot P_i \\ &= \sum_{i=0}^{n-1} n B_{i,n-1}(u) \cdot (P_{i+1} - P_i) \end{aligned} \quad A.3$$

The derivatives of the Bezier curve at beginning  $u = 0$  and end points  $u = 1$  are:

$$C(0) = n \cdot (P_1 - P_0) \quad A.2$$

$$C(1) = n \cdot (P_n - P_{n-1})$$

Hence, the first and last polygon control lines of the Bezier curve are parallel to the tangent direction at the beginning and end points.

## Appendix B: Value of rational shape function of RCST method at the trailing edge

The RCST method is written as:

$$\xi(\psi) = C_{N_2}^{N_1}(\psi) \cdot SR(\psi) + \psi \cdot \Delta\xi_{te} \quad B.1$$

where

$$C_{N_2}^{N_1}(\psi) = \psi^{N_1} \cdot (1 - \psi)^{N_2} \quad B.2$$

$$SR(\psi) = \frac{\sum_{i=0}^n A_i \cdot W_i \cdot K_{i,n} \cdot \psi^i \cdot (1 - \psi)^{n-i}}{\sum_{i=0}^n W_i \cdot K_{i,n} \cdot \psi^i \cdot (1 - \psi)^{n-i}} = \frac{\sum_{i=0}^n A_i \cdot W_i \cdot B_{i,n}(\psi)}{\sum_{i=0}^n W_i \cdot B_{i,n}(\psi)} \quad B.3$$

The first derivative of RCST with  $N_1 = 0.5$  and  $N_2 = 1.0$  is then written as:

$$\xi'(\psi) = \sqrt{\psi}(1 - \psi) \cdot SR'(\psi) + [-\sqrt{\psi} + \frac{1}{2}\psi^{-\frac{1}{2}}(1 - \psi)]SR(\psi) + \Delta\xi_{te} \quad B.4$$

At the trailing edge,  $\psi = 1$ , the derivative of RCST is given by:

$$\xi'(1) = SR(1) + \Delta\xi_{te} \quad B.5$$

where

$$SR(1) = \frac{\sum_{i=0}^n A_i \cdot W_i \cdot B_{i,n}(1)}{\sum_{i=0}^n W_i \cdot B_{i,n}(1)} = \frac{A_n \cdot W_n}{W_n} = A_n \quad B.6$$

Therefore,

$$A_n = -\xi'(1) + \Delta\xi_{te} \quad B.7$$

Therefore, the last coefficient is still equal to the sum of the tangential value and trailing edge thickness position, which is same as with standard CST methods. The weight of the rational equation will not affect the physical meaning of the last coefficient.

## Value of rational shape function of RCST method at the leading edge

The second derivative of RCST with  $N_1 = 0.5$  and  $N_2 = 1.0$  is written as:

$$\begin{aligned}\xi''(\psi) &= \sqrt{\psi}(1-\psi) \cdot SR''(\psi) + \left[ -\sqrt{\psi} + \frac{1}{2}\psi^{-\frac{1}{2}}(1-\psi) \right] SR'(\psi) \\ &\quad + \left[ -\psi^{-\frac{1}{2}} - \frac{1}{4}\psi^{-\frac{3}{2}}(1-\psi) \right] SR''(\psi) \\ &\quad + \left[ -\sqrt{\psi} + \frac{1}{2}\psi^{-\frac{1}{2}}(1-\psi) \right] SR'(\psi)\end{aligned}\quad \text{B.8}$$

The general equation for radius of curvature of the surface is:

$$\rho(\psi) = -\frac{\{1 + [\xi'(\psi)]^2\}^{3/2}}{\xi''(\psi)} \quad \text{B.9}$$

Equations B.4 and B.8 are substituted into Equation B.0:

$$-\rho(\psi) = \frac{\{1 + [\sqrt{\psi}(1-\psi) \cdot SR'(\psi) + [-\sqrt{\psi} + \frac{1}{2}\psi^{-\frac{1}{2}}(1-\psi)]SR''(\psi) + \Delta\xi_{te}]^2\}^{3/2}}{\sqrt{\psi}(1-\psi) \cdot SR''(\psi) + \left[ -\sqrt{\psi} + \frac{1}{2}\psi^{-\frac{1}{2}}(1-\psi) \right] SR'(\psi) + \left[ -\psi^{-\frac{1}{2}} - \frac{1}{4}\psi^{-\frac{3}{2}}(1-\psi) \right] SR''(\psi) + \left[ -\sqrt{\psi} + \frac{1}{2}\psi^{-\frac{1}{2}}(1-\psi) \right] SR'(\psi)} \quad \text{B.10}$$

At the leading edge,  $\psi = 0$ , and Equation B.10 could be simplified by omitting the term with  $\sqrt{\psi}$ .

$$\begin{aligned}-\rho(\psi)|_{\psi=0} &= \frac{\left\{1 + \left[\left[\frac{1}{2}\psi^{-\frac{1}{2}}(1-\psi)\right]SR''(\psi) + \Delta\xi_{te}\right]^2\right\}^{3/2}}{\left[\frac{1}{2}\psi^{-\frac{1}{2}}(1-\psi)\right]SR'(\psi) + \left[-\psi^{-\frac{1}{2}} - \frac{1}{4}\psi^{-\frac{3}{2}}(1-\psi)\right]SR''(\psi) + \left[\frac{1}{2}\psi^{-\frac{1}{2}}(1-\psi)\right]SR'(\psi)} \\ &= \frac{\left\{1 + \left[\left[\frac{1}{2}\psi^{-\frac{1}{2}}(1-\psi)\right]SR''(\psi) + \Delta\xi_{te}\right]^2\right\}^{3/2}}{\left[\psi^{-\frac{1}{2}}(1-\psi)\right]SR'(\psi) + \left[-\psi^{-\frac{1}{2}} - \frac{1}{4}\psi^{-\frac{3}{2}}(1-\psi)\right]SR''(\psi)} \\ &= \frac{\left\{1 + \left(\frac{1}{2}\psi^{-\frac{1}{2}}SR''(\psi)\right)^2 \left[(1-\psi) + \frac{2\psi^{0.5}\Delta\xi_{te}}{SR''(\psi)}\right]^2\right\}^{3/2}}{-\frac{1}{4}\psi^{-\frac{3}{2}}SR''(\psi) \left\{(1-\psi) - \frac{4\psi^{\frac{1}{2}}(1-\psi)SR'(\psi)}{SR''(\psi)} + 4\psi^{\frac{1}{2}}SR''(\psi)\right\}} \\ &= \frac{\left(\frac{1}{4}\psi^{-1}SR^2(\psi)\right)^{3/2} \left\{\frac{4\psi}{SR^2(\psi)} + \left[(1-\psi) + \frac{2\psi^{0.5}\Delta\xi_{te}}{SR''(\psi)}\right]^2\right\}^{3/2}}{-\frac{1}{4}\psi^{-\frac{3}{2}}SR''(\psi) \left\{(1-\psi) - \frac{4\psi^{\frac{1}{2}}(1-\psi)SR'(\psi)}{SR''(\psi)} + 4\psi^{\frac{1}{2}}SR''(\psi)\right\}}\end{aligned}\quad \text{B.11}$$

Then  $\psi = 0$  is substituted to complete Equation B.11. The radius of curvature at the leading edge is then equal to:

$$-\rho(0) = -\frac{SR^2(0)}{2} = -\frac{1}{2} \left( \frac{A_0 W_0}{W_0} \right)^2 = -\frac{A_0^2}{2} \quad \text{B.12}$$

Therefore, the first coefficient still implies the leading edge radius

$$A_0 = \sqrt{2R_{Le}} \quad \text{B.13}$$

This shows that the rational equation will not affect the physical meaning of the first coefficient.

## **Appendix C Value at boundary of CST function with class parameters N<sub>1</sub> =1.0 and N<sub>2</sub>=1.0**

The CST method with class parameters N<sub>1</sub> =1.0 and N<sub>2</sub>=1.0 is written as:

$$\xi(\psi) = \psi \cdot (1 - \psi) \cdot S(\psi) \quad \text{C.1}$$

Therefore, it is obvious that the values at  $\psi = 0$  and  $\psi = 1$  are zero.

$$\xi(0) = C_{1.0}^{1.0} \cdot S(0) = 0 \quad \xi(0) = C_{1.0}^{1.0} \cdot S(1) = 0 \quad \text{C.2}$$

The first derivative is then written as:

$$\xi'(\psi) = \psi(1 - \psi) \cdot SR'(\psi) + [-\psi + (1 - \psi)]SR(\psi) \quad \text{C.3}$$

Therefore, the first derivatives at beginning and ending points  $\psi = 0$  and  $\psi = 1$  are:

$$\xi'(0) = SR(0) = A_0 \quad \text{C.4}$$

$$\xi'(1) = SR(1) = A_1 \quad \text{C.5}$$

## **Appendix D Partial differentiation of geometry of fuselage and belly-fairing**

The CST method for belly-fairing is written as:

$$R(\bar{\theta}, \psi) = C_{1.0}^{1.0} \bullet \sum_i^{N_x} \sum_j^{N_y} B_{i,j} \bullet Sy_j(\psi) \bullet Sx_i(\bar{\theta}) + f(\bar{\theta}) \bullet R_{keel}(\psi) + g(\bar{\theta}) \bullet R_{width}(\psi) \quad \text{D.1}$$

$$x = \psi \bullet T_{length} \quad D.2$$

$$y = R(\bar{\theta}, \psi) \bullet \cos(\theta) \quad D.3$$

$$z = R(\bar{\theta}, \psi) \bullet \sin(\theta) \quad D.4$$

The partial differential of belly-fairing with respect to  $\bar{\theta}$  are:

$$\frac{\partial x}{\partial \bar{\theta}} = 0 \quad D.2$$

$$\frac{\partial y}{\partial \bar{\theta}} = \frac{\partial R(\bar{\theta}, \psi)}{\partial \bar{\theta}} \cos(\theta) + 0.5\pi \cdot R(\bar{\theta}, \psi) \sin(\theta) \quad D.3$$

$$\frac{\partial z}{\partial \bar{\theta}} = \frac{\partial R(\bar{\theta}, \psi)}{\partial \bar{\theta}} \sin(\theta) - 0.5\pi \cdot R(\bar{\theta}, \psi) \cos(\theta) \quad D.4$$

The partial differential of belly-fairing with respect to  $\psi$  are:

$$\frac{\partial x}{\partial \psi} = T_{length} \quad D.5$$

$$\frac{\partial y}{\partial \psi} = \frac{\partial R(\bar{\theta}, \psi)}{\partial \psi} \cos(\theta) \quad D.6$$

$$\frac{\partial z}{\partial \psi} = \frac{\partial R(\bar{\theta}, \psi)}{\partial \psi} \sin(\theta) \quad D.7$$

where

$$\begin{aligned} \frac{\partial R(\bar{\theta}, \psi)}{\partial \bar{\theta}} &= (1 - 2\bar{\theta}) \sum_i^N \sum_j^N B_{i,j} S y_j(\psi) S x_i(\bar{\theta}) \\ &+ \theta(1 - \bar{\theta}) \sum_i^N \sum_j^N B_{i,j} S y_j(\psi) S x'_i(\bar{\theta}) + f'(\bar{\theta}) R_{keel}(\psi) \\ &+ g'(\bar{\theta}) R_{width}(\psi) \end{aligned} \quad D.8$$

$$\begin{aligned} \frac{\partial R(\bar{\theta}, \psi)}{\partial \psi} &= \theta(1 - \bar{\theta}) \sum_i^N \sum_j^N B_{i,j} S y'_j(\psi) S x_i(\bar{\theta}) + f(\bar{\theta}) R'_{keel}(\psi) \\ &+ g(\bar{\theta}) R'_{width}(\psi) \end{aligned} \quad D.9$$

and

$$S x'_i(\bar{\theta}) = K_{i,j}(i\bar{\theta}^{i-1}(1 - \bar{\theta})^{N_x - i} - \bar{\theta}^i(1 - \bar{\theta})^{N_x - i - 1}) \quad D.10$$

$$S_{Y_i}(\psi) = K_{i,j}(j\psi^{j-1}(1-\psi)^{Ny-j} - \psi^j(1-\psi)^{Ny-j-1}) \quad D.11$$

$$f'(\bar{\theta}) = -6\bar{\theta}^2 + 6\bar{\theta} \quad D.12$$

$$g'(\bar{\theta}) = 6\bar{\theta}^2 - 6\bar{\theta} \quad D.13$$

The CST method for belly-fairing is written as:

$$X(\psi) = \psi \times T_{length} \quad 0 \leq \eta \leq 1 \quad D.14$$

$$Y(\eta) = \eta \times W_{Fuselage} \quad 0 \leq \eta \leq 1 \quad D.15$$

$$Z_U(\eta) = 2 \times H_{Fuselage} \times \eta^{0.5} \bullet (1-\eta)^{0.5} \quad 0 \leq \eta \leq 1 \quad D.16$$

The partial differentials of belly-fairing with respect to  $\psi$  are:

$$\frac{\partial X}{\partial \psi} = T_{length} \quad D.17$$

$$\frac{\partial Y}{\partial \psi} = 0 \quad D.18$$

$$\frac{\partial Z}{\partial \psi} = 0 \quad D.19$$

The partial differentials of belly-fairing with respect to  $\eta$  are:

$$\frac{\partial X}{\partial \eta} = 0 \quad D.20$$

$$\frac{\partial Y}{\partial \eta} = W_{Fuselage} \quad D.21$$

$$\frac{\partial Z}{\partial \eta} = H_{Fuselage}(\eta^{-0.5}(1-\eta)^{0.5} - \eta^{0.5}(1-\eta)^{-0.5}) \quad D.22$$

## Appendix E Objective function using target lift iteration

In order to remove the  $C_l$  constraint, the target lift iteration is employed in flow solver to output results which automatically satisfy the target required lift. However, the link between  $C_l$  and the angle of attack  $\alpha$  have to be exploited to modify the objective function.

The flow solver must perform an internal fix-point iteration on the angle of attack  $\alpha$  to simultaneously solve:

$$R(W, X, \tilde{\alpha}) = 0 \quad E.1$$

$$C_l(W, X, \tilde{\alpha}) - C_{l\text{target}} = 0 \quad E.2$$

where  $C_{l\text{target}}$  is the target required lift coefficient and  $\tilde{\alpha}$  represents the iterated angle of attack. The flow simulation starts at  $\tilde{\alpha} = \alpha_0$ , and after flow simulation is converged, the  $\tilde{\alpha}$  has to reach  $\alpha^*$  which indicates  $C_{l\text{target}}$ .

The gradient  $\frac{dI_i}{dD}$  computed by adjoint approaches in Equation 6.23 is then corrected with respect to E.1 and E.2. The modified cost function  $\tilde{I}_i$ , ( $I_i$  could be  $C_d$ ,  $C_m$ ) depends on  $D$  and  $\tilde{\alpha}$ , and  $\tilde{\alpha}$  depend on  $D$ .

$$I_i(D, \tilde{\alpha}) \quad \tilde{\alpha}(D) \quad E.3$$

It should satisfy:

$$\frac{\partial \tilde{I}_i}{\partial D} = \frac{\partial I_i}{\partial D} \quad \frac{\partial \tilde{I}_i}{\partial \tilde{\alpha}} = \frac{\partial I_i}{\partial \alpha} \quad E.4$$

and it is then derived by:

$$\frac{d\tilde{C}_l}{dD} = \frac{\partial \tilde{C}_l}{\partial \tilde{\alpha}} \frac{d\tilde{\alpha}}{dD} + \frac{\partial C_l}{\partial D} = \frac{dC_l}{d\tilde{\alpha}} \frac{d\tilde{\alpha}}{dD} + \frac{dC_l}{dD} = \frac{dC_{l\text{target}}}{dD} = 0 \quad E.5$$

$$\frac{d\tilde{I}_i}{dD} = \frac{\partial \tilde{I}_i}{\partial \tilde{\alpha}} \frac{d\tilde{\alpha}}{dD} + \frac{\partial I_i}{\partial D} = \frac{dI_i}{d\tilde{\alpha}} \frac{d\tilde{\alpha}}{dD} + \frac{dI_i}{dD} \quad E.6$$

$\frac{d\tilde{\alpha}}{dD}$  could be derived from Equation E.5 and substituted into E.6. Therefore, the final equation of the modified gradient is:

$$\frac{d\tilde{I}_i}{dD} = \frac{dI_i}{dD} - \frac{\left(\frac{\partial I_i}{\partial \alpha}\right)}{\left(\frac{\partial C_l}{\partial \alpha}\right)} \frac{dC_l}{dD} \quad E.7$$

where  $\frac{\partial I_i}{\partial \alpha}$  and  $\frac{\partial C_l}{\partial \alpha}$  are the sensitivities with respect to angle of attack, which could be calculated after solving the adjoint equation and obtaining  $\lambda_{flow}^i$  as well.

$$I_i = I_i(W, X, \alpha) + \lambda_{flow}^i R(W, X, \alpha) \quad E.8$$

$$\begin{aligned} \frac{dI_i}{d\alpha} &= \frac{\partial I_i}{\partial W} \frac{dW}{d\alpha} + \frac{\partial I_i}{\partial X} \frac{dX}{d\alpha} + \lambda_{flow}^i \frac{\partial R}{\partial W} \frac{dW}{d\alpha} + \lambda_{flow}^i \frac{\partial R}{\partial X} \frac{dX}{d\alpha} + \lambda_{flow}^i \frac{\partial R}{\partial \alpha} \\ &= \left( \frac{\partial I_i}{\partial W} + \lambda_{flow}^i \frac{\partial R}{\partial W} \right) \frac{dW}{d\alpha} + \frac{\partial I_i}{\partial X} + \lambda_{flow}^i \frac{\partial R}{\partial X} \\ &= \frac{\partial I_i}{\partial \alpha} + \lambda_{flow}^i \frac{\partial R}{\partial \alpha} \end{aligned} \quad E.9$$

The terms  $\frac{\partial I_i}{\partial \alpha}$  and  $\frac{\partial R}{\partial \alpha}$  are calculated analytically in TAU.  $\lambda_{flow}^i$  is solved by the flow adjoint equation. The sensitivities of cost function with respect to angle of attack are then computed.