

**University of Strathclyde**

**Department of Naval Architecture and Marine Engineering**

**APPLICATION OF MORPHING TECHNIQUE  
IN RAPID GENERATION OF SHIP HULLS AND  
OTHER FORMS**

by

**Ju Young Kang**

**A thesis presented in fulfilment of the requirements  
for the degree of Doctor of Philosophy**

**January**

**2011**

This thesis is the result of the author's original research. It has been composed by the author and has not been previously submitted for examination which has led to the award of a degree.

The copyright of this thesis belongs to the author under the terms of the United Kingdom Copyright Acts as qualified by University of Strathclyde Regulation 3.50. Due acknowledgement must always be made of the use of any material contained in, or derived from, this thesis.

*To my parents, Man Sun Kang & Kyung Jo Kim*

*To my sister, Mi Young Kang*

*To my fiancé, Myung Sook Jang*

## **ABSTRACT**

With the fast improving numerical analysis techniques and enormous computing power readily at hand, various performance evaluation tools have been introduced for product designs to such an extent that in many cases automatic optimisation (or optimisation ‘on the fly’) is a serious possibility. For any optimisation to be successful, a vast number of design alternatives should be made available. For automatic optimisation the designs will have to be available or generated ‘on demand’. For complicated shapes, such as ship hull forms, therefore, the form generation has become a new bottleneck of computer-based optimisation.

This thesis describes the research work done to address this problem and proposes morphing technique as an effective rapid form generation tool. Morphing is a geometric interpolation technique and is capable of producing an infinite number of ‘intermediate’ forms between two ‘extreme’ forms.

Starting with relatively simple 2-dimensional (2D) curve forms, the technique is extended to 3-dimensional (3D) surface modelling. Two methods of establishing the correspondence between the vertices of the two parent forms are developed and their practical implementation is illustrated through worked examples.

In an effort to provide the designers with a higher degree of flexibility and control than a simple morphing technique can provide, multi-target morphing technique has been put forward. In this technique, more than two forms are used as parents, and this technique has been illustrated using 2D airfoil designs in this thesis. A few issues associated with practical implementation have been identified and some of them have been addressed, albeit superficially in some cases.

It was found that the morphing technique proposed in this thesis does go some way to resolving the problem of rapid hull form generation requirements in design optimisation.

## **ACKNOWLEDGEMENTS**

I would like to express my immense gratitude to many people who helped me to complete this thesis and hence to achieve one of my most important lifetime goals.

I am deeply indebted to my parents, Man Sun Kang and Kyung Jo Kim, my sister Mi Young Kang for their endless love and support. Phone call to them on every Saturday was one of the most delightful activities in my Glasgow life. Even though they are a thousand miles away from me, I receive comfort and encouragement from them just by being there. My father working for Korean Register (KR) is also an excellent advisor for me to study naval architecture and ocean engineering.

I would like to express my deep gratitude and appreciation to my supervisor, Dr. B. S. Lee. It was he who introduced the concept of 'Morphing' to me for the first time. His assistance throughout my years of study here has been invaluable. I really appreciate the trouble that he has taken for my research and even my poor English. His many comments and suggestions have undoubtedly improved things.

I would also like to thank Prof. Dr.-Ing. Soo Young Kim (Pusan national university) who offered me the first opportunity to study in the field of computer aided ship hull design. He also recommended me to study abroad in the UK, and he introduced Dr. Lee to me for the first time. Whenever I meet him, he is willing to give me sincere advices.

I owe many thanks to Dr. Joonyun Kang (Samsung heavy industries), Mr. Seoung Chan Kang (Simulation Tech Inc.) and Mr. Bong Kung Chun (Daewoo shipbuilding & Marine Engineering) for their tips on C++ programming and their advices on naval architecture and ocean engineering.

I would also like to thank Ms. Thelma Will and Ms. Fiona Cameron for their administrative supports and genuine kindness.

I express sincere gratitude to Mr. Nabile Hifi and Dr. Hyowon Kim (Renault F1) for checking my English. Especially Dr. Kim, who is an expert on aerodynamics, gave much advice on airfoil designs.

Finally, I am deeply grateful to my beautiful fiancé, Myung Sook Jang for her endless love, patience, understanding, encouragement, trust, and support. Although I could concentrate on my study and complete this thesis thanks to her so far, I know I gave her a very hard time because of being far away for a long time. Now I would like to make her the happiest lady in the world. Myung Sook, would you marry me?

**CONTENTS**

ABSTRACT ..... I

ACKNOWLEDGEMENTS ..... II

CONTENTS ..... IV

LIST OF FIGURES ..... VII

LIST OF TABLES ..... XII

NOMENCLATURE ..... XIV

**CHAPTER 1. INTRODUCTION ..... 1**

    1.1 BACKGROUND AND CONTENTS OF THE THESIS ..... 1

    1.2 AIMS OF PROJECT ..... 6

    1.3 APPROACH ADOPTED ..... 7

**CHAPTER 2. LITERATURE REVIEW ..... 8**

    2.1 GENERAL REMARKS ..... 8

    2.2 HULL FORM DESIGN ..... 8

    2.3 MORPHING ..... 15

**CHAPTER 3. MORPHING METHODOLOGY ..... 17**

    3.1 GENERAL REMARKS ..... 17

    3.2 WHAT IS MORPHING ..... 17

    3.3 CURVE MORPHING ..... 19

    3.4 SURFACE MORPHING ..... 21

    3.5 GENERATING EXTREME TARGET MODELS ..... 37

    3.6 CONCLUDING REMARKS ..... 40

**CHAPTER 4. APPLICATION OF MORPHING TO AIRFOIL DESIGN ..... 41**

    4.1 GENERAL REMARKS ..... 41

    4.2 A BRIEF REVIEW OF AIRFOILS ..... 41

    4.3 AIRFOILS FOR TURBINES ..... 44

    4.4 MORPHING AIRFOILS ..... 45

    4.5 MULTI-TARGET MORPHING ..... 48

4.6 GENERATING EXTREME AIRFOILS FOR USE AS PARENTS .....	51
4.7 CONCLUDING REMARKS .....	58
<b>CHAPTER 5. APPLICATION OF MORPHING TECHNIQUE IN AIRFOIL</b>	
<b>DESIGN OPTIMISATION.....</b>	<b>59</b>
5.1 GENERAL REMARKS .....	59
5.2 BASIC PRINCIPLES OF AIRFOILS .....	60
5.3 AIRFOILS FOR WIND TURBINE.....	64
5.4 AIRFOILS FOR CURRENT TURBINE.....	73
5.5 CONCLUDING REMARKS .....	79
<b>CHAPTER 6. APPLICATION OF MORPHING TO HULL FORM</b>	
<b>GENERATION .....</b>	<b>80</b>
6.1 GENERAL REMARKS .....	80
6.2 HULL FORM MORPHING USING REGULARISING METHOD.....	81
6.3 HULL FORM MORPHING USING MERGING METHOD .....	91
6.4 GENERATING EXTREME HULL FORMS.....	104
6.5 PARAMETRIC HULL FORM DESIGN USING MORPHING .....	107
6.6 A FURTHER EXAMPLE OF MORPHING – BETWEEN A MONO HULL AND A CATAMARAN HULL .....	111
6.7 CONCLUDING REMARKS .....	117
<b>CHAPTER 7. DISCUSSION.....</b>	
7.1 NOVELTY OF WORK .....	118
7.2 PREDICTABILITY OF MORPHED FORMS.....	119
7.3 EASE OF USE.....	120
7.4 SPEED OF GENERATION .....	121
7.5 SIMPLICITY OF DEFINING MODELS.....	121
7.6 VERSATILITY OF MORPHING TECHNIQUES .....	122
7.7 CREATIVITY OF DESIGN.....	122
7.8 SUGGESTIONS FURTHER WORK.....	123
<b>CHAPTER 8. CONCLUSIONS.....</b>	
	<b>126</b>



**REFERENCES** ..... **128**

**APPENDICES** ..... **138**

    APPENDIX A. COORDINATES OF THE OPTIMISED AIRFOILS ..... 138

    APPENDIX B. CAVITATION OF CURRENT TURBINE AIRFOIL ..... 160

## LIST OF FIGURES

Figure 2.1 A hull form representation using cubic spline [15].	11
Figure 2.2 A hull form representation using Bézier curve [15].	11
Figure 2.3 A hull form representation using B-spline [15].	11
Figure 2.4 Example of Lackenby method.	13
Figure 2.5 Design spiral of ship design.	15
Figure 2.6 A group of shapes extrapolated from the 1979 and 1983 Honda Civics [46].	16
Figure 3.1 Butterfly metamorphosis (from [52]).	18
Figure 3.2 An example of generating different intermediate models according to the correspondence. Each point $p_n$ of the source models corresponds to $p_n'$ of the target models.	21
Figure 3.3 An example of a local parameterisation.	24
Figure 3.4 An example of calculating convex combination coefficients $\lambda_{i,k}$ ( $c_i = 3$ ). .....	25
Figure 3.5 An example of calculating convex combination coefficients $\lambda_{i,k}$ ( $c_i > 3$ ). .....	26
Figure 3.6 Straight lines through $p_{jk}$ and $p_i$ intersects a line segment ( $c_i = 5$ ).	28
Figure 3.7 Example of parameterising nodes on a boundary.	30
Figure 3.8 Procedure of regularising: (a) Original parameterised mesh; (b) Triangular borders; (c) Regularised mesh after first bisecting ( $4^1$ triangles); (d) Regularised mesh after second bisecting ( $4^2$ triangles); (e) Original parameterised mesh superimposed on the regularised mesh; (f) Representing regularised mesh with barycentric coordinates of original parameterised mesh.	34
Figure 3.9 Procedure of mesh merging: (a) source mesh; (b) target mesh; (c) overlaid meshes; (d) merged mesh.	36
Figure 3.10 Generating a new boundary $AB'C'$ from the boundary $ABC$ by moving vertex $C(1, 0)$ to vertex $C'(1, 2)$ .	39

Figure 4.1 Airfoil shapes.....	43
Figure 4.2 A source and a target models: (a) NACA 6409 for source; (b) Eppler 1098 for target.....	45
Figure 4.3 Original models and newly generated models: (a) source model ( $t = 0$ ); (b) $t = 0.1$ ; (c) $t = 0.2$ ; (d) $t = 0.3$ ; (e) $t = 0.4$ ; (f) $t = 0.5$ ; (g) $t = 0.6$ ; (h) $t = 0.7$ ; (i) $t = 0.8$ ; (j) $t = 0.9$ ; (k) target model ( $t = 1$ ). .....	47
Figure 4.4 Concept of multi-target morphing. ....	48
Figure 4.5 A source and two target models used for multi-target morphing: (a) source model: NACA 6409; (b) target model 1: Eppler 1098; (c) target model 2: FX 082512.....	49
Figure 4.6 Models generated from multi-target morphing: (a) $t_0 : 0.34, t_I : 0.33, t_2 : 0.33$ ; (b) $t_0 : 0.10, t_I : 0.05, t_2 : 0.85$ ; (c) $t_0 : 0.50,$ $t_I : 0.15, .....$	51
Figure 4.7 Original airfoil model for generating a new airfoil. ....	53
Figure 4.8 Generating a new upper surface by replacing a guide point ( $P4$ ).....	55
Figure 4.9 Generating new extreme airfoils using Laplacian coordinates.....	57
Figure 5.1 Fluid flow and circular disk of turbine. ....	61
Figure 5.2 Velocity (a) and forces (b) for an airfoil.....	62
Figure 5.3 NREL airfoils (S826, S825, S814 and S815). ....	64
Figure 5.4 Lift drag ratio of airfoils benchmarked S826 ( $Rn = 1.5 \times 10^6$ ). ....	67
Figure 5.5 Lift drag ratio of airfoils benchmarked S825 ( $Rn = 2.0 \times 10^6$ ). ....	68
Figure 5.6 Lift drag ratio of airfoils benchmarked S814 ( $Rn = 1.5 \times 10^6$ ). ....	69
Figure 5.7 Lift drag ratio of airfoils benchmarked S815 ( $Rn = 1.2 \times 10^6$ ). ....	70
Figure 5.8 Maximum lift-drag ratios against morphing parameter.....	71
Figure 5.9 Maximum lift-drag ratios of airfoils as radius of blade.....	72
Figure 5.10 Original models and morphed airfoils. ....	74
Figure 5.11 Lift-drag ratio of the parent models.....	75
Figure 5.12 Cavitation inception bucket of the parent models. ....	75
Figure 5.13 Optimised airfoils. ....	77
Figure 5.14 Lift-drag ratio of the original and optimised models.....	78
Figure 5.15 Cavitation inception bucket of the original and optimised models. ....	78

Figure 6.1 Two parent hull forms. ....	81
Figure 6.2 Extracting boundaries and internal nodes for bulbous bow: (a) source model; (b) target model. ....	83
Figure 6.3 Subdivision of the source and target models (34 patches each). ....	83
Figure 6.4 Procedure of remeshing for bulbous bow: (a) Original 3D mesh; (b) Mapping 3D points onto 2D plane; (c) Regularisation on 2D plane; (d) Remapping from 2D to 3D. ....	84
Figure 6.5 Examples of 3D patches with regularised meshes. ....	87
Figure 6.6 The regularised parent hull forms. ....	87
Figure 6.7 Hull form generated through morphing ( $t = 0, 0.25, 0.5, 0.75, 1$ ). ....	89
Figure 6.8 Morphing for the bulbous bow (a and b) and the forecastle (c). ....	90
Figure 6.9 Overview of algorithm used for the hull form morphing: (a) Original 3D meshes; (b) Projecting 3D meshes on 2D plane; (c) Merging two 2D meshes; (d) Remapping 2D mesh to each original 3D surface. ....	92
Figure 6.10 An example of selecting (a) feature points on 3D hull form meshes and (b) a 2D parameter polygon. ....	93
Figure 6.11 Parameterised meshes onto a common 2D polygon: (a) source model; (b) target model. ....	94
Figure 6.12 Merging boundaries. ....	95
Figure 6.13 An example of additional intersection vertices and consequent new triangular faces. ....	96
Figure 6.14 Parameterised meshes of source model (a) and target model (c), and their merged mesh (b). ....	97
Figure 6.15 Original source and target meshes and remapped meshes: (a) original source; (b) original target; (c) re-meshed source; (d) re-meshed target. ....	99
Figure 6.16 Hull form morphing using the merged method. ....	102
Figure 6.17 Selecting upper transom for the boundary. ....	103
Figure 6.18 Strange transom resulting from improper boundary. ....	103
Figure 6.19 A bulbous bow model (a) and its boundary (b). ....	104
Figure 6.20 Generating new boundaries of the bulbous bow. ....	105

Figure 6.21 Newly generated bulbous bow: (a) Original bulbous bow: (b) New bulbous bow from the red boundary in Figure 6.20; (c) New bulbous bow from the blue boundary in Figure 6.20. ....	106
Figure 6.22 Volume of a polygon column. ....	108
Figure 6.23 Comparison between (a) the normal and (b) constrained hull form morphings. ....	110
Figure 6.24(a) A mono hull and (b) a catamaran hull as the source and target models of the morphing. ....	111
Figure 6.25 Establishing boundaries for the parameterisation. ....	112
Figure 6.26 Parameterised mono hull (a) and catamaran hull (b). ....	113
Figure 6.27 Regularised meshes ( $2 \times 4^6$ faces for each). ....	113
Figure 6.28 Morphing between the mono hull and the catamaran hull. ....	116
Figure 7.1 A model of the turtle ship. ....	123
Figure 7.2 Example of imposing different morphing speed. ....	125
Figure A.1 Profile of S826. ....	139
Figure A.2 Lift and drag coefficients of S826 ( $R_n = 1.5 \times 10^6$ ). ....	139
Figure A.3 Profile of N6409-0.140. ....	140
Figure A.4 Lift and drag coefficients of N6409-0.140 ( $R_n = 1.5 \times 10^6$ ). ....	140
Figure A.5 Profile of E1098-0.140. ....	141
Figure A.6 Lift and drag coefficients of E1098-0.140 ( $R_n = 1.5 \times 10^6$ ). ....	141
Figure A.7 Profile of G652-0.140. ....	142
Figure A.8 Lift and drag coefficients of G652-0.140 ( $R_n = 1.5 \times 10^6$ ). ....	142
Figure A.9 Profile of M0.45-0.140. ....	143
Figure A.10 Lift and drag coefficients of M0.45-0.140 ( $R_n = 1.5 \times 10^6$ ). ....	143
Figure A.11 Profile of M0.45_0.4_0.05-0.140. ....	144
Figure A.12 Lift and drag coefficients of M0.45_0.4_0.05-0.140 ( $R_n = 1.5 \times 10^6$ ). .....	144
Figure A.13 Profile of S825. ....	145
Figure A.14 Lift and drag coefficients of S825 ( $R_n = 2.0 \times 10^6$ ). ....	145
Figure A.15 Profile of N6409-0.170. ....	146
Figure A.16 Lift and drag coefficients of N6409-0.170 ( $R_n = 2.0 \times 10^6$ ). ....	146
Figure A.17 Profile of E1098-0.170. ....	147

Figure A.18 Lift and drag coefficients of E1098-0.170 ( $R_n = 2.0 \times 10^6$ ).....	147
Figure A.19 Profile of G652-0.170. ....	148
Figure A.20 Lift and drag coefficients of G652-0.170 ( $R_n = 2.0 \times 10^6$ ). ....	148
Figure A.21 Profile of M0.70-0.170. ....	149
Figure A.22 Lift and drag coefficients of M0.70-0.170 ( $R_n = 2.0 \times 10^6$ ). ....	149
Figure A.23 Profile of M0.0_0.65_0.35-0.170. ....	150
Figure A.24 Lift and drag coefficients of M0.0_0.65_0.35-0.170 ( $R_n = 2.0 \times 10^6$ ). .....	150
Figure A.25 Profile of S814. ....	151
Figure A.26 Lift and drag coefficients of S814 ( $R_n = 1.5 \times 10^6$ ). ....	151
Figure A.27 Profile of N6409-0.240. ....	152
Figure A.28 Lift and drag coefficients of N6409-0.240 ( $R_n = 1.5 \times 10^6$ ). ....	152
Figure A.29 Profile of E1098- 0.240 (M1.00, M0.0_1.0_0.0).....	153
Figure A.30 Lift and drag coefficients of E1098- 0.240 (M1.00, M0.0_1.0_0.0) ( $R_n = 1.5 \times 10^6$ ).....	153
Figure A.31 Profile of G652-0.240. ....	154
Figure A.32 Lift and drag coefficients of G652-0.240 ( $R_n = 1.5 \times 10^6$ ). ....	154
Figure A.33 Profile of S815. ....	155
Figure A.34 Lift and drag coefficients of S815 ( $R_n = 1.2 \times 10^6$ ). ....	155
Figure A.35 Profile of N6409-0.260 (M1.00).....	156
Figure A.36 Lift and drag coefficients of N6409-0.260 (M1.00) ( $R_n = 1.2 \times 10^6$ ). ....	156
Figure A.37 Profile of E1098-0.260. ....	157
Figure A.38 Lift and drag coefficients of E1098-0.260 (M1.00) ( $R_n = 1.2 \times 10^6$ ). ....	157
Figure A.39 Profile of G652-0.260. ....	158
Figure A.40 Lift and drag coefficients of G652-0.260 ( $R_n = 1.2 \times 10^6$ ). ....	158
Figure A.41 Profile of M0.0_0.95_0.05-0.260. ....	159
Figure A.42 Lift and drag coefficients of M0.0_0.95_0.05-0.260 ( $R_n = 1.2 \times 10^6$ ). .....	159
Figure B.1 Generation of Cavitation.....	160
Figure B.2 Flow and pressure around an airfoil.....	161
Figure B.3 Cavitation bucket diagram. ....	163

## LIST OF TABLES

Table 5.1 Design specification.....	64
Table 5.2 Comparison of the maximum lift/drag of various airfoils at 95% radius. .	67
Table 5.3 Comparison of the maximum lift/drag of various airfoils at 75% radius. .	68
Table 5.4 Comparison of the maximum lift/drag of various airfoils at 40% radius. .	69
Table 5.5 Comparison of the maximum lift/drag of various airfoils at 30% radius. .	70
Table 5.6 Maximum lift-drag ratio and cavitation number of original sections. ....	76
Table 5.7 Constraints for the optimisation of the current turbine airfoil. ....	76
Table 5.8 Performances of the optimised airfoils. ....	78
Table 6.1 Calculated dimensions of the hull forms generated through basic morphing. .....	108
Table 6.2 Calculated dimensions during the advanced hull form morphing in which the target displacement is 75,000 m <sup>3</sup> .....	109
Table A.1 Coordinates of S826.....	139
Table A.2 Coordinates of N6409-0.140.....	140
Table A.3 Coordinates of E1098-0.140. ....	141
Table A.4 Coordinates of G652-0.140.....	142
Table A.5 Coordinates of M0.45-0.140. ....	143
Table A.6 Coordinates of M0.45_0.4_0.05-0.140. ....	144
Table A.7 Coordinates of S825.....	145
Table A.8 Coordinates of N6409-0.170.....	146
Table A.9 Coordinates of E1098-0.170. ....	147
Table A.10 Coordinates of G652-0.170.....	148
Table A.11 Coordinates of M0.70-0.170. ....	149
Table A.12 Coordinates of M0.0_0.65_0.35-0.170. ....	150
Table A.13 Coordinates of S814.....	151
Table A.14 Coordinates of N6409-0.240.....	152
Table A.15 Coordinates of E1098- 0.240 (M1.00, M0.0_1.0_0.0).....	153
Table A.16 Coordinates of G652-0.240.....	154
Table A.17 Coordinates of S815.....	155

Table A.18 Coordinates of N6409-0.260 (M1.00).....	156
Table A.19 Coordinates of E1098-0.260 (M1.00). ....	157
Table A.20 Coordinates of G652-0.260.....	158
Table A.21 Coordinates of M0.0_0.95_0.05-0.260. ....	159



## NOMENCLATURE

### For morphing process

$\mathbf{A}$	$n \times n$ matrix, and $n$ is the number of internal nodes
$ang(a,b,c)$	Angle of the edges $ab$ and $bc$
$area(a,b,c)$	Area of the triangle $abc$
$b_1$	Multiplication of matrix $\mathbf{A}$ and column vector $\mathbf{u}$
$b_2$	Multiplication of matrix $\mathbf{A}$ and column vector $\mathbf{v}$
$c_i$	Number of neighbours of an internal node
$d_i$	Distances between the two nodes of the outermost mesh elements
$d_i'$	Distances between two nodes of parameterised outermost mesh elements
$E$	Edge set
$L_i$	Length of a side of parameterised outermost mesh elements
$L$	Laplacian operator
$L'$	Modified Laplacian operator
$L_x$	x-axis Laplacian operator
$L_y$	y-axis Laplacian operator
$L(s_i)$	Set of Laplacian differential coordinates
$M(t)$	Morphed model
$\mathbf{M}$	Vertex set of merged mesh
$N$	Total number of nodes
$n$	Number of internal nodes
$\mathbf{P}$	Vertex set of parameterised mesh
$P_i$	i-th local parameterisation
$p_i$	Internal vertex of i-th local parameterisation
$p_{jk}$	k-th neighbour vertex of local parameterisation

$p_l$	Starting point of straight line during the procedure to calculate convex combination coefficient
$p_{r(l)}$	The first endpoint of a line segment intersected with the straight line through $p_i$ and $p_l$ during the procedure to calculate convex combination coefficient
$p_{r(l)+1}$	The second endpoint of a line segment intersected with the straight line through $p_i$ and $p_l$ during the procedure to calculate convex combination coefficient
$R$	Blade radius
$R_0$	Source model for morphing
$R_n$	n-th target model morphing
$r_0$	Vertex of regularised mesh on parameterisation
$S$	Vertex set of the 3D surface
$S_i$	i-th Local surface
$s_i$	Internal vertex of i-th local surface
$s_{jk}$	k-th neighbour vertex of local surface
$t$	Morphing parameter
$t_0$	Weight for source model
$t_n$	Weight for n-th target model
$U_i$	Parameterised nodes
$u_i$	x-axis component of $U_i$
$\mathbf{u}$	column vectors of $(u_1, \dots, u_n)$
$v_i$	y-axis component of $U_i$
$\mathbf{v}$	column vectors of $(v_1, \dots, v_n)$
$w_{ij}$	Weighting factor in Laplacian coordinates

---

$\delta_1, \delta_2, \delta_3$	Weighting factors applied to $p_l$ , $p_{r(l)}$ and $p_{r(l)+1}$ respectively
$\varepsilon + 1$	Number of nodes on a side of parameterised polygon
$\theta_i$	Total sum of the angles between two consecutive edges of i-th local surface
$\lambda$	Convex combination coefficient
$\mu$	Weighting factor for neighbour points during the procedure to calculate convex combination coefficient

### **For airfoil and hull form design**

$A$	Area of rotor sweep (m <sup>2</sup> )
$E_k$	Kinetic energy (Joules)
$C_b$	Block coefficient
$C_d$	Coefficient of drag
$C_l$	Coefficient of lift
$C_p$	Power coefficient
$c$	Chord length (m)
$D$	Drag force (N)
$L$	Lift force (N)
$m$	Mass (kg)
$N$	Blade number
$P_w$	Power in the wind (watt)
$Q$	Torque (N·m)
$r$	Position of the airfoil concerned on the blade
$R$	Radius of rotor (m)
$Re$	Reynolds number
$t$	Time (second)
	Maximum thickness of airfoil (m)
$T$	Thrust (N)

$v$	Fluid flow speed (m/s)
$v_r$	Relative fluid speed (m/s)
$W$	Electricity produced by turbine (watt)
$\alpha$	Angle of attack
$\rho$	Fluid density (kg/m <sup>3</sup> )
$\pi$	Ratio of circle's circumference to its diameter, 3.142
$\phi$	Incidence angle of the relative fluid speed in respect to the rotor plane (°)
$\omega$	Angular velocity (rad/s)

### **Acronyms**

2D	2-dimensional
3D	3-dimensional
AF	After perpendicular
CFD	Computation fluid dynamics
DWT	Dead weight tonnage
FP	Forward perpendicular
GZ	Righting arm (m)
LCB	Longitudinal position of centre of buoyancy
Lpp	Length between perpendiculars
NURBS	Non-Uniform Rational B-Spline
SMF	Garland's Simple Mesh Format
TEU	Twenty-foot equivalent unit
VLCC	Very large crude carrier

# Chapter 1.

## INTRODUCTION

---

### 1.1 BACKGROUND AND CONTENTS OF THE THESIS

Over the past several decades, there has been a steady pressure on ship design process to become faster and more efficient. This is primarily due to a commercial environment which is becoming increasingly competitive as the years go by. Shipowners want ships which are the best suited to their trade and have the potential to return the maximum profit for the circumstances that they envisage their ships will operate in. This means that the design process is no longer a simple search for a feasible solution but has become a process of optimisation often involving a vast number of feasible solutions. Many ship types which are optimised for specific types of trade have been developed and in this process many old traditional concepts of what a ship looks like have been often broken in the never-ending search for a better ship. This trend, of course, is not confined to ships, but the same can be said of many industrial or consumer products.

The nature of optimisation is such that inevitably a large number of designs within the feasible solution space need to be created and assessed according to a set of performance parameters. Full-scale testing of ships is almost out of the question at the design stage, unlike a motor vehicle, for example. The best that can be expected in ship design in this regard is scale model tests. Indeed, ground-breaking design work often relies upon model experiments. However, model tests are far too

expensive and time-consuming to be included in a day-to-day design process at all stages. Relying on scale model tests of ships for design assessment for optimisation purpose will undoubtedly limit the number of alternatives that can be evaluated. Scale model tests are also not so simple or easy to perform, and consequently the experiments with a number of different designs are limited. It is, therefore, hardly surprising that the use of computer simulation for evaluating performance of a design has seen marked increase in recent years. Rapid increase in speed and capability of computing facility accelerates the development of new tools and techniques such as performance simulation, FEA (finite element analysis) and CFD (computational fluid dynamics).

These tools are still very time- and resource-intensive. Nevertheless, they are fast and powerful enough to shift the bottleneck of the ship design process from the assessment stage to the design generation stage. Since 1970s when computers began their tentative, albeit exciting, career of application in hull form design, the techniques of representing and generating complex surfaces, such as a ship's hull, have improved beyond recognition. Despite this, most ship hull design software are still clumsy and time-intensive. The time needed to generate a hull form ready to be turned into a physical scale model may have been reduced from months to a few days, but they are still too long. Thus, rapid hull form generation became the goal of many researchers in this field. In order to provide efficient and speedy designs, several approaches have been introduced for a hull form representation.

The coefficient-based modelling method is one of the first approaches to define the hull form systematically using principal dimensions and coefficients as the input design variables. The greatest merit of this method is the similarity to the traditional concept design. On the other hand, it has also a drawback when the offset data are needed. There is no problem in seeking unique coefficients from one set of offset data. However, it is impossible to generate unique offset data from a single set of coefficients since these coefficients and principal dimensions are insufficient to represent a hull form specifically. In addition, the offset data generated using a set of coefficients may not yield a hull form that is practically feasible. Wire-frame models

are represented by specifying each edge of the physical object or by connecting object's surface vertices with spatial curves, usually parametrically orthogonal to each other. This modelling method is very practical for visualisation of 3D models, and it has been used widely and successfully for representing hull forms especially in CAD software. Although this is a relatively fast and robust method, it requires much human intervention due to the difficulties in the automation of the fairing process. More recently, surface models such as NURBS have become very popular for geometric modelling. This method can be immensely flexible in generating a wide range of shapes through the manipulation of control points, and capable of local modification without affecting other parts. Of course, the price for this flexibility and power is loss of speed.

With these representation methods developed, there have been also several different approaches for hull generation techniques. Conventionally, a new hull form has been developed by transforming existing hull forms, and this method is still widely used in commercial organisations and research centres. Manual manipulation is the classic example of this. Through this method, designers can handle the surface definition data directly, such as offset data and control polygon for a curve or a surface. This method can modify detailed features relatively easily; however, it could be difficult to develop a good hull shape if the designers are not sufficiently well-trained or experienced in the art of hull generation.

There are also transformation methods which can change hull forms based on original design without manual manipulations. One of these methods was introduced by Lackenby in 1950 [1]. Although it is unable to control some areas of hull surface such as bow and stern closely because of its relative simplicity, Lackenby's methods are still used in modern hull design applications.

The search for methods of rapid hull form generation has recently focused on parametric generation method, with NURBS as the actual representation method. However, hull forms can have many different characteristics according to ship types, and it is very difficult to find a mathematical formulation which is capable of

specifying a hull surface using these characteristics only. Furthermore, it may be not be possible to define a set of parameters that can define unique and practically feasible forms during automatic generating process in an optimisation program.

This thesis presents research work done to explore the idea of utilising ‘morphing’ to rapid hull form generation and modification. Originally the morphing technology was developed as a way of interpolating two images in order to figure out the transformation process or to expect the future results from the pictures of mechanical defects or the earth’s surface from satellite in the early 1960s. More recently, morphing became an essential tool of computer-graphics animators by providing an effective means of producing a sequence of forms with small variations which, when strung together, appear to present a seemingly seamless transition from one form to another. The earliest attempts of image morphing were seen in a music video, for instance, Michael Jackson’s ‘Black or White’, which fluidly transforms male faces to female ones, while altering the ethnicity and structure of the faces. Although that video was made in the early of 1990’s and the image morphing was applied for the visual effect, the transition is very smooth and natural. Currently, image morphing has become a commonplace tool and numerous programs have been released on the world wide web. Many can be downloaded for free and are fairly easy to use. However, the image morphing technique is not enough to perform more exacting functions of hull form design or any form design in engineering context. Due to the lack of information on the models’ spatial configuration, the method is unable to handle geometric transformation accurately. For example, the teeth in the closed mouth or the pupil in the closed eyes cannot be recognised in a 2D image and thus a satisfactory morphing to an image containing open mouth or eye is beyond the scope of the technique. This is one of the main reasons why 3D morphing, direct geometric interpolation of 3D objects, has received more attention in recent years. With this technique, it is possible to morph 3D models and this technique is now widely used for animated movies.

It would be impossible for a person to change into a dog in the real world, but a visual effect of a human being transformed to a dog very smoothly and naturally can



be created using the morphing technique. The principle of the visual effect is very simple. It is well known that human eye can perceive around 15 frames per second (fps), and images seem to move fluidly and naturally enough at 24 fps. If, thus, a sufficient number of interpolated models are shown between a starting model and an ending model, the models would seem to change naturally. Graphically, the morphing technique shows a smooth transition between two models by inserting enough numbers of interpolated models between them. In other words, morphing generates newly interpolated models from existing two ‘parent’ models. This idea can be applied to product design where interpolated forms between two extreme forms at the boundary of feasible solution space can reasonably be considered to be within the feasible solution space.

The morphing technique has many potential benefits for the product design including predictability of the outcome and simplicity of the concept. However, by far the most attractive feature of the technique is the speed with which the forms can be generated. This allows a huge number of designs to be produced more or less on demand. In addition, morphing has another big attraction as a design tool, for it generates mesh-based representation of the models directly. Morphed models, therefore, can be easily and directly used in mesh-based performance assessment programs such as FEA and CFD. These attributes of the morphing technique would be undoubtedly useful for product design optimisation process.

The thesis begins by examining existing literature on hull form design and morphing techniques (Chapter 2). Morphing technique including its origin is introduced, and several different types of morphing are briefly explored with some examples (Chapter 3). A simpler application of morphing to 2D form design is explored (Chapters 4 and 5) before its application to the 3D surface modelling is explored (Chapter 6). The algorithms and procedures of using the method for form design are developed and its potential is illustrated through worked examples. Some examples of 3D surface morphing are given in executable file format in the CD accompanying this thesis.

## 1.2 AIMS OF PROJECT

The main aim of this project is to contribute to form design by investigating the applicability of the morphing technique in design process to make it easier and more productive.

More specifically the objectives are as follows:

- To understand the nature of morphing and to identify how this technique can be applied to form design for products of naval architecture and ocean engineering, such as ship hulls and airfoil sections (Chapter 3)
- To develop a method of implementing the morphing technique in 2D form design (Chapter 4)
- To extend the 2D curve morphing to 3D surface morphing (Chapter 6)
- To test the soundness of the developed algorithms and methods by carrying out a number of appropriate case studies (Chapters 4, 5 and 6)
- To evaluate the usefulness or otherwise of the technique in the whole design process (Chapters 4,5 and 6)
- To seek improvement and refinement of the methods and algorithms thus developed (Chapters 4, 5 and 6)
- To evaluate the work carried out in the project and put forward recommendations for further research in this area (Chapter 7)

### **1.3 APPROACH ADOPTED**

The main aim of this research is to develop an improved method of generating high quality models as rapidly as possible. The concept of morphing is introduced and developed for this purpose in this thesis. In order to demonstrate that the technique has the potential of becoming an additional powerful tool at the disposal of the designers, the following approach is taken:

- Investigate the various morphing techniques, particularly those which can be used for handling 3D surfaces.
- Apply the method to morphing 2D curves and identify any problems which need to be addressed. Undertake a case study to become familiar with and conversant with the technique.
- Extend the method to 3D surfaces, with emphasis on ship hull forms. Identify any problems or weaknesses and address these. Garland's Simple Mesh Format (SMF) for QSLIM [2] will be used for the input and output mesh structure, as it is very easy and simple to manipulate. The format consists of two main parts: vertex and face set. A vertex is defined with coordinates [ $\langle x \rangle \langle y \rangle \langle z \rangle$ ], and a face is identified by three vertices [ $\langle v1 \rangle \langle v2 \rangle \langle v3 \rangle$ ] which are listed in counter-clockwise order.
- Examine the issues arising from applying the technique to actual ship design process and address these.

# Chapter 2.

## LITERATURE REVIEW

---

### 2.1 GENERAL REMARKS

The main objective of this Chapter is to review several topics which are relevant to this thesis, namely hull form design and morphing techniques. Other relevant subjects such as turbine airfoil design will be discussed as and where necessary.

### 2.2 HULL FORM DESIGN

Even in the early 20<sup>th</sup> century, many of the new ship designs had traditionally been based on previous hulls. One of the first researchers to pay attention to a systematic study of hull form generation was Taylor [3]. He carried out a systematic study of ship resistance through model experiments and developed one of the first techniques to produce a hull form using a mathematical formulation, based parabolic and hyperbolic functions. This method yielded a series of hull forms that could be used to investigate the effect of certain hull form shape on ship resistance. The Taylor's Standard Series was the first of the methodical series of ship hull forms, and received wide attention and usage. Taylor's research led others to develop the technique that could produce hull forms concerned with ship hydrodynamic analysis. In 1940, Benson [4] developed Taylor's formulation further into a tool that could represent more realistic hull forms. The method was found to be practical, efficient and quick

compared to existing methods of trial and error. Even though Benson's method did not require a large amount of computation, his work was limited by the available computational facilities in those days.

With more widespread availability of digital computers and the increasing availability of computing technology since 1960s, many methods were developed to define the surface of the hull form mathematically. In 1960 Kerwin [5] attempted to develop a system to express a hull surface with a complete mathematical representation. In order to describe a hull form, a polynomial surface was fitted to the hull offset data by using a least squares method. The polynomial techniques could be used for hydrodynamic analysis such as wave-making resistance and ship motions; however, the polynomial equation required was of a very high degree resulting from too many input points.

In order to calculate hydrodynamic properties of a hull form much more efficiently, Von Kerczek [6] described hull shapes using conformal mapping techniques. The conformal mapping is a transformation which maps arbitrary shapes in one plane to a circle in another plane, and this method maps wide range of ship-like sections to the unit circle. As the hydrodynamic properties of circles are known, the properties of the hull section could be found using the transformation quickly. However, the conformal mapping technique had problems in representing certain hull sections, particularly in the region of the bulbous bow. This drawback might not have significant influence to compute only an approximation of hydrodynamic properties for the hull form shape; however, this method could not offer satisfactorily accurate hull representation.

In 1970 a new approach to hull form representation, viz, parametric hull form generation, was introduced by Kuiper [7]. He introduced a method of generating waterline, profile, and section area curves using initial parameters describing hull characteristics rather than using hull form offset data. Although this method was limited in representing complex hull forms, it was the first attempt of the parametric design for a hull form representation. Kuiper's method could produce a hull form

which more or less satisfies a number of key owner's requirements at the preliminary design stage. Loosely based on Kuiper's parametric approach to hull form generation, Reed and Nowacki [8] developed a combined method which uses the conformal mapping technique for the underwater hull and polynomial functions generated from design parameters for the hull above the waterline. Despite the fact that this hybrid technique allowed a greater range of hull shapes to be represented, it was not entirely satisfactory from the viewpoint of hull form representation for practical design purpose.

In order to represent more complex hull shapes precisely, many researchers focused on curve generation techniques, and splines had gained importance through the advent of computers. The first mathematical reference to splines was made by Schoenberg [9, 10] in 1946. Before he mentioned the word spline as a piecewise polynomial curve, this word meant only a form of thin and long beams with constant flexural rigidity used by drafters to draw 'smooth' lines. The mathematical splines had great strength in the simplicity of their construction, the ease of evaluation, and the ability to approximate complex shapes. Similar spline techniques for different applications were developed and discussed by many researchers such as Mehlum [11], Forrest [12], Denman [13], Cline [14] but to name a few. These cubic spline techniques were excellent for hull form representation as shown in Figure 2.1; however, these methods had difficulties in local modification or transformation. In addition, it was essential to know in advance the coordinates of the various points on the curve. This becomes a crucial limitation when applied to curve generation, as the curve is yet to be generated and therefore the points are not known. It can be said that the cubic spline in its most common form is really suited to interpolating between a set of known points rather than to generating a new curve.

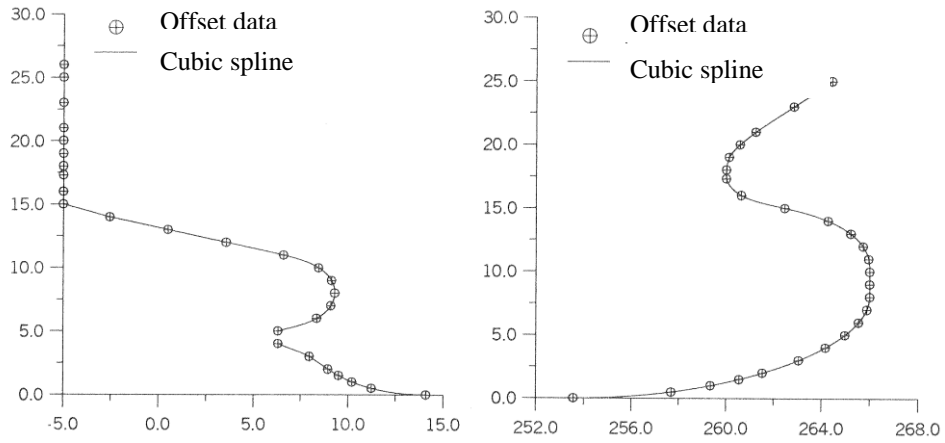


Figure 2.1 A hull form representation using cubic spline [15].

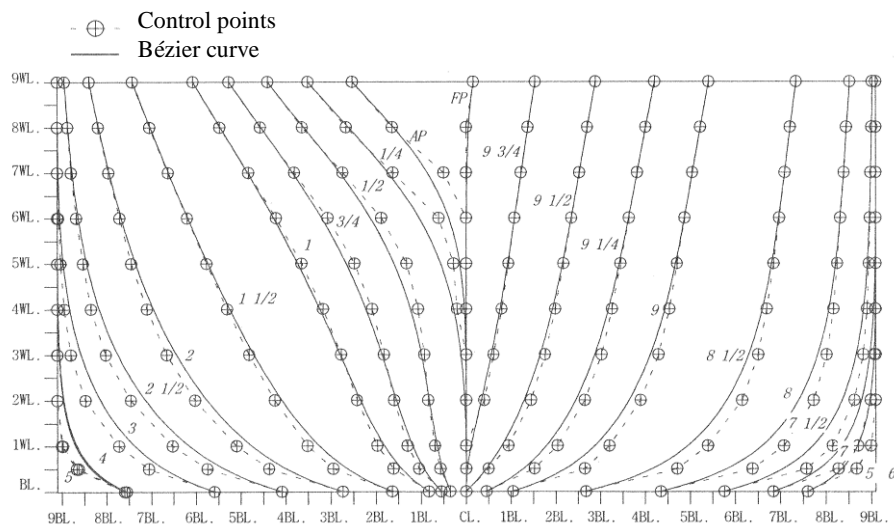


Figure 2.2 A hull form representation using Bézier curve [15].

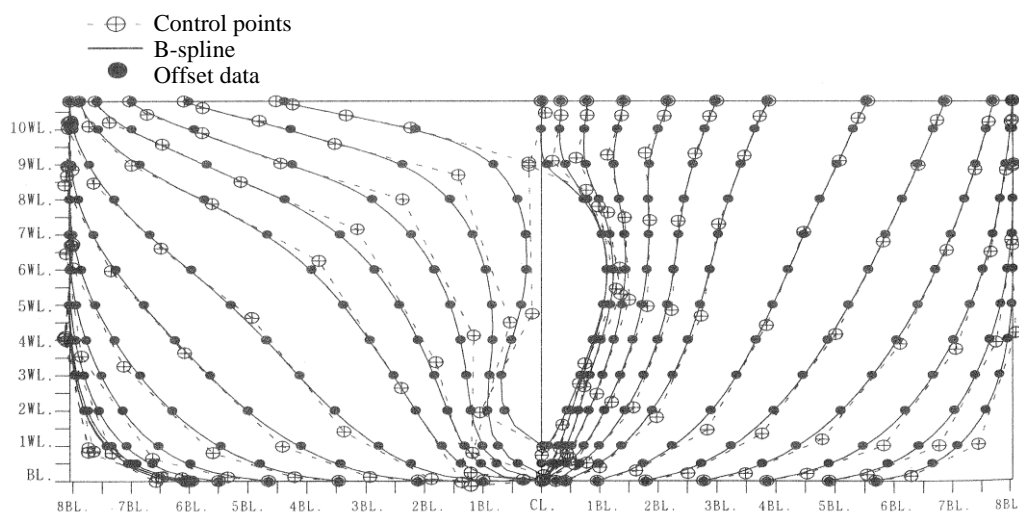


Figure 2.3 A hull form representation using B-spline [15].

On the other hand Bézier [16] developed a free form curve generation technique which has Bernstein polynomial basis while working as an engineer at the car manufacturer Renault. Figure 2.2 shows a generated body plan of a hull form using Bézier curves. This curve is determined by vertices in the control polygon, and these vertices manipulate curve shapes intuitively. However, the order of Bézier curve is completely dependent on the number of control vertices and this characteristic limits the flexibility of this technique in curve generation. For example, a polygon with seven vertices will always generate a curve of order 7 (degree 6). In addition, lack of local control is another flaw of this curve, so that movement of any vertex in the control polygon leads to a change throughout the entire curve.

After de Boor [17] and Cox [18] independently discovered a recursive property of the normalised B-spline basis functions which are suitable for numerical computation, Riesenfeld [19] and Gordon and Riesenfeld [20] applied the B-spline basis to curve definition that became known as B-spline curves. The vertices in control polygon of the B-spline affect only a certain range, so local control is possible without changing overall shape. After further work on the curves, Non-Uniform Rational B-Spline, normally called NURBS was introduced. NURBS can offer great flexibility and precision with weighted control points, and they have been used to represent hull forms by many researchers. (e.g., Rogers and Adams [21]). Wu *et al.* [22], Zhang [23], Lu *et al.* [24] and Zhang *et al.* [25] have researched on representation of hull forms, and they have shown that it is possible to represent simple forms such as submarines and Wigley hull forms with a single NURBS. However, there is still difficulty in representing the stern and stem parts of hull forms precisely.

Creutz and Schubert [26] developed a B-spline method using form parameters. They proposed an approach which generates B-spline curves using seven control points which are derived from fourteen form parameters (offsets, slopes, curvatures, area, centroid and so on). Munchmeyer *et al.* [27] extended the form parameter methods, and introduced a hull design system which allows knuckle lines and bulbous bows to be created. Moreover, Yuille [28] presented one of the first integrated programs for ship design. The program allows displaying both main hull forms and layout of other



parts such as superstructures, decks and bulkhead. It provides a facility to calculate the usual hydrostatic properties and estimate the effect of icing and high winds. In addition to these studies, many similar techniques related to parametric form generation using B-spline have been researched and developed [29-38]. Figure 2.3 shows a generated body plan of a hull form using the B-spline. However, most of the researches have focused on accurate representation rather than rapidity or efficiency of generation.

Keane [39] attempted a method allowing rapid generation of hull forms using conformal mapping technique. The shape of a hull form is defined by using up to 19 parameters, and it is possible to analyse the stability of a parametrically produced hull form. Although the system allows showing the effects of hull form parameter on the shape of the GZ curve, the method is limited to certain types of hull forms such as frigate type forms.

Systematic variation of a hull form was already attended to by Lackenby [1] decades ago. This well-known method, called '1- $C_p$ ' method, transforms hull forms systematically by changing prismatic coefficient ( $C_p$ ), longitudinal centre of buoyancy (LCB), and length of parallel middle body (See Figure 2.4).

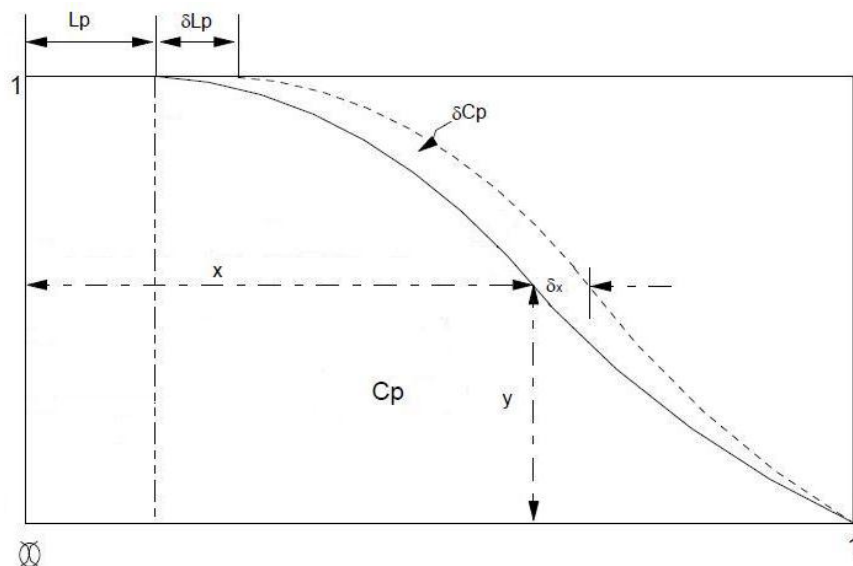


Figure 2.4 Example of Lackenby method.

The Lackenby method was extended by many researchers. Alef and Collatz [40] transformed hull forms numerically by controlling various geometric parameters. The method allowed block coefficient ( $C_b$ ) to be varied simply, and the hull form lines to be derived easily. Versluis [41] and Rabien [42] also developed a transformation method for ship hull forms by changing the LCB and  $C_b$ . However, these Lackenby-based methods had shortcomings of being unable to achieve systematic transformation of steep curvatures such as bow and stern.

Duvigneau and Visonneau [43] have attempted a mesh transformation method for hull forms using a Free Form Deformation (FFD) technique, which was introduced by Sederberg and Parry [44]. This technique allows modifying an object by deforming the space in which the object is embedded, rather than modifying the object itself. Duvigneau and Visonneau [43] used a  $4 \times 3$  control points net in the horizontal and vertical directions for the parameterisation and deformation of a structured grid of  $100 \times 60 \times 50$  nodes around the hull in the stream-wise, radial and girth-wise directions respectively. Duvigneau *et al.* [45] extended the 2D parameterisation to 3D box parameterisation using  $5 \times 6 \times 3$  control points for a structured mesh of  $121 \times 81 \times 41$  nodes. By managing the control points rather than the points of the object directly, these FFD techniques can allow the modified region to be transformed smoothly, and it is also possible to reduce the number of design variables. However, this technique may generate impractical hull forms, and this can be wasteful and inefficient for optimisation process.

Although many representation and generation methods have been developed and applied to the process of designing a ship's hull, the process is still widely based on a 'trial and error' methodology, and it broadly follows the traditional design spiral as shown in Figure 2.5.

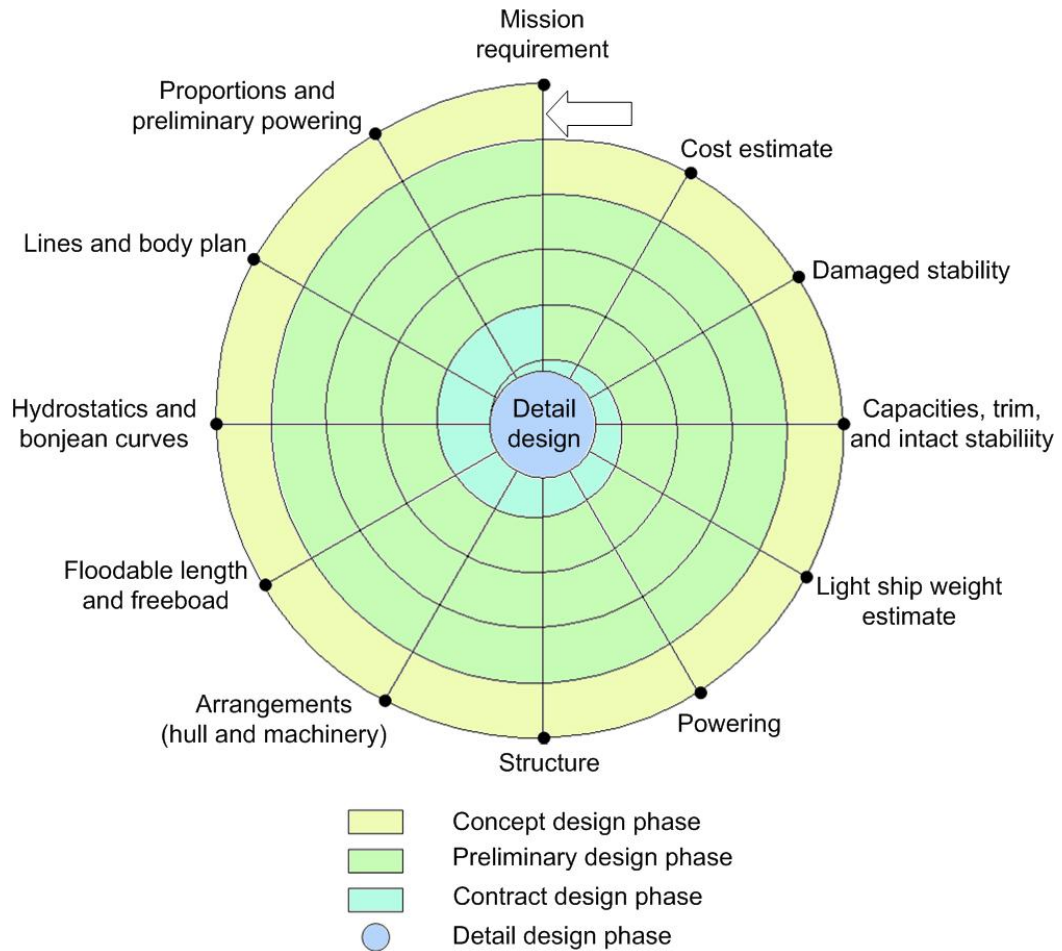


Figure 2.5 Design spiral of ship design.

## 2.3 MORPHING

Although little attention has been given to the application of morphing for hull form design, the morphing techniques have been in use for graphical manipulation for some time. Nevertheless, there has been a surprisingly few documents dealing with application of morphing for product design.

Chen and Parent [46] presented a transformation algorithm for 2D planar contours, and addressed briefly an extension for 3D lofted objects using planar contours. They applied shape-averaging method which was assumed to preserve the characteristics of the original shapes so that the result is useful in predicting trends in design,

extracting forms from a set of related shapes or creating new models by blending features of existing different shapes. However, there is a limit in the transformation of 3D objects. Since the 3D objects of this method are the integration of planar contours and each contour corresponds to another, only two axes in a Cartesian coordinate system can be transformed. Figure 2.6 shows an example of predicting trends of a car model performed by Chen and Parent. They extrapolated the 1971, 1975, 1987, and 1991 Honda Civics models from the 1979 and 1983 models.

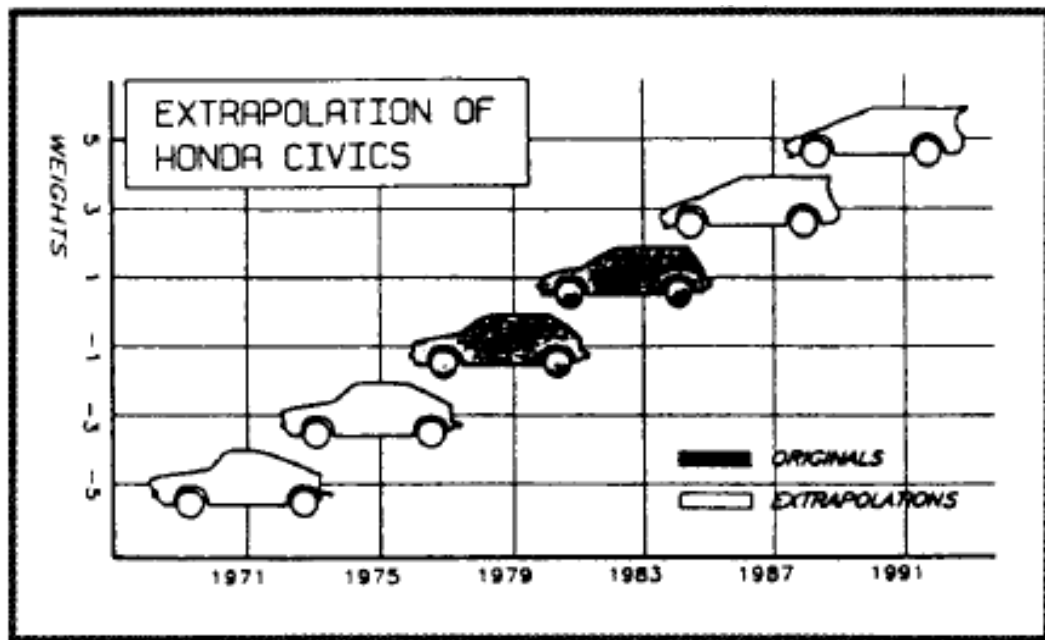


Figure 2.6 A group of shapes extrapolated from the 1979 and 1983 Honda Civics [46].

In 2003, Hsiao and Chang [47] developed a program which allows transformation of 3D objects. However, the 3D configurations are obtained from 2D images and the feature degrades the efficiency and quality of 3D visualisation. The first attempt of morphing application to substantial 3D models of a product was made by Hsiao and Liu [48]. They proposed a feature-based morphing approach for product-shape design based on existing products. However, this technique has not been extended to specific hull form design. Kang and Lee [49, 50] applied mesh-based morphing to rapid generation of ship hull forms. The application of the morphing for form design will be dealt with in the thesis in detail in later chapters.

# Chapter 3.

## MORPHING METHODOLOGY

---

### 3.1 GENERAL REMARKS

The main object of this chapter is to introduce the concept of morphing and how it is implemented in this thesis. A definition of morphing is presented including its linguistic origin, and several different types of morphing (e.g. curves and surface morphings) are explored with some examples. Although image morphing is in widespread use, the methodology of this type of morphing is not discussed in this chapter as it is not directly relevant to the design of forms that we are concerned with here.

### 3.2 WHAT IS MORPHING

The word morphing originates from the word metamorphosis which means a transformation of an insect or an amphibian from an immature form or a larva to an adult form in distinct stages. It also means a change in a form, or nature [51]. For example, to grow into an adult butterfly from an egg it goes through four stages: egg, larva, pupa, and adult as shown in Figure 3.1.

## Butterfly Metamorphosis

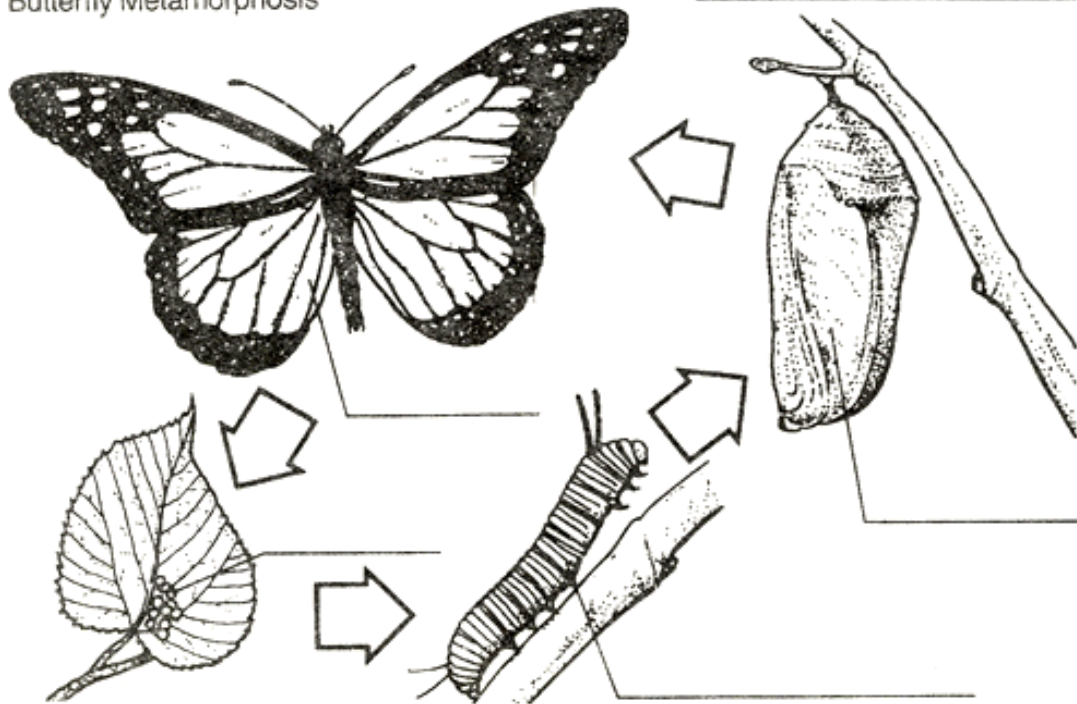


Figure 3.1 Butterfly metamorphosis (from [52]).

The process starting with larva, passing through pupa, and ending as adult is designated metamorphosis. Through the metamorphosis, the insect is transformed not only in form but also in nature. The primary function of the larva is to eat as much as possible and needs to store needed nourishment while the primary function of the adult butterfly is to find a mate and reproduce its offspring. A plant also has these growth processes. It starts from a single seed, which makes roots in order to settle down and grow to a complete tree with stems, leaves, roots, flowers and so on. In addition, the flower has new seeds for the reproduction like an adult butterfly.

As the word ‘morphing’ is derived from or related to the word ‘metamorphosis’, it is reasonable to guess that it has a similar meaning of change or transformation. Morphing, in the context of graphical manipulation, is usually defined as a geometric interpolation technique that transforms an object into another through a seamless transition. Broadly speaking, morphing is thus also known as a transitional technique of graphical objects.

In 1960s, the morphing technology was developed as a means of predicting the transitional process or future results of mechanical defects. Machinery appliances are rarely disassembled for inspection of its parts and of mechanical defects, because it carries enormous direct and indirect costs, including lost hours and added labour. Similarly, morphing was also applied to images processing of the earth's surface taken from satellites. It was prohibitively expensive to take many satellite photos and, therefore, the interim figures between two photographs were often obtained using morphing, an interpolation technique. In addition, image morphing has been used in order to correct camera distortion from the early days of image processing.

Recently, morphing techniques have been developed and applied very successfully in various areas, particularly in the field of computer graphics. Commercial advertisements, films and animations using morphing are often seen in today's media. Although the technique has seldom been applied in product design process so far [47 – 50, 53 – 55], it is possible that the method holds significant potential for rapid generation of shape modelling.

### **3.3 CURVE MORPHING**

Curve morphing is a geometric interpolation carried out between two curves. These curves can be either 2D polyhedral models or 3D polygonal curves. A polygonal curve is a set of line segments  $p_0p_1, p_1p_2, \dots, p_{n-1}p_n$ , where  $p_i (i = 0, 1, 2, \dots, n)$  is a vertex and each segment  $p_i p_{i+1}$  is an edge that represents a curve. Some objects can be defined purely as curves (e.g. cross section outline of an airfoil) or a group of curves (e.g. a mesh representing a surface). Thus curve morphing can be directly applied to curve designs, or applied to definition of boundaries of surface meshes.

The curve morphing is conducted by interpolating correspondent vertices of two different curves, one of which can be termed the 'source' and the other the 'target'. The linear morphing is defined by:

$$M(t) = (1-t) \cdot R_0 + t \cdot R_1, \quad t \in [0, 1] \quad (3-1)$$

where  $t$  is the morphing parameter,  $M(t)$  is the generated model and  $R_0$  and  $R_1$  denote the source and target models respectively. The model at  $t = 0$  is equivalent to the source model (i.e.  $M(0) = R_0$ ), and the model at  $t = 1$  is equivalent to the target model ( $M(1) = R_1$ ). When  $t$  is close to zero, the generated model  $M(t)$  is close to the source model  $R_0$ ; conversely, when  $t$  is close to one, the generated model  $M(t)$  is close to the target model  $R_1$ .

One of the most important and difficult problems of the curve morphing, as well as other forms of morphing, is to establish a correspondence between the vertices of the source and target objects. It is unusual for the two models to have the same number of vertices and with proper correspondence to each other, unless the two models are intentionally prepared. It is important to note that undesired intermediate models are generated from improper correspondence even though they may have the same number of vertices.

Figure 3.2 shows an example of generating different intermediate models by morphing between two triangular-shape polygonal curves. In the figure, each vertex  $p_n$  of the source model corresponds to a vertex  $p_n'$  of the target model. Morphing can be achieved by interpolating between the two corresponding vertices. Even though identical source and target models are used, the correspondence is different for the cases of (a), (b), (c) and (d) of Figure 3.2. This will become apparent when the first triangle and the last triangle are compared to each other for each of these cases. It can be seen that entirely different intermediate models are generated according to differing correspondences as indicated in the figures. Therefore, it is crucial that appropriate correspondence is chosen according to what a designer wants to generate. This does mean, on the other hand, that the correspondence feature provides another handle for controlling the generated shapes.



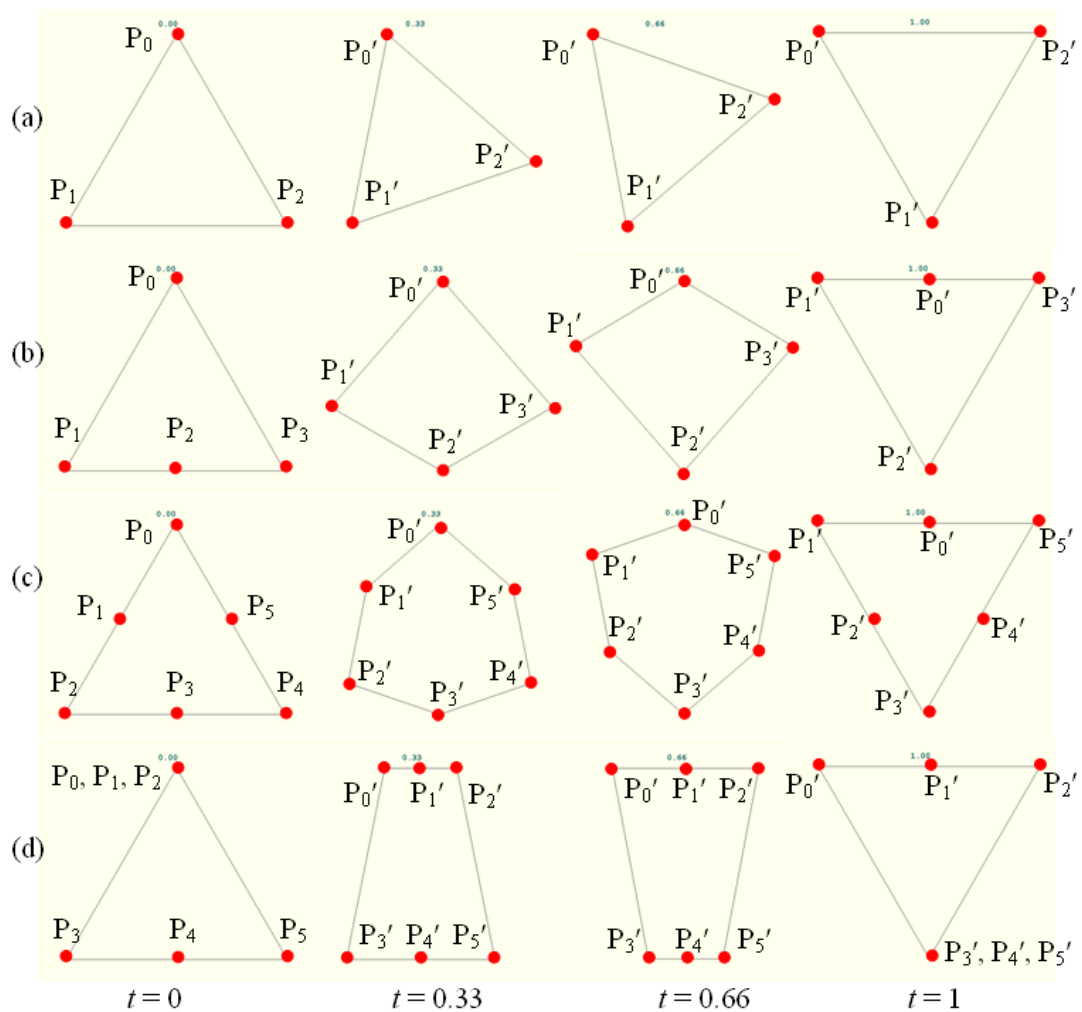


Figure 3.2 An example of generating different intermediate models according to the correspondence. Each point  $p_n$  of the source models corresponds to  $p_n'$  of the target models.

### 3.4 SURFACE MORPHING

Most of the objects found in the real world are 3D shapes. One of the methods of representing these objects is through defining their surfaces, for example ships, airplanes and cars. A surface in turn can be approximated by a series of joined up polygonal surfaces such as triangles. The number and size of these surface patches determine the degree of approximation. In most cases these patches are 2D polygons.

This method of representation is in widespread use for modelling an object, and it is called polygonal mesh, or, simply, mesh. In this thesis, a triangular mesh is used to model 3D surfaces since it allows the mesh structures to be handled simply and the mesh allows more detailed expression than the quadrilateral mesh. For example, the former can express a curved surface patch with only four vertices while the latter requires six vertices at least unless the quadrilateral is bilinear patch. Consequently, the approach adopted for surface morphing is also based on triangular elements here.

Correspondence between the vertices of the source and the target models is again one of the most important problems in surface morphing. In this thesis, two methods are introduced in order to solve the correspondence problem. One method is based on regularising, and another is on merging the meshes.

The former is usually mentioned in connection with mesh quality in order to obtain better numerical results rather than for correspondence problems. Nevertheless, when the source and target models are regularised to identically structured meshes, the vertices of the two models can be corresponded each other. In addition, if the source and target models have similar shapes to each other, this method can establish a good correspondence. Although two hull forms are of different types such as a container ship and a very large crude carrier (VLCC), their shapes are very similar especially for those who are not familiar with ships. Thus, the regularisation method can be a good method for establishing correspondence between source and target models of hull form morphing. Morphing between different types such as a container ship and a VLCC, may provide new hull forms which take on the compound characteristics of both models in factors such as resistance, sea-keeping and stability properties.

The latter, merging, method has been mostly used for establishing correspondences between vertices of the models as a first step in a 3D morphing process. One of the first studies on the 3D surface morphing was discussed by Kent *et al.* [56, 57], and they introduced the merging method for the correspondences. They merged the vertices using a projection on a circle for polygonal curves, and this allows

transformations from one object to another to be easily computed by interpolating between corresponding vertex positions.

A common goal of these methods is to create identically structured meshes. Unfortunately, these methods are very complex to implement directly in 3D. This difficulty can be overcome by mapping the meshes on to 2D planes first (parameterisation process). The parameterised meshes are then converted to identically structured meshes, and then remapped back to 3D surfaces. In this way a proper correspondence is established between the source and target models.

### 3.4.1 Mapping a 3D surface onto a 2D plane

The basic idea of parameterisation is that the boundary nodes of a surface are mapped to the boundary of a convex polygon, and each internal node is defined using the centre of mass of its neighbours known as convex combination (coefficient). This is also known as Tutte's barycentric method [58]. By considering certain particular convex combinations, there are several ways of finding a parameterisation  $P_i$  of a local 3D surface  $S_i$ . These include the method of least squares method which maps  $S_i$  onto the least squares plane through an internal point  $s_i$  and its neighbours, or average normal method that projects  $S_i$  onto the plane, whose normal is an average of the normals of the triangular facets in  $S_i$ . However, these methods can be unsuitable when there exist obtuse triangles or large angles between the triangular facets in  $S_i$ . The weights of neighbours might be negative in these cases.

To deal with this problem, Floater [59] described a method of shape-preserve parameterisation of a 3D surface onto a 2D plane by adopting geodesic polar map first introduced by Welch and Witkin [60]. In this thesis, the Floater's method has been used to obtain the parameterisation of 3D surfaces for both regularising and merging method. The vertex  $s_i$  and its neighbours are mapped to an isomorphic 1-ring in the 2D parameter plane so that the edge length in each radial direction is exactly preserved, and the angles between two consecutive edges are preserved as

well. This satisfies the following two equations for  $k = 1, \dots, c_i$  when a local surface  $S_i$ , consisting of an internal point  $s_i$  and its neighbours  $s_{jk}$ , is mapped to a local parameterisation  $P_i$  consisting of an internal point  $p_i$  and its neighbours  $p_{jk}$ .

$$\|p_{jk} - p_i\| = \|s_{jk} - s_i\| \quad (3-2)$$

$$\text{ang}(p_{jk}, p_i, p_{jk+1}) = 2\pi \cdot \frac{\text{ang}(s_{jk}, s_i, s_{jk+1})}{\theta_i} \quad (3-3)$$

where  $c_i$  indicates the number of neighbours, and  $\text{ang}(a, b, c)$  means the angle between the edges  $ab$  and  $bc$ .  $\theta_i$  denotes total sum of the angles between two consecutive edges of a local surface  $S_i$ . (i.e.  $\theta_i = \sum_{k=1}^{c_i} \text{ang}(s_{jk}, s_i, s_{jk+1})$ ).

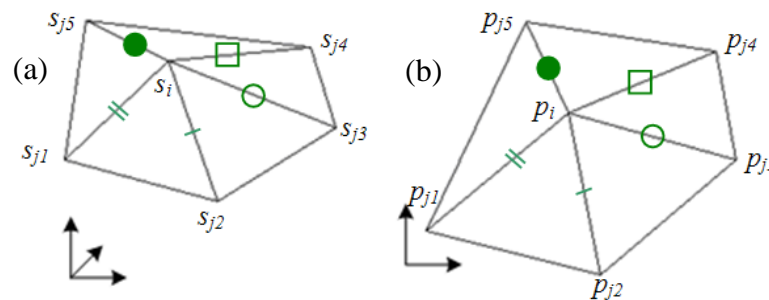


Figure 3.3 An example of a local parameterisation.

Figure 3.3 shows a local surface  $S_i$  that has five triangular facets and its parameterisation  $P_i$ . The first step of the parameterisation is to find the internal angles  $\text{ang}(s_{j1}, s_i, s_{j2})$ ,  $\text{ang}(s_{j2}, s_i, s_{j3})$ ,  $\dots$ , and  $\text{ang}(s_{j5}, s_i, s_{j1})$ . The sum of the angles is  $\theta_i$ . As shown in Equation (3-3), each  $\text{ang}(s_{jk}, s_i, s_{jk+1})$  can be converted to a new angle occupying the parameterised 2D domain by considering the ratio of each angle to the total angle  $\theta_i$ . After choosing any position of  $p_i$  for  $s_i$ , its neighbours  $p_{jk}$  can be translated to a parameterised plane as shown in Figure 3.3 (b), with the same distance between the internal point and each neighbour as shown in Equation (3-2) and the new angles as shown in Equation (3-3).

Again, the convex combination is the centre of mass of neighbours to an internal point. Thus an internal point  $p_i$  can be represented by its neighbours  $p_{jk}$  and their convex combination coefficients  $\lambda_{i,k}$ , and the convex combination coefficients are non-negative and sum up to one as follows:

$$p_i = \sum_{k=1}^{c_i} \lambda_{i,k} p_{jk} \quad (3-4)$$

$$\sum_{k=1}^{c_i} \lambda_{i,k} = 1 \quad \text{and} \quad \lambda_{i,k} > 0 \quad (3-5)$$

where  $c_i$  denotes the number of neighbours,  $p_i$  indicates the  $i^{\text{th}}$  internal point and  $p_{jk}$  indicates the  $k^{\text{th}}$  neighbour of the internal point. In addition, when the parameterisation of a local surface is calculated from Equation (3-2) and (3-3), it is then possible to calculate convex combination coefficient  $\lambda_{i,k}$  from the internal point  $p_i$  and its neighbours  $p_{jk}$ .

For  $c_i = 3$ , the convex combination coefficients  $\lambda_{i,k}$  can be easily calculated as

$$\lambda_{i,1} = \frac{\text{area}(p_i, p_{j2}, p_{j3})}{\text{area}(p_{j1}, p_{j2}, p_{j3})}, \lambda_{i,2} = \frac{\text{area}(p_{j1}, p_i, p_{j3})}{\text{area}(p_{j1}, p_{j2}, p_{j3})}, \lambda_{i,3} = \frac{\text{area}(p_{j1}, p_{j2}, p_i)}{\text{area}(p_{j1}, p_{j2}, p_{j3})} \quad (3-6)$$

where  $\text{area}(a,b,c)$  means the area of the triangle  $abc$ .

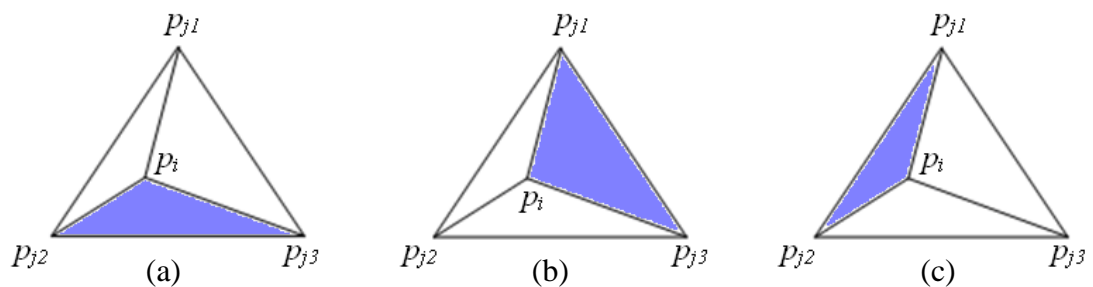


Figure 3.4 An example of calculating convex combination coefficients  $\lambda_{i,k}$  ( $c_i = 3$ ).

As shown in Figure 3.4, they are the unique barycentric coordinates of  $p_i$  with respect to the triangle  $p_{j1}p_{j2}p_{j3}$ , and the internal point will be expressed from Equation (3-4) as:

$$p_i = \lambda_{i,1}p_{j1} + \lambda_{i,2}p_{j2} + \lambda_{i,3}p_{j3} \quad (3-7)$$

For  $c_i > 3$ , the procedure to calculate the convex combination coefficients  $\lambda_{i,k}$  becomes more complex. For each neighbouring node  $p_l$ ,  $l \in \{1, \dots, c_i\}$ , the straight line through  $p_l$  ( $p_{j2}$  in Figure 3.5 (b)) and  $p_i$  intersects a line segment with endpoint  $p_{r(l)}$  and  $p_{r(l)+1}$  as shown in Figure 3.5 (c) ( $p_{j4}$  and  $p_{j5}$  in Figure 3.5 (b)).  $p_l$ ,  $p_{r(l)}$  and  $p_{r(l)+1}$  are then taken as the neighbouring points. The weighting factors applied to these points  $\delta_1$ ,  $\delta_2$  and  $\delta_3$  are such that  $\delta_1 > 0$ ,  $\delta_2 > 0$ ,  $\delta_3 > 0$ , and  $\delta_1 + \delta_2 + \delta_3 = 1$ .

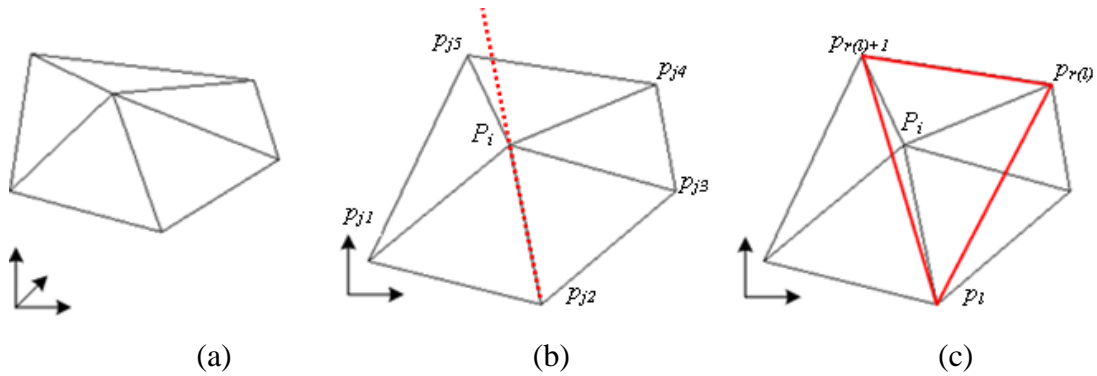


Figure 3.5 An example of calculating convex combination coefficients  $\lambda_{i,k}$  ( $c_i > 3$ ).

The internal point  $p_i$  can be found in a similar manner with Equation (3-7) by the following expression:

$$p_i = \delta_1 p_l + \delta_2 p_{r(l)} + \delta_3 p_{r(l)+1}, \quad l \in \{1, \dots, c_i\} \quad (3-8)$$

which can be rewritten as:

$$p_i = \delta_1 p_l + \delta_2 p_{r(l)} + (1 - (\delta_1 + \delta_2)) p_{r(l)+1}, \quad l \in \{1, \dots, c_i\} \quad (3-9)$$

From the Equation (3-9), there are two unknown variables ( $\delta_1, \delta_2$ ) and two equations because of two components  $U = \{u, v\}$  in 2D, so  $\delta_1, \delta_2$  and  $\delta_3$ , can be found. In order to simplify the equations a more general notation is adopted to apply to all neighbours, by substituting  $\mu_{l,l} = \delta_1$ ,  $\mu_{r(l),l} = \delta_2$ ,  $\mu_{r(l)+1,l} = \delta_3$ , and  $\mu_{k,l} = 0$  otherwise. In case that the neighbour  $p_{j2}$  is selected for  $p_l$  as shown in Figure 3.5 (b) and (c),  $\mu_{k,l}$  and  $p_i$  would be:

$$\mu_{1,2} = 0, \mu_{2,2} = \delta_1, \mu_{3,2} = 0, \mu_{4,2} = \delta_2 \text{ and } \mu_{5,2} = \delta_3$$

$$\begin{aligned} p_i &= \mu_{1,2} \cdot p_{j1} + \mu_{2,2} \cdot p_{j2} + \mu_{3,2} \cdot p_{j3} + \mu_{4,2} \cdot p_{j4} + \mu_{5,2} \cdot p_{j5} \\ &= \mu_{2,2} \cdot p_{j2} + \mu_{4,2} \cdot p_{j4} + \mu_{5,2} \cdot p_{j5} \end{aligned} \quad (3-10)$$

For each  $l$ ,

$$p_i = \sum_{k=1}^{c_i} \mu_{k,l} p_{jk} \quad (3-11)$$

$$\sum_{k=1}^{c_i} \mu_{k,l} = 1 \text{ and } \mu_{k,l} \geq 0 \quad (3-12)$$

Finally, the convex combination coefficients of the  $i^{\text{th}}$  internal point  $p_i$  and its neighbours  $p_{jk}$  are:

$$\lambda_{i,k} = \frac{1}{c_i} \sum_{l=1}^{c_i} \mu_{k,l}, \quad k \in \{1, \dots, c_i\} \quad (3-13)$$

For  $c_i > 3$ , Equations (3-4) and (3-5) are conformed to the following equations derived from Equations (3-11), (3-12) and (3-13).

$$p_i = \frac{1}{c_i} \sum_{l=1}^{c_i} p_i = \frac{1}{c_i} \sum_{l=1}^{c_i} \sum_{k=1}^{c_i} \mu_{k,l} p_{jk} = \sum_{k=1}^{c_i} \frac{1}{c_i} \sum_{l=1}^{c_i} \mu_{k,l} p_{jk} = \sum_{k=1}^{c_i} \lambda_{i,k} p_{jk} \quad (3-14)$$

$$\sum_{k=1}^{c_i} \lambda_{i,k} = \sum_{k=1}^{c_i} \frac{1}{c_i} \sum_{l=1}^{c_i} \mu_{k,l} = \frac{1}{c_i} \sum_{l=1}^{c_i} \sum_{k=1}^{c_i} \mu_{k,l} = \frac{1}{c_i} \sum_{l=1}^{c_i} 1 = 1 \quad (3-15)$$

Also  $\mu_{k,k}(\delta_1)$  is nonzero for every  $k$ , and hence  $\lambda_{i,k} > 0$  for all values of  $k$ .

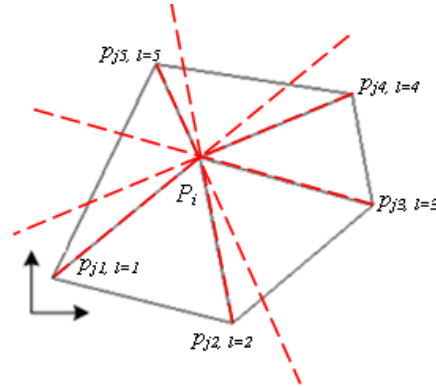


Figure 3.6 Straight lines through  $p_{jk}$  and  $p_i$  intersects a line segment ( $c_i=5$ ).

The following equations are the result of the example shown in Figure 3.6.

	$l=1$	$l=2$	$l=3$	$l=4$	$l=5$	
$k=1$	$\mu_{1,1}(\delta_1)$	0	$\mu_{1,3}(\delta_3)$	$\mu_{1,4}(\delta_3)$	0	$\rightarrow \sum_{l=1}^5 \mu_{1,l}$
$k=2$	0	$\mu_{2,2}(\delta_1)$	0	0	$\mu_{2,5}(\delta_2)$	$\rightarrow \sum_{l=1}^5 \mu_{2,l}$
$k=3$	0	0	$\mu_{3,3}(\delta_1)$	0	$\mu_{3,5}(\delta_3)$	$\rightarrow \sum_{l=1}^5 \mu_{3,l}$
$k=4$	$\mu_{4,1}(\delta_2)$	$\mu_{4,2}(\delta_2)$	0	$\mu_{4,4}(\delta_1)$	0	$\rightarrow \sum_{l=1}^5 \mu_{4,l}$
$k=5$	$\mu_{5,1}(\delta_3)$	$\mu_{5,2}(\delta_3)$	$\mu_{5,3}(\delta_2)$	$\mu_{5,4}(\delta_2)$	$\mu_{5,5}(\delta_1)$	$\rightarrow \sum_{l=1}^5 \mu_{5,l}$
	↓	↓	↓	↓	↓	↓
	$\sum_{k=1}^5 \mu_{k,1} = 1$	$\sum_{k=1}^5 \mu_{k,2} = 1$	$\sum_{k=1}^5 \mu_{k,3} = 1$	$\sum_{k=1}^5 \mu_{k,4} = 1$	$\sum_{k=1}^5 \mu_{k,5} = 1$	5



From the relationships shown above following equations are obtained.

$$\sum_{k=1}^5 \mu_{k,1} + \sum_{k=1}^5 \mu_{k,2} + \sum_{k=1}^5 \mu_{k,3} + \sum_{k=1}^5 \mu_{k,4} + \sum_{k=1}^5 \mu_{k,5} = 5 \quad (3-16)$$

$$\sum_{l=1}^5 \mu_{1,l} + \sum_{l=1}^5 \mu_{2,l} + \sum_{l=1}^5 \mu_{3,l} + \sum_{l=1}^5 \mu_{4,l} + \sum_{l=1}^5 \mu_{5,l} = 5 \quad (3-17)$$

For the  $i^{\text{th}}$  internal point, Equation (3-17) can be rewritten using Equation (3-13), which satisfies Equation (3-5).

$$\begin{aligned} & \frac{1}{5} \cdot \sum_{l=1}^5 \mu_{1,l} + \frac{1}{5} \cdot \sum_{l=1}^5 \mu_{2,l} + \frac{1}{5} \cdot \sum_{l=1}^5 \mu_{3,l} + \frac{1}{5} \cdot \sum_{l=1}^5 \mu_{4,l} + \frac{1}{5} \cdot \sum_{l=1}^5 \mu_{5,l} = \frac{1}{5} \cdot 5 \\ \Rightarrow & \lambda_{i,1} + \lambda_{i,2} + \lambda_{i,3} + \lambda_{i,4} + \lambda_{i,5} = 1 \quad (3-18) \\ \Rightarrow & \sum_{k=1}^{c_i} \lambda_{i,k} = 1 \end{aligned}$$

In order to map vertices of 3D surface to 2D plane with convex combination coefficients  $\lambda_{i,j}$ , boundary nodes of the 3D surface should be parameterised to edges of a 2D plane. This can be done based on identical length ratios, and the shape of the parameterised plane can be any kinds of polygons such as triangular or quadrilateral polygons. If a surface will be refined to a regularised triangular mesh later, then a triangular parameterised domain would be better than other shapes. For example, for the surface shown in Figure 3.7 (a) to be mapped onto a triangular polygon, say A'C'D' as shown in Figure 3.7 (b), three boundaries of both the surface and parameterised domain should be established. The polygonal curves ABC, CD, and DA are selected for the boundaries of the surface here, and vertices A', C', and D' are assigned to similar shape-position with the corresponding vertices of the surface. Alternatively, vertices A', C', and D' can be allocated the same positions as vertices A, C, and D because the three points construct a unique 2D plane.

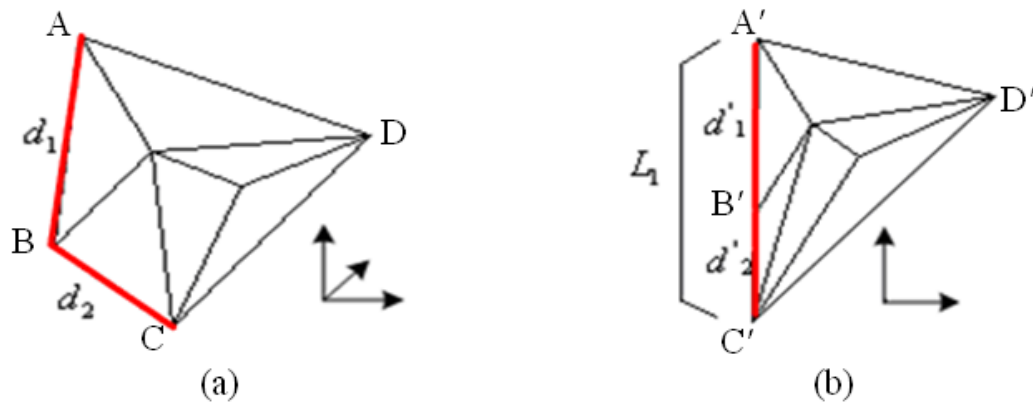


Figure 3.7 Example of parameterising nodes on a boundary.

Let the lengths of the straight lines linking two nodes of the outermost mesh elements on line ABC be  $d_1$  and  $d_2$ . Then the distances between the two nodes of the outermost mesh elements resting on line A'B'C', viz  $d'_1$  and  $d'_2$ , are calculated as follows:

$$d'_1 = L_1 \cdot \frac{d_1}{d_1 + d_2}, \quad d'_2 = L_1 \cdot \frac{d_2}{d_1 + d_2} \quad (3-19)$$

where  $L_1 = d'_1 + d'_2$  or the length of A'C'.

Equation (3-19) can be generalised as follows:

$$d'_1 = L_1 \cdot \frac{d_1}{\sum_{k=1}^{\varepsilon} d_k}, \quad d'_2 = L_1 \cdot \frac{d_2}{\sum_{k=1}^{\varepsilon} d_k}, \quad \dots, \quad d'_\varepsilon = L_1 \cdot \frac{d_\varepsilon}{\sum_{k=1}^{\varepsilon} d_k} \quad (3-20)$$

where  $\varepsilon + 1$  is the number of nodes on a side of parameterised polygon, and  $L_1$  is equivalent to the length of a side of the parameterised polygon,  $d'_1 + d'_2 + \dots + d'_\varepsilon$ .

The nodes lying on other boundaries are similarly mapped onto the 2D plane. With the convex combination coefficients and parameterised nodes on the boundaries thus obtained, internal points of the 3D surface can be finally mapped to the 2D plane. If

the total number of nodes is  $N$ , and the number of the internal nodes is  $n$ , we can represent the parameterised internal nodes with  $U_1, \dots, U_n$ , and the parameterised boundary nodes with  $U_{n+1}, \dots, U_N$ . For the  $i^{\text{th}}$  internal node,  $i \in \{1, \dots, n\}$ , the convex combination coefficients  $\lambda_{i,k}$  can be rewritten as:

$$\begin{cases} \lambda_{i,k} > 0 & \text{if } (i,k) \in E \\ \lambda_{i,k} = 0 & \text{if } (i,k) \notin E \end{cases}, \quad \sum_{k=1}^N \lambda_{i,k} = 1 \quad (3-21)$$

where a node  $U_k$  is a neighbour of node  $U_i$  if  $(i,k) \in E$  (i.e.  $E$  is an edge set).

And each internal node  $U_i$  can be represented by its neighbouring nodes  $U_k$  and convex combination coefficients  $\lambda_{i,k}$ :

$$U_i = \sum_{k=1}^N \lambda_{i,k} U_k \quad (3-22)$$

$$U_i = \sum_{k=1}^n \lambda_{i,k} U_k + \sum_{k=n+1}^N \lambda_{i,k} U_k, \quad i = 1, \dots, n \quad (3-23)$$

The first term of the right side of Equation (3-23) represents parameterised internal nodes, which are unknown, and the second term signifies parameterised boundary nodes, which are known. By grouping the unknown terms to the left side and the known term to the right side, Equation (3-23) can be rewritten in the following form:

$$U_i - \sum_{k=1}^n \lambda_{i,k} U_k = \sum_{k=n+1}^N \lambda_{i,k} U_k, \quad i = 1, \dots, n \quad (3-24)$$

In 2D,  $U_i$  has two components  $u_i$  and  $v_i$ , and therefore it can be expressed in the form of two matrix equations:

$$\mathbf{A}\mathbf{u} = \mathbf{b}_1, \quad (3-25)$$

$$\mathbf{A}\mathbf{v} = \mathbf{b}_2 \quad (3-26)$$

Here  $\mathbf{u}$  and  $\mathbf{v}$  are the column vectors of  $(u_1, \dots, u_n)$  and  $(v_1, \dots, v_n)$ , respectively, and  $\mathbf{A}$  is an  $n \times n$  matrix whose elements are:

$$\begin{aligned} a_{i,k} &= 1 && \text{if } i = k \\ a_{i,k} &= -\lambda_{i,k} && \text{if } i \neq k \end{aligned}, \quad i = 1, \dots, n, \quad k = 1, \dots, n \quad (3-27)$$

$b_1$  and  $b_2$  are the multiplication of an  $n \times (N-n)$  matrix by the column vectors of  $(u_{n+1}, \dots, u_N)$  and  $(v_{n+1}, \dots, v_N)$  respectively. The elements of the  $n \times (N-n)$  matrix are equivalent to  $\lambda_{i,k}$  for  $i = 1, \dots, n$  and  $k = n + 1, \dots, N$ .

Thus, Equation (3-24) with three internal nodes ( $n = 3$ ) and four boundary nodes ( $N = 7$ ) becomes:

$$\begin{aligned} U_1 - \lambda_{1,1}U_1 - \lambda_{1,2}U_2 - \lambda_{1,3}U_3 &= \lambda_{1,4}U_4 + \lambda_{1,5}U_5 + \lambda_{1,6}U_6 + \lambda_{1,7}U_7 \\ U_2 - \lambda_{2,1}U_1 - \lambda_{2,2}U_2 - \lambda_{2,3}U_3 &= \lambda_{2,4}U_4 + \lambda_{2,5}U_5 + \lambda_{2,6}U_6 + \lambda_{2,7}U_7 \\ U_3 - \lambda_{3,1}U_1 - \lambda_{3,2}U_2 - \lambda_{3,3}U_3 &= \lambda_{3,4}U_4 + \lambda_{3,5}U_5 + \lambda_{3,6}U_6 + \lambda_{3,7}U_7 \end{aligned} \quad (3-28)$$

This can be written in a matrix form, as in Equation (3-25) and (3-26):

$$\begin{aligned} \begin{bmatrix} 1 & -\lambda_{1,2} & -\lambda_{1,3} \\ -\lambda_{2,1} & 1 & -\lambda_{2,3} \\ -\lambda_{3,1} & -\lambda_{3,2} & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} &= \begin{bmatrix} \lambda_{1,4} & \lambda_{1,5} & \lambda_{1,6} & \lambda_{1,7} \\ \lambda_{2,4} & \lambda_{2,5} & \lambda_{2,6} & \lambda_{2,7} \\ \lambda_{3,4} & \lambda_{3,5} & \lambda_{3,6} & \lambda_{3,7} \end{bmatrix} \begin{bmatrix} u_4 \\ u_5 \\ u_6 \\ u_7 \end{bmatrix} \\ \mathbf{A} \quad \times \quad \mathbf{u} &= \quad b_1 \end{aligned} \quad (3-29)$$

$$\begin{aligned} \begin{bmatrix} 1 & -\lambda_{1,2} & -\lambda_{1,3} \\ -\lambda_{2,1} & 1 & -\lambda_{2,3} \\ -\lambda_{3,1} & -\lambda_{3,2} & 1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} &= \begin{bmatrix} \lambda_{1,4} & \lambda_{1,5} & \lambda_{1,6} & \lambda_{1,7} \\ \lambda_{2,4} & \lambda_{2,5} & \lambda_{2,6} & \lambda_{2,7} \\ \lambda_{3,4} & \lambda_{3,5} & \lambda_{3,6} & \lambda_{3,7} \end{bmatrix} \begin{bmatrix} v_4 \\ v_5 \\ v_6 \\ v_7 \end{bmatrix} \\ \mathbf{A} \quad \times \quad \mathbf{v} &= \quad b_2 \end{aligned} \quad (3-30)$$

By substituting the values of  $\lambda_{i,j}$  according to Equation (3-27), the internal nodes  $U_i = (u_i, v_i)$  mapped onto the 2D plane can be obtained using the Gauss-elimination method.

### 3.4.2 Generating regularised mesh

A method of obtaining the parameterised meshes is discussed in Section 3.4.1, and these meshes should be of an isomorphic mesh structure to establish a correspondence between source and target models. By regularising the two meshes to have the same number of vertices, edges and faces, the two models can be made to have an identical mesh structure.

As the first step in regularising the meshes, the given meshes need to be subdivided. The primary reason for the subdivision is to ensure that the corresponding vertices between the two models are broadly from the same region of the geometry. The secondary reason for subdivision is to make mapping processes easier in both directions between 3D and 2D. When bow parts of hull form are projected onto horizontal plan, the parameterised meshes of bulbous bow and forecastle might be superimposed onto each other. Several researchers [61, 62] have paid attention to the possible folding problem (essentially a multi-value problem), and subdivision of regions with high curvature into smaller patches is one of the easiest ways to avoid this problem. Since regularised triangular meshes can be generated conveniently from triangular patches, all the subdivided patches are made triangular in this thesis.

With the parameterisation of the subdivided triangular shape patches obtained, regularisation is conducted by repeatedly bisecting each edge of the mesh elements, starting with the triangle formed by the original borders. The original triangle is divided into 4 triangles by bisecting each edge and connecting the bisecting points. Each triangle is then divided further into 4 triangles and so on. In this way, after  $i^{\text{th}}$  repetition,  $4^i$  ( $i = 1, 2, 3, \dots$ ) regularised triangles are generated.

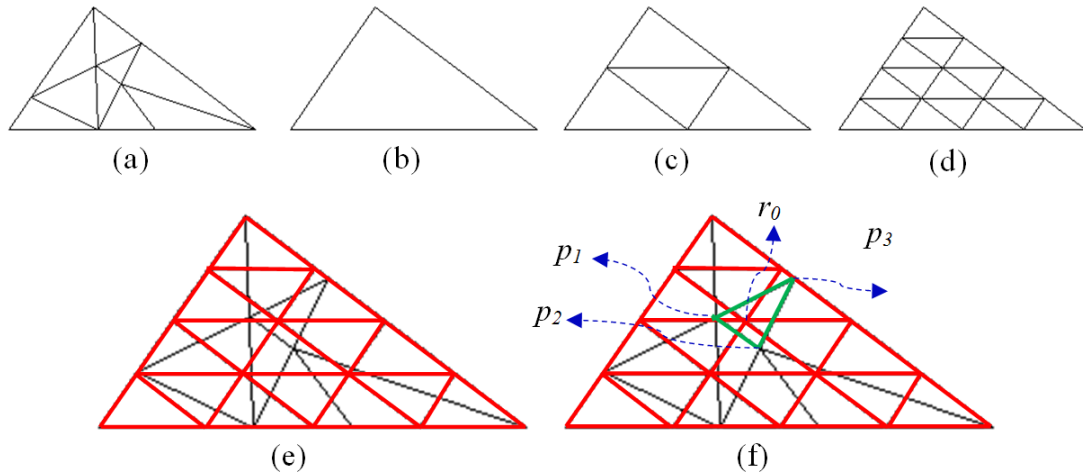


Figure 3.8 Procedure of regularising: (a) Original parameterised mesh; (b) Triangular borders; (c) Regularised mesh after first bisecting ( $4^1$  triangles); (d) Regularised mesh after second bisecting ( $4^2$  triangles); (e) Original parameterised mesh superimposed on the regularised mesh; (f) Representing regularised mesh with barycentric coordinates of original parameterised mesh.

With a surface parameterised onto a 2D triangle as shown in Figure 3.8 (a), the method described above produces a regularised triangular mesh such as shown in Figure 3.8 (d). The original parameterised mesh mapped from 3D surface and the regularised triangular mesh are then superimposed on the same 2D domain triangle as shown Figure 3.8 (e). The vertices of the regularised triangles can be represented with barycentric coordinates of parameterised mesh using a point set  $\mathbf{P} = (\zeta_i, \xi_i)$ . A vertex of the regularised mesh  $r_o$  can be represented with surrounding parameterised facet  $\Delta p_1 p_2 p_3$  in a similar way with Equation (3-7) of Section 3.4.1.

$$r_o = \alpha \cdot p_1 + \beta \cdot p_2 + (1 - \alpha - \beta) \cdot p_3 \quad (3-31)$$

Using the same barycentric coordinates with a vertex set  $\mathbf{S} = (x_i, y_i, z_i)$  of the 3D surface, the regularised mesh can be mapped back onto the 3D surface.

### 3.4.3 Generating merged mesh

Another method of establishing a correspondence between two models is mesh merging. One of the first researchers to discuss this method was Kent et al. [56, 57], and the merging method is still widely used for establishing a correspondence between two models during morphing process in the field of computer graphics. The merging method saves much time by reducing the need for manual manipulations, and is possible to preserve the original mesh shape. In the case of regularisation method, meshes need to be subdivided first in order to ensure that the corresponding vertices between the two models are from similar regions of the respective models. This process of subdivision, however, is not required in the merging method. In addition, a regularised mesh may not keep the vertices of the original mesh so that the 3D surface represented by the regularised mesh may not be exactly the same as the original one, although they may be similar enough to each other for most purposes.

The merging method requires the source and the target models to be parameterised onto a common plane domain. The algorithm of this method is described with an example of two simple parameterised meshes here. Figure 3.9 illustrates the basic idea of the mesh merging method. Figure 3.9 (a) is a 2D parameterised source model mesh and Figure 3.9 (b) is a 2D parameterised target model mesh, where both the source and the target have identical boundary. Since the boundaries ABCE in Figure 3.9 (a) and ABCDE in (b) are geometrically identical, they can be overlaid. In addition, new intersected vertices G and F are created as shown in Figure 3.9 (c). In essence, a set of new faces {AGE, ABG, BFG, BCF, FCD, GFD, GDE} are generated as shown in Figure 3.9 (d).

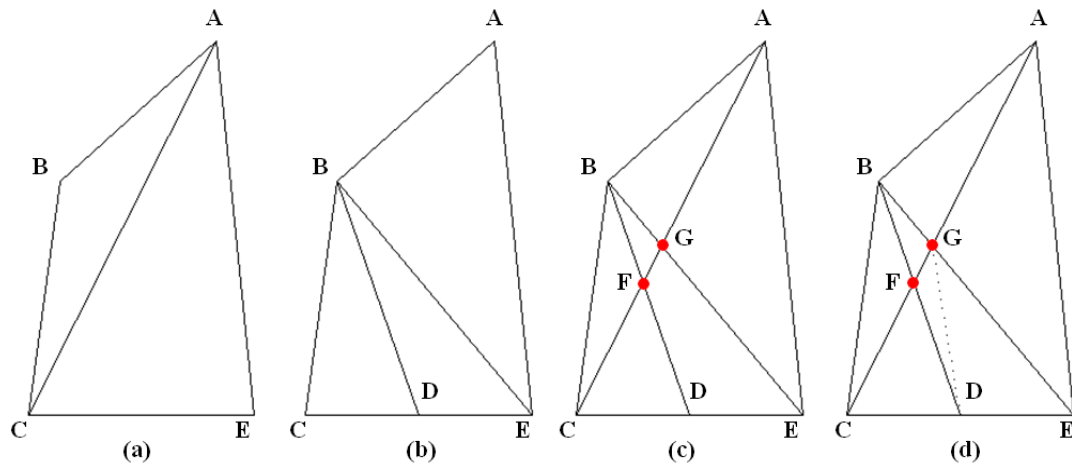


Figure 3.9 Procedure of mesh merging:  
 (a) source mesh; (b) target mesh; (c) overlaid meshes; (d) merged mesh.

The principal aim of merging is to establish a new set of vertices and faces as a new mesh which can be used to represent both models. In order to make a set of new vertices from two existing meshes, the original vertices of the two models are retained in the new vertex set. However, if there are shared or coincidental vertices, only one of them for each such pair is used to remove the redundancy. For example, vertex A in Figure 3.9 (a) and vertex A in Figure 3.9 (b) are the same point so that just one of them is saved to a new vertex set. The next step is to add new points of intersection which are generated from crossed edges of two meshes, such as vertices F and G of Figure 3.9 (c).

Establishing a new face set is somewhat more complex than the vertex set. In finding the new face set the first step is to determine an edge set. In the similar manner with the new vertex set, the original edges of both models are saved to a new edge set. Again, duplicated edges are added once only. The resulting edge set from Figure 3.9 (a) and Figure 3.9 (b) are then  $\{AB, BC, EA, AC, CD, DE, BD, BE\}$ . After removing the redundant edges, the crossing edges are split by the intersecting edges. In Figure 3.9 (c), for instance, the edge  $\{AC\}$  split  $\{BE\}$  and  $\{BD\}$  to  $\{BG, GE\}$  and  $\{BF, FD\}$  respectively, and the edge  $\{AC\}$  itself is split to  $\{AG, GF, FC\}$  by the intersecting edges  $\{BE\}$  and  $\{BD\}$ . From the edge set thus obtained, it is possible to find neighbouring vertices for each vertex, and a new set of



faces can be generated with the vertices taken in counter-clockwise direction. In the case of the vertex A whose neighbours are {B, G, E}, the newly generated faces are {ABG, AGE}. With the vertex F whose neighbouring vertices are {B, C, D, G}, the edge {GD} should be added to the new edge set to avoid non-triangular faces. The whole faces can be found from each vertex and its neighbours in the like manner.

The vertices of the merged meshes contains at least one vertex of both parameterised meshes because the vertices of merged mesh are shared by the two parameterised meshes. Thus, during the re-mapping of merged mesh to the source model, if a vertex  $(\zeta_i, \xi_i)$  belongs to the parameterised mesh from the source model, it is re-parameterised to the corresponding original 3D point  $(x_i, y_i, z_i)$ . If not, the point can be represented with barycentric coordinates of the parameterised mesh of source model using a point set  $\mathbf{P}_I = (\zeta_i, \xi_i)$ . Using the same barycentric coordinates with a vertex set  $\mathbf{S}_I = (x_i, y_i, z_i)$  of the 3D surface, the non-denoted points can now be mapped back onto the 3D surface. In the same manner, the parameterised mesh of the target model can be mapped back onto the 3D surface.

### 3.5 GENERATING EXTREME TARGET MODELS

Since morphing is based on interpolation between two models, and in order to ensure that the widest possible range of alternatives are generated, these parent models need to be at or beyond the boundary of realistically feasible solution domain. However, when there is no convenient target model available which can be used as the target model, or if a certain set of characteristics is desired in the target model, one such extreme model can be generated by ‘warping’ the source model in a suitable manner. In the strictest sense, the technique of warping deforms one object to form another and is distinctly different from interpolation-based morphing technique, although warping can also be regarded as one of the morphing techniques in its broader sense of metamorphosis. It can be said, therefore, that the warping method is an

extrapolation technique which deforms a single model. One of the models generated from warping process can be used for a target model of a morphing process.

An extreme model can be generated using Laplacian coordinates  $L(s_i)$  which represents a difference between a vertex and the weighted average of its neighbours. In the field of computer graphics, this coordinates have been widely used for mesh deformations [63-68]. As was the case for parameterisation, each internal node of a surface can be defined by its neighbours after defining the boundary. Similarly, a new extreme boundary is generated by moving a specific point of an existing boundary to another position, and then the internal nodes are defined in accordance with the new boundary and convex combination coefficients obtained from the original mesh. An example of how an extreme target model can be generated from a source model is shown below.

The Laplacian coordinates are the simplest form of differential coordinates and have been usually utilised in handling a set of vertices for the mesh deformation. Fundamentally, the coordinates represent a vertex  $s_i$  by using a set of differentials  $L(s_i)$  between  $s_i$  and the weighted average of its neighbours  $s_{jk}$  in the form:

$$L(s_i) = s_i - \sum_{k=1}^{c_i} w_{ik} s_{jk} \quad (3-32)$$

This can be rewritten:

$$s_i = L(s_i) + \sum_{k=1}^{c_i} w_{ik} s_{jk} \quad (3-33)$$

where  $c_i$  is the number of neighbours of each  $s_i$ , and  $w_{ik}$  is the weighting factor for vertex  $s_{jk}$  with  $\sum_{k=1}^{c_i} w_{ik} = 1$ , and  $L$  is the Laplacian operator. Many weighting schemes can be applied in different ways taking into consideration the neighbouring structures.

One simple scheme is to set the inverse of the number of neighbours ( $1/c_i$ ) for the weights. Here is an example case with a vertex  $(0, 0)$  and two neighbours  $(1, 0)$  and  $(-1.5, 0)$ . Laplacian operators from Equation (3-32) for x-axis  $L_x$  and y-axis  $L_y$  would be:

$$L_x = 0 - (0.5 \times 1 + 0.5 \times (-1.5)) = 0.25 \quad (3-34)$$

$$L_y = 0 - (0.5 \times 0 + 0.5 \times 0) = 0 \quad (3-35)$$

And if it is assumed that a neighbour  $(1, 0)$  moves to  $(1, 2)$  and another neighbour  $(-1.5, 0)$  is fixed, the vertex  $(0, 0)$  from Equation (3-33) would be move to :

$$0.25 + (0.5 \times 1 + 0.5 \times (-1.5)) = 0$$

$$0 + (0.5 \times 2 + 0.5 \times 0) = 1$$

$$\rightarrow (0, 1)$$

0.25 and 0 are  $L_x$  and  $L_y$  respectively obtained from Equation (3-34) and (3-35). Figure 3.10 shows the result of this example.

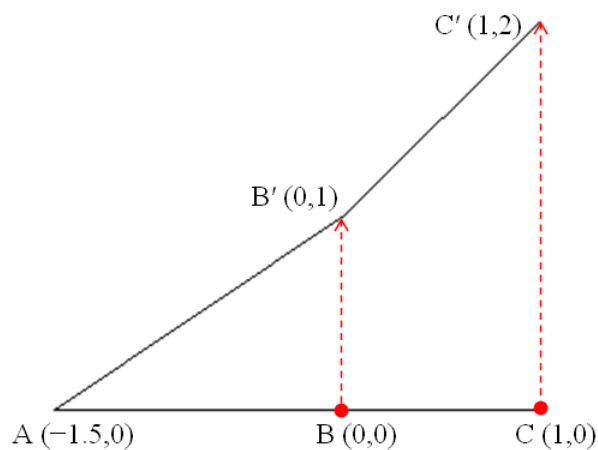


Figure 3.10 Generating a new boundary  $AB'C'$  from the boundary  $ABC$  by moving vertex  $C(1, 0)$  to vertex  $C'(1, 2)$ .

With a new boundary and convex combination coefficients of an original surface, a new surface, or internal nodes, can be generated with Equation (3-29) and (3-30) of Section 3.4.1.

### **3.6 CONCLUDING REMARKS**

In this Chapter, morphing was introduced including its concept and the methodology through its mathematical and geometric representation. Morphing is a geometric interpolation which transform an object into another. Curve morphing could transform 2D polygonal objects and curves in 2D or 3D, and it is closely related to surface morphing as well because a surface has boundaries of 3D curves. It was shown that, through morphing, various intermediate shapes between the source and the target models can be generated.

In this thesis, 3D triangular meshes are used for morphing 3D surfaces, and two methods of establishing correspondence between the vertices of the source and target models were introduced: regularising and merging. The regularising method generates new isomorphic triangular meshes by bisecting each edge of the triangular shape-patches repeatedly. The merging method produces a new mesh which has both mesh elements of the source and target models.

Although mesh quality problem such as absence of skinny triangles, has not been focused on in this thesis, it would be possible to easily avoid yielding skinny triangles with preparing non-skinny triangle patches from subdivision tasks in the regularising method. In the merging method, however, it would be necessary to employ an additional algorithm such as Delaunay triangulation algorithm in order to guarantee meshes of good quality. This work remains for future research.

Finally, a warping method was introduced for generating extreme models using Laplacian differential coordinates. This technique would be useful when there is no proper target model available.

# Chapter 4.

## APPLICATION OF MORPHING TO AIRFOIL DESIGN

---

### 4.1 GENERAL REMARKS

Before examining the application of morphing in the design of 3D surfaces, a simpler case of 2D curve morphing is explored in this Chapter. The airfoil shape is selected for this purpose, as such a shape can be expressed by the cross section shapes which are essentially 2D curves. First of all related publications and work on modelling airfoils are reviewed briefly before airfoils used for turbine blades are introduced. The Chapter continues with the discussion of how curve morphing can be used for modelling of the turbine airfoils. Multi-target morphing is introduced and its use is demonstrated through airfoil design. In addition, generating extreme airfoil shapes by using the warping method will be demonstrated.

### 4.2 A BRIEF REVIEW OF AIRFOILS

Airfoil is used as the cross section of wings, propellers or turbines. An airfoil-shaped body moving through a fluid produces lift-force in the perpendicular direction of motion, and this lift allows an airplane to fly, a yacht to sail and a helicopter to take off. The blades of wind/current turbines absorb the energy of the fluid passing through them and therefore do not act in exactly the same way as the lifting devices.

The performance of these devices is influenced by the characteristics of the airfoils, and thus it is important to adopt an airfoil suitable for their purpose and operating requirements.

Early airplanes, such as the first powered airplane designed by Wright brothers in 1903, Voisin's biplane in 1908, and Bleriot's monoplane which crossed the English Channel in 1909, all had thin airfoil sections along their wings. During the First World War, RAF-6 and RAF-15 were developed through numerous experiments in the United Kingdom. The theoretical study of an airfoil had been gradually organised and high performance airfoils have been designed since that time. Other notable airfoils developed in that time are Göttingen-387 designed at the University of Göttingen in 1919, and Clark Y designed by Virginus E. Clark in 1922. In 1924, Max Munk [69] introduced thin airfoil theory that relates angle of attack to lift, and he designed M-6 and M-12. The British RAF series airfoils were also updated using this theory. In the 1930s, the National Advisory Committee for Aeronautics (NACA), the predecessor of the present National Aeronautics and Space Administration (NASA), designed airfoils systematically, and developed NACA airfoil four-digit series. The first digit of the denotation indicates maximum camber as percentage of the chord, and the second digit describes the distance of maximum camber from the airfoil leading edge in tens of percentage of the chord. The last two digits indicate maximum thickness of the airfoil as percentage of the chord. The NACA had led the research on airfoil characteristics since that time, and also developed the five-digit series, including modified four-/five-digit, one series, six series, seven series, and eight series. Figure 4.1 shows some of airfoils mentioned above.

Recently, renewable energy has been attracting much attention in pursuit of sustainability and environmental friendliness, and many wind turbines and ocean current turbines have been developed and constructed. The overall performance of such turbines is highly dependent on the aerodynamic characteristics of the airfoil, and there has been some work on the characteristics of the airfoil for wind turbines and ocean current turbines. Griffiths [70] presented an outline of the general design procedure for a wind turbine sections with a review of how the aerodynamic

characteristics of the airfoil affect the power output, and found, not surprisingly, that airfoil characteristics have much influence on performance. Morcos [71] analysed

(a) Wright brothers' airplane airfoil



(b) RAF-15



(c) Göttingen-387



(d) Clark Y



(e) Munk M-6



(f) NACA four-digit series NACA 0015



Figure 4.1 Airfoil shapes.

performances of three highly idealised airfoil blade sections (flat-plate, symmetric and circular-arc airfoils) for wind turbines, and found that flat-plate and symmetric

airfoils are recommended for small and large wind turbines respectively, because the airfoils are operated at a wider range of tip-speed ratios than that of circular-arc airfoil. Kamoun *et al.* [72] developed an accurate airfoil analysis code for wind turbine applications based on singularity method. The results obtained have shown considerable agreement with experimental data at low angles of attack; however, the agreement deteriorates as the angle of attack increases. Perez and Pérez-Rojas [73] attempted to design rotor blade for ocean current turbines taking into account hydrodynamic considerations, and produced an airfoil section that is optimised for maximum lift with a minimum drag. However, research on those blades is still progressing, and there is room for design improvement in airfoils in order to obtain maximum output from the turbines.

### 4.3 AIRFOILS FOR TURBINES

In recent years, the focus of the search for green and sustainable energy has been on the sun, wind, ocean, bio-fuel and geothermal heat. Wind and ocean current especially have attracted much attention as their energy can be readily converted into a usable form by using turbines. Offshore wind and current farms have become favourite subjects of study in naval architecture and ocean engineering these days. However, it is probably fair to say that most of renewable energy technologies, including these two turbines, are at an early stage of development, and they have room for improvement. Here the morphing method will be applied to the design of the airfoil profiles for the ocean current and wind turbines.

The principles of generating energy from both turbines are very similar in that the kinetic energy of fluid flow is converted into mechanical energy. However, the density of the fluid and the speed of flow would be different, and therefore, the optimum shapes of the airfoils cannot be the same. A method of using this technique of rapidly generating airfoil forms in the optimisation process will be discussed in



Chapter 5. However, we need to establish the method of using the morphing technique for airfoil shape generation first.

#### 4.4 MORPHING AIRFOILS

As has been mentioned a number of times, airfoil profile is very important for a good performance of a turbine since the total thrust and torque of rotor depends on the lift and drag of the airfoil. In order to examine the feasibility of using morphing for generating airfoils, NACA 6409 airfoil (NACA 6409) and Eppler E1098 general aviation airfoil (Eppler 1098) are selected for the source and target models respectively here. As shown in Figure 4.2, there is quite a difference between the two models: NACA 6409 is very thin and Eppler 1098 is much thicker. In addition, other geometrical properties such as the location of maximum thickness and camber are also different, thus diverse characteristics of airfoils can be expected from morphing.

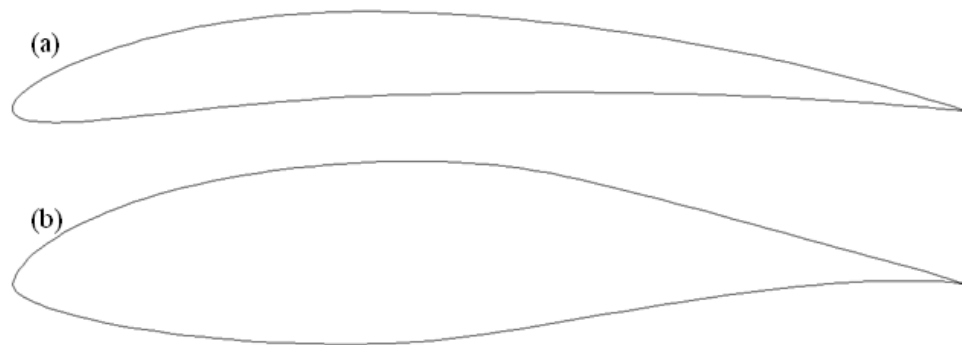


Figure 4.2 A source and a target models:  
(a) NACA 6409 for source; (b) Eppler 1098 for target.

The two original airfoil models have different number of nodes representing the shapes; for example, the source model has 100 nodes and the target 120. As mentioned in Chapter 3, both models should be made to have identical numbers of nodes in order to establish a correspondence. For this task, the same number of nodes can be set up by interpolating between existing nodes (e.g. by using cubic spline), or an entirely new number of nodes can be generated using an airfoil analysis program,

Xfoil [74]. Usually 100 nodes are enough for defining an airfoil, and each model was made to have 160 nodes using Xfoil program in this research. In order to have the same structure of data, the order of data for both models should be identical as well. This task can be carried out using Xfoil or text editor program, UltraEdit [75].

Consequently, the nodes for both source and target models were made to start from the trailing edge of the airfoil, through the upper part to the leading edge and through lower part back to the starting node at all times maintaining vertex correspondence at least for key points, e.g. trailing edge, leading edge and so on. With the source and the target models thus prepared, the morphing between two models can finally take place by interpolating correspondent nodes according to Equation (3-1).

The code for 2D morphing procedure has been developed using C++. This code does not include the procedure for establishing a correspondence since these tasks can be easily carried out using Xfoil or UltraEdit. For the input data of 2D morphing, thus, source and target models which have an identical data structure need to be defined with DAT files which have x-y coordinates. The generated intermediate models from this procedure are also formed with DAT files.

Through morphing, numerous intermediate models can be generated instantaneously, and Figure 4.3 illustrates some of these newly generated airfoils (b, c, d, e, f, g, h, i, j) form the NACA 6409 (a) and the Eppler 1098 (k) airfoils. Here, the morphing parameter is  $t \in [0, 1]$ , and the model at  $t = 0$  is equivalent to the source model (a) while the model at  $t = 1$  is equivalent to the target model (k). The results of morphing for the interpolation parameter  $t$  at the intervals of 0.1 are shown in this figure. It is not difficult to see that, since  $t$  can take on an infinite number of values between 0 and 1, an infinite number of these intermediate forms can be generated.

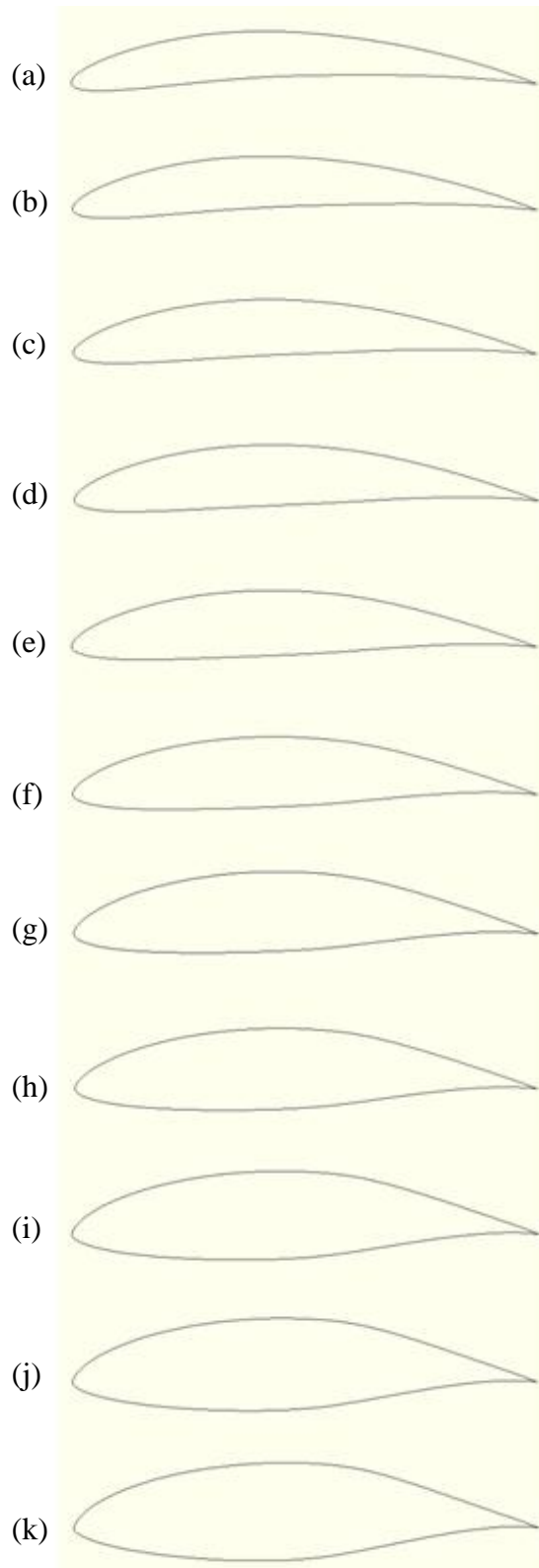


Figure 4.3 Original models and newly generated models: (a) source model ( $t = 0$ ); (b)  $t = 0.1$ ; (c)  $t = 0.2$ ; (d)  $t = 0.3$ ; (e)  $t = 0.4$ ; (f)  $t = 0.5$ ; (g)  $t = 0.6$ ; (h)  $t = 0.7$ ; (i)  $t = 0.8$ ; (j)  $t = 0.9$ ; (k) target model ( $t = 1$ ).

## 4.5 MULTI-TARGET MORPHING

The previous section presented how numerous airfoils can be generated from two original models through morphing. However, the new models are derived from these two models and are, therefore, predictable. With the two-model morphing so far discussed it is impossible to generate models which are entirely different from both of the parents. One could argue that the lack of possible mutation of forms is a key drawback of this technique in generating forms for optimisation process.

In order to mitigate this drawback and allow wider range of shapes to be generated, more than two models can be used as parents. For lack of better terminology, we could call this ‘multi-target morphing’. With the multi-target morphing proposed here, it is possible to use as many target models as required, thus making the technique much more versatile and providing the designer one more handle. Indeed it can be said that morphing in general uses any number (provided there are two or more) of parent models and the single-target morphing is a special case of morphing where there are just two parent models. Figure 4.4 illustrates the concept of multi-target morphing.

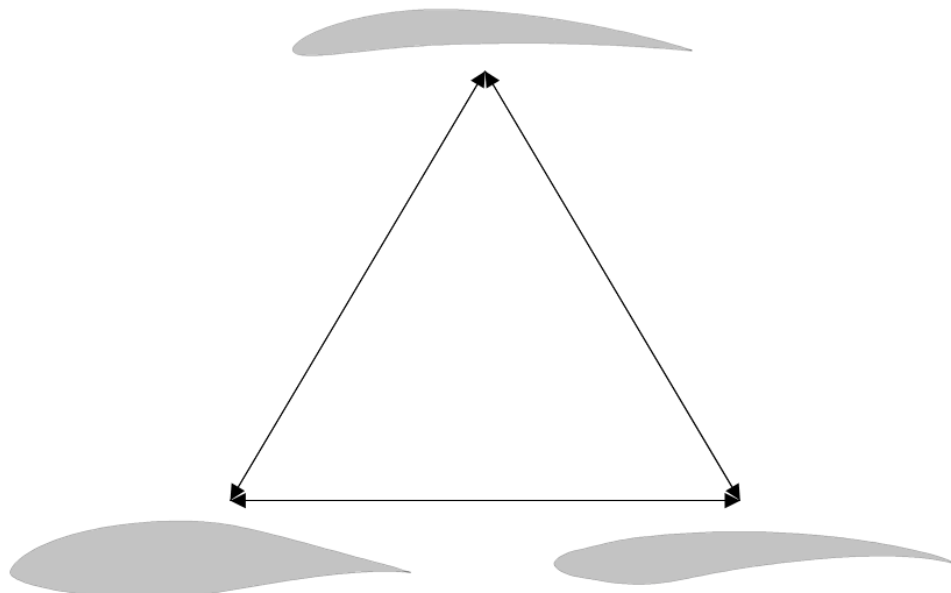


Figure 4.4 Concept of multi-target morphing.

The algorithm for the general morphing  $n$  number of target model is extended from the single-target morphing algorithm, and the morphing parameters are equivalent to weights of the target models  $(t_1, t_2, \dots, t_n)$ . The generated intermediate models can be expressed as follows:

$$M(t_1, t_2, \dots, t_n) = t_0 \cdot R_0 + t_1 \cdot R_1 + t_2 \cdot R_2 + \dots + t_n \cdot R_n \quad (4-1)$$

where,  $t_0$ : weight for a source model,  $1 - (t_1 + t_2 + \dots + t_n)$

$R_0$ : source model

$t_1, t_2, \dots, t_n$ : weights for target models,  $t_1 + t_2 + \dots + t_n \leq 1$

$R_1, R_2, \dots, R_n$ : target models

A case study is conducted with a source model and two target models. The source airfoil model is NACA 6409 airfoil (NACA 6409), and the first target model is Eppler E1098 general aviation airfoil (Eppler 1098) as were used for the first case study in Section 4.4. In addition, Wortmann FX 082-512 airfoil (FX 082512) is newly added as the second target model. The three parent models are illustrated in Figure 4.5 (a), (b) and (c).

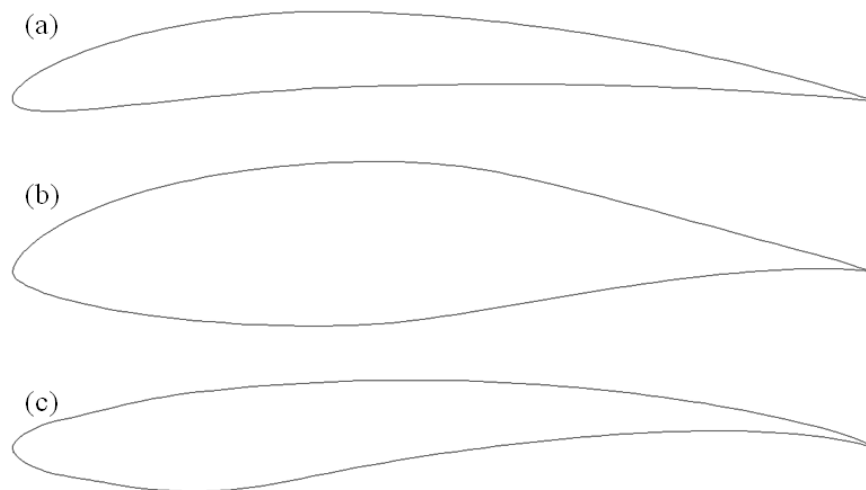


Figure 4.5 A source and two target models used for multi-target morphing: (a) source model: NACA 6409; (b) target model 1: Eppler 1098; (c) target model 2: FX 082512.

Just as in the single-target morphing discussed in Section 4.4, a proper correspondence should be established for the multi-target morphing. The number of nodes of the three models is made to be the same using Xfoil program. Moreover, the data structure of the node of each model needs to be aligned to the others, and in this example they were made to start from the trailing edge of the airfoil through the upper part to the leading edge returning through the lower part to the trailing edge.

The morphing parameters of the two-target morphing,  $t_1$  and  $t_2$ , are equivalent to weights of the target models in the interpolation, where,  $t_1 + t_2 \leq 1$ . The weight of the source model ( $t_0$ ) can be calculated by subtracting sum of weights of target models from one, i.e.

$$t_0 = 1 - (t_1 + t_2)$$

For example, if the weight of the first target model ( $t_1$ ) is 0.2 and the weight of the second target model ( $t_2$ ) is 0.3, the weight of the source model ( $t_0$ ) would be 0.5. Figure 4.6 shows some of the models generated in this manner.

As expected the intermediate models take on the characteristics of the parents to the extent indicated by the weighting factors applied. Thus, various airfoil shapes can be generated by varying the combinations of weights, and it is also expected that a myriad of models can be produced by adding more target models.

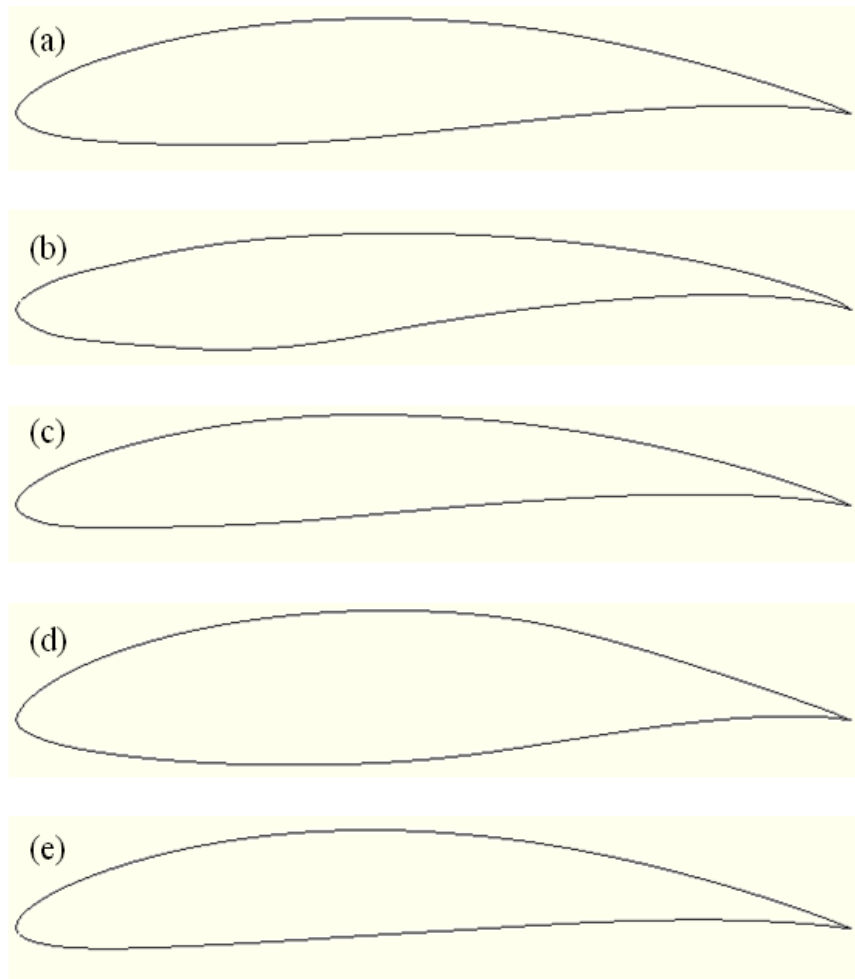


Figure 4.6 Models generated from multi-target morphing:  
(a)  $t_0 : 0.34, t_I : 0.33, t_2 : 0.33$ ; (b)  $t_0 : 0.10, t_I : 0.05, t_2 : 0.85$ ; (c)  $t_0 : 0.50, t_I : 0.15,$   
 $t_2 : 0.35$ ; (d)  $t_0 : 0.15, t_I : 0.75, t_2 : 0.10$ ; (e)  $t_0 : 0.65, t_I : 0.25, t_2 : 0.10$ .

## 4.6 GENERATING EXTREME AIRFOILS FOR USE AS PARENTS

It was shown that an infinite number of intermediate airfoils can be generated by morphing between the source and target models, and the newly generated shapes take on the compound characteristics of the parents. If, on the other hand, there is no model available suitable for the target, an extreme model can be generated at or beyond the boundary of realistic feasibility by deforming (warping) the source model.

As discussed in Chapter 3, this can be achieved by applying Laplacian differential coordinates. Fundamentally, the method establishes relationship between the vertices using the coordinates, and relocates the vertex set to new positions whilst preserving this relationship. The target of the new model can be established by replacing a ‘guide point’ of the mesh to a new position, and then the other vertices are transferred to their respective new positions according to the Laplacian differential coordinates.

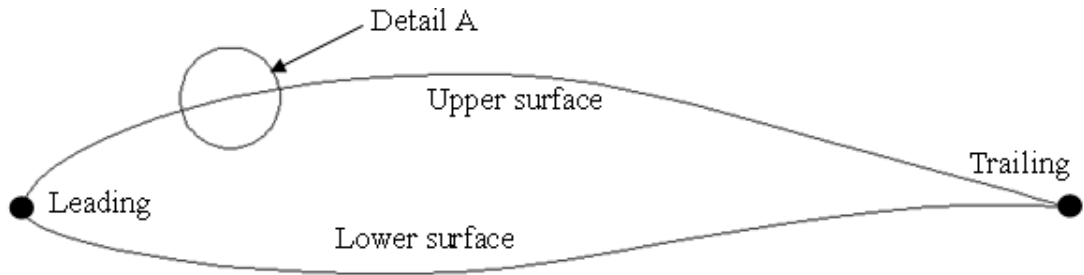
The Laplacian differential coordinates represent a point  $s_i$  by using a set of differentials  $L(s_i)$  between  $s_i$  and the average of its neighbours  $s_{jk}$ :

$$L(s_i) = s_i - \sum_k w_{ik} s_{jk} \quad (4-2)$$

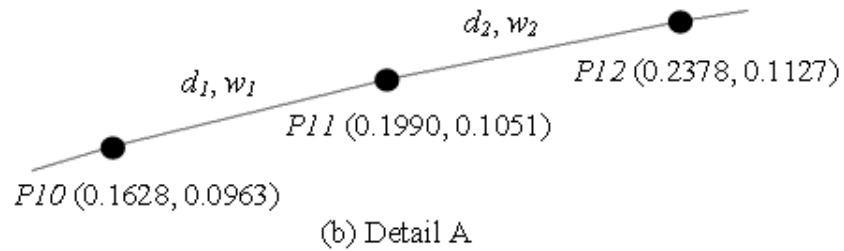
where  $w_{ik}$  is the weighting factor for point  $s_{jk}$  with  $\sum_k w_{ik} = 1$ , and the  $L$  is Laplacian operator. Many weighting schemes can be applied in different ways taking into consideration the neighbouring structures. One simple scheme is to set the inverse of the number of neighbours for the weights. Since a vertex of a polygonal curve such as airfoil shapes always has two neighbours, the weights would be always 0.5 with this scheme. Another simple scheme for a polygonal curve is to use the distances between a vertex and its two neighbours, in which the closer neighbour has greater weight.

Figure 4.7 illustrates Eppler 1908 airfoil prepared as the original model for generating a new airfoil, and the region of the 10<sup>th</sup>, 11<sup>th</sup> and 12<sup>th</sup> vertices is enlarged in Figure 4.7 (b) in order to show an example calculation of the Laplacian operator.





(a) Original airfoil model for generating a new airfoil (Eppler 1098)



(b) Detail A

Figure 4.7 Original airfoil model for generating a new airfoil.

In order to calculate the Laplacian operator for the vertex  $P11$  in Figure 4.7 (b), the distances between the vertex  $P11$  and its neighbours  $P10$  and  $P12$  should be obtained so as to calculate the weighting factors of each neighbouring vertex. Since the distances  $d_1$  and  $d_2$  are 0.0373 and 0.0395 respectively, the weighting factors would be:

$$\begin{aligned}
 w_1 &= \frac{d_2}{d_1 + d_2} \\
 &= \frac{0.0395}{0.0373 + 0.0395} = 0.5143
 \end{aligned}
 \tag{4-3}$$

$$\begin{aligned}
 w_2 &= \frac{d_1}{d_1 + d_2} \\
 &= \frac{0.0373}{0.0373 + 0.0395} = 0.4857
 \end{aligned}
 \tag{4-4}$$

Then, according to Equation (4-2) the Laplacian differential coordinates of the vertex  $P11$  would be:

x-component:

$$\begin{aligned}
 L_x &= s_{i-x} - \sum_k w_{ik} s_{jk-x} \\
 &= 0.1990 - (0.5143 \times 0.1628 + 0.4857 \times 0.2378) \\
 &= -0.0002
 \end{aligned} \tag{4-5}$$

y-component:

$$\begin{aligned}
 L_y &= s_{i-y} - \sum_k w_{ik} s_{jk-y} \\
 &= 0.1051 - (0.5143 \times 0.0963 + 0.4857 \times 0.1127) \\
 &= 0.0008
 \end{aligned} \tag{4-6}$$

In a similar manner, the Laplacian operator of the other vertices can be calculated. Since the Laplacian operators are based on the original model, these operators should be modified for the new target model. The modified operators can be obtained by multiplying the ratio of difference between the original position of guide point and its new position to the operator. For example, when the vertex *P10* (0.1628, 0.0963) is selected as the guide point and is moved to (0.1628, 0.15), the x-component of the modified operator does not change; however, the y-component would be the original Laplacian operator multiplied by  $\frac{0.15}{0.0963} = 1.5576$ .

It was mentioned in Chapter 3 that internal vertices of a 3D surface mesh are parameterised onto a 2D plane using the known boundary nodes and the convex combination coefficients. The vertices of the airfoil can be similarly transferred to new positions with known boundary nodes, weighting factors and Laplacian operators. In this thesis, it is assumed that the leading and trailing edges of the airfoil are fixed at the same positions, so the known boundary nodes consist of the positions of the leading and trailing edges and the new position of the guide point. By fixing the leading and trailing edges, it is also possible to transform the upper surface and the lower surface of the airfoil separately. Let us take an example case where the number of upper surface nodes is 10 and the 4<sup>th</sup> node is the guide point which is to be moved to a new position as shown in Figure 4.8.

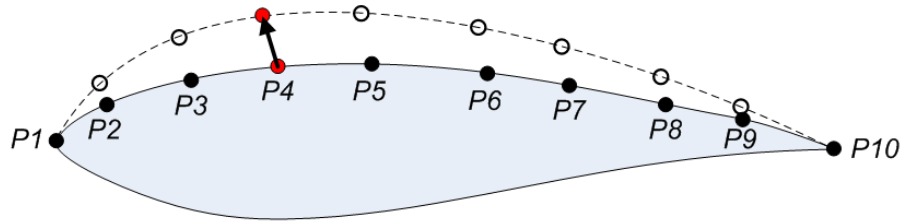


Figure 4.8 Generating a new upper surface by replacing a guide point ( $P4$ ).

When the guide point  $P4$  is moved to a new position, the left side of the guide point,  $P1$ ,  $P2$ , and  $P3$ , and the right side,  $P5$ ,  $P6$ , ...  $P10$  might require different modified Laplacian operator. Thus, x-components of Equation (4-2) for the left side and the right side about the guide point  $P4$  are:

For the left side points,  $P2$  and  $P3$ :

$$\begin{aligned} L_{2x} &= P_{2x} - w_{2,1}P_{1x} - w_{2,3}P_{3x} \\ L_{3x} &= P_{3x} - w_{3,2}P_{2x} - w_{3,4}P_{4x} \end{aligned} \quad (4-7)$$

For the right side points,  $P5$ ,  $P6$ ,  $P7$ ,  $P8$  and  $P9$ :

$$\begin{aligned} L_{5x} &= P_{5x} - w_{5,4}P_{4x} - w_{5,6}P_{6x} \\ L_{6x} &= P_{6x} - w_{6,5}P_{5x} - w_{6,7}P_{7x} \\ L_{7x} &= P_{7x} - w_{7,6}P_{6x} - w_{7,8}P_{8x} \\ L_{8x} &= P_{8x} - w_{8,7}P_{7x} - w_{8,9}P_{9x} \\ L_{9x} &= P_{9x} - w_{9,8}P_{8x} - w_{9,10}P_{10x} \end{aligned} \quad (4-8)$$

By substituting y-components of the points and the Laplacian operators to the Equations (4-7) and (4-8), the y-components of the Equations for both sides can also be obtained with ease.

In order to transform the model by moving the guide point to the new position, the modified Laplacian operator  $L'$  and new position of the guide point  $P_{4x}'$  should be substituted to the original operator  $L$  and the original guide point  $P_{4x}$  respectively. By

grouping the unknown terms to the left side and the known terms to the right after the substitution, Equations (4-7) and (4-8) can be rewritten in the following form:

$$\begin{aligned} P_{2x} - w_{2,3}P_{3x} &= w_{2,1}P_{1x} + L_{2x}' \\ P_{3x} - w_{3,2}P_{2x} &= w_{3,4}P_{4x}' + L_{3x}' \end{aligned} \quad (4-9)$$

$$\begin{aligned} P_{5x} - w_{5,6}P_{6x} &= w_{5,4}P_{4x}' + L_{5x}' \\ P_{6x} - w_{6,5}P_{5x} - w_{6,7}P_{7x} &= L_{6x}' \\ P_{7x} - w_{7,6}P_{6x} - w_{7,8}P_{8x} &= L_{7x}' \\ P_{8x} - w_{8,7}P_{7x} - w_{8,9}P_{9x} &= L_{8x}' \\ P_{9x} - w_{9,8}P_{8x} &= w_{9,10}P_{10x}' + L_{9x}' \end{aligned} \quad (4-10)$$

This can be written in a matrix form as:

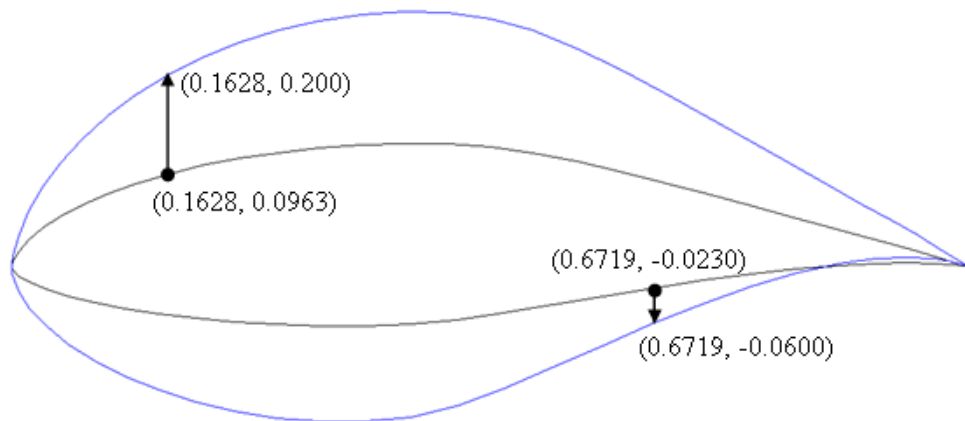
$$\begin{bmatrix} 1 & -w_{2,3} \\ -w_{3,2} & 1 \end{bmatrix} \begin{bmatrix} P_{2x} \\ P_{3x} \end{bmatrix} = \begin{bmatrix} w_{2,1} & 0 \\ 0 & w_{3,4} \end{bmatrix} \begin{bmatrix} P_{1x} \\ P_{4x}' \end{bmatrix} + \begin{bmatrix} L_{2x}' \\ L_{3x}' \end{bmatrix} \quad (4-11)$$

$$\begin{bmatrix} 1 & -w_{5,6} & 0 & 0 & 0 \\ -w_{6,5} & 1 & -w_{6,7} & 0 & 0 \\ 0 & -w_{7,6} & 1 & -w_{7,8} & 0 \\ 0 & 0 & -w_{8,7} & 1 & -w_{8,9} \\ 0 & 0 & 0 & -w_{9,8} & 1 \end{bmatrix} \begin{bmatrix} P_{5x} \\ P_{6x} \\ P_{7x} \\ P_{8x} \\ P_{9x} \end{bmatrix} = \begin{bmatrix} w_{5,4} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & w_{9,10} \end{bmatrix} \begin{bmatrix} P_{4x}' \\ P_{10x}' \end{bmatrix} + \begin{bmatrix} L_{5x}' \\ L_{6x}' \\ L_{7x}' \\ L_{8x}' \\ L_{9x}' \end{bmatrix} \quad (4-12)$$

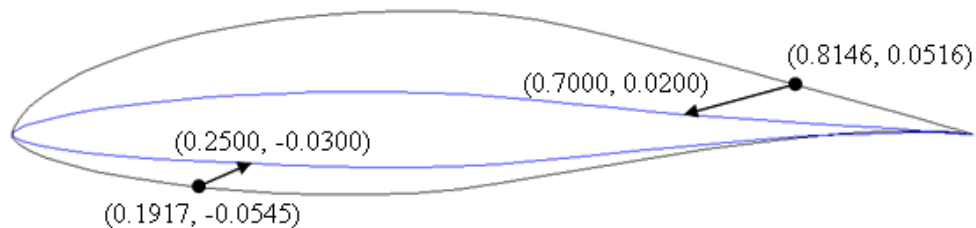
By using Gauss-elimination method, the unknown values  $P_{2x}$ ,  $P_{3x}$ ,  $P_{5x}$ ,  $P_{6x}$ ,  $P_{7x}$ ,  $P_{8x}$ , and  $P_{9x}$  are finally obtained, and y-components of the points can also be calculated in a similar way.

Figure 4.9 illustrates newly generated airfoils from Eppler 1908 model according to new positions of the guide point. The model in black is the original and the one in blue lines is the newly generated model. The upper surface consists of 33 nodes while the lower surface is modelled with 30 nodes. In Figure 4.9 (a), a thicker airfoil

is generated from the original model by moving the guide points  $(0.1628, 0.0963)$  and  $(0.6719, -0.0230)$  to  $(0.1628, 0.200)$  and  $(0.6719, -0.0600)$  for the upper and lower side surfaces. Similarly, Figure 4.9 (b) shows a newly generated airfoil thinner than the original one by moving the guide points  $(0.8146, 0.0516)$  and  $(0.1917, -0.0545)$  to  $(0.7000, 0.0200)$  and  $(0.2500, -0.0300)$  respectively for each side. As shown in Figure 4.9, any vertex can be chosen as a guide point, and the guide point can move to anywhere. Thus, it is possible to produce diverse models based on the original shape by selecting the new position of the guide point.



(a) Generating a new thicker airfoil from Eppler 1908 airfoil.



(b) Generating a new thinner airfoil from Eppler 1908 airfoil.

Figure 4.9 Generating new extreme airfoils using Laplacian coordinates.

This technique will also be useful when certain desirable characteristics of any of the parent models are to be retained as much as possible without being diluted too much during the interpolation process. Extreme models can be generated by deforming the original parent models so that the desired characteristics are exaggerated.

## **4.7 CONCLUDING REMARKS**

This Chapter briefly introduced airfoils of turbine blades and demonstrated a practical application of the morphing technique to the design of airfoils.

2D morphing method was used with two somewhat different shapes selected for the source and target airfoils, and it produced potentially an infinite number of feasible forms more or less instantaneously. In order to give more diversity to the design, multi-target morphing was also introduced. Depending on the combination of the weighting factors, shapes of many different characteristics can be generated through this method. In addition, a warping method based on Laplacian differential coordinates was used to create new extreme airfoils, adding a little more power to the morphing technique as a design tool.

# **Chapter 5.**

## **APPLICATION OF MORPHING TECHNIQUE IN AIRFOIL DESIGN OPTIMISATION**

---

### **5.1 GENERAL REMARKS**

The method of generating an infinite number of airfoil forms more or less instantaneously was discussed in Chapter 4. There are many strategies of optimisation. However, a common feature of design optimisation of any product or system involves generating a feasible design followed by the performance evaluation of the design. In many cases this procedure has to be repeated many times before an optimum or near-optimum solution is reached. Often both the design generation and analysis are long-winded, thus making the optimisation a time consuming task. This sometimes obliges the designers to limit the number of designs to examine, leading to less perfect optimisation.

By making the design generation task quick and instantaneous, the overall time required for optimisation can be reduced. This benefit will become more apparent, the more complex the design. Even for relatively simple forms it opens up the possibility of automatic optimisation. One could also imagine an ‘on-line’ optimisation process where the designer quickly picks a design and enquire how its performance is likely to be. With the fast improving numerical analysis technique and improving computer power, this is not just a fanciful notion.

This chapter will demonstrate the application of design generation with morphing to the process of optimisation of foil forms. The performance of the generated airfoils is analysed using Xfoil. This program is a low Reynolds number airfoil analysis code, and it estimates lift and drag coefficients of the airfoils. The case study covers the airfoils for wind turbine and ocean current turbine blades.

## 5.2 BASIC PRINCIPLES OF AIRFOILS

When an object is in motion, it is said to have kinetic energy, and the kinetic energy can be given by the following equation:

$$E_k = \frac{1}{2} \cdot m \cdot v^2 \quad (5-1)$$

where  $m$  and  $v$  denote mass (kg) and speed (m/s).

Wind and ocean current are the motions of air and water respectively so that they also have kinetic energies. When the fluid flows with speed  $v$  through a turbine whose blade radius is  $R$ , the turbine can be illustrated as shown in Figure 5.1. Area of the rotor sweep is a circular disk and the area  $A$  will be  $\pi \cdot R^2$ . The volume of the fluid passing through the rotor in time  $\delta t$  can be found by multiplying the area with the thickness  $\delta x = v \cdot \delta t$ . The mass of fluid will therefore be  $\rho \cdot A \cdot \delta x$ , or  $\rho \cdot A \cdot v \cdot \delta t$  where  $\rho$  is fluid density ( $\text{kg/m}^3$ ) and  $t$  is time (s).

Thus, the total amount of kinetic energy that passes through a circular disk in  $\delta t$  is:

$$\delta(E_k) = \frac{1}{2} \cdot (\rho \cdot A \cdot v \cdot \delta t) \cdot v^2 = \frac{1}{2} \cdot \rho \cdot A \cdot v^3 \cdot dt \quad (\text{Joules}) \quad (5-2)$$



The power in the wind  $P_w$  is the time derivative of the kinetic energy, and is written as:

$$P_w = \frac{\delta(E_k)}{\delta t} = \frac{1}{2} \cdot \rho \cdot A \cdot v^3 \quad (\text{watt}) \quad (5-3)$$

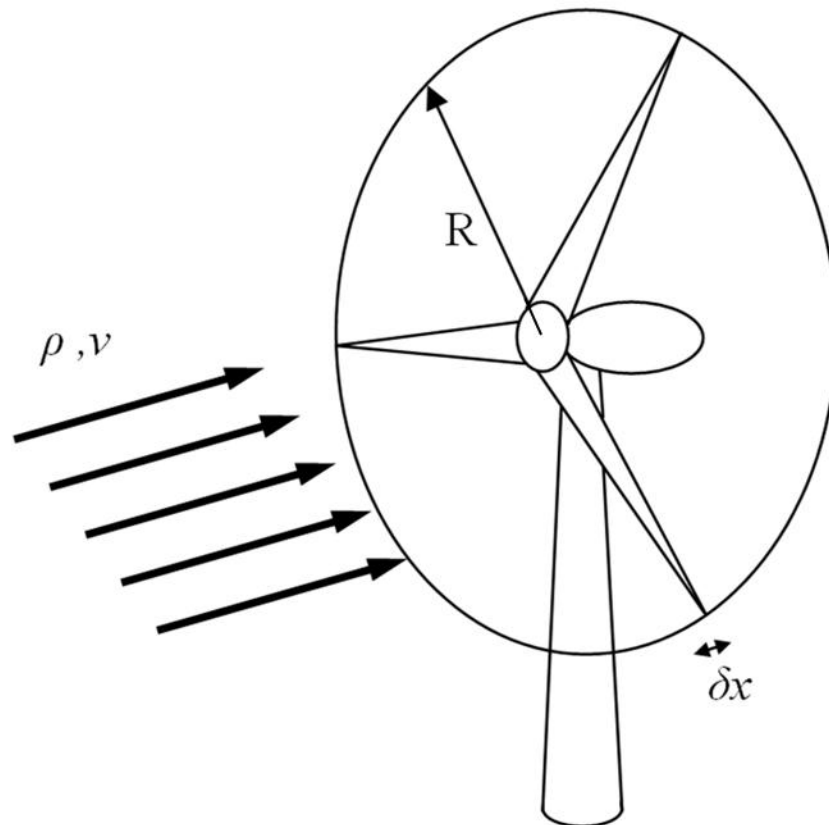


Figure 5.1 Fluid flow and circular disk of turbine.

As shown in Equation (5-3), the density of fluid and its speed have a direct influence on the power that can be generated by a turbine for a given turbine size. It is known that the seawater density at 10 °C is approximately 1023 kg/m<sup>3</sup>, and the air density at 20 °C is approximately 1.205 kg/m<sup>3</sup>. Although the density of water is much higher than air, the wind speed is much higher than that of ocean current. Therefore, one cannot argue one turbine is better or more efficient than the other. However, the important point is that proper shapes of airfoils should be chosen for the blades in order to maximise the power take-off for each environment.

It will be very nice if one could obtain all the power available from the wind or current as shown in Equation (5-3), but this is not possible for a variety of reasons. In 1919, Betz [76] proved that a wind turbine cannot extract power beyond 59.26% from the available power. In practice, it is known that turbines are well below the Betz Limit with possible values of 0.35 ~ 0.45 considering electrical and mechanical losses, and energy loss from generating wake or cavitation. This efficiency of power is referred as the power coefficient  $C_p$  and can be calculated by dividing the electricity produced by the power available in the fluid:

$$C_p = \frac{\text{Electricity produced by turbine}}{\text{The power available in fluid}} = \frac{W}{\frac{1}{2} \cdot \rho \cdot A \cdot v^3} \quad (5-4)$$

Conversely, the electrical power output can be obtained by multiplying the available power in the fluid with the power coefficient. Despite the low power coefficient, the available power is still one of the most important factors determining the power output. The fluid flow applies torque on the turbine and this is converted to electrical power. The torque is related to lift ( $L$ ) and drag ( $D$ ) of the blade as shown in Figure 5.2.

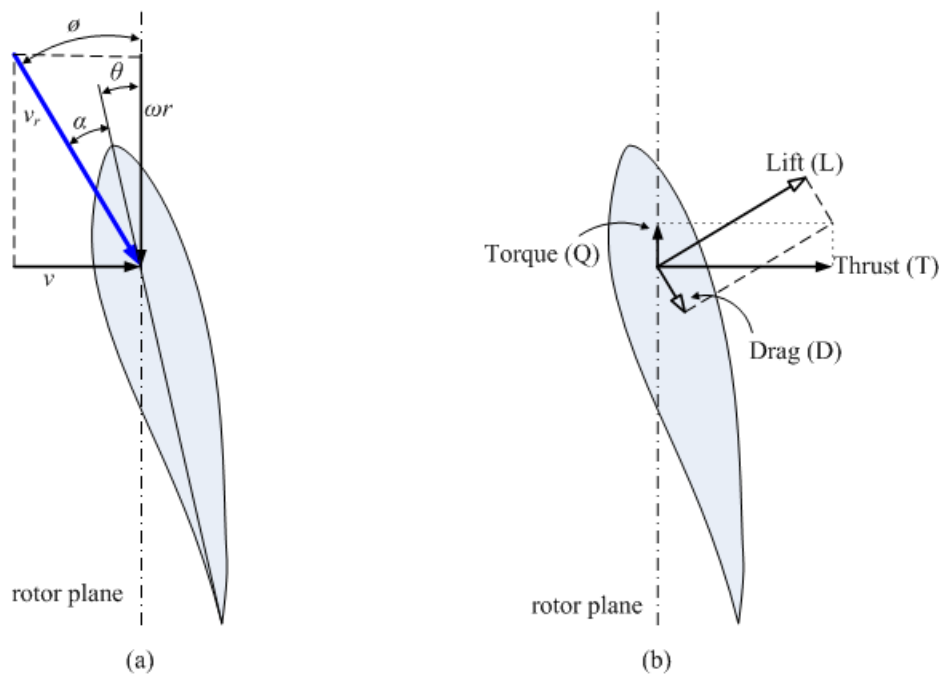


Figure 5.2 Velocity (a) and forces (b) for an airfoil.

The elemental lift ( $dL$ ) and drag ( $dD$ ) depend on the airfoil shape and angle of attack, and can be represented by the airfoil element and sectional coefficients of lift ( $C_l$ ) and drag ( $C_d$ ):

$$dL = \frac{1}{2} \cdot \rho \cdot v_r^2 \cdot C_l \cdot c \cdot dr \quad (5-5)$$

$$dD = \frac{1}{2} \cdot \rho \cdot v_r^2 \cdot C_d \cdot c \cdot dr \quad (5-6)$$

where  $\rho$ ,  $r$ ,  $v_r$ , and  $c$  denote density of fluid, radius of the rotor, relative fluid speed at the radius  $r$  of the rotor, and chord length respectively. The relative fluid speed  $v_r$  can be calculated as  $v_r = \sqrt{v^2 + (\omega r)^2}$ , where  $v$  and  $\omega r$  denote fluid flow speed and linear speed of the rotor element at radius  $r$  due to blade rotation respectively.

With the use of the lift and drag from the Equation (5-5, 5-6), the total thrust ( $dT$ ) and torque ( $dQ$ ) on  $N$  blade elements of a turbine can be calculated as follows:

$$dT = \frac{1}{2} \cdot \rho \cdot v_r^2 \cdot (C_l \cdot \cos\phi + C_d \cdot \sin\phi) \cdot c \cdot N \cdot dr \quad (5-7)$$

$$dQ = \frac{1}{2} \cdot \rho \cdot v_r^2 \cdot (C_l \cdot \sin\phi - C_d \cdot \cos\phi) \cdot c \cdot N \cdot r \cdot dr \quad (5-8)$$

where,  $\phi$  denotes the incidence angle of the relative fluid speed in respect to the rotor plane as shown in Figure 5.2 (a). The total thrust and torque of the rotor can be calculated by integrating Equations (5-7, 5-8) respectively. The thrust and torque depend on the sectional coefficient of lift ( $C_l$ ) and drag ( $C_d$ ), and the lift should be maximised and the drag should be minimised in order to maximise the total torque ( $dQ$ ) as shown in Equations (5-7, 5-8). Therefore, it is important to design airfoil suited to specific environment such as fluid density and flow speed in order to increase all the power available.

### 5.3 AIRFOILS FOR WIND TURBINE

As mentioned in the previous section, lift and drag are the key factors which determine the performance of a turbine airfoil. In particular the lift to drag ratio is the main performance indicator and, the larger this ratio, the better the performance.

An NREL airfoil family (S826, S825, S814 and S815) [77] for large blades with thick airfoil, designed by the National Renewable Energy Laboratory are benchmarked in this study. Figure 5.3 shows the airfoils. The design specifications of the airfoils are shown in Table 5.1.

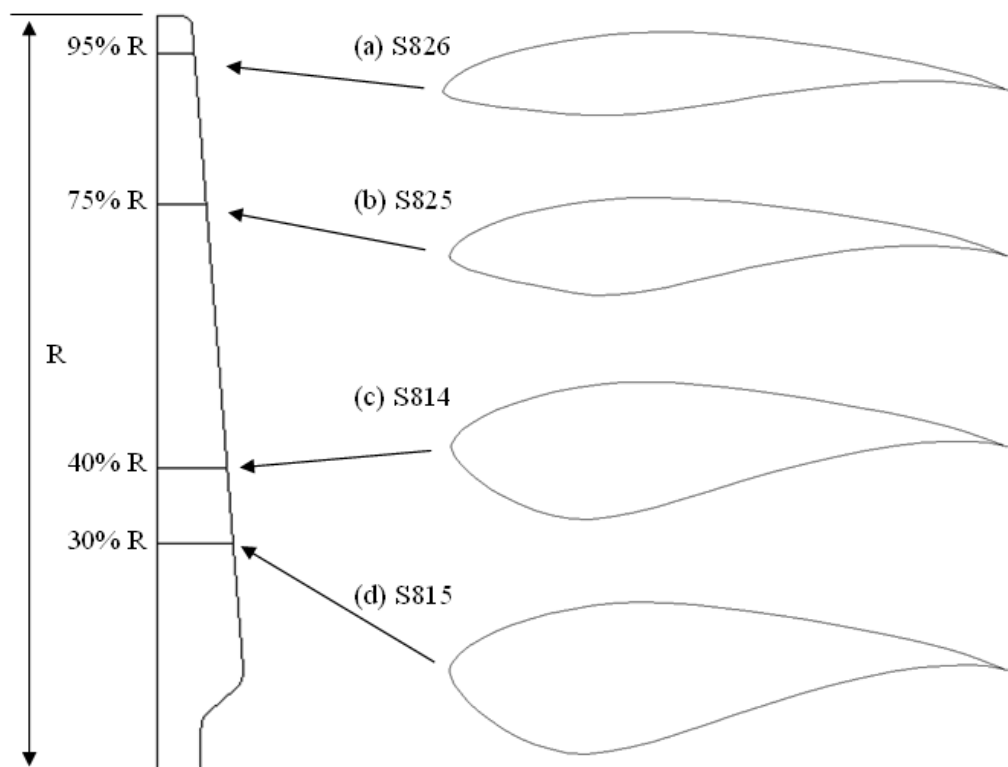


Figure 5.3 NREL airfoils (S826, S825, S814 and S815).

Table 5.1 Design specification.

<i>Airfoil</i>	<i>r/R</i>	<i>Re No. (×10<sup>6</sup>)</i>	<i>t/c</i>
<b>S826</b>	0.95	1.5	0.140
<b>S825</b>	0.75	2.0	0.170
<b>S814</b>	0.40	1.5	0.240
<b>S815</b>	0.30	1.2	0.260

In Table 5.1  $r/R$  denotes the position of the airfoil concerned on the blade as a fraction of the blade radius, and  $t/c$  is the ratio of the maximum thickness to chord length of the airfoil.

In this study, three methods are applied to generate airfoils. The first method is just to change the thickness of a source airfoil model by scaling y-coordinates so as to have a specific ratio of the maximum thickness to chord length. NACA 6409, Eppler E1098 and Gottingen 652 airfoils are used for the source airfoils for this method, and they are modified to each specific thickness to chord length ratio ( $t/c$ ) corresponding to each target airfoil. The second and the last methods are to generate airfoil models through single-target morphing and multi-target morphing respectively. For the single-target morphing, NACA 6409 (N6409) and Eppler E1098 airfoils are used for the source and the target models respectively, and Gottingen 652 (G652) airfoil is used for the additional target model for the multi-target morphing. The newly generated airfoils are also modified to the specific thickness ratio ( $t/c$ ) corresponding to each target airfoil.

The process of finding the optimum, or at least a near-optimum, airfoil for the particular position along the blade as discussed above will be illustrated by the case of 95% radius below before summarising the results obtained for the rest of the positions.

### **An Example Optimisation at 95% radius**

The S826 airfoil was originally designed for operation in the region of Reynolds number  $1.5 \times 10^6$  and the maximum thickness-chord length ratio ( $t/c$ ) of 0.140, as shown in Table 5.1. Thus, the single airfoils (N6409, E1098 and G652), the maximum thickness-chord length ratios ( $t/c$ ) of which are 0.090, 0.190 and 0.171 respectively, were converted to the  $t/c$  of 0.140 for this study. Through the single-target morphing between N6409 and E1098, 19 new intermediate models were generated at the morphing parameter intervals of 0.05. Moreover, 261 models were







generated from the multi-target morphing (N6409, E1098 and G652) at the morphing parameter intervals of 0.05. All the airfoils newly generated through morphing were converted to the  $t/c$  of 0.140, and the lift-drag ratio of each airfoil was estimated using Xfoil program. With the result of the estimation thus obtained, the models that show the maximum lift-drag ratio can be found.

At the Reynolds number of  $1.5 \times 10^6$ , the maximum lift-drag ratio of S826 was estimated at 151.777. In the same condition, the maximum lift-drag ratio of the converted N6409, E1098 and G652 were estimated at 147.209, 160.015 and 167.741 respectively. The single-target morphing between N6409 and E1098, produced a maximum performance at the morphing parameter of 0.45, and its lift-drag ratio was 181.859. The multi-target morphing with N6409, E1098 and G652 airfoils, on the other hand, gave a maximum performance model at the morphing parameter set of (0.55, 0.40, 0.05), where its lift-drag ratio was 182.370.

Even though no formal optimisation methodology was used and, thus it is likely that a better performance model could be obtained, the result of this little exercise is impressive enough. The best performing model from the single-target morphing has a significantly higher lift to drag ratio, and that from the multi-target morphing is marginally better still.

The profile and the maximum lift-drag ratio of each airfoil are shown in Table 5.2, and the lift-drag ratios are illustrated in Figure 5.4. In table 5.2, model S826 is the targeted airfoil of the test, and N6409-0.140, E1098-0.140 and G652-0.140 denote converted N6409, E1098 and G652 to the  $t/c$  of 0.140. M0.45 denotes the morphed model at the morphing parameter of 0.45. Similarly M0.55\_0.4\_0.05 denotes the model obtained from the multi-target morphing at the morphing parameter set of (0.55, 0.4 and 0.05).

Table 5.2 Comparison of the maximum lift/drag of various airfoils at 95% radius.

Model	Profile	Max Lift/Drag
(a) S826		151.777
(b) N6409-0.140		147.209
(c) E1098-0.140		160.015
(d) G652-0.140		167.741
(e) M0.45		181.859
(f) M0.55_0.4_0.05		182.370

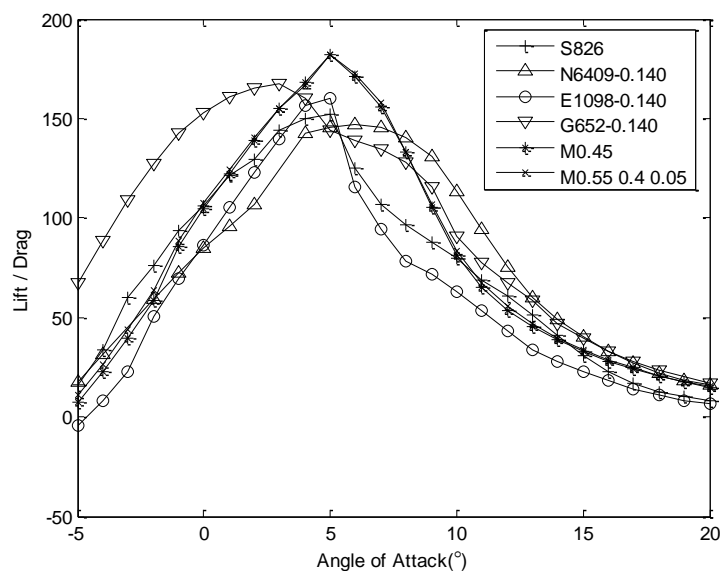








Figure 5.4 Lift drag ratio of airfoils benchmarked S826 ( $R_n = 1.5 \times 10^6$ ).

### Summary Results of Optimisation

The procedure used above is repeated for  $r/R = 0.3, 0.4$  and  $0.75$  taking into account the foil model,  $Re$  and  $t/c$  values specified. The results are given in Tables 5.3 – 5.5 and Figures 5.5 – 5.7.

Table 5.3 Comparison of the maximum lift/drag of various airfoils at 75% radius.

Model	Profile	Max Lift/ Drag
(a) S825		131.018
(b) N6409-0.170		141.098
(c) E1098-0.170		176.714
(d) G652-0.170		141.338
(e) M0.70		179.408
(f) M0.0_0.65_0.35		181.076

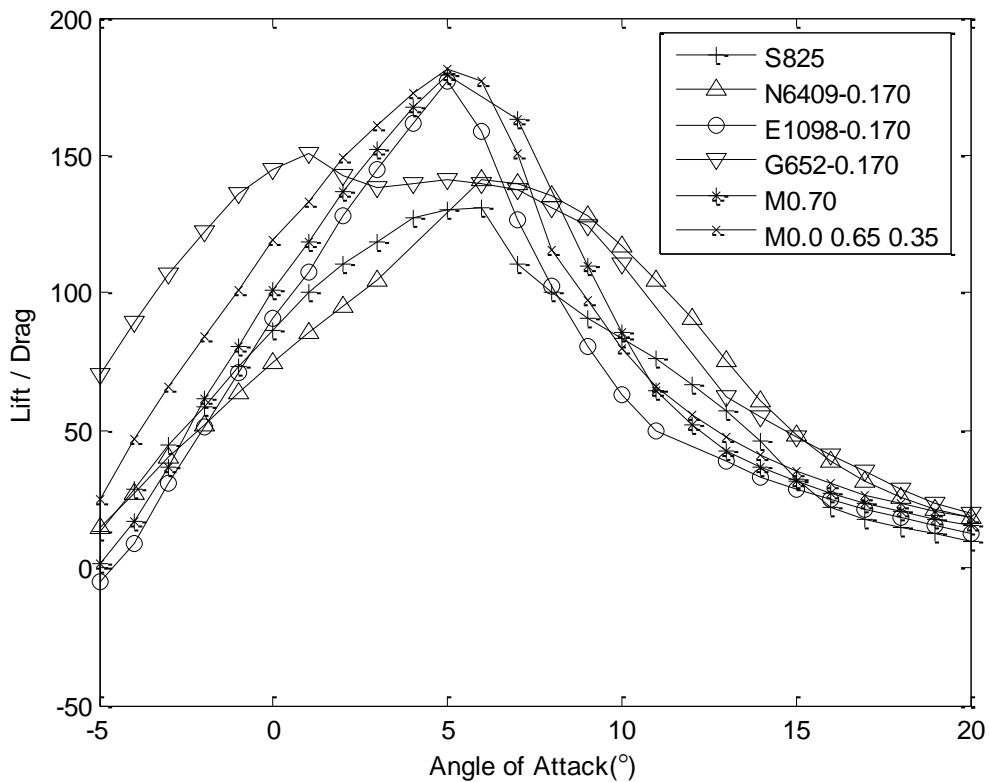
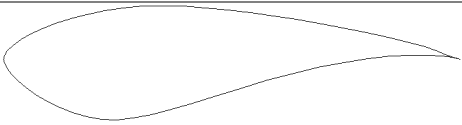



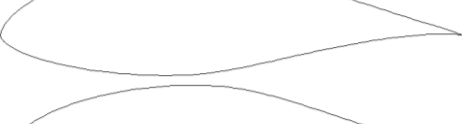



Figure 5.5 Lift drag ratio of airfoils benchmarked S825 ( $R_n = 2.0 \times 10^6$ ).



Table 5.4 Comparison of the maximum lift/drag of various airfoils at 40% radius.

Model	Profile	Max Lift/Drag
(a) S814		117.161
(b) N6409-0.240		78.503
(c) E1098-0.240		159.665
(d) G652-0.240		108.137
(e) M1.00		159.665
(f) M0.0_1.0_0.0		159.665

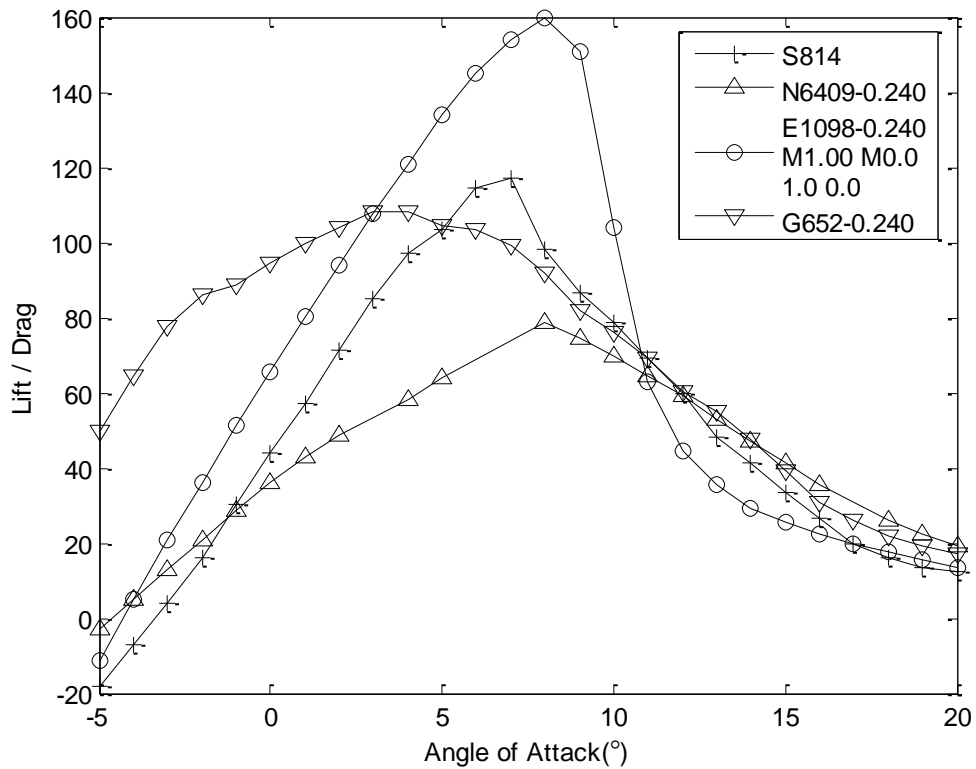
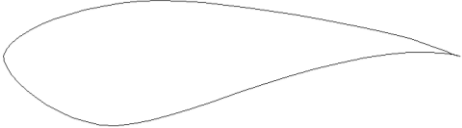




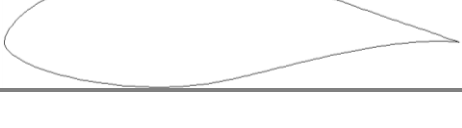


Figure 5.6 Lift drag ratio of airfoils benchmarked S814 ( $R_n = 1.5 \times 10^6$ ).

Table 5.5 Comparison of the maximum lift/drag of various airfoils at 30% radius.

Model	Profile	Max Lift/Drag
(a) S815		111.250
(b) N6409-0.260		64.9560
(c) E1098-0.260		133.628
(d) G652-0.260		93.062
(e) M1.00		133.628
(f) M0.0_0.95_0.05		134.192

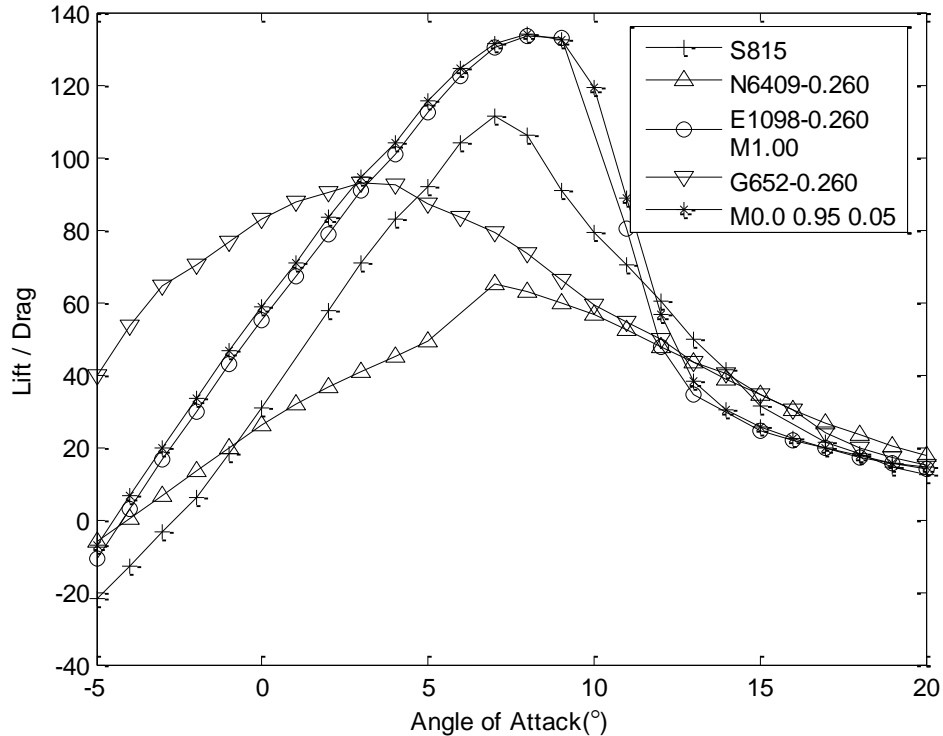


Figure 5.7 Lift drag ratio of airfoils benchmarked S815 ( $R_n = 1.2 \times 10^6$ ).

Figure 5.8 illustrates the maximum lift-drag ratios against the morphing parameter during normal morphing procedures, and ‘0.140’, ‘0.170’, ‘0.240’ and ‘0.260’ designate targeted  $t/c$ . As shown in Figure 5.8, morphing can generate intermediate models between source and target models parametrically, and it would be helpful to investigate the relationship between performances and parametrically generated models.

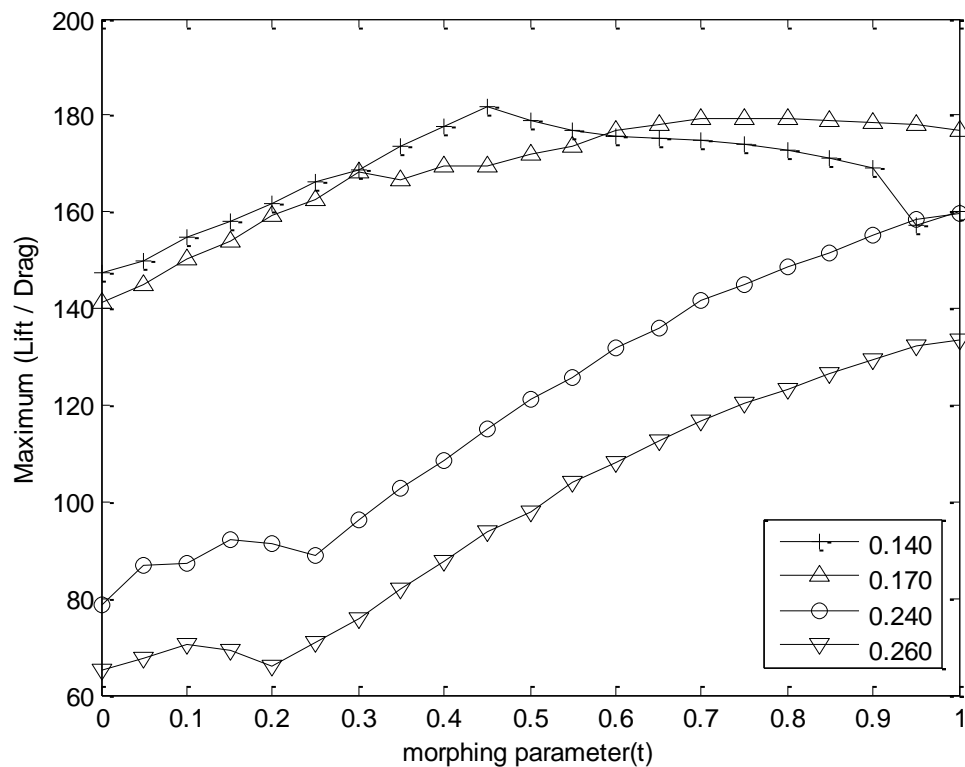


Figure 5.8 Maximum lift-drag ratios against morphing parameter

Figure 5.9 illustrates the maximum lift-drag ratios of optimised models from three generating methods. In the figure, ‘NERL S-’ denotes S815, S814, S825 and S826 at 0.3R, 0.4R, 0.75R and 0.95R respectively. ‘N6409-’, ‘E1098-’ and ‘G652-’ designate single airfoils converted to the specific  $t/c$ : 0.260 for 0.3R, 0.240 for 0.4R, 0.170 for 0.75R, and 0.140 for 0.95R. In addition, ‘Morphing’ and ‘Multi’ denote the optimized models for each radius of blade through the normal morphing and the multi-target morphing respectively. As shown in Figure 5.9, the optimised models from the multi-target morphing always have the best performances at all positions

along the radius. Even the normal morphing produced airfoil which appear to have better performance than other single airfoils. By the way, the NERL airfoils were originally designed considering factors other than lift to drag ratio, such as roughness effects or structural requirements, so they may not have been optimised for maximum aerodynamic performance.

The coordinates of the optimised models are presented in Appendix A together with their lift and drag coefficients.

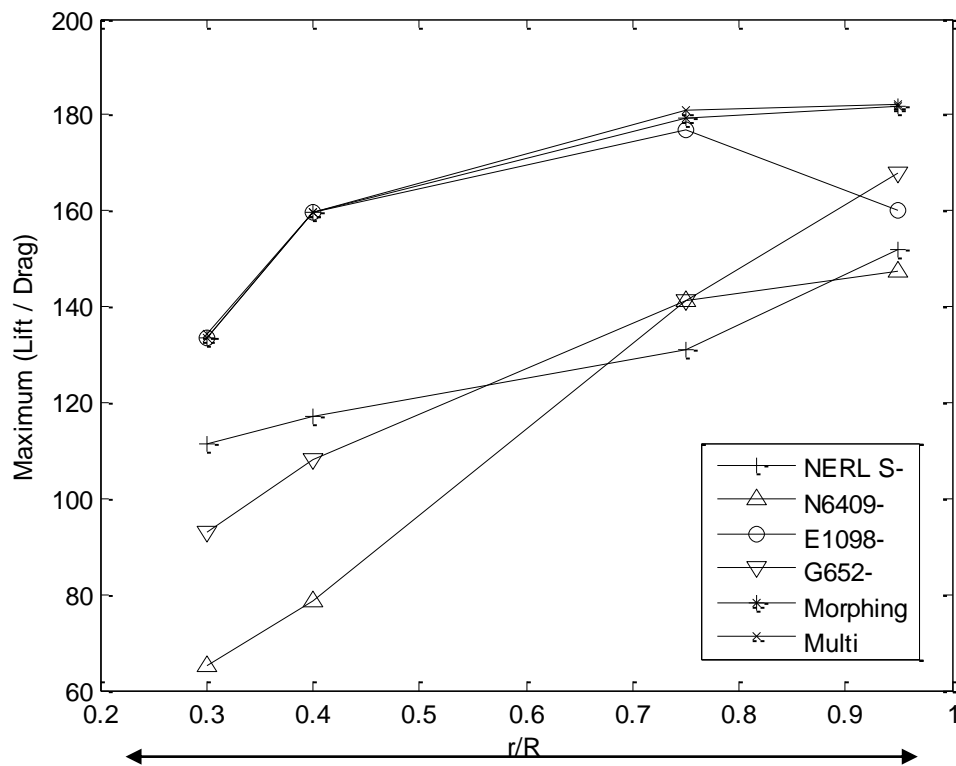


Figure 5.9 Maximum lift-drag ratios of airfoils as radius of blade.

## 5.4 AIRFOILS FOR CURRENT TURBINE

NACA 4415 has been frequently employed for current turbines ([73], [78] and [79]), and the parent airfoil models chosen for this study were NACA 4315 (source) and 4615 (target) sections. They differ from NACA 4415 only in the position of maximum camber.

For the NACA four-digit wing sections, the first digit describes maximum camber as percentage of the chord. The second digit denotes the distance of maximum camber from the airfoil leading edge in tens of percents of the chord. The last two digits describe maximum thickness of the airfoil as percentage of the chord. NACA 4615 section, for example, has a maximum camber of 4% (0.04 chord) located at 60% of the chord (0.6 chord) from the leading edge, and it has a maximum thickness equivalent to 15% of the chord length.

The two foil sections chosen for this study have different positions of maximum camber (30% and the other at 60 % from the leading edge) but the same maximum camber and maximum thickness (4% and 15% respectively).

The procedure followed for this case study is more or less the same as for the wind turbines. However, current turbines, unlike the wind turbines, can suffer from cavitation. Consequently cavitation inception has been included in the optimisation process. A brief examination of the basics of cavitation is given in Appendix B. The morphed models were generated at the morphing parameter intervals of 0.01 from zero to one. Figure 5.10 illustrates source, target, and some of their morphed models.

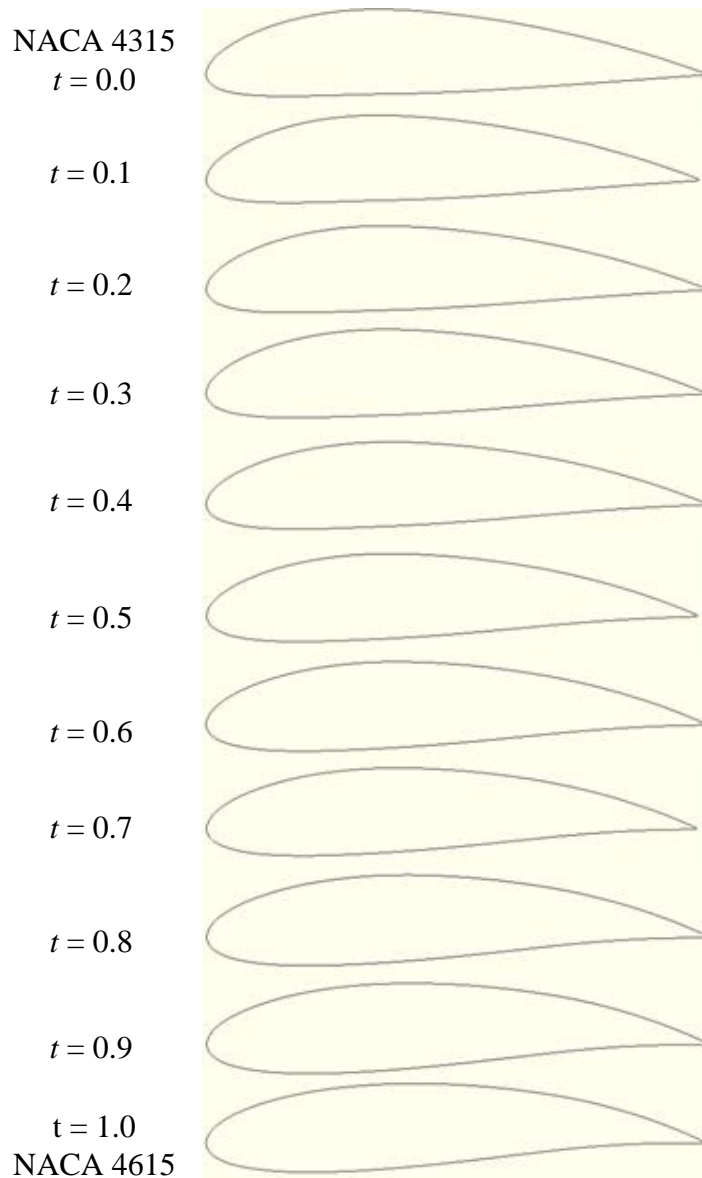


Figure 5.10 Original models and morphed airfoils.

First, the lift-drag ratio and the cavitation inception of source and target airfoils were estimated by the Xfoil program. The analysis was carried out with the Reynolds number of  $5.0 \times 10^6$ , which is within the realistic operational range of current turbines. The range of angle of attack ( $\alpha$ ) was from negative  $10^\circ$  to positive  $20^\circ$ , and the maximum thickness of airfoil was 15 % of chord. The lift-drag ratio and the cavitation inception of the two original models are illustrated in Figures 5.11 and 5.12.

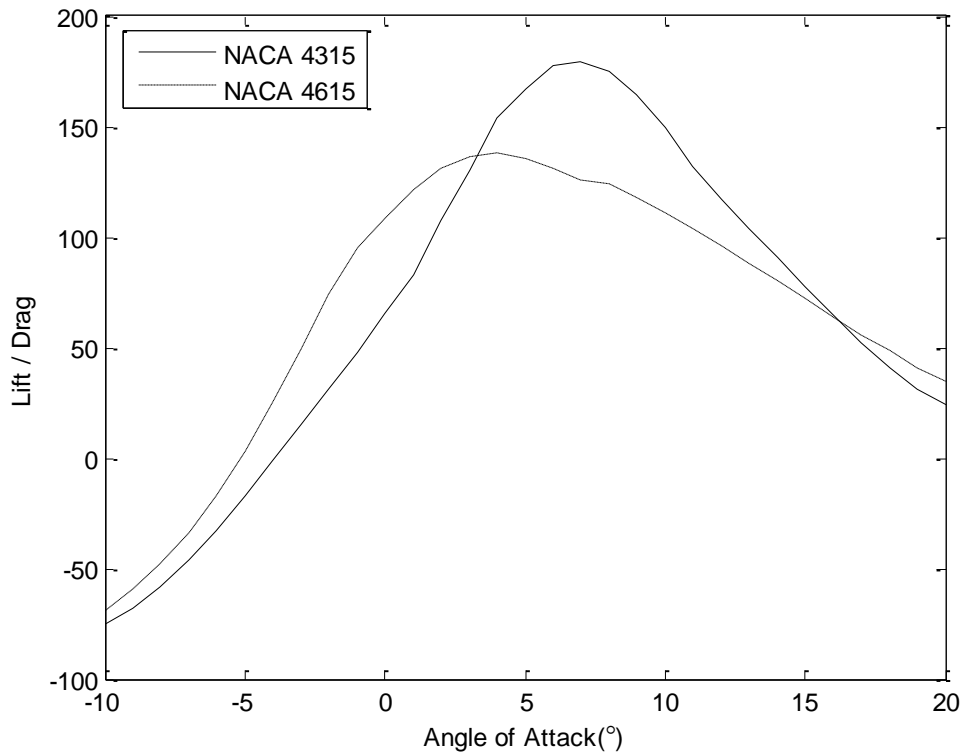


Figure 5.11 Lift-drag ratio of the parent models.

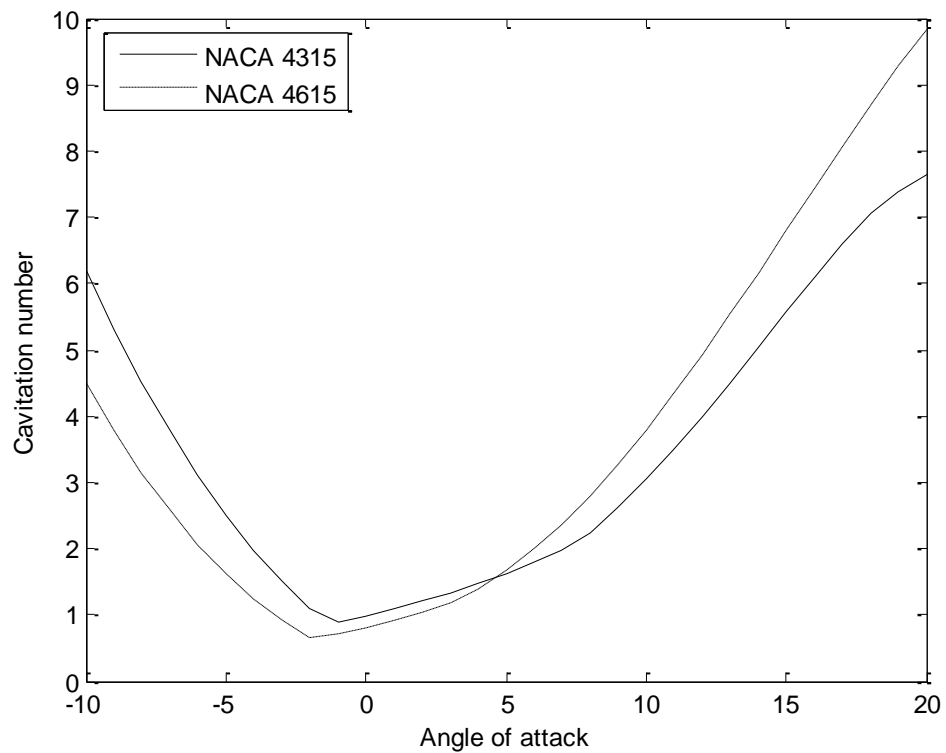


Figure 5.12 Cavitation inception bucket of the parent models.

As shown in Figure 5.11, the maximum lift to drag ratio of NACA 4315 is higher than that of NACA 4615, although at small angles of attack the opposite is true. NACA 4315 has a higher risk of face sheet cavitation and back bubble cavitation at negative and small angles of attack. However, NACA 4615 has a higher risk of back sheet cavitation at higher angles of attack. The cavitation inception at angles of attack  $-10^\circ$ ,  $0^\circ$ , and  $20^\circ$  is shown in Table 5.6.

Table 5.6 Maximum lift-drag ratio and cavitation number of original sections.

Model	Max Lift/Drag	Cavitation number		
		$-10^\circ$	$0^\circ$	$20^\circ$
NACA 4315	179.552	6.173	0.994	7.633
NACA 4615	138.173	4.485	0.812	9.837

In order to search for optimum airfoils considering the cavitation inception, the criterion of angle of attack for the face sheet, back bubble and back sheet cavitation are based on  $-10^\circ$ ,  $0^\circ$  and  $20^\circ$  respectively since these angles could be represented for each cavitation as shown in Figure 5.12. In this study, the analysis was carried out with three conditions of cavitation inception constraint as shown in Table 5.7.

Table 5.7 Constraints for the optimisation of the current turbine airfoil.

Condition	Cavitation number Constraint		
	$-10^\circ$	$0^\circ$	$20^\circ$
(a)	5	free	Free
(b)	free	0.9	Free
(c)	5.5	free	8.5

### **Condition (a): Constraint of face sheet cavitation**

The first analysis was carried out with the constraint of face sheet cavitation. The cavitation number smaller than 5 at  $-10^\circ$  angle of attack was selected for the constraint. With this condition, the maximum lift-drag ratio of 157.297 occurred at the morphing parameter of 0.66. The cavitation number at angles of attack of  $-10^\circ$ ,  $0^\circ$  and  $20^\circ$  was 4.988, 0.867, and 9.106 respectively.



**Condition (b): Constraint of back bubble cavitation**

The second optimisation was carried out with the constraint of back bubble cavitation, and the cavitation number smaller than 0.9 at  $0^\circ$  angle of attack was imposed. With this condition, maximum lift-drag ratio of 165.615 was obtained at the morphing parameter of 0.48. Cavitation number at angles of attack of  $-10^\circ$ ,  $0^\circ$  and  $20^\circ$  was 5.278, 0.899 and 8.707 respectively.

**Condition (c): Constraint of face and back sheet cavitation**

The third optimisation was conducted with two constraints of face and back sheet cavitation. A condition of cavitation number smaller than 5.5 at  $-10^\circ$  angle of attack and 8.5 at  $20^\circ$  angles of attack was imposed. With the condition given, maximum lift-drag ratio of 170.234 was obtained at the morphing parameter of 0.48. The cavitation numbers at  $-10^\circ$ ,  $0^\circ$  and  $20^\circ$  angle of attack were 5.486, 0.921, and 8.451 respectively.

The performance of the optimised airfoils for the above conditions is shown in Table 5.8 and the airfoils are illustrated in Figure 5.13. In addition, the lift-drag ratio and the cavitation inception bucket of each airfoil are illustrated in Figure 5.14 and 5.15 respectively.

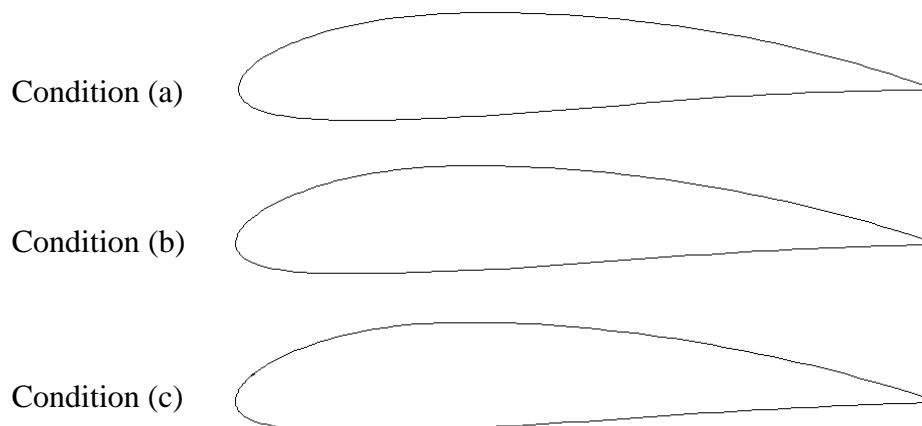


Figure 5.13 Optimised airfoils.

Table 5.8 Performances of the optimised airfoils.

Condition	Model	Max Lift/Drag	Cavitation number		
			-10°	0°	20°
(a)	0.66	157.297	4.988	0.867	9.106
(b)	0.48	165.615	5.278	0.899	8.707
(c)	0.36	170.234	5.486	0.921	8.451

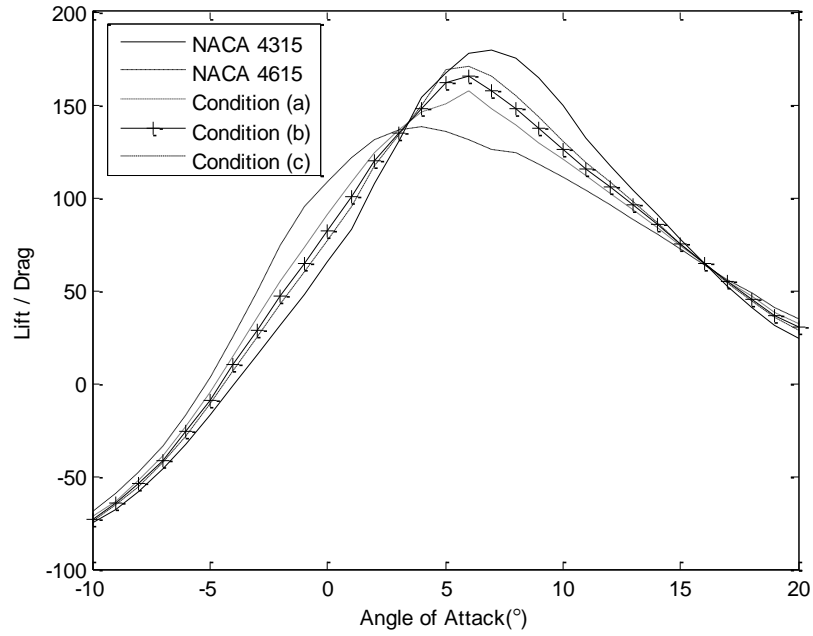


Figure 5.14 Lift-drag ratio of the original and optimised models.

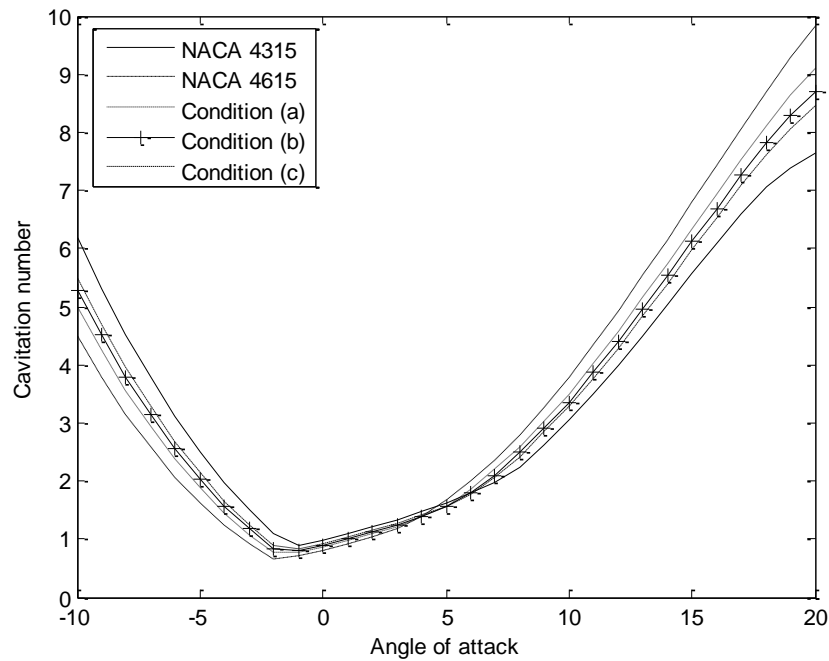


Figure 5.15 Cavitation inception bucket of the original and optimised models.

## **5.5 CONCLUDING REMARKS**

This chapter discussed a method of utilising the morphing technique to rapid generation of airfoil forms in the optimisation process. The performance of the generated airfoils is analysed using Xfoil.

It was shown that any number of airfoil forms could be produced more or less instantaneously as and when required by the optimisation process. Since the generated models can be defined with only one, or at most a very few, morphing parameters, the performance analysis program can refer to them simply and with ease. As expected, in addition, morphing could generate higher performance airfoils than the method of scaling y-coordinates of single airfoil.

# Chapter 6.

## APPLICATION OF MORPHING TO HULL FORM GENERATION

---

### 6.1 GENERAL REMARKS

With the availability of cheap and advanced computing capability, development in various (mostly numerical) ship performance evaluation tools has been taken to the next logical step, *i.e.* multi-objective optimisation techniques. If such an optimisation technique is to be successful, the system should be able to examine possibly a vast number of design alternatives which can be produced entirely at the beck and call of the optimisation system more or less instantaneously.

In Chapter 4 morphing of 2D curves was discussed and some examples of practical application in design were given in Chapter 5. In this chapter morphing of 3D surfaces, specifically ship hull forms, will be developed and a number of interesting case studies will be made.

Two methods of establishing correspondence between two parent models for the surface morphing was introduced in Chapter 3: regularising and merging. The regularising method will be applied to morphing between a 4,100 TEU container carrier and a 5,000 TEU container carrier hull forms and morphing between a container carrier and a very large crude carrier (VLCC) is attempted through the merging method.

In order for this technique to be applied to practical hull form design tasks successfully form morphing needs to be ‘constrained’ to produce a certain set of desired design requirements, e.g. block coefficient or displacement of the hull form. This issue will be dealt with towards the end of this chapter.

## 6.2 HULL FORM MORPHING USING REGULARISING METHOD

Two container ships of 4,100 TEU and 5,000 TEU capacity were selected as the parent models. The source form was represented by a mesh of 12,797 vertices and 24,899 faces, while the target form was modelled with a mesh of 19,817 vertices and 38,933 faces, and they are shown in Figure 6.1.



(a) Source model - 4,100 TEU container carrier (vertex: 12,797, face: 24,899)



(b) Target model - 5,000 TEU container carrier (vertex: 19,817, face: 38,933)

Figure 6.1 Two parent hull forms.

The two parent forms which represent the source and target forms in morphing must be made to have corresponding mesh vertices, as discussed in Chapter 3. As shown in this figure, however, it is most unusual to find two ship hulls represented in identical mesh structures so as to accommodate easy correspondence between vertices, unless, of course, the meshes were created for morphing in mind specifically. Therefore, usually the first task in morphing is to represent the two surfaces with identical mesh structures. The attributes of a mesh structure include the number of edges of polygon mesh, degree of coarseness or fineness and the number of vertices and faces. For this reason, it is usually necessary to ‘re-mesh’ the surfaces.

### 6.2.1 Remeshing

The two corresponding pair of vertices must be situated at approximately equivalent location on each patch, and consequently the first stage of remeshing is subdivision.

#### **Subdivision**

Subdivision is carried out to ensure that the corresponding vertices between the two models are from the same kind of region of the parent hulls. For example, the bulbous bow of a source model should be matched to the bulbous bow or forefoot region of the target hull, if the target has no bulbous bow, and not to the bow flare. In addition, the subdivided meshes make mapping processes easier in both direction between 3D and 2D. 2D parts of a hull form, such as parallel middle body, for example, can easily be refined. However, parts with steep curvature, such as stern, are difficult to handle without subdivision due to possible folding problem during 3D to 2D mapping of the mesh. In order to avoid this problem, smaller subdivided patches are required for these parts.

All the subdivision patches are made to be of triangular shape with three boundary lines in this work, since regularised and refined triangular meshes are generated by bisecting the boundaries of the triangular patch concerned. For the subdivision discussed here, some obvious features are selected as the boundary lines between patches. These include flat of side lines, bilge lines, transom, bossing lines, fore perpendicular line and so on. The most important thing in the subdivision is that the same rules are applied to both models. For example, if the third section line is used for subdividing a model, the same section line should also be used for the other, when they have the same number of sections.

The subdivision tasks in this work were carried out manually using Rhinoceros 4.0 program [80] which can handle mesh elements such as extracting nodes data, and saving them to a file with ease. Using the Rhinoceros program, three boundary nodes and internal nodes of each subdivided patch can be extracted

from the original mesh. Figure 6.2 shows an example of such subdivision with the extracted boundaries and internal nodes for the patch of the bulbous bow region.

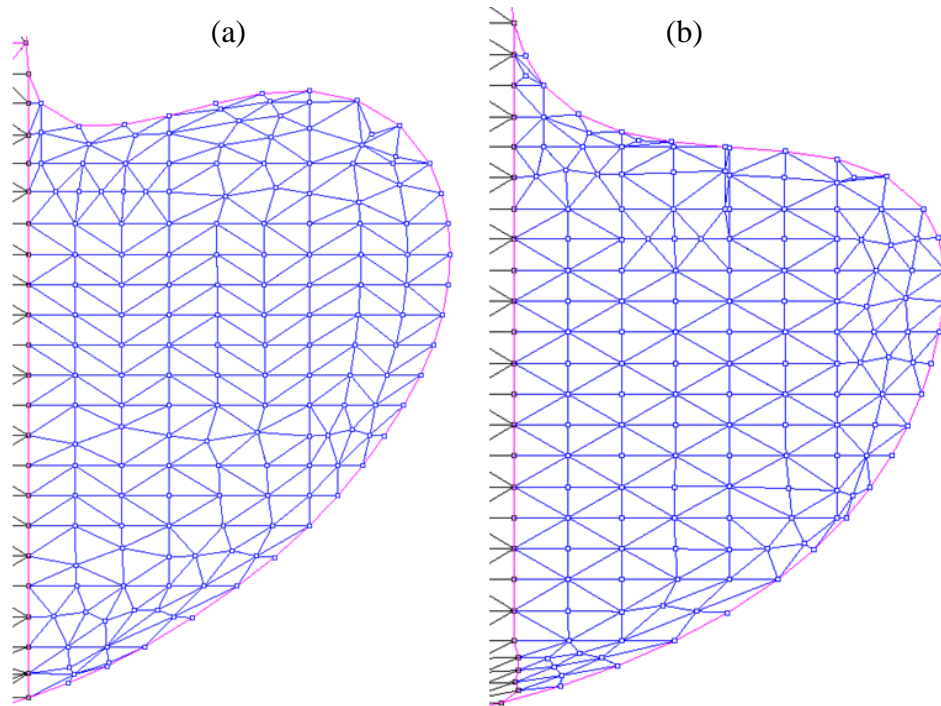


Figure 6.2 Extracting boundaries and internal nodes for bulbous bow: (a) source model; (b) target model.

An example of the two parent hull forms subdivided into roughly corresponding patches is shown in Figure 6.3. Although some of the patches do not appear to be triangular, they still can be regarded to have three boundary lines and that is all that matters for re-meshing.

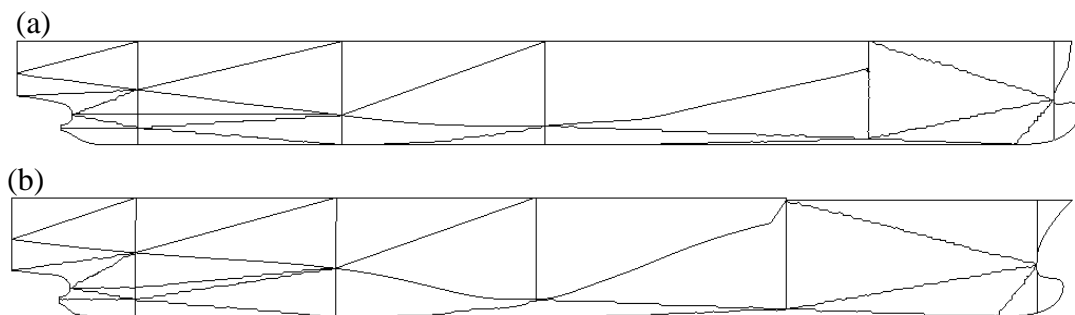


Figure 6.3 Subdivision of the source and target models (34 patches each).

### Parameterisation

With the subdivided patches thus obtained, each patch of the target model now roughly corresponds to a patch of the source model. Remeshing procedure proper can be divided into three stages and this is illustrated in Figure 6.4 with the patch of the bulbous bow region. Figure 6.4 (a) shows the original mesh representing a bulbous bow region.

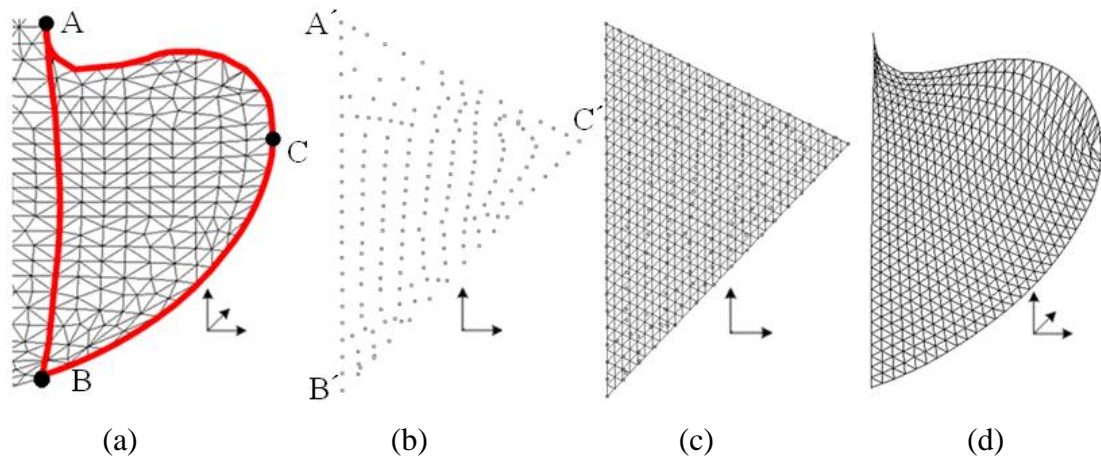


Figure 6.4 Procedure of remeshing for bulbous bow: (a) Original 3D mesh; (b) Mapping 3D points onto 2D plane; (c) Regularisation on 2D plane; (d) Remapping from 2D to 3D.

The regularising method is one of the re-meshing methods that makes irregular meshes into regular ones with the ultimate aim of establishing an identical mesh structure for both corresponding patches of source and target models. Here, the regularity means that all vertices have the same number of neighbours and all faces have the same number of sides. It will be far too complicated, if not impossible, to create a regular mesh structure in 3D. This difficulty can be overcome by first mapping the 3D surface mesh onto a 2D plane. This entails parameterising a vertex set  $\mathbf{S} = (x_i, y_i, z_i)$  of a 3D surface so that an equivalent vertex set  $\mathbf{P} = (\zeta_i, \xi_i)$  can be created on a 2D plane, which in effect is a parameter plane (Figure 6.4 (b)).



Since a regular triangular mesh can be easily created by repeatedly bisecting each edge of a triangular shape, a convenient set of three points are selected from the mesh nodes on the boundary curves of the surface in question. Obvious choices in the current problem are points A, B and C in Figure 6.4 (a): one at each of the two corner points and another approximately in the middle of the forward boundary curve. With the three points selected, the outline of the 2D mesh on the parameter plane should be established so that length ratios between the three sides are identical to the length ratios of the three 3D boundary curves. The length of each 3D boundary curve can be approximated as the sum of the shortest distances between the neighbouring mesh nodes on the boundary curve.

First, the boundary nodes on AB are mapped onto the straight line between A' and B' in Figure 6.4 (b), and so on. Starting from the point A, let the shortest distances between two neighbouring nodes of the outermost mesh elements resting on AB be  $d_1, d_2, \dots, d_n$ . Then the distances between the two nodes of the outermost mesh elements resting on line A'B',  $d'_1, d'_2, \dots, d'_n$ , are calculated according to Equation (3-20). In a similar manner the node lying on BC and CA are mapped onto the 2D plane.

As mentioned in Chapter 3, with the convex combination coefficients calculated from Equation (3-13) and parameterised nodes on the boundaries, internal points can thus be finally mapped to the 2D plane. With the procedure of the parameterisation, the 3D surface mesh of a patch can be parameterised to the 2D triangle, and other patches of both source and target models can also be mapped onto the 2D planes in a similar manner.

### **Regularisation**

This step generates a fine mesh on the 2D parameter plane. This is done by repeatedly bisecting each edge of the mesh elements, starting with the one triangle formed by the three borders (A B C in Figure 6.4 (b)). The original triangle is divided into four triangles by bisecting each edge and connecting the bisecting points. Each of the resulting triangles is then divided into four triangles and so on. In this way after the  $i^{\text{th}}$  repetition,  $4^i$  ( $i = 1, 2, 3, \dots$ ) regularised triangles are generated on the 2D plane.

### **Remapping**

The new triangular mesh mapped from the 3D surface and the regularised mesh now lie on the same 2D domain triangle, as shown in Figure 6.4 (c). In order to represent the 3D surface with the regularised triangular mesh, the vertices of the regularised triangles should be represented with barycentric coordinates of parameterised mesh using the relationship already established between the point set  $\mathbf{P} = (\zeta_i, \xi_i)$  and the vertex set  $\mathbf{S} = (x_i, y_i, z_i)$  of the 3D surface.

A node of the regularised triangles can be represented with its surrounding triangle of the parameterised mesh, so as to represent the node with barycentric coordinates with a point set  $\mathbf{P} = (\zeta_i, \xi_i)$  of the parameterised mesh. Using the same barycentric coordinates with a vertex set  $\mathbf{S} = (x_i, y_i, z_i)$  of the 3D surface, the regularised mesh can now be mapped back onto the 3D surface, as shown in Figure 6.4 (d). Other patches of the original meshes can be regularised in a similar manner, and Figure 6.5 shows some examples of the 3D regularised meshes for the target model. The regularised meshes in Figure 6.5 are generated by 4<sup>th</sup> repetition and each patch has  $4^4$  triangular faces while the regularised mesh in Figure 6.4 (d) is generated by 5<sup>th</sup> repetition and it has  $4^5$  triangular faces.

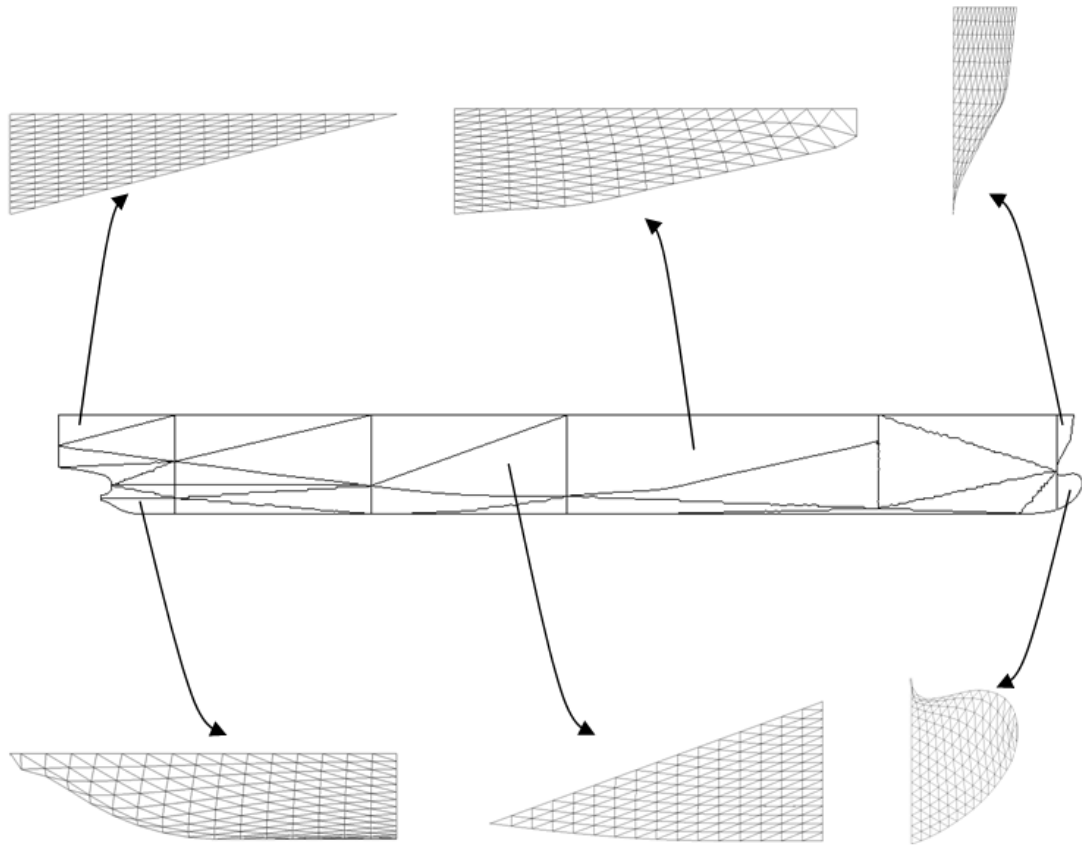


Figure 6.5 Examples of 3D patches with regularised meshes.

Since both source and target models were divided into similar shape patches with same rules and each patch was regularised with the same number of repetition into identically structured meshes, each patch of the source model has now a good correspondence with a patch of the target model. For morphing the whole hull form, all patches need to be put together again and Figure 6.6 shows the results of regularised meshes for both source and target hull forms.

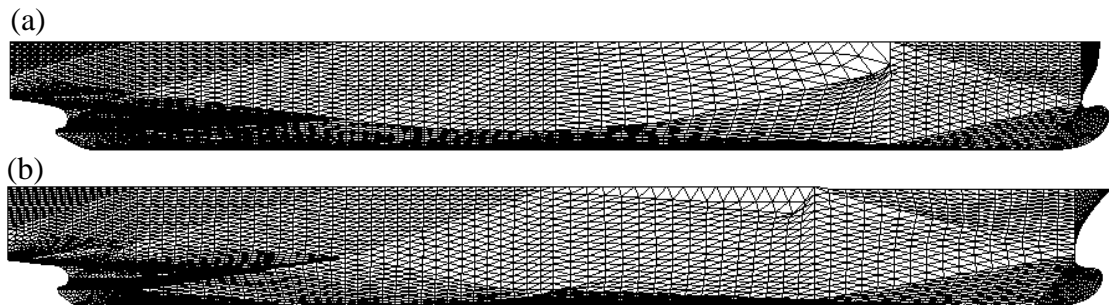


Figure 6.6 The regularised parent hull forms

### **6.2.2 Morphing**

The two original meshes which were originally of different structures, have now been made to correspond to each other so that they have the same number of faces, vertices, and edges. In addition, the vertices and faces of each mesh are given identical order. Therefore, there is no problem in relating the vertices on one mesh to those on the other. Now, morphing proper can take place.

The code for 3D surface morphing procedure has been also developed using C++, and it is based on triangular meshes. The algorithm consists mainly two parts: remeshing and deforming. For the input data, the remeshing procedure requires original mesh files (SMF), and boundaries of 3D surface and its corresponding parameterisation domain need to be defined with TXT files which have x-y-z coordinates. The generated remeshed models are formed with SMF files as well. For the deforming procedure, the remeshed models are used as input data, and the generated intermediate models from this procedure are also formed with SMF files.

The number of newly generated models can be controlled according to the interval of morphing parameter. Moreover, the morphing sequence can be checked through visualisation. The visualised display can be rotated or moved using mouse or keyboard. The developed code is very dependent on SMF file format. However, many other 3D graphics files can be converted from or to SMF files using IVCON program [81] which is available free on the internet. Thus, the models, expressed in SMF files, can be embedded in the developed code.

It was found that any number of hull forms corresponding to any value of the parameter  $t$  between 0 and 1 could be generated in a fraction of a second using a normal desktop PC. The model at  $t = 0$  is the source 4,100 TEU Panamax, and the model at  $t = 1$  is the target 5,000 TEU Post Panamax ship. Figure 6.7 shows some examples of hull forms generated through morphing.

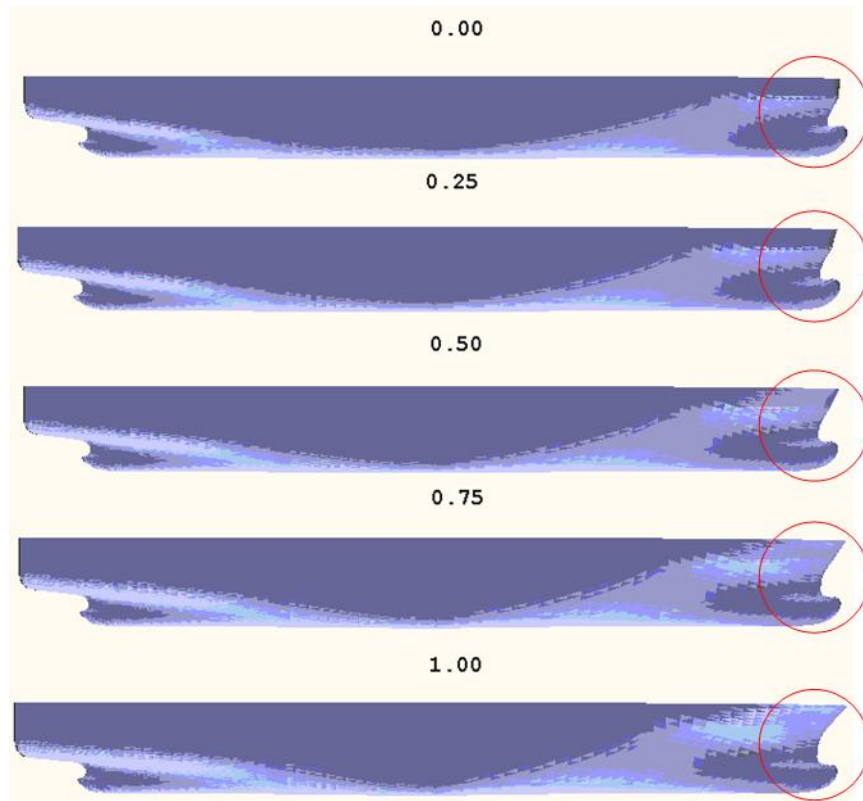


Figure 6.7 Hull form generated through morphing ( $t = 0, 0.25, 0.5, 0.75, 1$ ).

Since the two parent forms have a very similar shape, the result of morphing does not appear to be dramatic. However, the progression from one form to another could be clearly discerned, especially on the distinguishing features, such as shape of bulbous bow, and bow flare.

It is also possible to transform only some parts of hull form such as bulbous bow and flare since the mesh is already subdivided into patches. Figure 6.8 shows the morphing sequence of the bulbous bow and the forecastle ( $t$  varying from 0 to 1) in which the source (corresponding to  $t = 0$ ) and the target ( $t = 1$ ) models are of 4,100 TEU and 5,000 TEU container carriers. When the patches of the target model are assembled to the mesh of the main part, the patches should be deformed appropriately so as to fit the adjoined boundaries. Figure 6.8(a) illustrates morphing between two different bulbous bows without considering adjoined boundary of bulbous bow and main hull part, so the boundary shape might be changing during the morphing. On the other hand, Figure 6.8(b) shows another example where the

boundary of the target bulbous bow has been replaced by the one of source model. In this way the boundary is always identical during morphing sequence and the compatibility of newly generated bulbous bows to the main hull part is assured without any further manipulations, excepts the question of fairness which will be briefly discussed in Discussion. These deformation tasks are very similar to the procedure of generating extreme hull forms, and the method will be described in Section 6.4.

In this example the source and target meshes for the bulbous bow are remeshed to have  $4^4$  and  $4^5$  triangles for Figure 6.8(b) and Figure 6.8(a) respectively, and the meshes of the flare have  $4^4$  triangles.

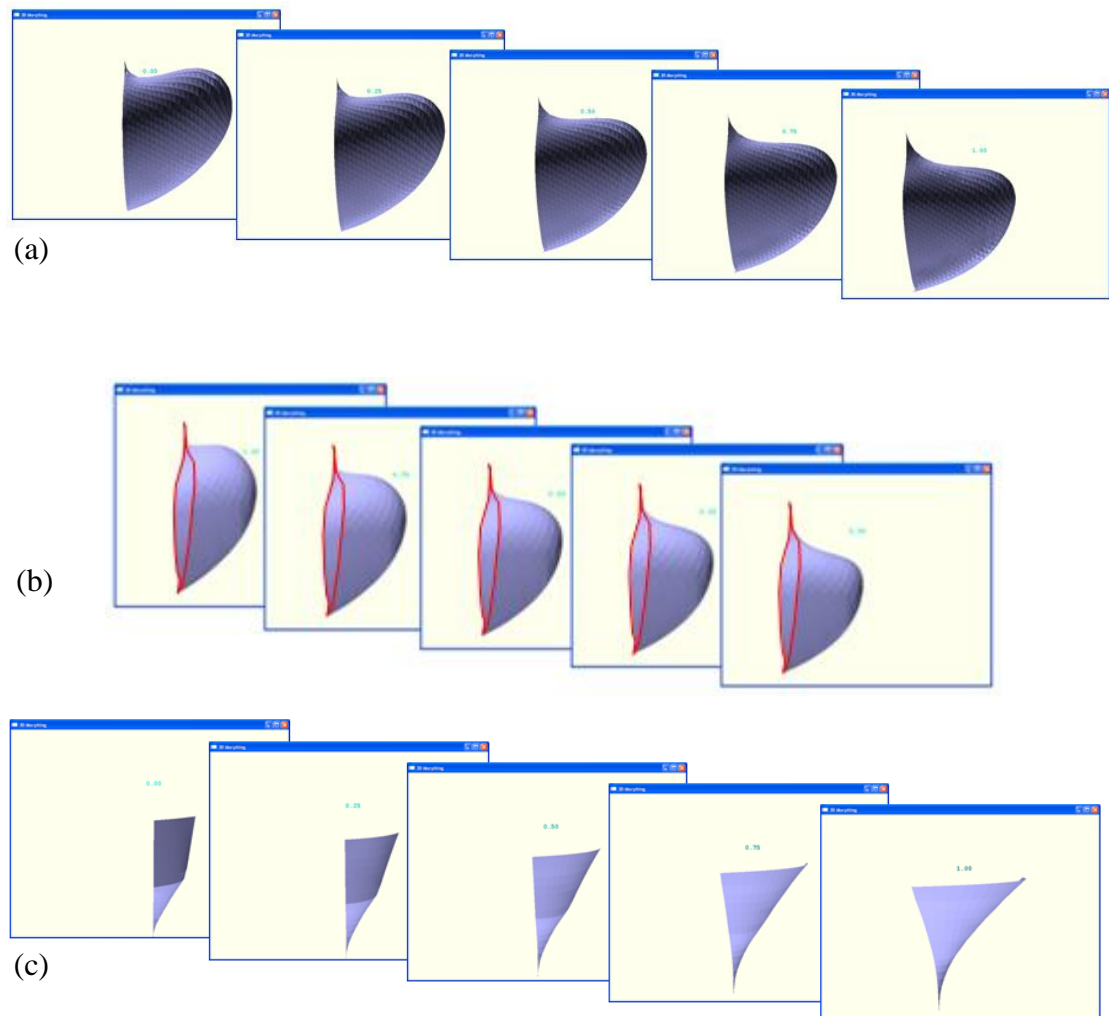


Figure 6.8 Morphing for the bulbous bow (a and b) and the forecastle (c).

## **6.3 HULL FORM MORPHING USING MERGING METHOD**

### **6.3.1 Remeshing**

It is no exaggeration to say that finding a proper correspondence of the vertices between two models is one of the most important tasks for the morphing procedure. The method of establishing the correspondence by creating a new mesh structure (regularising method) was discussed in Section 6.2. Another method of establishing the correspondence, a merging method, is introduced in this section. The merging method has two significant advantages over the regularising method. The first merit is that it minimises the need for manual manipulation. In the regularising method, it will be recalled that the original meshes need to be subdivided into several patches by extracting boundaries and internal nodes for easier re-meshing work, and they also need to be put back together for morphing proper. These tasks, however, are not required at all in the merging method. The second excellent point of the merging method is that it preserves exactly the original mesh data. In the regularising method, the vertices and faces of the regularised mesh are newly generated so that the meshes differ from the original ones. However, the merging method can keep the original vertices of the source and target meshes, and the redefined meshes through the merging method will inherit the original mesh characteristics.

Figure 6.9 shows the outline procedure of the merging method of preparing the hull form meshes prior to morphing. The first task for the merging method is mapping the two original meshes of the hull forms onto a 2D parameter plane in order to make it easier to handle the meshes. The next step is merging the projected meshes into a single mesh. When this 2D mesh is mapped back onto the original source and target 3D surfaces, they will have an identical mesh structure and the correspondence between the two meshes is automatically assured.

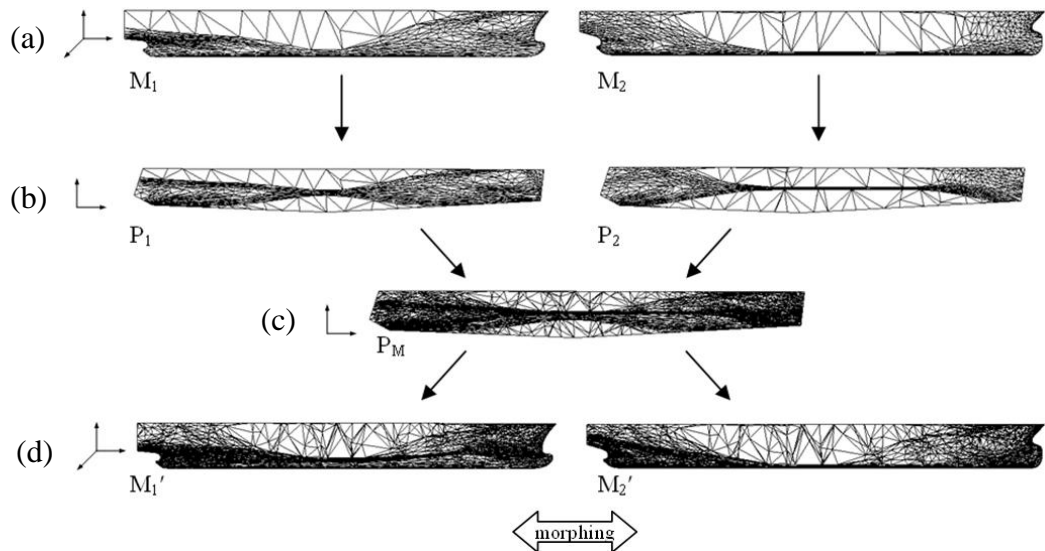


Figure 6.9 Overview of algorithm used for the hull form morphing:  
 (a) Original 3D meshes; (b) Projecting 3D meshes on 2D plane;  
 (c) Merging two 2D meshes; (d) Remapping 2D mesh to each original 3D surface.

In the case study based on the procedure described above and presented in this section a 5,000 TEU container carrier hull form and a 300,000 DWT very large crude carrier (VLCC) hull form are used as the two parent forms.

### Parameterisation

The method used for parameterising in the merging method is the same as that used for the regularising method. The 3D boundary nodes are first mapped to 2D boundary of the 2D plane, and then the internal nodes are mapped to the 2D plane with the convex combination coefficients. However, the way of choosing the 2D parameterised domain is somewhat different. In the regularising method, each subdivided 3D surface patch was mapped onto only a triangular patch on the 2D plane so as to accommodate generation of a triangular mesh during regularisation. In the merging method, however, original source and target meshes can be mapped onto any 2D polygonal plane.



For the parameterisation in the merging method, therefore, a 2D polygon, to which a 3D surface will be mapped, should be determined first. As mentioned above, any shape, size, or number of edges can be chosen; however, care should be taken to ensure that same features of both source and target models are assigned to a same vertex of the 2D parameterised domain. For example, bulbous bow nose tip of both source and target forms should be mapped to a same vertex on the 2D polygon. Figure 6.10 illustrates an example of setting a 2D polygon for the parameterisation of the 3D hull forms. Six vertices are selected for the hull form features in this example: (A) an apex of flare, (B, C) two transom edge points, (D) a bossing, and (E) keel point at midship, and (F) a bulbous bow nose tip. With these six feature points selected, the polygon would be a hexagon, of any proportion or size. However, it is helpful to have the polygon roughly resembling the outline of the hull shape so that the shape of the parametric mesh can be checked easily.

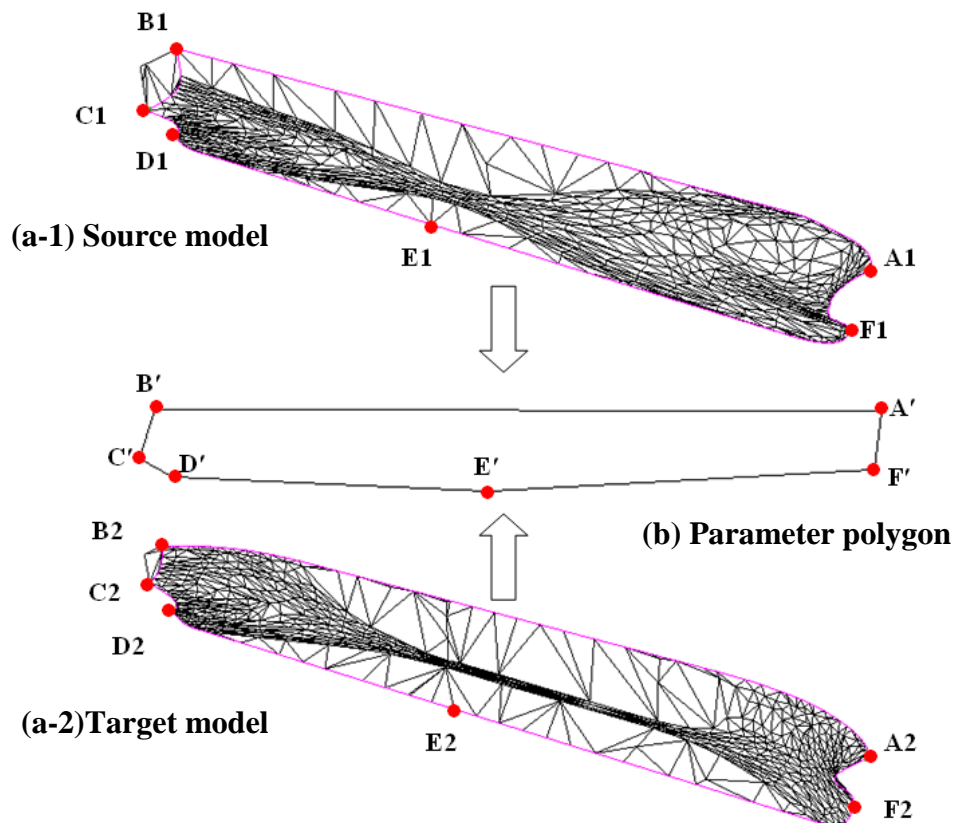


Figure 6.10 An example of selecting (a) feature points on 3D hull form meshes and (b) a 2D parameter polygon.

In Figure 6.10, the lines in magenta colour on the original meshes are the boundaries selected. The transom has been left out of parameterisation in order to ensure good correspondence. This point will be discussed in more detail later in Section 6.3.2.

With the boundaries of the 3D surface and the 2D polygon determined, each 3D boundary segment between two neighbouring feature points is mapped to the corresponding edge of the polygon. The boundary nodes on A1B1 of the source model and on A2B2 of the target model are mapped onto the same straight line between A' and B', and the nodes on A1B1 and on A2B2 are assigned to the nodes on A'B' so that their lengthwise relative positions on the lines are maintained. In the same manner, the nodes lying on other boundaries are mapped onto the corresponding edges of the 2D polygon.

As mentioned in Section 6.2, with the convex combination coefficients calculated from Equation (3-13) and parameterised nodes on the edges of the 2D polygon thus obtained, internal points can now be mapped onto the inside of the 2D polygon. Figure 6.11 shows the results of mapping onto the common 2D plane of the original source model 5,000 TEU container carrier and target model 300,000 DWT VLCC.

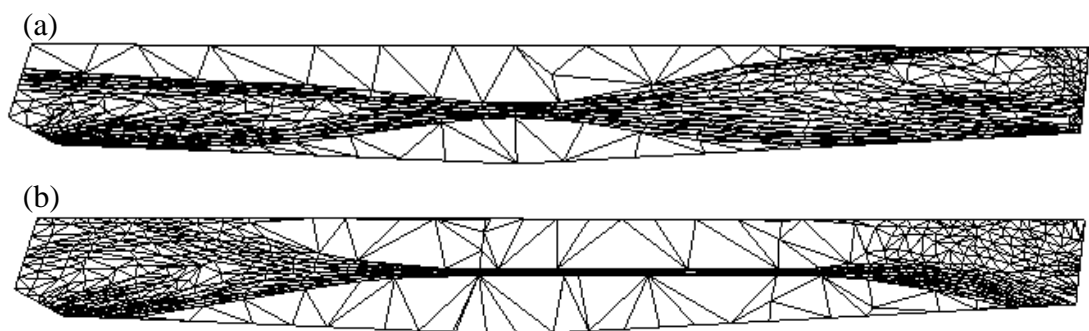


Figure 6.11 Parameterised meshes onto a common 2D polygon:  
(a) source model; (b) target model.

## Merging

The two parameterised meshes, thus obtained, are then merged into one mesh. The merging process is not quite as simple as superimposing the two parametric meshes. A systematic merging algorithm is required to ensure that the merged mesh is a triangular one inheriting both parent meshes.

The first task of merging is to establish a merged boundary edge set from the two boundary sets of both parameterised meshes. Since the nodes of both boundary sets are parameterised from different mesh structures, the number or positions of the nodes are usually not identical. Figure 6.12 illustrates the procedure of the boundary merging. Figure 6.12 (a) shows the example boundaries of two parameterised meshes overlaid. The boundary AD comes from one model and EI from the other. AD has four vertices, while EI has five vertices.

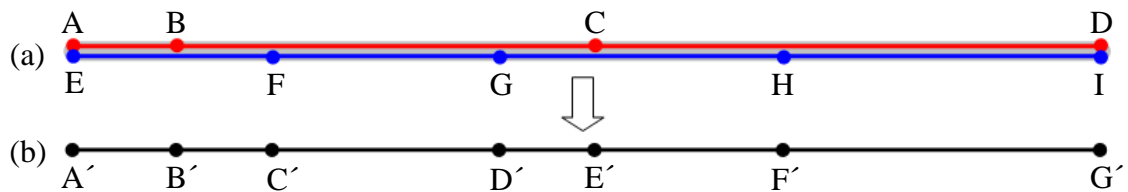


Figure 6.12 Merging boundaries.

In order to establish a new boundary edge set, a new point set should be prepared by removing one of duplicate vertices such as E and I (or A and D), and the nodes of the new vertex set are arranged in the order of occurrence from one end (from the left end in this case). The new vertex set is  $\{A', B', C', D', E', F', G'\}$ , and new edge set is  $\{A'B', B'C', C'D', D'E', E'F', F'G'\}$  as shown in Figure 6.12 (b).

The next task is to merge internal mesh elements of the two parameterised meshes. The concept is similar to the boundary merging, and the vertex and edge sets of two parameterised meshes are merged to a single vertex and edge sets.

The obvious constituents of the new vertex set are the existing vertices of the

two parametric meshes. Then intersection vertices are added to these. The intersection vertices are generated from crossing edges, such as vertices G and F as shown in Figure 6.13. Note the red and blue lines denote edges of source mesh and target mesh respectively, and the black lines designate common boundaries of the two meshes.

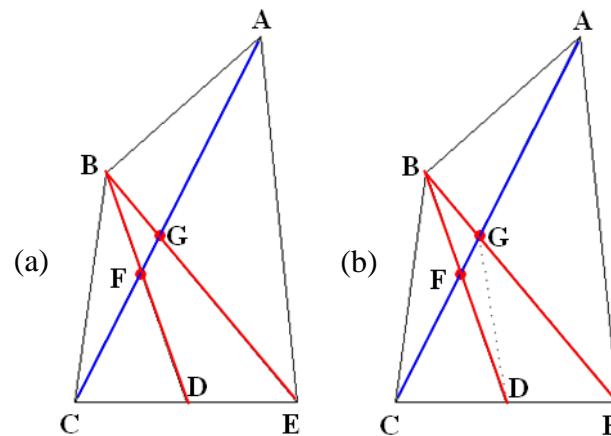


Figure 6.13 An example of additional intersection vertices and consequent new triangular faces.

It is essential, therefore, that each existing edge is checked to see if it is crossed by another edge (originally belonging to the other model). If the edge is crossed by any edge of the other mesh, the intersection point needs to be found and the edge is split into two by this point and these two edges become elements of the new edge set. If the edge is not crossed by any edge, it simply becomes an element of the new edge set. This edge set is as yet incomplete, because the newly established vertices may have introduced non-triangular faces. The edge DG in Figure 6.13 (b) is an example of this. As shown in Figure 6.13 (a), the merged mesh has a non-triangular face GFDE, and the face should be split into two triangular faces by generating edge GD or FE.

These new edges can be found by first checking if each of the neighbouring nodes of a vertex is joined to the next neighbour by an edge. For example, vertex A has three neighbours {B, G, E}, and they are connected by edges {BG, GE}. In the case of vertex F, however, its neighbours are {B, C, D, G}, but there is no

edge between vertices G and D. Consequently the edge GD is created and added to the edge set. Originally, the neighbouring nodes F and E of vertex G needed a new edge FE, but with the edge GD newly created, the neighbours of vertex G are now {A, B, F, D, E}. Each neighbour is connected to its next neighbour with edges {AB, BF, FD, DE, EA}, and thus a new edge FE is not required for the vertex G.

By operating these tasks repeatedly for all the vertices, the new edge set is established completely, and this process ensures that the resultant mesh is triangular with no non-triangular faces. A new face set can be constituted from the new vertex set and new edge set, and the merged parametric mesh is finally complete. Figure 6.14 (b) shows the parametric hull form mesh which is the result of merging two parameterised source and target models.

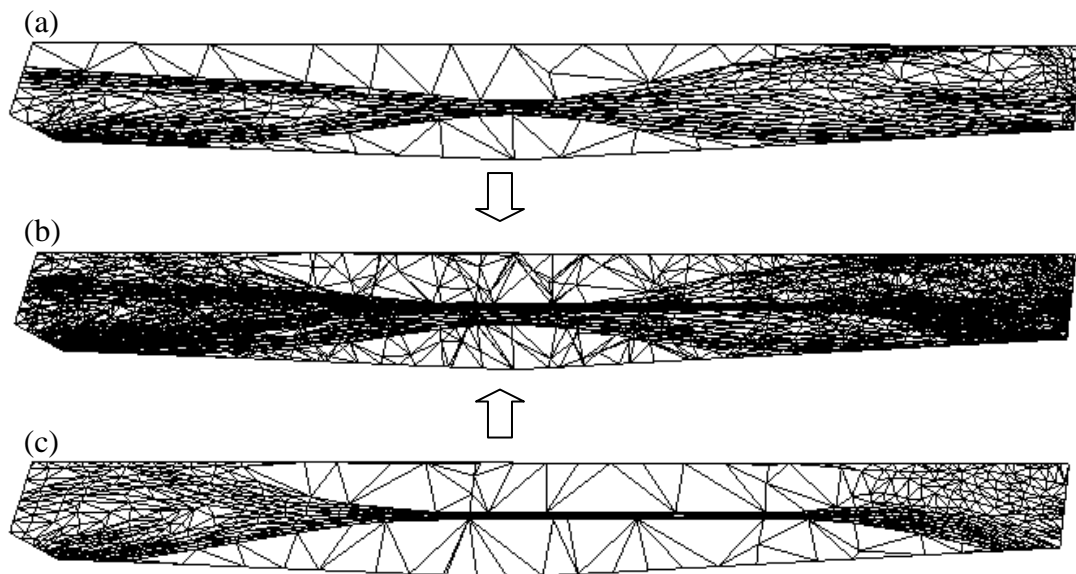


Figure 6.14 Parameterised meshes of source model (a) and target model (c), and their merged mesh (b).

The merged mesh contains all the vertices of the two parent meshes. In addition, all faces are triangular, and there is no overlapping edge or vertex.

### **Remapping**

In the regularising method, regularised triangular meshes are remapped back onto the 3D surface with barycentric coordinates of parameterised mesh of the original 3D surface. Remapping in the merging method is done on similar principles.

In order to remap the merged meshes to the two 3D surfaces, let us assume the vertex sets of the 3D original source and target model meshes as  $S_1 = (x_i, y_i, z_i)$  and  $S_2 = (x_j, y_j, z_j)$  respectively, and the vertex sets of their parameterised meshes as  $P_1 = (\zeta_i, \xi_i)$  and  $P_2 = (\zeta_j, \xi_j)$  respectively. Let the vertex set of the merged parameterised mesh be represented by  $M = (\zeta_k, \xi_k)$ . Since the vertex sets  $P_1$  and  $P_2$  are parameterised from the vertex sets of  $S_1$  and  $S_2$ ,  $P_1$  and  $P_2$  have accurate correspondence with  $S_1$  and  $S_2$ . In addition, the vertex set  $M$  clearly has the vertices of both  $P_1$  and  $P_2$ . When the number of vertices in  $S_1$ ,  $S_2$  and  $M$  are  $n_1$ ,  $n_2$  and  $n_3$  respectively, then  $(i = 0, 1, 2, \dots, n_1)$ ,  $(j = 0, 1, 2, \dots, n_2)$  and  $(k = 0, 1, 2, \dots, n_3)$ .

For remapping the parametric mesh onto the 3D source model, the vertices of the merged mesh  $M = (\zeta_k, \xi_k)$  can be divided into two groups: ‘ $P_1$  group’ and ‘non- $P_1$  group’. If a vertex of  $M = (\zeta_k, \xi_k)$  is an element of  $P_1 = (\zeta_i, \xi_i)$ , its corresponding vertex of  $S_1 = (x_i, y_i, z_i)$  can easily be found. If not, however, the vertex should be remapped to the 3D source surface with barycentric coordinates of  $P_1 = (\zeta_i, \xi_i)$ .

Remapping the merged mesh back onto the 3D target surface can be done in a similar manner except that this time the vertex set of the merged mesh  $M = (\zeta_k, \xi_k)$  is divided into ‘ $P_2$  group’ and ‘non- $P_2$  group’.

Unlike remapping of the vertex set, the edge and face sets of the parametric merged mesh can be directly used for both of the remapped meshes, since the structure of the meshes are identical.

Figure 6.15 illustrates comparison of the original source (a) and target meshes (b) to their remapped meshes (c and d respectively).

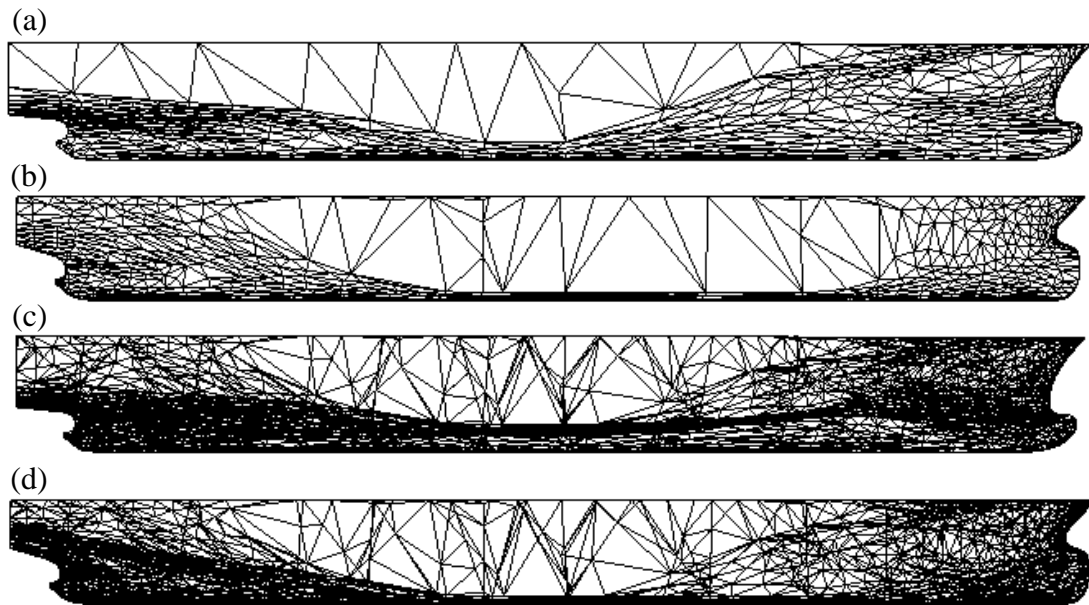


Figure 6.15 Original source and target meshes and remapped meshes: (a) original source; (b) original target; (c) re-meshed source; (d) re-meshed target.

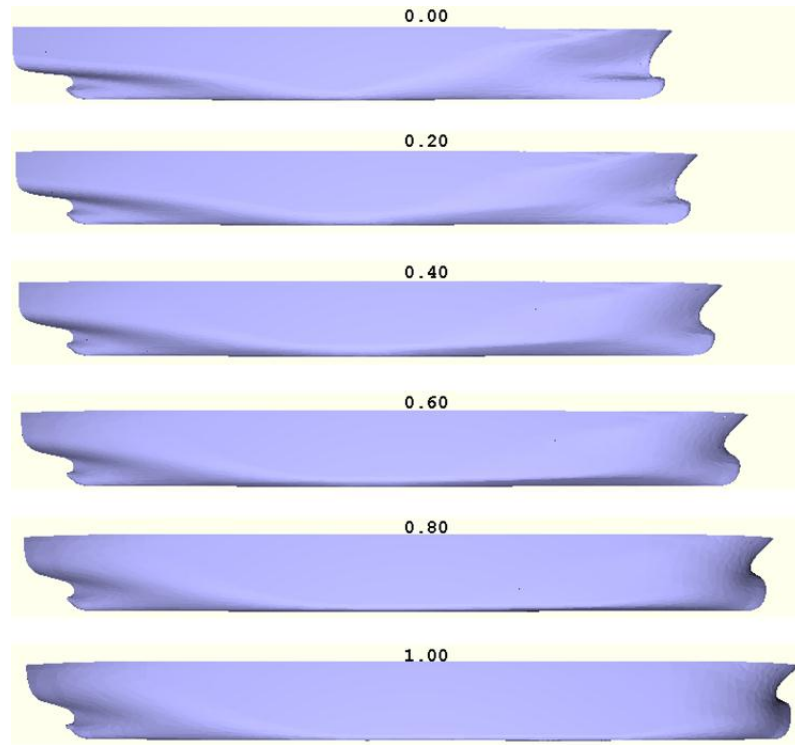
As shown in Figure 6.15, the mesh structures of the original source and target models are quite different; however, it is apparent that the two re-meshed models have an identical mesh structure.

### 6.3.2 Morphing

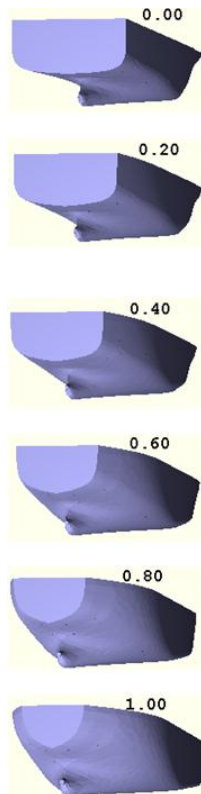
With the two re-meshed models that have an identical mesh structure, there is no problem now to carry out the linear morphing. This process is identical to the one used in the regularising method. Figure 6.16 shows examples of intermediate hull

forms in various views generated through hull form morphing ( $t = 0.0, 0.2, 0.4, 0.6, 0.8$  and  $1.0$ ) using the merging method. The source (corresponding to  $t = 0$ ) and the target ( $t = 1$ ) models are a 5,000 TEU container carrier and a 300,000 DWT VLCC respectively.

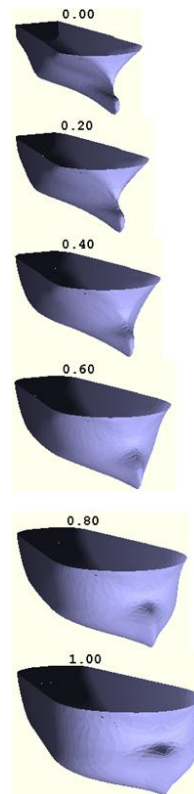




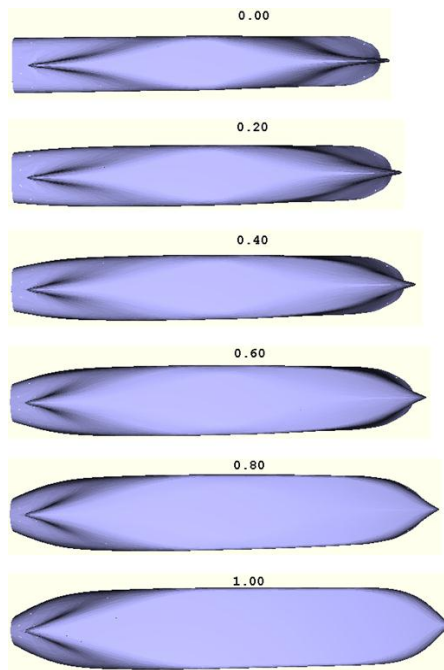
(a)



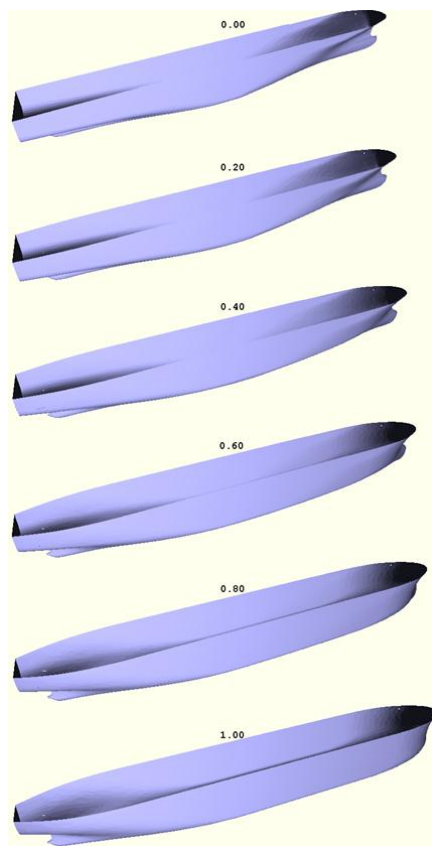
(b)



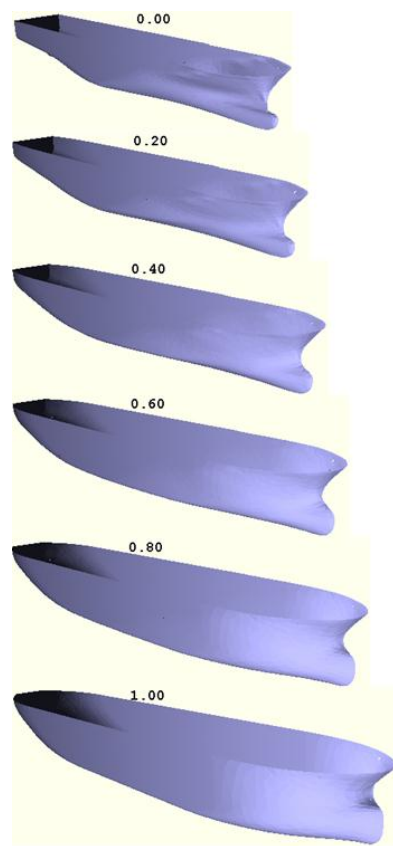
(c)



(d)



(e)



(f)

Figure 6.16 Hull form morphing using the merged method.

The boundary of the surface meshes must be selected with some care as strange results can be obtained from morphing otherwise. Consider the boundary of the original parent meshes which include the transom at deck followed by the transom at centreplane as shown in Figure 6.17. (It will be recalled that in the worked example of this section earlier the boundary used was the lower transom line.) As shown in Figure 6.18, the generated intermediate hull forms have chine lines on their transoms. It is quite obvious that something went wrong with the correspondence of the transom region. This happens because there was no specific correspondence for the chine line of the transom – the nodes on the chine line of one model may not necessarily correspond to those on the chine line of the other model. This problem would be easily solved by choosing these chine or knuckle lines as boundaries of surface meshes.

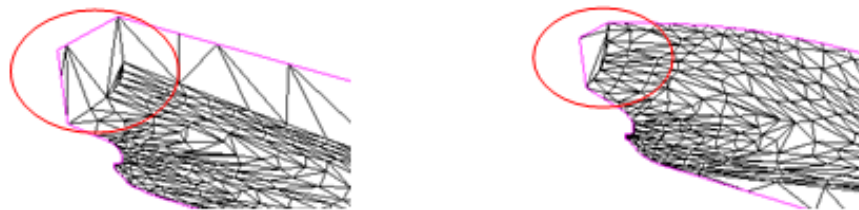


Figure 6.17 Selecting upper transom for the boundary.

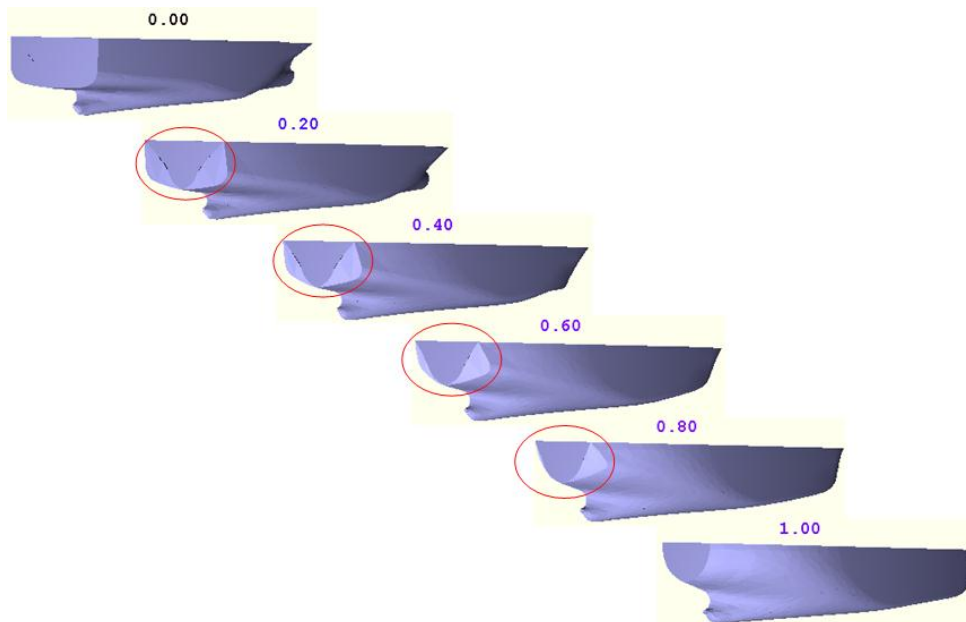


Figure 6.18 Strange transom resulting from improper boundary.

## 6.4 GENERATING EXTREME HULL FORMS

As has been the case with extreme airfoil forms, an ability to generate extreme hull forms can be useful, when there is no convenient hull form available for morphing. In Chapter 4 a method of generating a new extreme airfoil, represented by a polygonal curve, was introduced using the Laplacian differential coordinates. In this section this method is extended to generating a new extreme 3D surface from an existing one.

We need to consider two factors: boundary deformation and surface deformation. Since the boundaries of a surface consist of polygonal curves, the boundaries of the 3D surface can be deformed with the same method as used to generate extreme airfoils. With the convex combination coefficients of the internal nodes and the new boundaries thus obtained, the internal mesh properties are transferred to new positions so as to form a new surface. As the guide point pulls the points on a polygonal curve to their new positions to generate extreme shape of polygonal curve, the new boundaries pulls the surface to a new shape. Figures 6.19, 6.20 and 6.21 show an example of generating an extreme bulbous bow form from the original bulbous bow of 5,000 TEU container carrier.

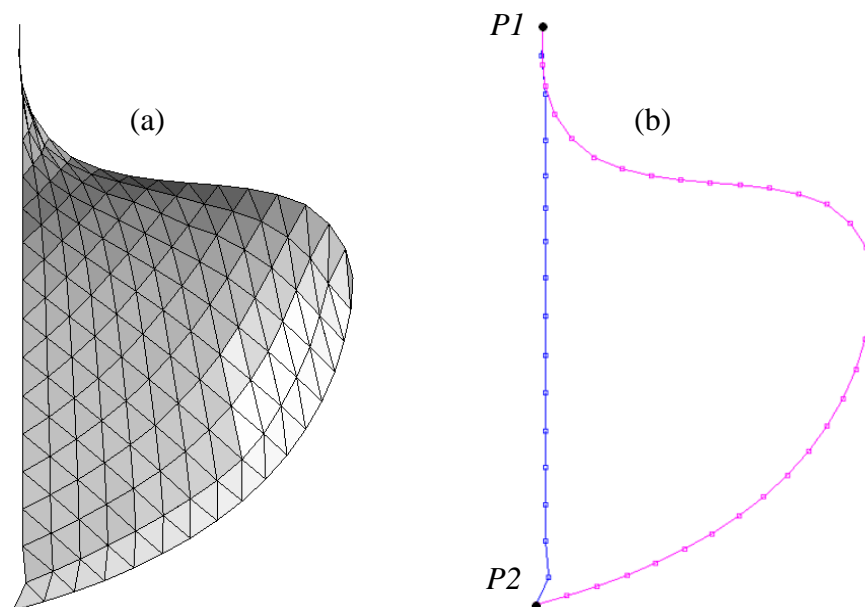


Figure 6.19 A bulbous bow model (a) and its boundary (b).

It will be recalled that in generating extreme airfoil shapes the leading and trailing edges were fixed at the original positions. In a similar manner points  $P1$  and  $P2$  are fixed at the positions shown, and these two points separate the boundary of the bulbous bow into two segments as shown in blue and magenta in Figure 6.19 (b). A new boundary for the extreme bulbous bow is generated by moving a point on the original boundary using the Laplacian differential coordinates. Figure 6.20 shows examples of newly generated boundaries.

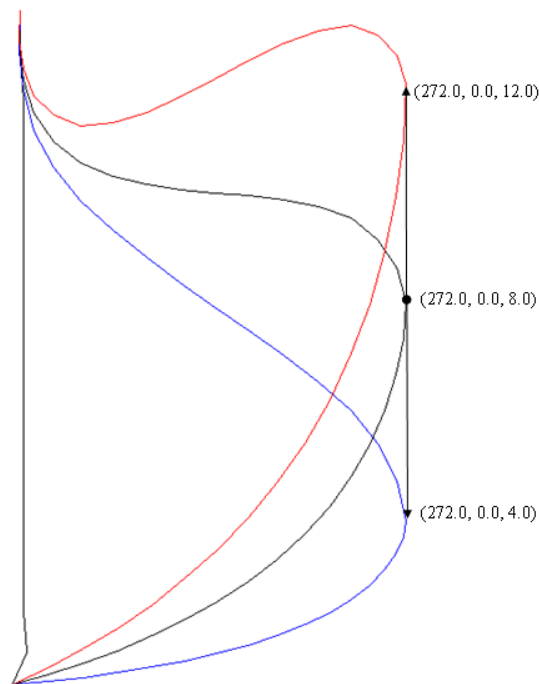


Figure 6.20 Generating new boundaries of the bulbous bow.

As shown in Figure 6.20, the red and blue boundaries are generated by moving a guide point  $(272.0, 0.0, 8.0)$  to  $(272.0, 0.0, 12.0)$  and  $(272.0, 0.0, 4.0)$  respectively. According to the positions of the guide point, it is possible to generate various new models.

The convex combination coefficients of the internal vertices of the surface can be calculated from Equation (3-13), and with the new boundaries thus created and the coefficients just obtained, the internal points of the surface can be calculated from

Equations (3-29) and (3-30). Figure 6.21 shows newly generated surface based on the boundaries created as shown in Figure 6.20.

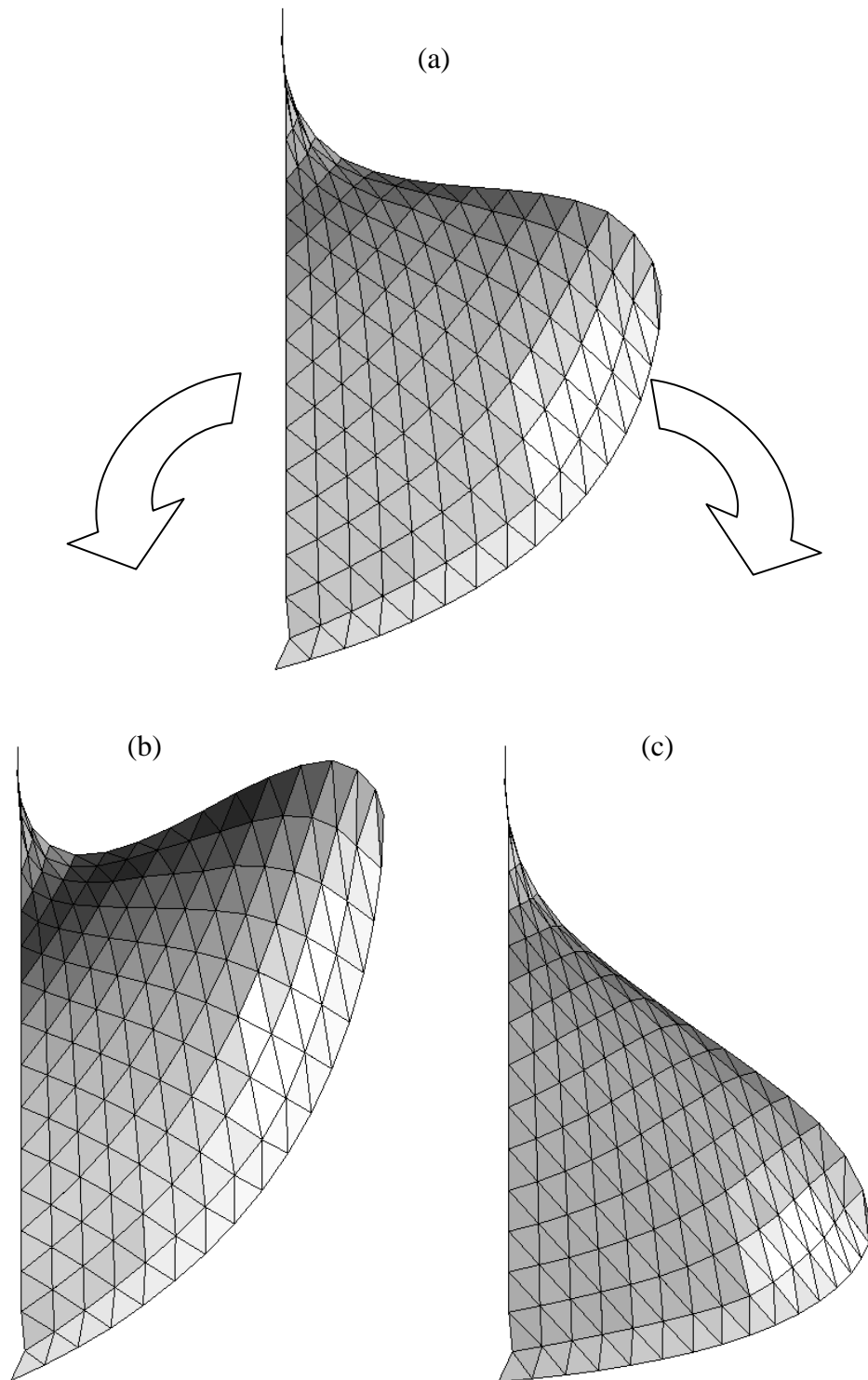


Figure 6.21 Newly generated bulbous bow: (a) Original bulbous bow:  
(b) New bulbous bow from the red boundary in Figure 6.20;  
(c) New bulbous bow from the blue boundary in Figure 6.20.

## 6.5 PARAMETRIC HULL FORM DESIGN USING MORPHING

It has been shown that numerous intermediate hull forms can be generated in a matter of seconds through morphing. By choosing a vessel with low block coefficient ( $C_b$ ) and another with high  $C_b$  as the source and target models respectively, it should then be possible to generate vessels of various  $C_b$  between the two values of block coefficient. Certain parameters, such as displacement, is decided at the early stages of design, and by the time a form is generated, such parameters may be more or less determined. It is, therefore, essential that the morphing technique described here should be able to fulfil the requirements of predetermined parameters whilst interpolating two extreme parent forms.

In this section a method of maintaining a predetermined volume displacement for all forms generated by morphing between two given parent hulls will be put forward. Essentially, this method relies on rescaling so that the resulting form has the specific displacement. First of all, morphing process is undertaken for a specific parameter  $t$  and the draught of the newly generated intermediate form must be determined. This can be done by interpolating between the draughts of the two parent hull forms and the draught will be a function of the morphing parameter  $t$ .  $L_{pp}$  can be calculated from the distance between the forward perpendicular (FP) and the after perpendicular (AP). AP and FP are found by identifying the points of intersection of the draught line with the facet edges, and then selecting the points with the minimum and the maximum x-component. The volume displacement of the ship is calculated by integrating the volume of each facet column. The facet column is the volume swept out by the facet through its projection onto the x-z plane, as shown in Figure 6.22.

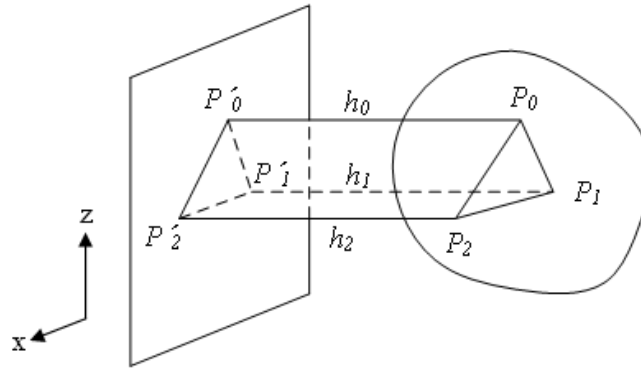


Figure 6.22 Volume of a polygon column.

Each column volume is evaluated by multiplying the projected area  $\Delta P'_0 P'_1 P'_2$  by the average offset of the facet ( $= \frac{h_0 + h_1 + h_2}{3}$ ).

Table 6.1 shows the obtained dimensions from the meshes during the basic morphing in which the source (corresponding to  $t = 0.00$ ) and the target ( $t = 1.00$ ) models are a 5,000 TEU containers carriers and a 300,000 DWT VLCC respectively.

Table 6.1 Calculated dimensions of the hull forms generated through basic morphing.

$T$	$C_b$	$L_{pp}$ (m)	Breadth (m)	Draught (m)	Displacement ( $m^3$ )
<b>0.00</b>	0.543	264.94	40.89	12.50	73,465
<b>0.01</b>	0.545	265.33	41.00	12.55	74,380
<b>0.02</b>	0.547	265.91	41.17	12.63	75,593
<b>0.03</b>	0.550	266.49	41.34	12.70	77,004
:	:	:	:	:	:
:	:	:	:	:	:
<b>0.98</b>	0.830	318.77	59.54	19.82	312,159
<b>0.99</b>	0.832	319.30	59.74	19.90	315,779
<b>1.00</b>	0.834	319.84	60.00	20.00	319,999

A myriad of hull forms are generated between  $C_b$ : 0.543 and  $C_b$ : 0.834, and each hull form has different volume displacement between  $73,465 m^3$  and  $319,999 m^3$ . In order to rescale the hull forms to have a predetermined displacement, it is easy to see that the vertices of the mesh for each hull form should be multiplied by the cubic root of the ratio of each displacement to the target displacement. If we wish to achieve a displacement of  $75,000 m^3$ , for example, for the intermediate hull form for morphing



parameter of 0.99, the displacement ratio is  $\frac{315,779}{75,000} = 4.210$ , and its cubic root is

$\sqrt[3]{4.210} = 1.615$ . Thus each vertex of the hull form on the morphing parameter ( $t=0.99$ ) is rescaled by multiplying the inverse of 1.615.

Table 6.2 shows newly obtained dimensions of the hull forms through the constrained hull form morphing, in which all hull forms are made to have a displacement of  $75,000\text{m}^3$ . Each dimension is calculated in the same way as used in compiling the Table 6.1.

Table 6.2 Calculated dimensions during the advanced hull form morphing in which the target displacement is  $75,000\text{ m}^3$ .

<i>t</i>	<i>Cb</i>	<i>Lpp (m)</i>	<i>Breadth (m)</i>	<i>Draught (m)</i>	<i>Displacement (m<sup>3</sup>)</i>
<b>0.00</b>	0.543	266.77	41.17	12.59	75,000
<b>0.01</b>	0.545	266.06	41.12	12.58	75,000
<b>0.02</b>	0.547	265.21	41.07	12.59	75,000
<b>0.03</b>	0.550	264.15	40.98	12.59	75,000
⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮
<b>0.98</b>	0.830	198.17	37.02	12.32	75,000
<b>0.99</b>	0.832	197.74	36.99	12.32	75,000
<b>1.00</b>	0.834	197.20	36.99	12.33	75,000

As shown in Tables 6.1 and 6.2, the block coefficients are dependent on the morphing parameter  $t$ ; length, breadth, and draught are all scaled to give the required displacement. In this thesis, it has been shown that displacement can be constrained during a hull form morphing; however, it would be also possible to constrain other hull form parameters such as  $L_{pp}$ , breadth, draught, depth and centres of flotation, buoyancy and gravity.

Figure 6.23 (a) shows the result of normal morphing between a 5,000 TEU container carrier and a 300,000 DWT VLCC, and Figure 6.23 (b) illustrates the example of the constrained morphing with the same source and target models with the target displacement of  $75,000\text{m}^3$ .

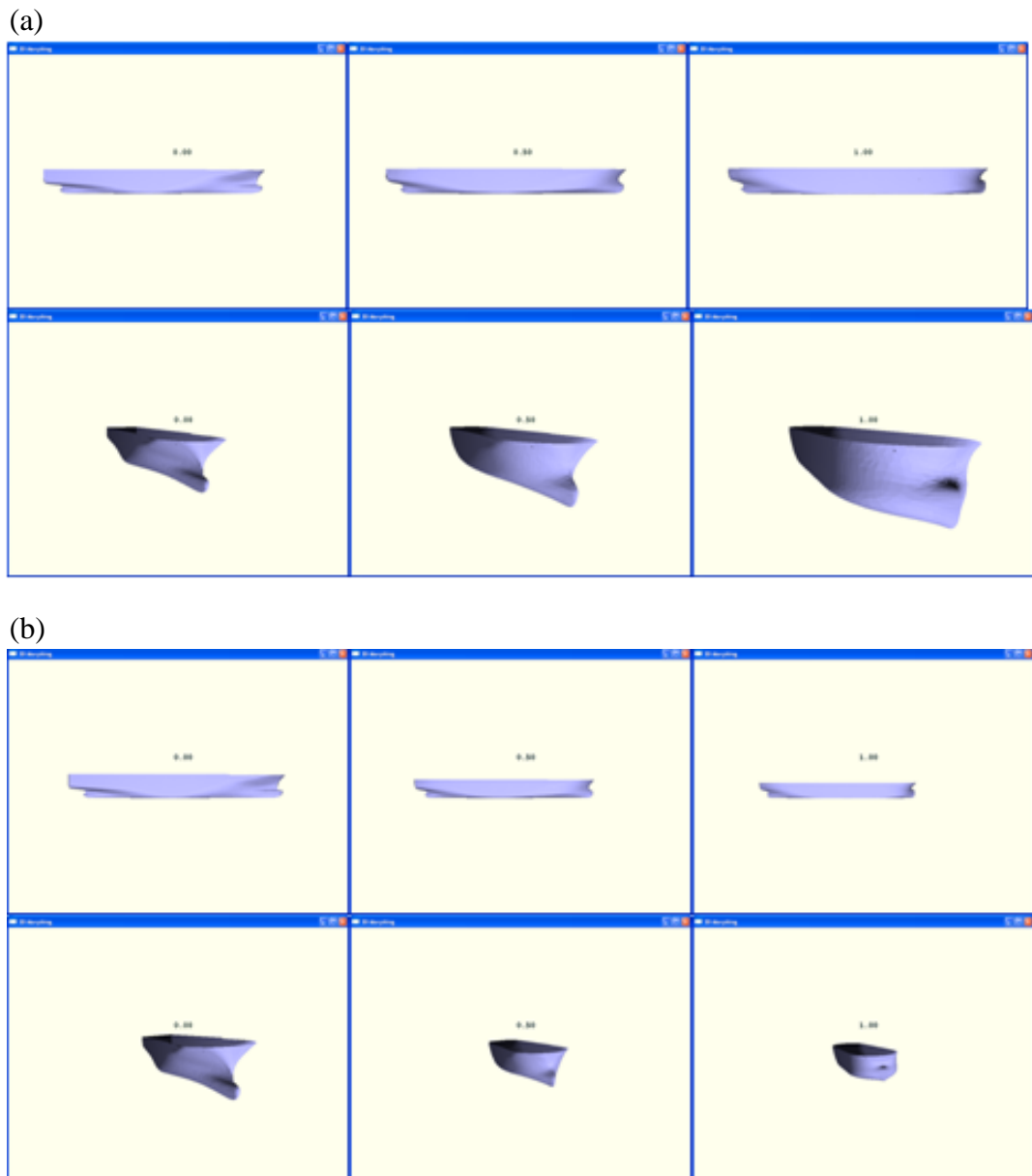


Figure 6.23 Comparison between (a) the normal and (b) constrained hull form morphings.

Alternatively, it is possible to ‘normalise’ the parent hull forms first, i.e. the main dimensions are made to unity. Once the intermediate form is generated for the required  $t$  value, the form can then be scaled back to give the correct displacement. Of course, this has a slight drawback in that it will be difficult to judge visually the appropriateness of the form thus generated.

## 6.6 A FURTHER EXAMPLE OF MORPHING – BETWEEN A MONO HULL AND A CATAMARAN HULL

So far the source and target models are of similar hull forms and consequently the result of morphing is not quite so dramatic. In order to illustrate how morphing can be used for generating ideas which are outside the ‘norm’ in ship design, two hull forms which differ widely are selected as source and target in this section: a mono hull and a catamaran hull.

The two hulls were panelised using Rhinoceros 4.0 program. The procedure of the morphing is exactly the same as used so far, and the regularising method is used for the parameterisation. In order to reduce the amount of manual manipulations required, only one triangular patch is used for the 2D parameterised domain of both models. The two models represented by a triangular mesh are shown in Figure 6.24.

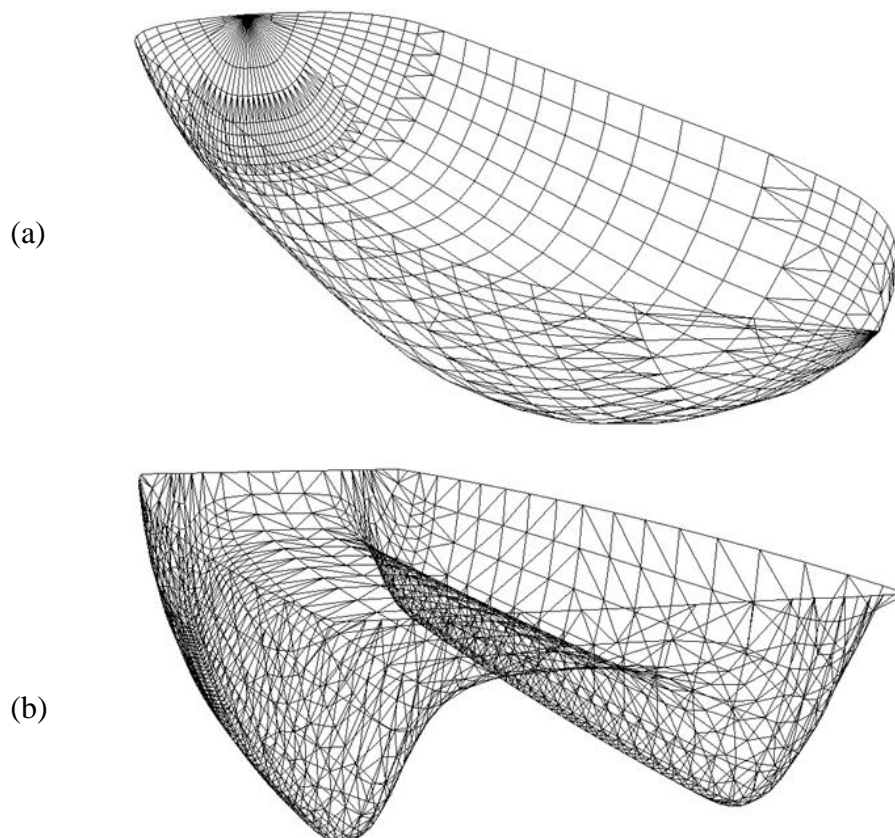


Figure 6.24(a) A mono hull and (b) a catamaran hull as the source and target models of the morphing.

In order to establish a proper correspondence between vertices of the two models, each original mesh needs to be parameterised onto a 2D triangular domain for the regularisation. Since the two hull forms are symmetrical about their centre plane, the port hull only is parameterised onto the 2D domain. Figure 6.25 shows the half hulls of the original models, where the magenta lines represent the corresponding boundaries of the 3D surfaces and the blue lines represent the 2D (parametric) triangular domain. Boundaries B1(a), B2(a), B3(a), B1(b), B2(b), and B3(b) are mapped onto the B'1(a), B'2(a), B'3(a), B'1(b), B'2(b), and B'3(b) respectively, and then the internal nodes are mapped onto the 2D domains. In order to reduce manual manipulations required in the subdivision tasks for the regularising method, only one 2D domain was used for parameterisation. Thus, the only logical candidate for the boundaries in this case was the outer boundaries of the mesh and these were used for the parameterisation. If, however, a designer wants to impose more boundaries, more subdivided patches would be required.

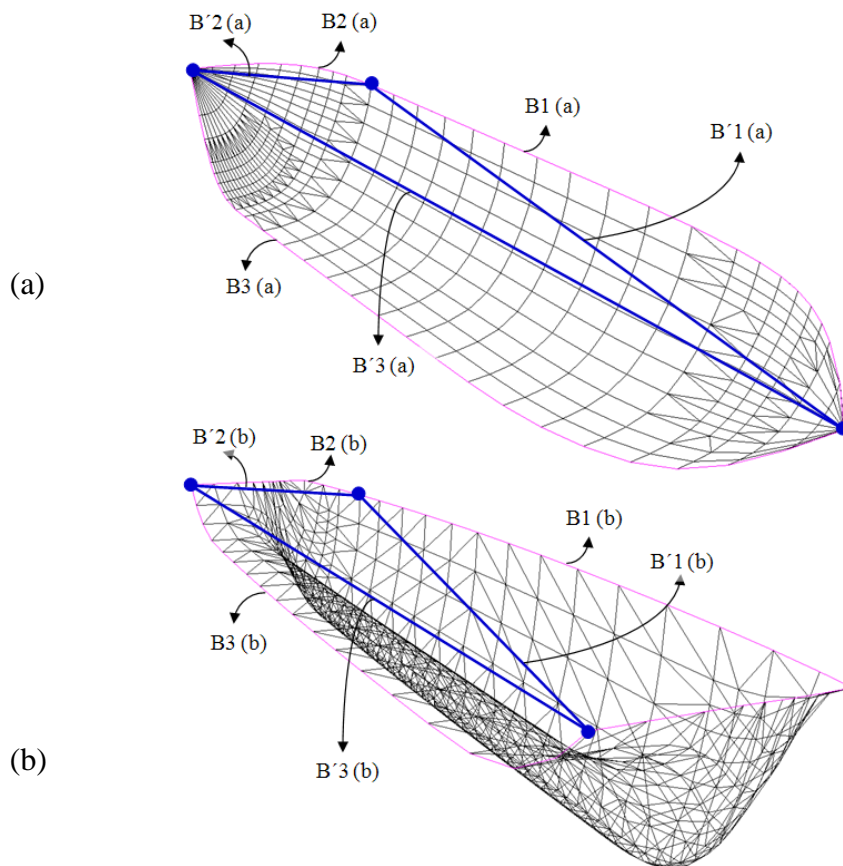


Figure 6.25 Establishing boundaries for the parameterisation.

The results of the parameterisation of each half-hull forms are shown in Figure 6.26.

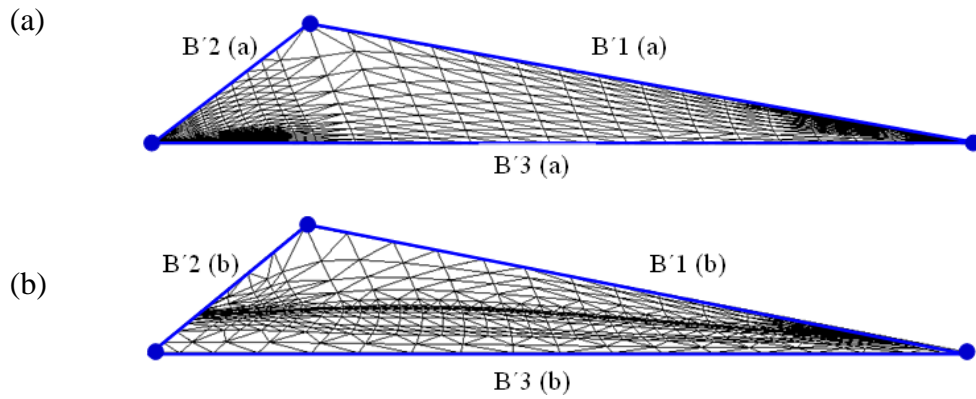


Figure 6.26 Parameterised mono hull (a) and catamaran hull (b).

From these, regularised meshes ( $4^6$  faces) in 2D are easily produced by repeatedly bisecting each edge as discussed earlier, and then each regularised mesh is re-mapped onto the 3D surfaces. Figure 6.27 shows the results of the re-mapping onto the 3D surface.

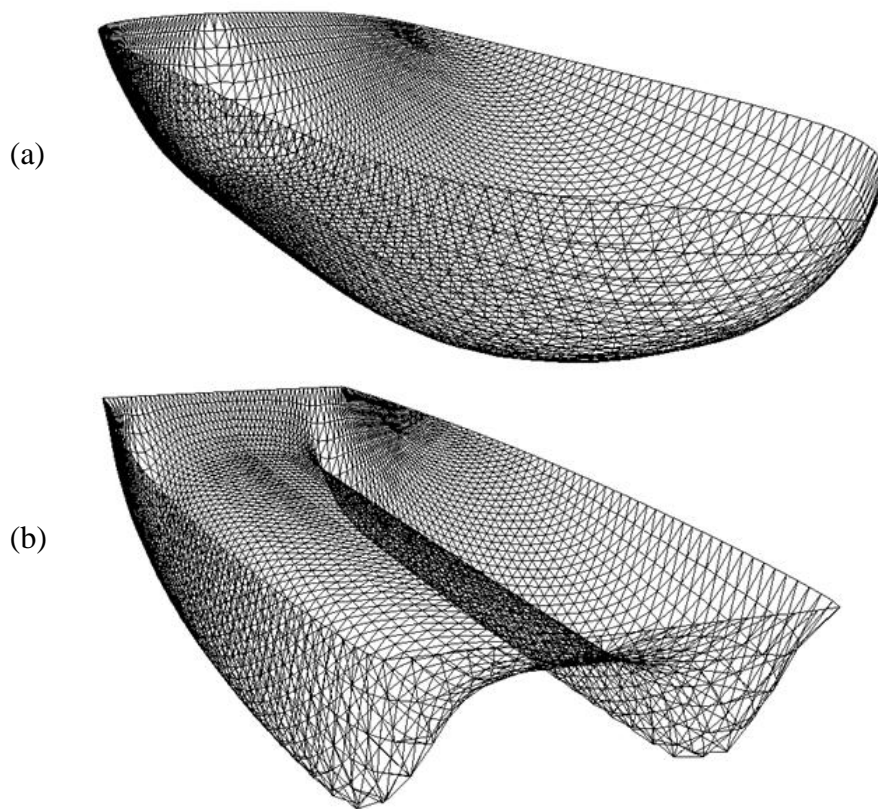
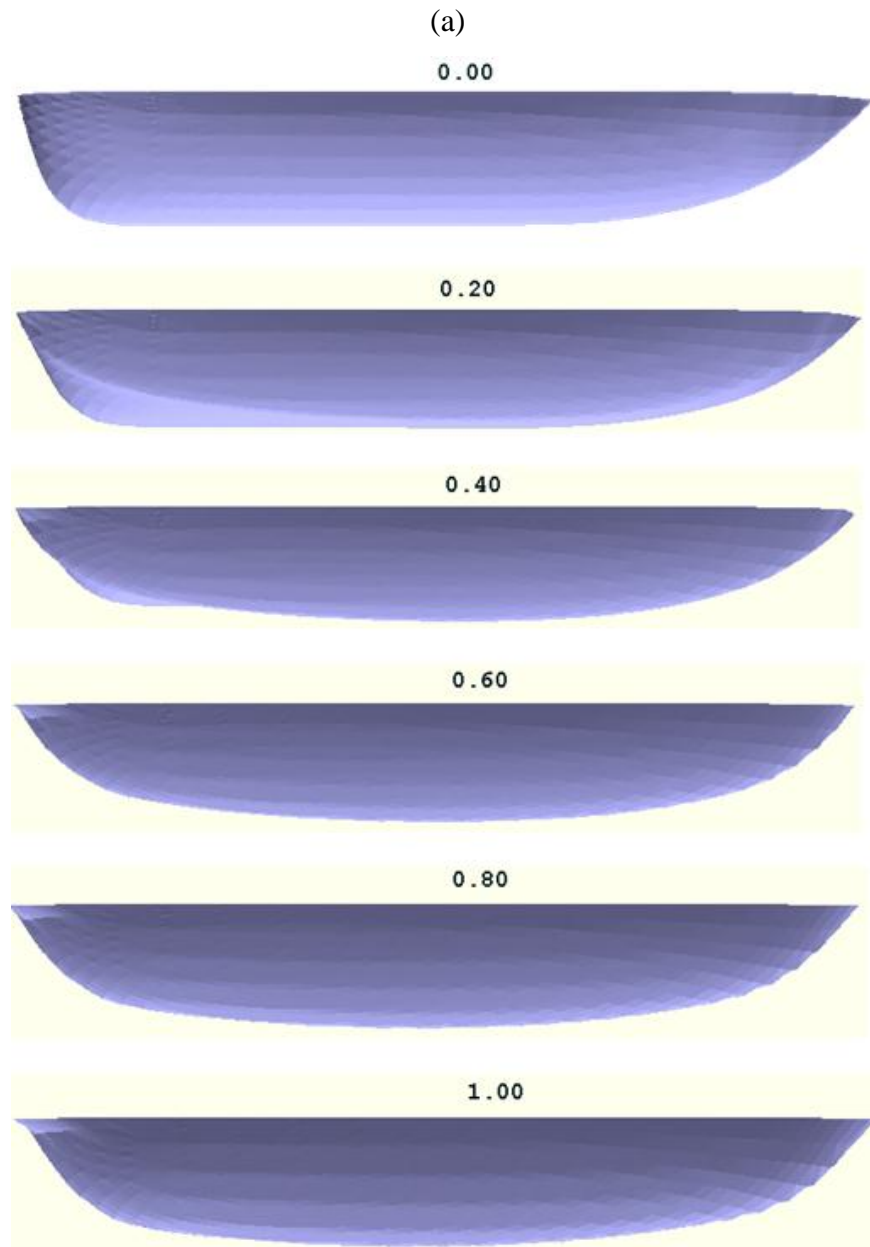
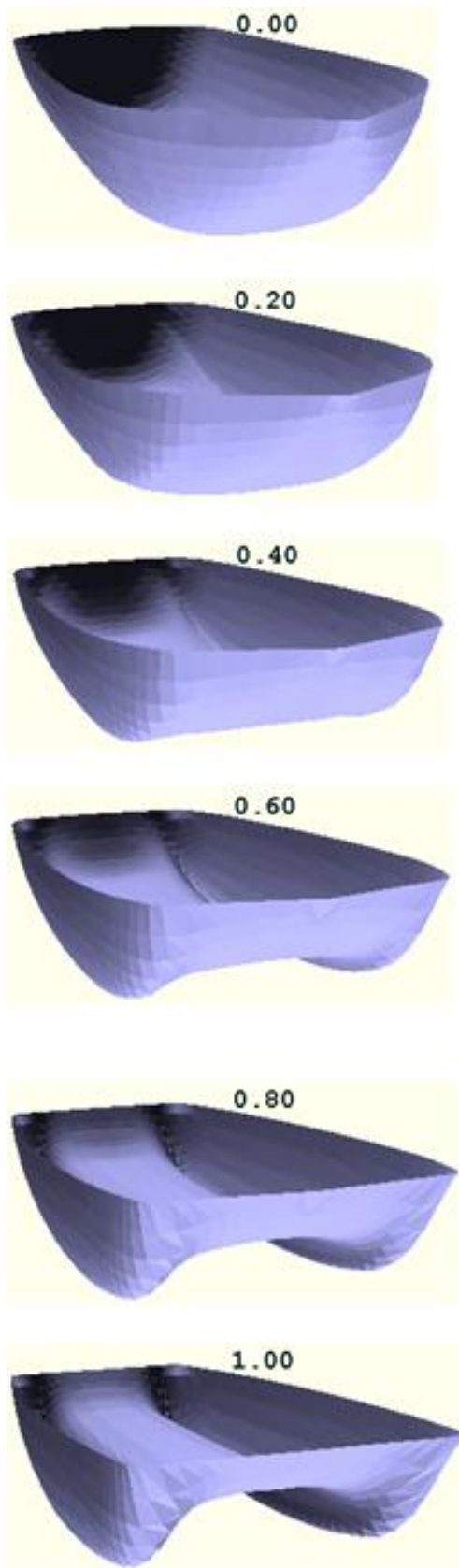


Figure 6.27 Regularised meshes ( $2 \times 4^6$  faces for each).

Morphing can now be carried out between these two models, and the results are shown in Figure 6.28 in a variety of views. Since these two meshes are re-mapped from only one 2D triangular domain without any subdivision as in the example of the previous hull form morphing between two container carriers, the correspondence of similar regions between two models are not so obvious. Nevertheless, this exercise will serve as an example of morphing as a powerful tool of idea generation.



(b)





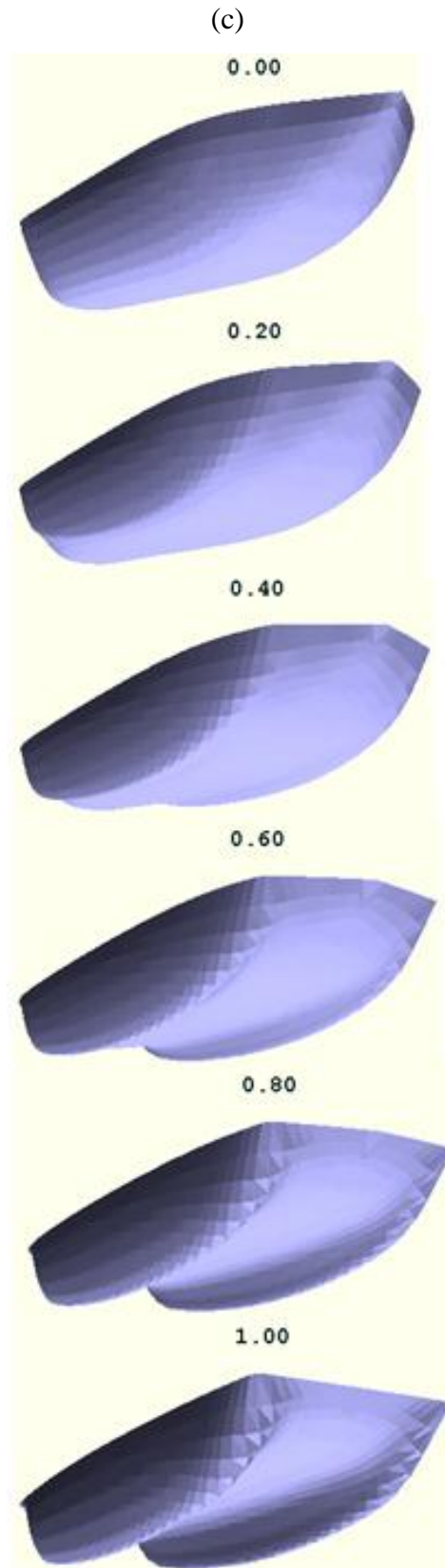


Figure 6.28 Morphing between the mono hull and the catamaran hull.



## 6.7 CONCLUDING REMARKS

This chapter demonstrated the application of the morphing technique to hull form generation. Through a few case studies, it was shown that a vast number of intermediate models can be rapidly generated through the proposed morphing processes.

It was found that the most time-consuming task of morphing procedure is remeshing process to establish correspondence between source and target models. When the corresponding meshes are ready, however, the intermediate models can be generated almost instantaneously on demand.

Using a PC with Intel Pentium 4 CPU 3.00GHz and 1GB of RAM, it took approximately 1, 3, 5, 28 and 68 seconds for regularising a subdivided patch of 125 vertices to  $4^2$ ,  $4^3$ ,  $4^4$ ,  $4^5$  and  $4^6$  triangles respectively. For merging a container ship model of 550 vertices and 993 faces and a VLCC model of 559 vertices and 993 faces, it took less than 3 minutes for two whole models to be remeshed to new 3D surfaces with 4900 vertices and 9788 faces.

It is an interesting prospect to consider morphing between two hull forms, one with very good resistance characteristics but moderate sea-keeping properties and the other with excellent sea-keeping qualities but mediocre resistance characteristics. Would one expect to achieve some hull forms with rather good qualities in both aspects? This idea leads us to consider morphing as a form generation tool equivalent or in parallel to the genetic algorithm in optimisation process.

# Chapter 7.

## DISCUSSION

---

### 7.1 NOVELTY OF WORK

Morphing is an interpolation technique normally used for creating a smooth transition between two objects, and consequently the technique has been regarded as more or less exclusive property of computer graphics. To the author's knowledge morphing technique has seldom been used for product design application in 3D shapes, and never been applied to systematic variations of ship hull forms. The main idea of this thesis was the possibility that the ability of this technique in rapid generation of intermediate objects could be used for form design. It has been amply shown that this idea is not only feasible, but can make a significant addition to the armoury of designers' tools.

The regularisation method is also a new concept for establishing a correspondence instead of using merging method. As mentioned in Chapter 6, the regularising method is quicker than the merging method as a remeshing algorithm itself. However, the former requires more manual manipulations, and longer time for the whole morphing procedure ultimately. Therefore, in terms of time taken, it can be said that the merging method is better than the regularising method, although the difference is really marginal. Since, on the other hand, the regularising method is more attractive from the viewpoint of data preparation and the resulting mesh structure is more useful for computational programs (hydrostatics, CFD, FEA, etc), it is believed that

the method has more potential. It will be well worth while, in this regard to explore and develop a method or a computer program which can undertake the regularising tasks with minimum manual manipulations.

## **7.2 PREDICTABILITY OF MORPHED FORMS**

In this thesis, the method of applying the morphing technique to generating forms, in particular hull forms, was explored. It has been shown through examples that morphing allows an infinite number of fairly ‘predictable’ forms to be generated very quickly. The forms thus generated are as detailed and complete as the parent forms are. Normally, generating even relatively simple ship hull forms would demand a great deal of effort and time. Morphing, therefore, has a good potential for removing one of the ‘bottlenecks’ of design optimisation.

Morphing works on a principle of interpolating between two extreme forms. Therefore, it is obvious that the ‘morphed’ forms will inherit the geometric characteristics of the parent forms. The proximity to either of the parents, of course, will depend upon the morphing parameter. This predictability can be quite helpful in design, as the designer will be able to control the shape. The expectation here is that in inheriting the geometrical characteristics of the parents, the morphed forms will also inherit their performance characteristics. One can use this feature and ‘cross’ many different hull forms which have shown to perform particularly well to produce superior hulls, in a very similar manner as plant or animal breeders work.

The predictability, whilst it is helpful in one sense, is disadvantageous in another way. The shapes produced from two parents will be within the bounds of what is expected, limiting the freedom of creating imaginative forms. Perhaps it can be concluded that the morphing technique should be used only when an approximate boundary of eligible forms are determined. Nevertheless, an effort was made to provide extra control ‘handles’ to the designer in the form of a multi-target morphing method. This does help, but it is thought that more work will be required to address this limitation.

### **7.3 EASE OF USE**

The morphing technique was found to be very easy to use for relatively simple shapes, such as airfoil cross sections. The fact that they are 2D polygonal shapes certainly helps in this respect. In the 2D case studies presented in Chapters 4 and 5 the positions of the nodes were determined by morphing. The smooth curves connecting these nodes were generated using cubic splines, since it is the most useful form for interpolating between a number of points.

Working with ship hull forms, it was found that the most onerous task was defining the parent hull forms with triangular panels. In order to represent the smooth curved surface with panels one would require quite a number of panels, although it was not explored in this thesis how many panels would be required to preserve the characteristic shapes. The next problem was establishing the correspondence between the nodes of the two parent forms.

These two difficulties necessitated the introduction of a ‘panel regularising’ stage into the process. The idea was simply to produce many more panels of similar dimensions in such a way as to enhance the form representation. The correspondence was easily established by simply making the number of panels of the two forms identical.

However, this process still has some pitfalls, such as having to subdivide the hulls in a broadly similar way for both parent forms. The panel merging method was introduced to simplify this process and it was largely successful. Because of the differences in the number and distribution of panels used, the morphed hull forms of the two methods may have some differences; however, both methods allow the designers to establish a correspondence of vertices between two different 3D free-form surfaces.

## **7.4 SPEED OF GENERATION**

In computer-generated animations, transition has to be smooth and as seamless as possible and the morphing technique is used to interpolate between a starting model and an ending model. This is possible, since the morphing technique can generate any number of intermediated models sufficiently rapidly.

In product design, this speed of generating the morphed models is very attractive and useful. In this thesis, it has been found that the morphed models could be generated almost instantaneously even for sophisticated hull forms.

Therefore, the technique is not only a good modelling tool for design optimisation as mentioned above, but also it can be an effective tool for design change management. Morphing can allow the designer to respond to consumer demand quickly, and by reducing the time required for design modification, the product development cycle can be shortened.

## **7.5 SIMPLICITY OF DEFINING MODELS**

Conventional parametric hull form generation methods require several parameters, for instance, longitudinal centre of buoyancy, prismatic coefficient, aft-body prismatic coefficient, and fore-body prismatic coefficient are needed for Lackenby method [1]. Still, the number of parameters required in the Lackenby method is nothing compared to more modern conventional parametric hull design methods ([29] – [38], [82], [83]). Some of them require more than 15 parameters only for bulbous bow alone. These many parameters can be helpful for representing the hull forms precisely; but, they will make generating many hull forms awkward and slow.

Morphing method, on the other hand, does not necessarily require as many parameters. It is possible to define a hull form from two parent models with only one morphing parameter. Even multi-target morphing requires only the same number of

parameters as the number of target models as described in Chapter 4. This simplicity of defining models may allow the algorithm to be easily associated with other programs for numerical performance analyses.

## **7.6 VERSATILITY OF MORPHING TECHNIQUES**

This thesis has examined the possibility of using the morphing technique for generating a large number of 2D airfoil section profiles and hull forms which are 3D free-form surfaces. This means that the morphing technique can be used for any shapes of objects which can be represented by meshes or polygonal curves. Therefore, the technique would be useful for rapid generation of not only hull forms but also any other surface forms.

Besides product designs, the technique would be very useful in various fields such as medical science, biology, geology and so on. Changes of geologic structures and biological, genetic or organic evolutions [84] in nature may be examined through the morphing technique. For example, Penska *et al.* [85] introduced a medical application of morphing for illustrating the success and/or failure of the physiological healing process as a dynamically morphed video.

Since the morphing technique described in this thesis is based on meshes, this method can be a useful data preparation tool for many computational programs such as finite element analysis, hydrostatics, CFD and possibly 3D rendered visualisation. In addition, the regularising method illustrated here can produce either high quality mesh or coarser meshes according to the requirements, and this can minimise numerical errors or save calculating time.

## **7.7 CREATIVITY OF DESIGN**

It is well-known that many inventors are inspired by nature. For example, in the 16<sup>th</sup> Century, a type of warship generally known as the ‘turtle ship’ was designed and

built in Korea, making a crucial contribution in repelling the Japanese invasion of Korea. It is easy to see that the outward appearance of this ship shown in Figure 7.1, of course, was inspired by turtles.



Figure 7.1 A model of the turtle ship.

The predictability of a morphed form has been discussed in Section 7.1. Despite this it was found that two widely different shape models, such as a mono hull and a catamaran hull forms, can be morphed as introduced in Chapter 6. Thus, it will also be possible that a hull form is interpolated with a dolphin, a shark, a whale or even any insects. The generated models from these morphings may not be suitable for a commercial ship; but, at least this does open up many possibilities.

## 7.8 SUGGESTIONS FURTHER WORK

In defining the 2D curves or 3D surfaces no curve/surface fitting/generation technique was used in this thesis. An interesting idea is to discover what would happen to morph B-spline curves. Morphing, of course, will be done on the corresponding sets of control vertices, in other words it will be an indirect morphing

in a way. One would guess that controlling the curve shapes will not be quite as straightforward as with cubic splines, but it will be worth investigating. A further development from this idea is to morph between two NURBS surfaces by morphing the control vertex nets and not the surfaces themselves directly. Unfortunately, time limitation did not allow this interesting possibility to be explored in this thesis.

It has been found that automating the mesh subdivision process for the regularising method is a difficult problem, and consequently it has been done manually using Rhinoceros 4.0 program in this thesis. However, it is worth paying some attention to in the future. This will undoubtedly save time and minimise the human error during the mesh subdivision process.

Ship hull form fairing has been discussed by many researchers ([86] – [113]). However, the fairing task is still one of costly processes in ship hull design stage and usually requires a manual procedure by trained and experienced personnel. In this thesis, unfortunately, the focus was on generating models, and the curvatures of morphed surfaces have not been considered here. Since there have been some work on the mesh smoothing and fairing ([114] – [107]), these methods can be developed and applied for the hull form fairing in the future. If the fairing tasks can be carried out automatically during the morphing process, it would be possible to reduce much time at hull design stage.

It has been described that original models can be warped using Laplacian differential coordinates in this thesis, and this method controlled a node to establish a new boundary. The new surface is generated from the new boundary, and therefore boundaries such as profile lines or half breadth lines are required for the warping. If, however, a warping method that can transform directly the surface without boundary procedure is developed, the warping procedure would be much simplified.

In this thesis, only the linear morphing was demonstrated, and the paths of vertices in the linear morphing are always straight lines. However, nonlinear morphing will be able to introduce curvi-linear paths, and these paths can offer yet more handles to the



designers. In addition, it may be possible to impose different morphing speeds to different regions of the model. For example, when morphing between two hull forms, fore-body and aft-body can be given different morphing speeds. Figure 7.2 illustrates an example of imposing different morphing parameter on fore-body and aft-body. When the morphing parameter is near to zero, the fore-body approaches the target model faster than the aft-body with increasing morphing parameter. On the other hand, when the morphing parameter is near to one, the rate of change of the aft-body is greater than the fore-body.

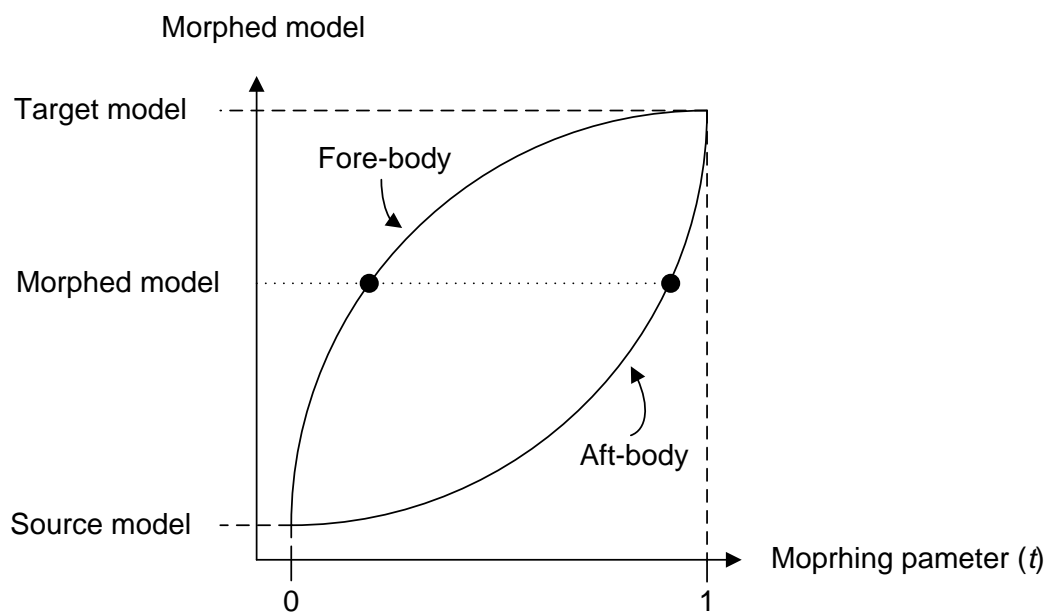


Figure 7.2 Example of imposing different morphing speed.

Similarly, it would be possible to apply a reverse morphing parameter; for instance, morphing starts with  $t = 0$  in the aft-body and  $t = 1$  in the fore-body, and goes to  $t = 1$  and  $t = 0$  respectively. Thus, this approach may be able to create a mixture of the two hulls over the length of the vessel.

Morphing is usually defined as graphical interpolation technique between two models. However, it would also be worth delving into the possibility of extrapolating based on two or more models. This could be used as one of warping algorithms, and this may be useful in generating model forms in its own right.

# Chapter 8.

## CONCLUSIONS

---

Based on the research work presented in the foregoing, the following conclusions can be drawn:

1. Morphing technique, which has not been used in product design to date, is introduced as a potentially powerful design tool. It has been found that the morphing technique can generate a large number of models rapidly, possibly enabling a large number of design alternatives to be considered for optimisation.
2. Establishing the correspondence between the vertices of the source and the target models has proved to be one of the most important problems in morphing process. This was found to be relatively straightforward for 2D polygonal curves, such as airfoil section profiles. However, for free form surfaces, such as ship hull forms, an additional process of mesh regularising or mesh merging is necessary. The methods developed and discussed in this thesis can provide a good correspondence between vertices of any shapes of models, and this allows a wide application of the morphing technique for product design.
3. The regularising method is based on repeated bisection of each edge of the mesh elements. By imposing the same rule to the source and target models, it

is possible to establish a correspondence between the two models. This method allows the control of the fineness of the mesh. Data preparation for mesh-based computational programs, such as 3D rendered visualisation, hydrostatics calculation, FEA and CFD, will be possible with this method.

4. Merging method establishes a correspondence between vertices of the source and target models by merging two meshes to form a new mesh which has both mesh elements of the two models. Thus, this method can preserve the original mesh shape and save much time by reducing the need for manual manipulations. The less manual manipulations will be quite helpful and potential for automatic design and optimum design.
5. The idea of multi-target morphing was introduced in order to overcome the predictability of morphing based on two parents only and to allow the designer more freedom to control the forms.
6. The warping method using Laplacian differential coordinates can be used to generate extreme forms which can be employed as one or more parent forms when a suitable model is not available. Furthermore, this technique can be also quite useful as a design tool in its own right. The warping method undoubtedly makes the morphing technique potentially more powerful.
7. There is no doubt that the morphing technique is very useful, powerful, reliable and efficient as a design tool, and it has a great potential for designs of ships and other products.

**REFERENCES**

- [1] Lackenby H. On the systematic geometrical variation of ship forms. Transaction INA, 92, p. 289-316. 1950.
- [2] SMF - Garland M. Garland's Simple Mesh Format (SMF). <http://www.mgarland.org/>
- [3] Taylor DW. Calculations of Ships' Forms and Light Thrown by Model Experiments upon Resistance, Propulsion Rolling of Ships, Intl Congress of Engineering, San Francisco, 1915.
- [4] Benson FW. Mathematical Ships' Lines. Trans R.I.N.A., Vol 82, 1940.
- [5] Kerwin J. Polynomial Surface Representation of Arbitrary Ship Forms. Journal of Ship Research, Vol. 4, No. 12, 1960.
- [6] Von Kerczek C. The Representation of Ship Hulls by Conformal Mapping Functions, Journal of Ship Research, Vol. 13, No. 4, December 1969.
- [7] Kuiper G. Preliminary Design of Ship Lines by Mathematical Methods. Journal of Ship Research, Vol. 14, 1970.
- [8] Reed AM, Nowacki H. Interactive creation of fair ship lines. Journal of Ship Research, Vol. 18(2), 1974.
- [9] Schoenberg IJ. Contributions to the problem of approximation of equidistant data by analytic functions. Part A: On the problem of smoothing or graduation. A first class of analytic approximation formulae. Quart. Appl. Math., Vol IV, No. 1, p. 45-99, 1946.
- [10] Schoenberg IJ. Contributions to the problem of approximation of equidistant data by analytic functions. Part B—on the problem of osculatory interpolation. A second class of analytic approximation formulae. Quart. Appl. Math., Vol IV, No. 2, p. 112-141, 1946.
- [11] Mehlum E. Nonlinear splines. Computer Aided Geometric Design, p. 173-205, 1974.
- [12] Forrest AR. Interactive Interpolation and Approximation by Bézier Polynomials. Computer Journal, Vol. 51, p. 71-79, 1971.

- [13] Denman HH. Smooth Cubic Spline Interpolation Functions. *Industrial Mathematics, J. Ind. Math. Soc.* Vol. 21, 1971.
- [14] Cline AK. Curve Fitting using Spline under Tension. *Atoms, Tech*, No. 3, 1973.
- [15] Kim SY. Park DH. Hull Form Representation using Mathematics representation and Computer Aided Design. In Korean, PNU Press, Busan, South Korea, 2002.
- [16] Bézier P. General distortion of an ensemble of biparametric patches. *Computer Aided Design*, 10(2): p. 116-20, 1978.
- [17] De Boor C. CADRE: An algorithm for numerical quadrautre. In *Mathematical software*, J. R. Rice, Ed. Academic Press, New York and London, p. 201-209.
- [18] Cox MG. The numerical evaluation of B-splines. *J. Inst. Math. Appl.*, Vol. 10, p.134-149, 1972.
- [19] Riesenfeld RF. Application of B-spline Approximation to Geometric Problems of Computer Aided Design, Doctoral dissertation, Syracuse University, 1972.
- [20] Cordon W. Riesenfeld RF. B-spline curves and surfaces. in: Barnhill R.E. Riesenfeld R.E. eds., *Computer Aided Geometric Design*, Academic Press, New York, p.95-126.
- [21] Rogers DF, Adams JA. *Mathematical Elements for Computer Graphics*, McGraw-Hill Inc, 1976.
- [22] Wu DW, Lin Y, Ji ZS. Ship hull surface's NURBS representation and design (in Chinese), *Journal of Dalian University of Technology*, 42(5), p. 569-573, 2002.
- [23] Zhang MX. Study of hull damage stability calculation with NURBS technology (in Chinese), Doctoral dissertation, Department of Naval Architecture and Ocean Engineering, Dalian University of Technology, Dalian, China, 2002.
- [24] Lu CH, Lin Y, Ji ZS. Ship hull representation with a single NURBS surface. The 15<sup>th</sup> international offshore and polar engineering conference. Seoul, Korea, 2005.

- [25] Zhang J, He Q, Feng S. Theory of NURBS and its application in submarine hull modelling. *Advanced Computer Control (ICACC), 2010 2<sup>nd</sup> International Conference, Shenyang, China, March, 2010.*
- [26] Creutz G, Schubert C. Interactive Curve Creation from Form Parameters by Means of B-splines, *Schiffstechnik, Vol. 25, 1978.*
- [27] Munchmeyer FC, Schubert C, Nowacki H. Interactive design of fair hull surfaces using B-splines. In: *Computer applications in the automation of shipyard operation and ship design III. IFIP. North-Holland publishing Company; 1979.*
- [28] Yuille IM. The Forward Design System for Computer Aided Ship Design using a Mini-Computer, *Trans R.I.N.A., Vol. 120, 1978.*
- [29] Harries S, Abt, C. Parametric Curve Design Applying Fairness Criteria. *International Workshop on Creating Fair and Shape-Preserving Curves and Surfaces, Network Fairshape, Berlin/Potsdam; published at Teubner, 1998.*
- [30] Harries S. Parametric Design and Hydrodynamic Optimisation of Ship Hull forms. *Doctoral Dissertation, Technische Universität Berlin; Mensch & Buch Verlag, 1998.*
- [31] Abt C, Bade SD, Birk L, Harries S. Parametric Hull Form Design – A Step Towards One Week Ship Design. *8<sup>th</sup> International Symposium on Practical Design of Ships and Other Floating Structures, PRADS 2001, Shanghai 2001.*
- [32] Harries S, Valdenazzi F, Abt C, Viviani U. Investigation on Optimization Strategies for the Hydrodynamic Design of Fast Ferries, *FAST'01, Southampton 2001.*
- [33] Valdenazzi F, Harries S, Abt C, Viviani U. Abt C. Seakeeping Optimisation of Fast Vessels by Means of Parametric Modeling. *HSMV02, Naples 2002.*
- [34] Maisonneuve JJ, Harries S, Raven HC, Marzi J, Viviani U. Towards optimal design of ship hull shapes. *IMDC 03, Athens 2003.*
- [35] Harries S. Systematic optimization – A Key for improving ship hydrodynamics. *Hansa, Vol. 12, December 2005.*
- [36] Abt C, Harries S. FRIENDSHIP Framework – integrating ship-design modelling, simulation, and optimization. *The Naval Architect, RINA, 2007.*

- [37] Abt C, Harries S. A new approach to integration of CAD and CFD for naval architects. The 6<sup>th</sup> International Conference on Computer Applications and Information Technology in the Maritime Industries (COMPIT 2007), Cortona, 2007.
- [38] Kim HC. Parametric Design of Ship Hull Forms with a Complex Multiple Domain Surface Topology. Doctoral dissertation, Technische Universität Berlin, 2004.
- [39] Keane AJ. A Computer Based Method for Hull Form Concept Design: Application to stability Analysis. Trans RINA., Vol. 130, 1988.
- [40] Alef WE, Collatz G. Computer aided design of ship's lines by nonlinear distortion of parent forms. ICCAS, 76, p. 157-163, 1976.
- [41] Versluis. Computer aided design of ship form by affine transformation. International Shipbuilding Progress, 24(274), p. 147-160, 1977.
- [42] Rabien V. Ship Surface design by transforming given mesh representations. In: ICCAS, p. 85-93, 1979.
- [43] Duvigneau R, Visonneau M. Single- and Multi-objective optimization for high-fidelity CFD using genetic algorithms. Eurogen 2001, Evolutionary Method for Design, Optimization and Control, September, Athens, Greece.
- [44] Sederberg TW, Parry SR. Free-Form Deformation of Solid Geometric Models. Computer Graphics, 20(4), 151-160, 1986.
- [45] Duvigneau R, Visonneau M, Deng GB. On the role played by turbulence closures for hull shape optimization at model and full scale. 4<sup>th</sup> ONR Symposium on Naval Hydrodynamics, Fukuoka, Japan, July 2002.
- [46] Chen E, Parent R. Shape Averaging and Its Applications to Industrial Design. IEEE Computer Graphics and Applications, 9, 1 p. 47-54, 1989.
- [47] Hsiao SW, Chang MS. A Semantic Recognition-based Approach for Car's Concept Design. International Journal of Vehicle Design, Vol. 18, p. 83-82, 1997.
- [48] Hsiao SW, Liu MC. A morphing method for shape generation and image prediction in product design. Design Studies 2002.
- [49] Kang JY, Lee BS. Mesh-based morphing method for rapid hull form generation, Computer-aided Design. Vol. 42, p. 970 – 976, 2010.

- [50] Kang JY, Lee BS. An investigation of parametric rolling and its reduction through geometric design. GreenTech - International Conference on Marine Science & Technology for Green Shipping, Glasgow, p. 131-141, 2009.
- [51] Compact Oxford English Dictionary, Free online dictionary resources from Oxford University Press. <http://www.askoxford.com/>
- [52] The Biology Corner - Biology Teaching Resources. <http://www.biologycorner.com/>
- [53] Chen L, Wang GF, Hsiao K, Liang J. Affective product shapes through image morphing. In: Proceedings of the 2003 international conference on designing pleasurable products and interfaces, 2003. p. 11-16.
- [54] Terzides C. Transformational Design. Knowledge Aided Architectural Problem Solving and Design, NSF Project #DMC-8609893, Final Report 1989.
- [55] Saitou K, Rusák Z, Horváth I. Rapid concept generation and evaluation based on vague discrete interval model and variational analysis. In Proceedings of TMCE 2006. 2006. p. 149-58
- [56] Kent J, Parent R, Carlson W. Establishing Correspondences by Topological Merging: A New Approach to 3-D Shape Transformation. Proceedings of Graphics Interface '91. p. 271-278.
- [57] Kent J. Carlson W. Parent R. Shape transformation for polyhedral objects. In: SIGGRAPH '92. p. 47-54, 1992.
- [58] Tutte WT. Convex Representations of Graphs. Proc. London Math. Soc. 10, p. 304-320.
- [59] Floater M. Parametrization and smooth approximation of surface triangulations. Computer Aided Geometric Design. Vol. 14, p. 231-250, 1997.
- [60] Welch W, Witkin A. Free-form shape design using triangulated surfaces. Computer Graphics, SIGGRAPH, p. 247-256, 1994.
- [61] Gotsman C, Sheffer A, Gu X. Fundamentals of Spherical Parameterization for 3D Meshes. ACM Trans. On Graphics, 22, 2003.
- [62] Steiner D, Fischer A. Planar Parameterization for Closed Manifold Genus-g Meshes Using Any Type of Positive Weights. Journal of Computing and Information Science in Engineering, Vol. 5(2), p. 118-126, June 2005.



- [63] Zorin D, Schröder P, Sweldens W. Interactive multiresolution mesh editing. In SIGGRAPH 97 Conference Proceedings, p. 259-268, 1997.
- [64] Shewchuk JR. Tetrahedral mesh generation by delaunay Refinement. In proceedings of the 14<sup>th</sup> Annual Symposium on Computational Geometry, p. 86-95, 1998.
- [65] Desbrun M, Meyer M, Schröder P, Barr A. Implicit fairing of irregular meshes using diffusing and curvature flow. In SIGGRAPH 99 Conference Proceedings, p. 317-324, 1999.
- [66] Alexa M. Differential coordinates for local mesh morphing and deformation. *The Visual Computer* Vol. 19(2), p. 105-114, 2003.
- [67] Lipman Y, Sorkine O, Cohen-OR D, Rössl C, Seidel HP. Differential coordinates for interactive mesh editing. In proceedings of Shape Modeling International, IEEE Computer Society Press, p. 181-190, 2004.
- [68] Sorkine O, Lipman Y, Cohen-OR D, Alexa M, Rössl C, Seidel HP. Laplacian surface editing. In Symposium on Geometry Processing, ACM SIGGRAPH / Eurographics, p. 179-188, 2004.
- [69] Munk MM. Elements of the Wing Section Theory and of the Wing theory. NACA Rep. No. 191, 1924.
- [70] Griffiths RT. The effect of aerofoil characteristics on windmill performance. *Aeronautical Journal*, Vol. 81, July 1977, p. 322-326.
- [71] Morcos VM. Aerodynamic performance analysis of horizontal axis wind turbines. *Renewable Energy*, Vol. 4(5), July 1994, p. 505-518.
- [72] Kamoun B, Afungchui D, Chauvin A. A wind turbine blade profile analysis code based on the singularities method. *Renewable Energy*, Vol. 30(3), March 2005, p. 339-352.
- [73] Perez F, Pérez-Rojas L. Hydrodynamic Considerations of Rotor Blades of Marine Current Turbines. *GreenTech - International Conference on Marine Science & Technology for Green Shipping*, 2009, p. 56-68.
- [74] Drela M, Youngren H. XFOil. <http://web.mit.edu/drela/Public/web/xfoil/>
- [75] UltraEdit – Text editor program. <http://www.ultraedit.com/>
- [76] Betz A. Schraubenpropeller mit Geringstem Engergieverlust. Doctoral dissertation, Göttingen, 1919.

- [77] Tangler JL, Somers DM. NREL Airfoil Families for HAWTs. NREL/TP-442-7109, Jan. 1995.
- [78] Coiro DP, Nicolosi F, De Marco A, Melon S, Montella F, Comstock JP. Dynamic Behavior of Novel Vertical Axis Tidal Current Turbine: Numerical and Experimental Investigations. ISOPE Conference, Seoul, June 2005.
- [79] Molland AF, Bahaj AS, Chaplin JR, Batten WMJ. Measurement and prediction of forces, pressures and cavitation on 2-D sections suitable for marine current turbines. Proc I Mech, Part M 2004.
- [80] Rhinoceros – NURBS modelling for Windows. <http://www.rhino3d.com/>
- [81] IVCON – 3D Graphics File conversion. [http://orion.math.iastate.edu/burkardt/g\\_src/ivcon/ivcon.html](http://orion.math.iastate.edu/burkardt/g_src/ivcon/ivcon.html)
- [82] Kracht AM. Design of Bulbous Bows. SNAME Transactions, Vol. 86, 1978.
- [83] Zhang P. Parametric Approach to Design of Hull Forms. Journal of Hydrodynamics, Vol. 20, No. 6, p. 804-810, 2008.
- [84] Darwin C. On the Origin of Species, UK, 24 Nov 1859.
- [85] Penska K, Folio Les, Bungler R. Medical Applications of Digital Image Morphing. Journal of digital imaging, Vol. 20, No. 3, p. 279-283, 2007.
- [86] Miller NS, Kuo C. The Mathematical Fairing of Ship Lines. European Shipbuilding, Vol. 12, No. 4, 1963.
- [87] Hattori Y, Matida Y. Some problems in practical improvement of mathematical fairing. Proceedings of Computer Aided Hull Surfaces Definition Symposium, SCHAD Vol. 77, p. 61-70, 1977.
- [88] Kjellander JAP. Smoothing of Cubic Parametric Splines. Computer Aided Design, Vol. 15, p. 175-179, 1983.
- [89] Nowacki H, Reese D. Design and fairing of ship surfaces. In: Barnhill, R.E., Boehm, W. (Eds.), Surfaces in Computer Aided Geometric Design. North-Holland, Amsterdam, p. 121-134, 1983.
- [90] McCallum KJ, Zhang JM. Curve-smoothing techniques using B-splines. The Computer Journal, Vol. 29, No. 6, p. 564-569, 1986.
- [91] Horsham, W. Recent Advances in Fairing Technology. Marine and Offshore Computer Applications, p. 401-411, 1988.

- [92] Farin G, Sapidis N. Curvature and the Fairness of Curves and Surfaces. *IEEE Computer Graphics and Applications*, Vol. 9. p. 52-57, 1989.
- [93] Nowacki H, Dingyuan L, Xinmin L. Mesh Fairing GC Surface Generation Method. In: *Theory and Practice of Geometric Modelling*, SpringerVerlag, Berlin p. 93-108, 1989.
- [94] Sapidis N, Farin G. Automatic Faring Algorithm for B-Spline Curves. *Computer Aided Design*, Vol. 22, p. 121-129, 1990.
- [95] Rando T, Roulier JA. Designing faired parametric surfaces, *Computer-Aided Design*, Vol. 23, No. 7, p. 492-497, 1991.
- [96] Liu JP, Koyama T, Yamato H. Constrained Smoothing B-spline Curve Fitting for Ship Hull Generation and Fairing. *Proceedings of Computer Applications in the Automation of Shipyard Operation and Ship design VI*, Amsterdam: Elsevier Science Publishers, p. 221-232, 1991.
- [97] Huanzong R, Gang C, Weirong Z. Nonuniform B-Spline Mesh Fairing Method. *Proceedings of Computer Applications in the Automation of Shipyard Operation and Ship design VI*, Amsterdam: Elsevier Science Publishers, p. 261-272, 1991.
- [98] Moreton HP, Sequin CH. Functional Optimisation for Fair Surface Design. *Computer Graphics*, Vol. 26, p. 167-176, 1992.
- [99] Park S, Lee K. Fairing B-spline surfaces using optimization technique, *Proceedings of 6<sup>th</sup> International Conference on Engineering Computer Graphics and Descriptive Geometry*, p. 209-213, ASEE, 1994.
- [100] Roulier JA, Rando T. Measures of Fairness for Curves and Surfaces. *Designing Fair Curves and Surfaces: Shape Quality in Geometric Modelling and Computer Aided Design*, Philadelphia: SIAM, p. 75-122, 1994.
- [101] Nowacki H, Lü X. Fairing Composite Polynomial Curves with Constraints. *Computer Aided Geometric Design*, Vol. 11, p. 1-15, 1994.
- [102] Pigounakis KG, Sapidis N, Kaklis PD. Fairing Spatial B-Spline Curves. *Journal of Ship Research*, Vol. 40. p. 351-367, 1996.
- [103] Pigounakis KG, Sapidis N, Kaklis PD. Convexity Preserving Fairing. *Computer Aided Design*, Vol. 28, p. 981-994, 1996.

- [104] Narli E, Sariöz K. Fairing of High Speed Displacement Hull Forms by B-spline Approximation and Fitting. *Naval Engineers Journal*, Vol. 110, p. 35-47, 1998.
- [105] Hahmann S. Shape Improvement of Surfaces. *Journal of Computing*, Vol. 13, p. 135-152, 1998.
- [106] Harries S, Nowacki H. Form parameter approach to the design of fair hull shapes, *Proceedings of 10<sup>th</sup> international conference on computer applications in shipbuilding*, 1999.
- [107] Polkakoff JF, Wong YK, Thomas PD. An Automated Curve Fairing Algorithm for Cubic B-spline Curves. *Journal of Computational and Applied Mathematics*, Vol. 102, p. 73-85, 1999.
- [108] Kantorowitz E, Drabkin R, Krits A. Fitting correctly shaped splines to ship lines given by inaccurate points. *Ship Technology Research* Vol. 47, No. 2, p. 63-66, 2000.
- [109] Yang X, Wang G. Planar Point Set Fairing and Fitting by Arc Splines. *Computer Aided Design*, Vol. 33, p. 35-43, 2001.
- [110] Narli E, Sariöz K. The automated fairing of ship hull lines using formal optimisation methods. *Turkish Journal Engineering and Environmental Sciences*, Vol. 28, p. 157-166, 2004.
- [111] Kovibia A, Pasadas M. Approximation of surfaces by fairness bicubic splines. *Advances in Computational Mathematics* Vol. 20, p. 87-103, 2004.
- [112] Katsui T, Norisada K. Fundamental Study for Automatic Ship Hull Form Fairing. In Japanese, *Journal of the Japan Society of Naval Architects and Ocean Engineers*, No. 2, p. 161-168, 2005.
- [113] Sariöz E. An optimization approach for fairing of ship hull forms. *Ocean Engineering*, Vol. 33, p. 2105-2118, 2006.
- [114] Taubin G. A signal Processing Approach to Fair Surface Design. In *Proceedings of Siggraph'95*, p. 351-358, 1995.
- [115] Desbrun M, Meyer M, Shroder P, Barr AH. Implicit Fairing of Irregular Meshes Using Diffusion and Curvature Flow. In *Proceedings of Siggraph'99*, p. 317-324, 1999.

- [116] Desbrun M, Meyer M, Shroder P, Barr AH. An isotropic Feature-preserving Denosing of Height Fields and Bivariate Data. In Proceedings of Graphics Interface'00 p. 145-152, 2000.
- [117] Ohtake Y, Belyaev A, Bogaeovski I. Mesh Regularization and Adaptive Smoothing. Computer-Aided Design, Vol. 33, No. 11, p. 789-800, 2001.
- [118] Liu X, Bao H, Shum HY, Peng. A Novel Volume Constrained Smoothing Method for Meshes. Graphics Models, Vol. 64, No. 3-4, p. 169-182, 2002.
- [119] Jones T, Durand F, Desbrun M. Non-iterative, feature preserving mesh smoothing. In Proceedings of SIGGRAPH' 03, San Diego, p. 943-949, 2003.
- [120] Zhou K, Bao H, Shi J. 3D Surface Filtering Using Spherical Harmonics. Computer-Aided Design, Vol. 34, No. 4, p. 363-375, 2004.
- [121] Qin X, Chen X, Zhang S, Wang W. EMD based fairing algorithm for mesh surface. Computer –Aided Design and Computer Graphics, CAD/Graphics '09, 11<sup>th</sup> IEEE International Conference, p. 606-609, 2009.
- [122] Comstock JP. Principles of naval architecture (Revised). Society of Naval Architects and Marine Engineers. New York. 1967.

# APPENDICES

---

## **APPENDIX A. COORDINATES OF THE OPTIMISED AIRFOILS**

Here, the tables represent the coordinates of the benchmarked and optimised airfoils for each case as mentioned in Chapter 5. Their coefficients of lift and drag are also illustrated.

**A.1 Targeting S826 (95% radius)**

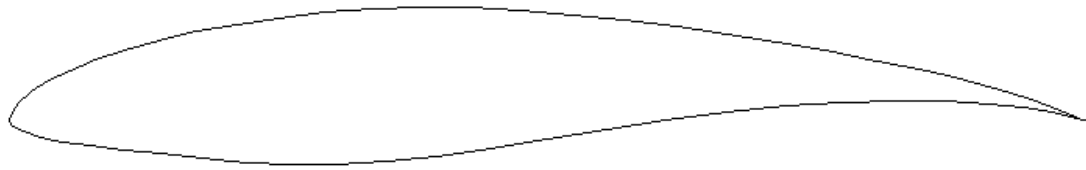


Figure A.1 Profile of S826.

Table A.1 Coordinates of S826.

x/c	y/c	x/c	y/c	x/c	y/c	x/c	y/c
1.00000	0.00000	0.42801	0.10281	0.00000	0.00001	0.44764	-0.02565
0.99664	0.00099	0.38225	0.10326	0.00020	-0.00142	0.50009	-0.01711
0.98706	0.00427	0.33732	0.10158	0.00095	-0.00269	0.55462	-0.00827
0.97231	0.00983	0.29357	0.09830	0.00217	-0.00399	0.61036	0.00005
0.95333	0.01704	0.25146	0.09364	0.00367	-0.00522	0.66629	0.00722
0.93053	0.02495	0.21142	0.08779	0.01364	-0.01034	0.72131	0.01279
0.90379	0.03277	0.17391	0.08088	0.02918	-0.01517	0.77428	0.01647
0.87274	0.04040	0.13930	0.07308	0.04994	-0.01958	0.82406	0.01811
0.83765	0.04810	0.10796	0.06453	0.07577	-0.02357	0.86955	0.01769
0.79901	0.05584	0.08022	0.05539	0.10632	-0.02722	0.90956	0.01522
0.75733	0.06354	0.05631	0.04581	0.14129	-0.03072	0.94270	0.01128
0.71316	0.07108	0.03649	0.03599	0.17988	-0.03473	0.96824	0.00697
0.66706	0.07831	0.02084	0.02613	0.22007	-0.03865	0.98610	0.00328
0.61962	0.08506	0.00955	0.01655	0.26153	-0.04077	0.99657	0.00085
0.57143	0.09112	0.00255	0.00758	0.30497	-0.04063	1.00000	0.00000
0.52305	0.09628	0.00017	0.00165	0.35010	-0.03808		
0.47506	0.10028	0.00000	0.00010	0.39764	-0.03282		

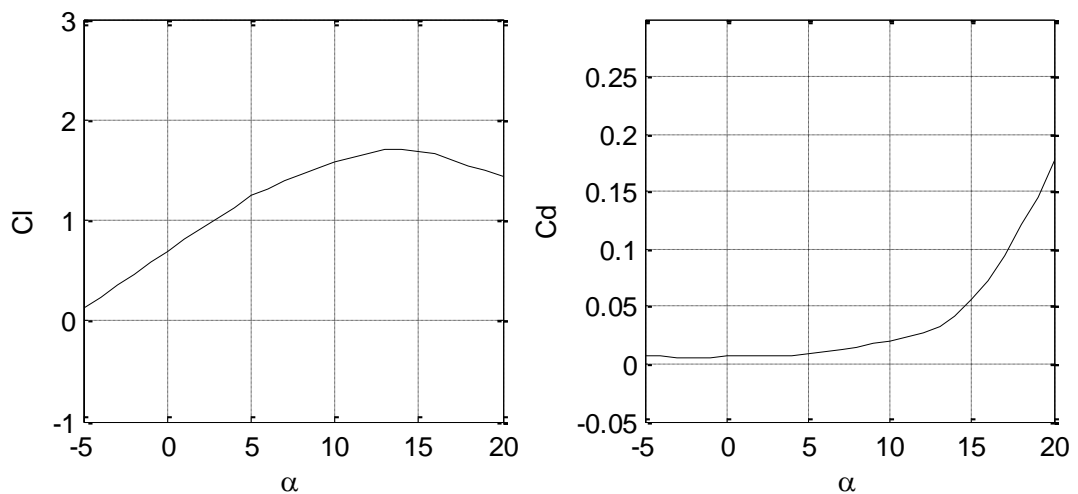


Figure A.2 Lift and drag coefficients of S826 ( $R_n = 1.5 \times 10^6$ ).

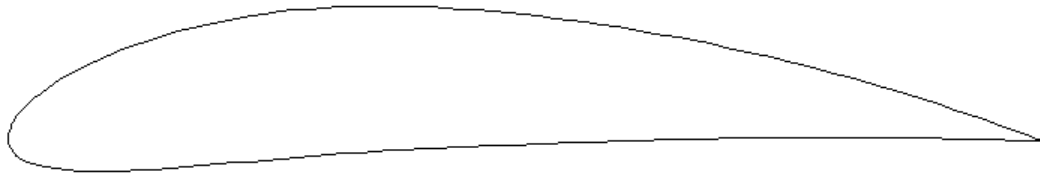


Figure A.3 Profile of N6409-0.140.

Table A.2 Coordinates of N6409-0.140.

x/c	y/c	x/c	y/c	x/c	y/c	x/c	y/c	x/c	y/c	x/c	y/c
1.0000	0.0000	0.6177	0.1035	0.2202	0.1165	0.0000	0.0021	0.2339	-0.0193	0.6480	0.0018
0.9945	0.0021	0.6027	0.1060	0.2061	0.1139	0.0001	0.0003	0.2491	-0.0180	0.6635	0.0021
0.9841	0.0059	0.5877	0.1084	0.1922	0.1110	0.0005	-0.0021	0.2644	-0.0167	0.6789	0.0024
0.9716	0.0104	0.5727	0.1107	0.1782	0.1078	0.0013	-0.0048	0.2796	-0.0154	0.6943	0.0026
0.9579	0.0153	0.5577	0.1129	0.1644	0.1043	0.0025	-0.0073	0.2947	-0.0142	0.7097	0.0029
0.9436	0.0202	0.5427	0.1149	0.1507	0.1004	0.0040	-0.0097	0.3099	-0.0130	0.7251	0.0031
0.9291	0.0251	0.5276	0.1168	0.1372	0.0963	0.0059	-0.0120	0.3250	-0.0119	0.7405	0.0032
0.9145	0.0299	0.5126	0.1185	0.1238	0.0918	0.0081	-0.0141	0.3401	-0.0108	0.7559	0.0033
0.8999	0.0346	0.4976	0.1202	0.1106	0.0870	0.0109	-0.0161	0.3551	-0.0099	0.7713	0.0034
0.8853	0.0392	0.4826	0.1217	0.0976	0.0819	0.0142	-0.0180	0.3702	-0.0090	0.7867	0.0034
0.8706	0.0437	0.4675	0.1230	0.0848	0.0763	0.0181	-0.0198	0.3853	-0.0082	0.8021	0.0034
0.8559	0.0481	0.4524	0.1243	0.0726	0.0705	0.0229	-0.0216	0.4005	-0.0074	0.8175	0.0033
0.8411	0.0524	0.4374	0.1254	0.0609	0.0644	0.0288	-0.0234	0.4158	-0.0067	0.8329	0.0032
0.8263	0.0566	0.4224	0.1264	0.0499	0.0581	0.0361	-0.0250	0.4313	-0.0060	0.8483	0.0031
0.8115	0.0606	0.4076	0.1271	0.0399	0.0517	0.0451	-0.0265	0.4468	-0.0054	0.8637	0.0029
0.7967	0.0646	0.3928	0.1277	0.0314	0.0456	0.0560	-0.0276	0.4624	-0.0047	0.8791	0.0027
0.7819	0.0685	0.3782	0.1281	0.0244	0.0401	0.0684	-0.0284	0.4780	-0.0040	0.8945	0.0025
0.7671	0.0722	0.3637	0.1283	0.0188	0.0351	0.0819	-0.0287	0.4935	-0.0034	0.9099	0.0022
0.7522	0.0758	0.3492	0.1282	0.0143	0.0306	0.0961	-0.0286	0.5090	-0.0028	0.9252	0.0019
0.7373	0.0794	0.3348	0.1279	0.0107	0.0265	0.1109	-0.0282	0.5244	-0.0021	0.9405	0.0016
0.7224	0.0828	0.3203	0.1274	0.0078	0.0228	0.1259	-0.0275	0.5399	-0.0016	0.9556	0.0012
0.7075	0.0861	0.3059	0.1266	0.0056	0.0195	0.1412	-0.0267	0.5554	-0.0010	0.9701	0.0008
0.6926	0.0893	0.2915	0.1256	0.0038	0.0165	0.1567	-0.0256	0.5708	-0.0005	0.9834	0.0005
0.6776	0.0923	0.2772	0.1243	0.0024	0.0136	0.1722	-0.0245	0.5863	0.0001	0.9942	0.0002
0.6627	0.0953	0.2628	0.1227	0.0014	0.0108	0.1877	-0.0232	0.6017	0.0005	1.0000	0.0000
0.6477	0.0982	0.2486	0.1209	0.0007	0.0079	0.2032	-0.0219	0.6172	0.0010		
0.6327	0.1009	0.2343	0.1189	0.0002	0.0052	0.2186	-0.0206	0.6326	0.0014		

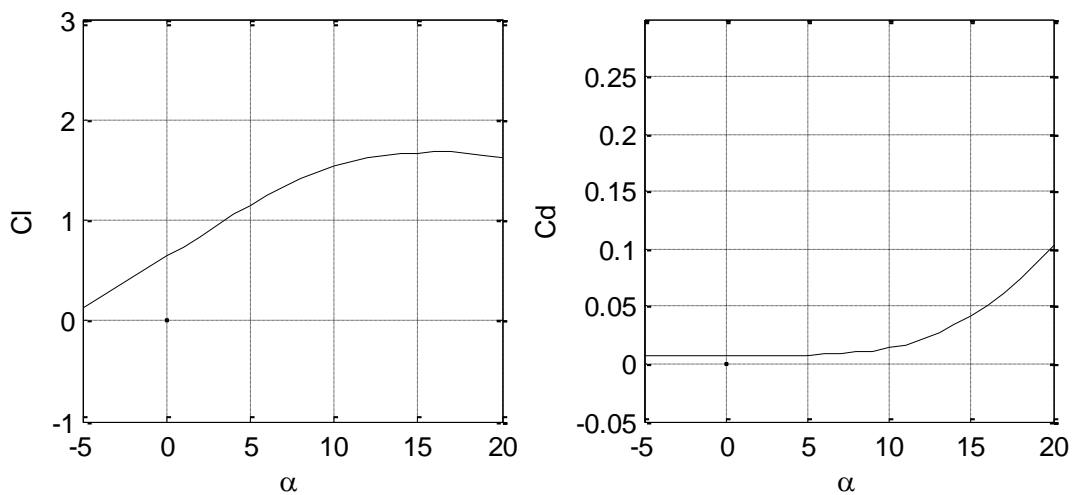


Figure A.4 Lift and drag coefficients of N6409-0.140 ( $R_n = 1.5 \times 10^6$ ).



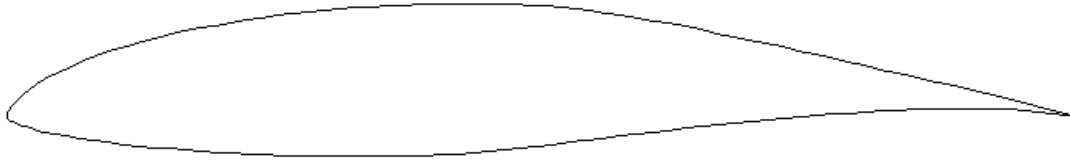


Figure A.5 Profile of E1098-0.140.

Table A.3 Coordinates of E1098-0.140.

x/c	y/c	x/c	y/c	x/c	y/c	x/c	y/c	x/c	y/c	x/c	y/c
1.0000	0.0000	0.6071	0.0879	0.2102	0.0860	0.0000	0.0003	0.2428	-0.0363	0.6536	-0.0101
0.9938	0.0016	0.5922	0.0904	0.1958	0.0835	0.0001	-0.0013	0.2580	-0.0369	0.6692	-0.0084
0.9829	0.0046	0.5775	0.0928	0.1816	0.0808	0.0006	-0.0026	0.2732	-0.0373	0.6847	-0.0068
0.9703	0.0081	0.5629	0.0950	0.1675	0.0780	0.0016	-0.0037	0.2884	-0.0377	0.7002	-0.0052
0.9569	0.0117	0.5485	0.0970	0.1536	0.0749	0.0029	-0.0048	0.3037	-0.0379	0.7157	-0.0038
0.9427	0.0152	0.5342	0.0987	0.1399	0.0716	0.0047	-0.0059	0.3189	-0.0381	0.7311	-0.0023
0.9280	0.0187	0.5200	0.1001	0.1265	0.0681	0.0070	-0.0070	0.3341	-0.0381	0.7466	-0.0010
0.9128	0.0222	0.5058	0.1012	0.1132	0.0644	0.0097	-0.0082	0.3493	-0.0380	0.7619	0.0002
0.8975	0.0257	0.4915	0.1021	0.1003	0.0604	0.0130	-0.0093	0.3645	-0.0378	0.7773	0.0014
0.8821	0.0292	0.4771	0.1028	0.0877	0.0563	0.0169	-0.0106	0.3795	-0.0375	0.7926	0.0024
0.8668	0.0326	0.4625	0.1032	0.0756	0.0519	0.0218	-0.0119	0.3945	-0.0370	0.8079	0.0033
0.8514	0.0361	0.4477	0.1034	0.0641	0.0473	0.0279	-0.0135	0.4092	-0.0364	0.8232	0.0042
0.8360	0.0396	0.4329	0.1035	0.0533	0.0426	0.0355	-0.0153	0.4239	-0.0355	0.8384	0.0049
0.8207	0.0430	0.4180	0.1034	0.0436	0.0379	0.0447	-0.0172	0.4385	-0.0345	0.8536	0.0054
0.8053	0.0465	0.4030	0.1032	0.0351	0.0333	0.0554	-0.0191	0.4532	-0.0333	0.8687	0.0058
0.7899	0.0499	0.3881	0.1028	0.0278	0.0289	0.0676	-0.0211	0.4679	-0.0319	0.8838	0.0061
0.7745	0.0533	0.3731	0.1023	0.0215	0.0247	0.0807	-0.0231	0.4828	-0.0303	0.8987	0.0062
0.7591	0.0568	0.3581	0.1016	0.0166	0.0210	0.0944	-0.0249	0.4978	-0.0286	0.9136	0.0061
0.7437	0.0601	0.3432	0.1007	0.0128	0.0179	0.1085	-0.0266	0.5131	-0.0268	0.9282	0.0059
0.7283	0.0635	0.3282	0.0997	0.0098	0.0151	0.1230	-0.0281	0.5285	-0.0250	0.9427	0.0054
0.7130	0.0668	0.3133	0.0986	0.0074	0.0126	0.1376	-0.0295	0.5440	-0.0231	0.9567	0.0046
0.6978	0.0700	0.2984	0.0972	0.0053	0.0102	0.1524	-0.0308	0.5596	-0.0211	0.9701	0.0036
0.6826	0.0732	0.2836	0.0958	0.0037	0.0080	0.1673	-0.0320	0.5753	-0.0192	0.9826	0.0023
0.6674	0.0763	0.2688	0.0941	0.0024	0.0059	0.1822	-0.0331	0.5910	-0.0174	0.9936	0.0009
0.6522	0.0794	0.2540	0.0924	0.0014	0.0042	0.1973	-0.0341	0.6067	-0.0155	1.0000	0.0000
0.6371	0.0823	0.2393	0.0904	0.0006	0.0027	0.2124	-0.0349	0.6224	-0.0137		
0.6221	0.0851	0.2247	0.0883	0.0002	0.0014	0.2276	-0.0357	0.6380	-0.0119		

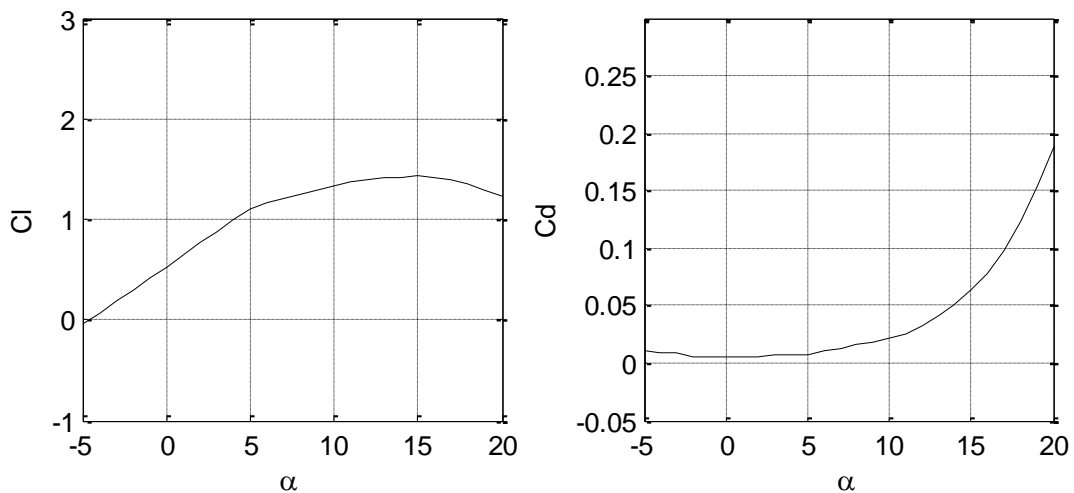


Figure A.6 Lift and drag coefficients of E1098-0.140 ( $Rn = 1.5 \times 10^6$ ).

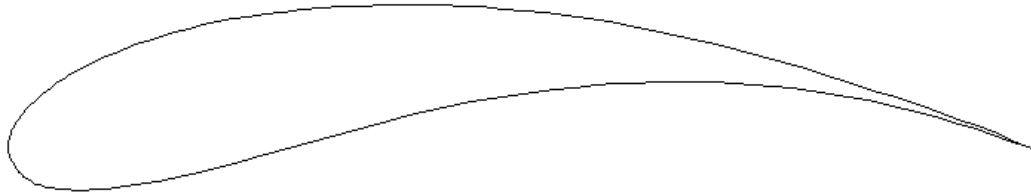


Figure A.7 Profile of G652-0.140.

Table A.4 Coordinates of G652-0.140.

x/c	y/c	x/c	y/c	x/c	y/c	x/c	y/c	x/c	y/c	x/c	y/c
1.0000	0.0000	0.4630	0.1358	0.0646	0.0728	0.0007	-0.0043	0.0975	-0.0384	0.5740	0.0607
0.9913	0.0037	0.4433	0.1370	0.0582	0.0691	0.0013	-0.0065	0.1060	-0.0375	0.5939	0.0621
0.9773	0.0095	0.4240	0.1378	0.0523	0.0655	0.0021	-0.0087	0.1151	-0.0364	0.6136	0.0632
0.9622	0.0156	0.4049	0.1382	0.0469	0.0620	0.0031	-0.0110	0.1249	-0.0350	0.6331	0.0639
0.9459	0.0218	0.3857	0.1382	0.0419	0.0586	0.0042	-0.0133	0.1355	-0.0333	0.6524	0.0643
0.9282	0.0282	0.3662	0.1378	0.0374	0.0552	0.0056	-0.0156	0.1469	-0.0312	0.6715	0.0642
0.9090	0.0349	0.3463	0.1370	0.0331	0.0519	0.0071	-0.0179	0.1592	-0.0287	0.6904	0.0638
0.8887	0.0419	0.3261	0.1359	0.0293	0.0486	0.0089	-0.0202	0.1726	-0.0258	0.7091	0.0629
0.8672	0.0489	0.3060	0.1345	0.0257	0.0455	0.0109	-0.0224	0.1873	-0.0222	0.7278	0.0617
0.8452	0.0561	0.2865	0.1329	0.0224	0.0424	0.0130	-0.0246	0.2035	-0.0182	0.7464	0.0601
0.8228	0.0631	0.2681	0.1312	0.0194	0.0394	0.0154	-0.0267	0.2213	-0.0136	0.7650	0.0581
0.8004	0.0699	0.2509	0.1292	0.0166	0.0364	0.0179	-0.0287	0.2409	-0.0084	0.7836	0.0557
0.7781	0.0766	0.2348	0.1270	0.0140	0.0334	0.0207	-0.0306	0.2618	-0.0027	0.8021	0.0529
0.7560	0.0829	0.2195	0.1246	0.0117	0.0306	0.0236	-0.0323	0.2837	0.0033	0.8205	0.0498
0.7342	0.0889	0.2049	0.1218	0.0096	0.0277	0.0269	-0.0338	0.3059	0.0094	0.8387	0.0463
0.7126	0.0946	0.1907	0.1186	0.0078	0.0249	0.0304	-0.0351	0.3281	0.0154	0.8568	0.0425
0.6912	0.1000	0.1767	0.1151	0.0061	0.0222	0.0341	-0.0362	0.3498	0.0212	0.8748	0.0384
0.6701	0.1050	0.1629	0.1113	0.0047	0.0196	0.0382	-0.0371	0.3711	0.0267	0.8925	0.0339
0.6491	0.1096	0.1495	0.1074	0.0035	0.0171	0.0426	-0.0379	0.3919	0.0318	0.9101	0.0291
0.6284	0.1138	0.1368	0.1035	0.0024	0.0146	0.0473	-0.0385	0.4123	0.0364	0.9275	0.0240
0.6077	0.1177	0.1252	0.0997	0.0016	0.0121	0.0524	-0.0390	0.4326	0.0407	0.9445	0.0187
0.5870	0.1213	0.1144	0.0959	0.0009	0.0097	0.0577	-0.0394	0.4528	0.0446	0.9610	0.0134
0.5662	0.1245	0.1046	0.0921	0.0004	0.0075	0.0633	-0.0396	0.4730	0.0481	0.9765	0.0082
0.5454	0.1275	0.0954	0.0882	0.0001	0.0051	0.0693	-0.0397	0.4932	0.0513	0.9910	0.0032
0.5245	0.1300	0.0869	0.0843	0.0000	0.0030	0.0756	-0.0397	0.5135	0.0542	1.0000	0.0000
0.5037	0.1323	0.0789	0.0804	0.0001	0.0003	0.0824	-0.0394	0.5337	0.0567		
0.4831	0.1342	0.0715	0.0765	0.0003	-0.0021	0.0897	-0.0390	0.5539	0.0589		

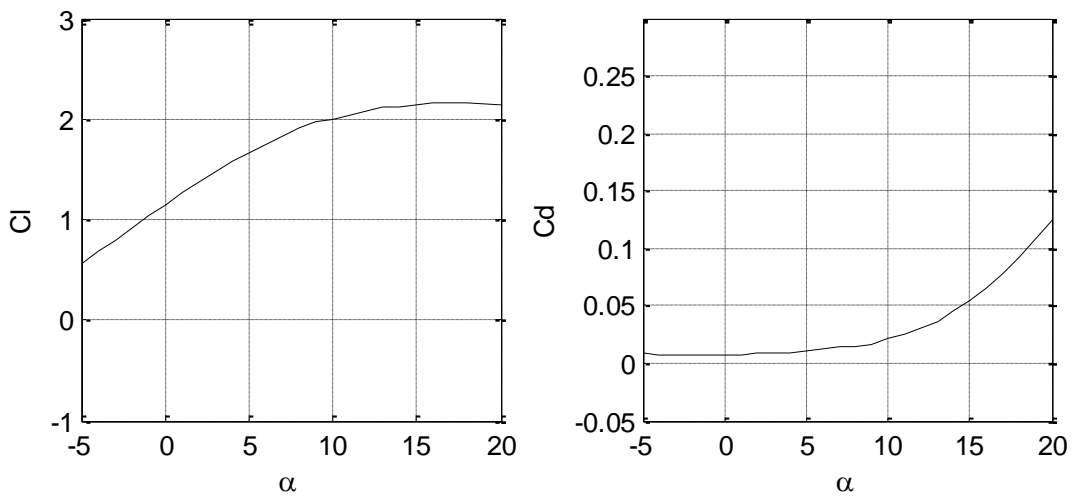


Figure A.8 Lift and drag coefficients of G652-0.140 ( $Rn = 1.5 \times 10^6$ ).

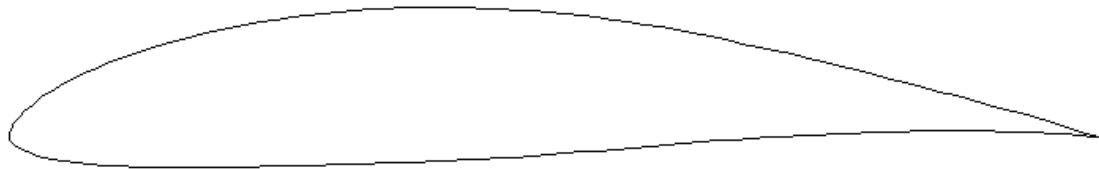


Figure A.9 Profile of M0.45-0.140.

Table A.5 Coordinates of M0.45-0.140.

x/c	y/c	x/c	y/c	x/c	y/c	x/c	y/c	x/c	y/c	x/c	y/c
1.0000	0.0000	0.6130	0.0969	0.2157	0.1018	0.0000	0.0014	0.2379	-0.0265	0.6505	-0.0033
0.9941	0.0017	0.5980	0.0996	0.2015	0.0991	0.0001	-0.0003	0.2531	-0.0261	0.6660	-0.0022
0.9835	0.0050	0.5831	0.1021	0.1874	0.0961	0.0006	-0.0020	0.2683	-0.0258	0.6815	-0.0011
0.9710	0.0089	0.5683	0.1045	0.1734	0.0929	0.0014	-0.0037	0.2835	-0.0254	0.6970	-0.0002
0.9574	0.0131	0.5536	0.1067	0.1596	0.0894	0.0027	-0.0054	0.2988	-0.0250	0.7124	0.0008
0.9432	0.0172	0.5389	0.1086	0.1459	0.0856	0.0043	-0.0070	0.3139	-0.0245	0.7278	0.0016
0.9286	0.0213	0.5242	0.1104	0.1324	0.0816	0.0064	-0.0087	0.3291	-0.0241	0.7432	0.0024
0.9138	0.0255	0.5096	0.1119	0.1190	0.0774	0.0088	-0.0102	0.3443	-0.0236	0.7586	0.0031
0.8988	0.0295	0.4949	0.1133	0.1060	0.0728	0.0118	-0.0118	0.3594	-0.0231	0.7740	0.0037
0.8838	0.0336	0.4801	0.1144	0.0931	0.0680	0.0154	-0.0133	0.3744	-0.0226	0.7894	0.0043
0.8689	0.0376	0.4653	0.1154	0.0807	0.0629	0.0198	-0.0148	0.3894	-0.0220	0.8047	0.0047
0.8539	0.0415	0.4503	0.1161	0.0688	0.0576	0.0252	-0.0165	0.4044	-0.0214	0.8201	0.0051
0.8388	0.0455	0.4354	0.1167	0.0575	0.0521	0.0318	-0.0183	0.4195	-0.0207	0.8354	0.0054
0.8238	0.0493	0.4204	0.1171	0.0471	0.0465	0.0399	-0.0200	0.4345	-0.0199	0.8507	0.0056
0.8087	0.0532	0.4055	0.1174	0.0378	0.0411	0.0498	-0.0217	0.4497	-0.0191	0.8660	0.0056
0.7937	0.0570	0.3907	0.1174	0.0297	0.0359	0.0612	-0.0232	0.4649	-0.0181	0.8812	0.0056
0.7786	0.0607	0.3759	0.1173	0.0231	0.0311	0.0739	-0.0245	0.4802	-0.0170	0.8964	0.0054
0.7635	0.0644	0.3612	0.1170	0.0178	0.0268	0.0875	-0.0255	0.4955	-0.0158	0.9115	0.0051
0.7484	0.0680	0.3465	0.1164	0.0136	0.0230	0.1017	-0.0262	0.5108	-0.0146	0.9266	0.0047
0.7333	0.0715	0.3318	0.1156	0.0103	0.0197	0.1163	-0.0267	0.5263	-0.0133	0.9415	0.0041
0.7182	0.0750	0.3172	0.1147	0.0076	0.0168	0.1312	-0.0271	0.5417	-0.0120	0.9561	0.0034
0.7031	0.0784	0.3025	0.1135	0.0055	0.0141	0.1463	-0.0272	0.5573	-0.0107	0.9701	0.0026
0.6881	0.0818	0.2879	0.1121	0.0037	0.0116	0.1614	-0.0273	0.5728	-0.0094	0.9839	0.0016
0.6730	0.0850	0.2734	0.1105	0.0024	0.0093	0.1767	-0.0273	0.5884	-0.0081	0.9939	0.0006
0.6580	0.0881	0.2589	0.1086	0.0014	0.0071	0.1920	-0.0272	0.6040	-0.0069	1.0000	0.0000
0.6430	0.0912	0.2444	0.1066	0.0006	0.0051	0.2073	-0.0270	0.6195	-0.0056		
0.6279	0.0941	0.2300	0.1043	0.0002	0.0032	0.2226	-0.0267	0.6350	-0.0044		

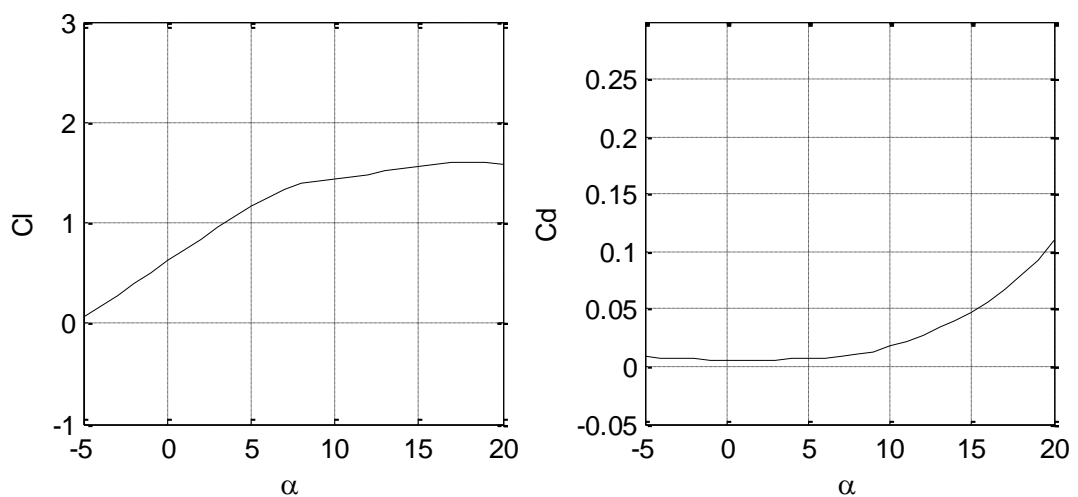


Figure A.10 Lift and drag coefficients of M0.45-0.140 ( $R_n = 1.5 \times 10^6$ ).

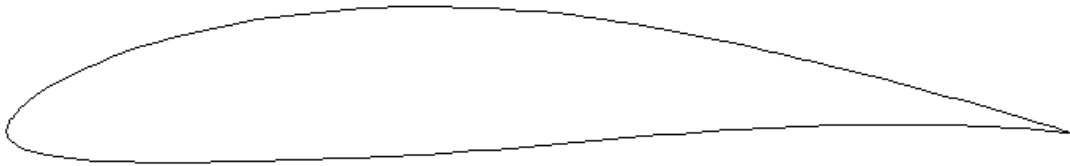


Figure A.11 Profile of M0.45\_0.4\_0.05-0.140.

Table A.6 Coordinates of M0.45\_0.4\_0.05-0.140.

x/c	y/c	x/c	y/c	x/c	y/c	x/c	y/c	x/c	y/c	x/c	y/c
1.0000	0.0000	0.6058	0.0993	0.2084	0.1012	0.0000	0.0011	0.2306	-0.0267	0.6466	0.0003
0.9940	0.0018	0.5906	0.1019	0.1946	0.0984	0.0002	-0.0007	0.2455	-0.0263	0.6623	0.0014
0.9833	0.0053	0.5754	0.1044	0.1809	0.0954	0.0007	-0.0025	0.2604	-0.0259	0.6780	0.0024
0.9706	0.0093	0.5604	0.1067	0.1674	0.0922	0.0015	-0.0042	0.2754	-0.0254	0.6936	0.0034
0.9569	0.0135	0.5454	0.1087	0.1540	0.0887	0.0027	-0.0060	0.2903	-0.0249	0.7092	0.0042
0.9425	0.0178	0.5305	0.1106	0.1407	0.0849	0.0044	-0.0077	0.3054	-0.0243	0.7248	0.0050
0.9277	0.0221	0.5155	0.1123	0.1277	0.0809	0.0064	-0.0094	0.3204	-0.0238	0.7404	0.0057
0.9126	0.0264	0.5006	0.1137	0.1148	0.0767	0.0088	-0.0110	0.3354	-0.0231	0.7560	0.0063
0.8973	0.0307	0.4856	0.1149	0.1022	0.0722	0.0117	-0.0126	0.3505	-0.0225	0.7715	0.0068
0.8820	0.0349	0.4706	0.1160	0.0899	0.0674	0.0152	-0.0142	0.3656	-0.0217	0.7871	0.0072
0.8667	0.0391	0.4555	0.1168	0.0779	0.0624	0.0195	-0.0158	0.3808	-0.0210	0.8026	0.0075
0.8513	0.0432	0.4405	0.1175	0.0664	0.0571	0.0247	-0.0175	0.3960	-0.0201	0.8181	0.0078
0.8359	0.0473	0.4254	0.1180	0.0555	0.0518	0.0311	-0.0193	0.4114	-0.0192	0.8336	0.0078
0.8206	0.0513	0.4105	0.1183	0.0455	0.0463	0.0389	-0.0210	0.4268	-0.0181	0.8490	0.0078
0.8052	0.0553	0.3956	0.1184	0.0365	0.0409	0.0483	-0.0226	0.4423	-0.0170	0.8645	0.0077
0.7898	0.0592	0.3808	0.1184	0.0287	0.0358	0.0593	-0.0241	0.4579	-0.0157	0.8799	0.0074
0.7744	0.0630	0.3661	0.1181	0.0224	0.0310	0.0716	-0.0253	0.4735	-0.0144	0.8952	0.0071
0.7590	0.0667	0.3514	0.1176	0.0172	0.0268	0.0847	-0.0262	0.4891	-0.0130	0.9105	0.0065
0.7437	0.0704	0.3368	0.1169	0.0132	0.0231	0.0984	-0.0269	0.5048	-0.0116	0.9257	0.0059
0.7283	0.0740	0.3223	0.1160	0.0099	0.0198	0.1125	-0.0274	0.5205	-0.0102	0.9407	0.0051
0.7129	0.0775	0.3078	0.1149	0.0073	0.0168	0.1269	-0.0277	0.5362	-0.0088	0.9555	0.0041
0.6976	0.0810	0.2933	0.1135	0.0052	0.0141	0.1415	-0.0278	0.5519	-0.0074	0.9697	0.0030
0.6823	0.0843	0.2790	0.1120	0.0036	0.0116	0.1562	-0.0278	0.5677	-0.0060	0.9827	0.0019
0.6669	0.0875	0.2647	0.1103	0.0023	0.0092	0.1711	-0.0278	0.5835	-0.0046	0.9938	0.0007
0.6516	0.0907	0.2505	0.1083	0.0013	0.0070	0.1859	-0.0276	0.5993	-0.0033	1.0000	0.0000
0.6363	0.0937	0.2364	0.1062	0.0006	0.0050	0.2008	-0.0273	0.6151	-0.0020		
0.6210	0.0966	0.2224	0.1038	0.0002	0.0030	0.2157	-0.0270	0.6308	-0.0008		

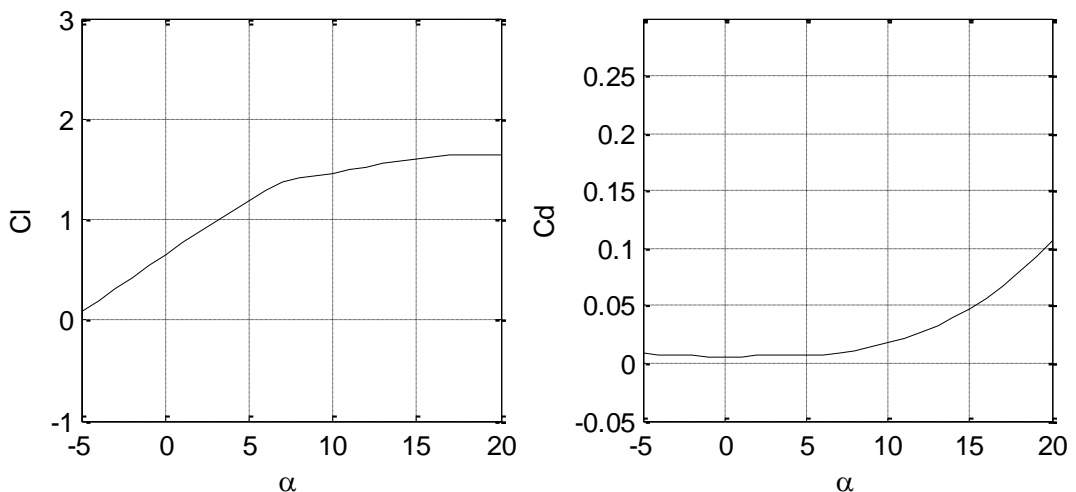


Figure A.12 Lift and drag coefficients of M0.45\_0.4\_0.05-0.140 ( $Rn = 1.5 \times 10^6$ ).

**A.2 Targeting S825 (75% radius)**

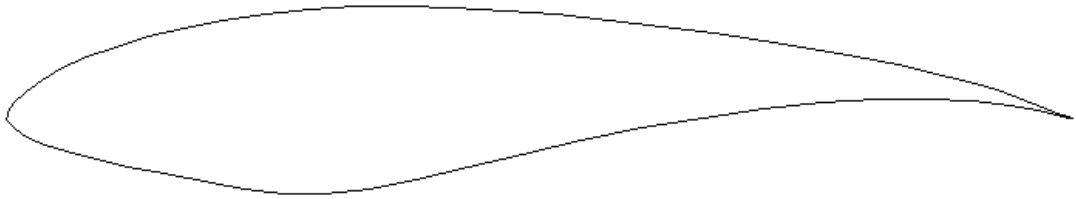


Figure A.13 Profile of S825.

Table A.7 Coordinates of S825.

x/c	y/c	x/c	y/c	x/c	y/c	x/c	y/c
1.00000	0.00000	0.41916	0.10211	0.00013	-0.00141	0.43276	-0.04522
0.99657	0.00103	0.37216	0.10348	0.00040	-0.00221	0.48470	-0.03298
0.98682	0.00445	0.32695	0.10321	0.00082	-0.00304	0.53947	-0.02066
0.97188	0.01023	0.28388	0.10085	0.00198	-0.00475	0.59608	-0.00914
0.95272	0.01766	0.24254	0.09643	0.00653	-0.00920	0.65337	0.00087
0.92975	0.02575	0.20325	0.09049	0.01822	-0.01650	0.71012	0.00884
0.90283	0.03365	0.16644	0.08330	0.03507	-0.02380	0.76505	0.01442
0.87153	0.04127	0.13252	0.07508	0.05673	-0.03091	0.81689	0.01744
0.83611	0.04888	0.10186	0.06603	0.08301	-0.03775	0.86438	0.01790
0.79707	0.05646	0.07481	0.05630	0.11354	-0.04435	0.90620	0.01581
0.75490	0.06394	0.05161	0.04609	0.14797	-0.05090	0.94077	0.01186
0.71014	0.07122	0.03253	0.03562	0.18527	-0.05824	0.96729	0.00736
0.66335	0.07816	0.01767	0.02512	0.22271	-0.06521	0.98574	0.00346
0.61511	0.08461	0.00725	0.01493	0.26014	-0.06927	0.99649	0.00089
0.56601	0.09041	0.00130	0.00546	0.29868	-0.06904	1.00000	0.00000
0.51661	0.09538	0.00028	0.00227	0.33993	-0.06436		
0.46748	0.09935	0.00001	0.00033	0.38449	-0.05615		

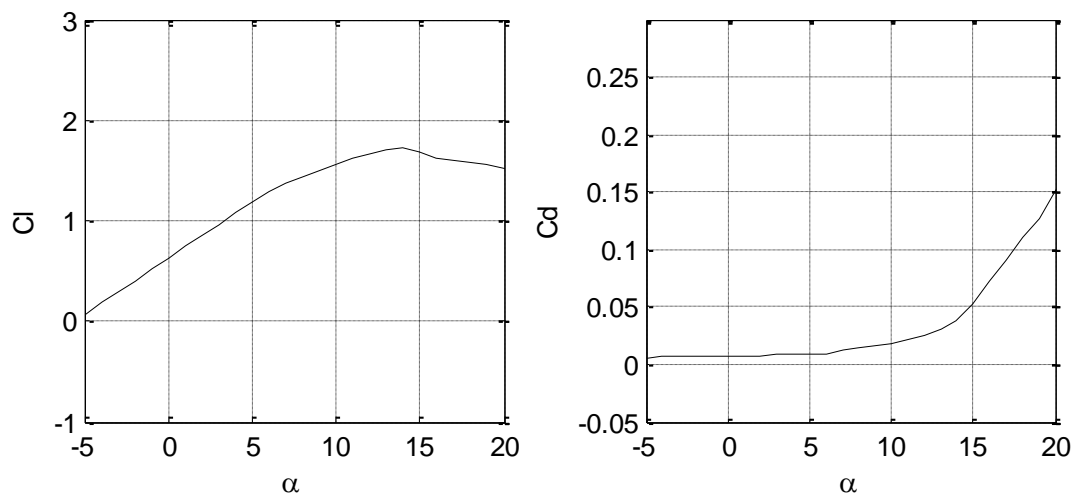


Figure A.14 Lift and drag coefficients of S825 ( $R_n = 2.0 \times 10^6$ ).

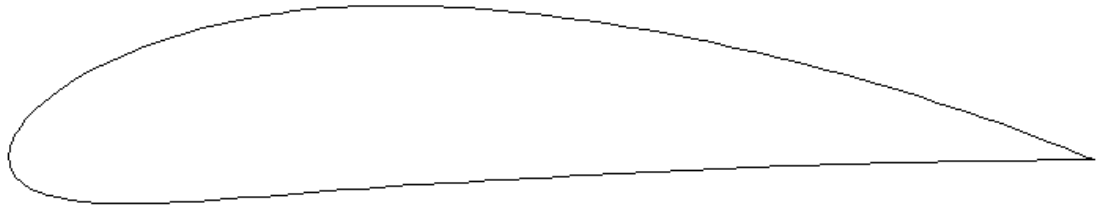


Figure A.15 Profile of N6409-0.170.

Table A.8 Coordinates of N6409-0.170.

x/c	y/c	x/c	y/c	x/c	y/c	x/c	y/c	x/c	y/c	x/c	y/c
1.0000	0.0000	0.6177	0.1145	0.2202	0.1312	0.0000	0.0018	0.2339	-0.0341	0.6480	-0.0086
0.9945	0.0023	0.6027	0.1173	0.2061	0.1284	0.0001	-0.0001	0.2491	-0.0328	0.6635	-0.0079
0.9841	0.0065	0.5877	0.1200	0.1922	0.1253	0.0005	-0.0030	0.2644	-0.0316	0.6789	-0.0072
0.9716	0.0114	0.5727	0.1226	0.1782	0.1219	0.0013	-0.0064	0.2796	-0.0304	0.6943	-0.0066
0.9579	0.0168	0.5577	0.1250	0.1644	0.1181	0.0025	-0.0095	0.2947	-0.0292	0.7097	-0.0060
0.9436	0.0222	0.5427	0.1273	0.1507	0.1140	0.0040	-0.0125	0.3099	-0.0280	0.7251	-0.0054
0.9291	0.0276	0.5276	0.1295	0.1372	0.1095	0.0059	-0.0154	0.3250	-0.0268	0.7405	-0.0049
0.9145	0.0329	0.5126	0.1315	0.1238	0.1046	0.0081	-0.0181	0.3401	-0.0257	0.7559	-0.0044
0.8999	0.0381	0.4976	0.1334	0.1106	0.0994	0.0109	-0.0207	0.3551	-0.0247	0.7713	-0.0039
0.8853	0.0432	0.4826	0.1351	0.0976	0.0937	0.0142	-0.0232	0.3702	-0.0237	0.7867	-0.0034
0.8706	0.0481	0.4675	0.1367	0.0848	0.0876	0.0181	-0.0256	0.3853	-0.0227	0.8021	-0.0030
0.8559	0.0530	0.4524	0.1382	0.0726	0.0811	0.0229	-0.0281	0.4005	-0.0219	0.8175	-0.0026
0.8411	0.0577	0.4374	0.1395	0.0609	0.0743	0.0288	-0.0305	0.4158	-0.0210	0.8329	-0.0023
0.8263	0.0623	0.4224	0.1406	0.0499	0.0672	0.0361	-0.0330	0.4313	-0.0202	0.8483	-0.0020
0.8115	0.0668	0.4076	0.1415	0.0399	0.0600	0.0451	-0.0352	0.4468	-0.0193	0.8637	-0.0017
0.7967	0.0712	0.3928	0.1423	0.0314	0.0531	0.0560	-0.0372	0.4624	-0.0184	0.8791	-0.0014
0.7819	0.0754	0.3782	0.1428	0.0244	0.0468	0.0684	-0.0387	0.4780	-0.0176	0.8945	-0.0011
0.7671	0.0796	0.3637	0.1430	0.0188	0.0411	0.0819	-0.0398	0.4935	-0.0167	0.9099	-0.0009
0.7522	0.0836	0.3492	0.1431	0.0143	0.0358	0.0961	-0.0404	0.5090	-0.0158	0.9252	-0.0007
0.7373	0.0875	0.3348	0.1429	0.0107	0.0310	0.1109	-0.0406	0.5244	-0.0149	0.9405	-0.0005
0.7224	0.0913	0.3203	0.1424	0.0078	0.0268	0.1259	-0.0404	0.5399	-0.0141	0.9556	-0.0004
0.7075	0.0950	0.3059	0.1416	0.0056	0.0229	0.1412	-0.0400	0.5554	-0.0132	0.9701	-0.0002
0.6926	0.0986	0.2915	0.1406	0.0038	0.0193	0.1567	-0.0393	0.5708	-0.0124	0.9834	-0.0001
0.6776	0.1020	0.2772	0.1393	0.0024	0.0158	0.1722	-0.0385	0.5863	-0.0116	0.9942	0.0000
0.6627	0.1053	0.2628	0.1377	0.0014	0.0124	0.1877	-0.0375	0.6017	-0.0108	1.0000	0.0000
0.6477	0.1085	0.2486	0.1358	0.0007	0.0089	0.2032	-0.0364	0.6172	-0.0100		
0.6327	0.1116	0.2343	0.1337	0.0002	0.0057	0.2186	-0.0353	0.6326	-0.0093		

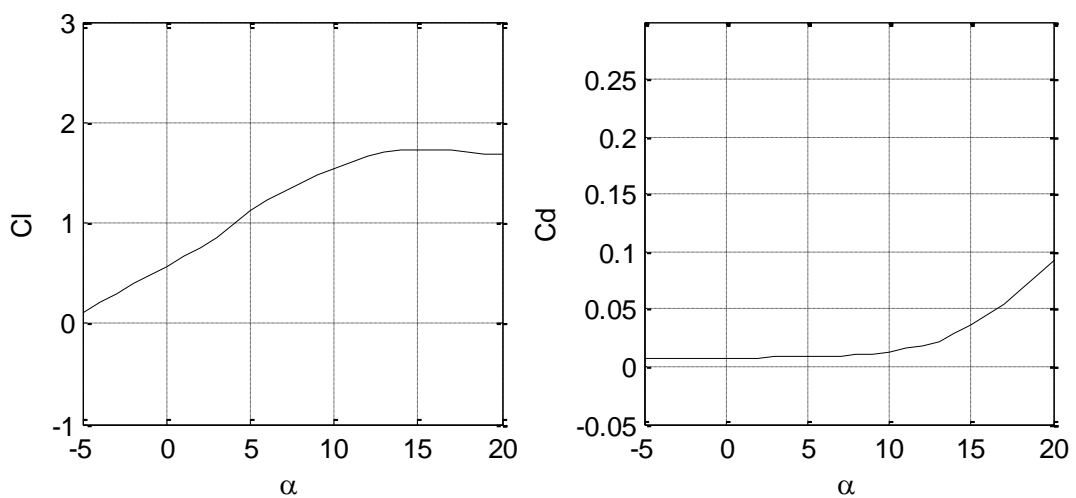


Figure A.16 Lift and drag coefficients of N6409-0.170 ( $Rn = 2.0 \times 10^6$ ).

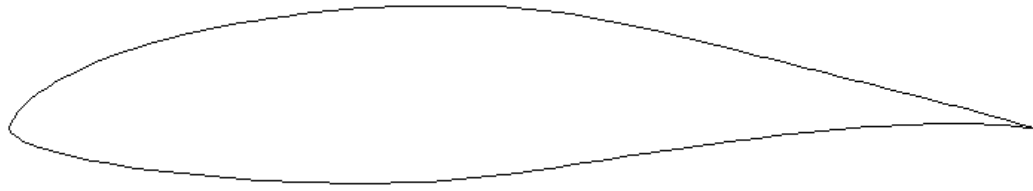


Figure A.17 Profile of E1098-0.170.

Table A.9 Coordinates of E1098-0.170.

x/c	y/c	x/c	y/c	x/c	y/c	x/c	y/c	x/c	y/c	x/c	y/c
1.0000	0.0000	0.6071	0.0989	0.2102	0.0989	0.0000	0.0001	0.2428	-0.0500	0.6536	-0.0197
0.9938	0.0016	0.5922	0.1020	0.1958	0.0961	0.0001	-0.0015	0.2580	-0.0508	0.6692	-0.0175
0.9829	0.0048	0.5775	0.1048	0.1816	0.0930	0.0006	-0.0031	0.2732	-0.0515	0.6847	-0.0153
0.9703	0.0086	0.5629	0.1074	0.1675	0.0898	0.0016	-0.0046	0.2884	-0.0520	0.7002	-0.0133
0.9569	0.0124	0.5485	0.1098	0.1536	0.0863	0.0029	-0.0061	0.3037	-0.0525	0.7157	-0.0113
0.9427	0.0163	0.5342	0.1118	0.1399	0.0825	0.0047	-0.0076	0.3189	-0.0527	0.7311	-0.0093
0.9280	0.0201	0.5200	0.1136	0.1265	0.0785	0.0070	-0.0091	0.3341	-0.0529	0.7466	-0.0075
0.9128	0.0239	0.5058	0.1150	0.1132	0.0742	0.0097	-0.0106	0.3493	-0.0529	0.7619	-0.0058
0.8975	0.0278	0.4915	0.1162	0.1003	0.0697	0.0130	-0.0122	0.3645	-0.0528	0.7773	-0.0041
0.8821	0.0316	0.4771	0.1171	0.0877	0.0649	0.0169	-0.0140	0.3795	-0.0525	0.7926	-0.0026
0.8668	0.0355	0.4625	0.1177	0.0756	0.0598	0.0218	-0.0159	0.3945	-0.0520	0.8079	-0.0012
0.8514	0.0394	0.4477	0.1181	0.0641	0.0546	0.0279	-0.0181	0.4092	-0.0513	0.8232	0.0001
0.8360	0.0433	0.4329	0.1183	0.0533	0.0492	0.0355	-0.0205	0.4239	-0.0504	0.8384	0.0012
0.8207	0.0472	0.4180	0.1184	0.0436	0.0438	0.0447	-0.0231	0.4385	-0.0493	0.8536	0.0022
0.8053	0.0511	0.4030	0.1182	0.0351	0.0385	0.0554	-0.0259	0.4532	-0.0480	0.8687	0.0030
0.7899	0.0550	0.3881	0.1178	0.0278	0.0335	0.0676	-0.0286	0.4679	-0.0464	0.8838	0.0037
0.7745	0.0589	0.3731	0.1172	0.0215	0.0287	0.0807	-0.0313	0.4828	-0.0446	0.8987	0.0042
0.7591	0.0628	0.3581	0.1165	0.0166	0.0244	0.0944	-0.0338	0.4978	-0.0426	0.9136	0.0044
0.7437	0.0667	0.3432	0.1156	0.0128	0.0208	0.1085	-0.0361	0.5131	-0.0405	0.9282	0.0045
0.7283	0.0706	0.3282	0.1145	0.0098	0.0176	0.1230	-0.0383	0.5285	-0.0383	0.9427	0.0043
0.7130	0.0744	0.3133	0.1132	0.0074	0.0147	0.1376	-0.0403	0.5440	-0.0360	0.9567	0.0039
0.6978	0.0781	0.2984	0.1117	0.0053	0.0119	0.1524	-0.0421	0.5596	-0.0336	0.9701	0.0031
0.6826	0.0818	0.2836	0.1101	0.0037	0.0094	0.1673	-0.0438	0.5753	-0.0313	0.9826	0.0021
0.6674	0.0854	0.2688	0.1082	0.0024	0.0070	0.1822	-0.0453	0.5910	-0.0289	0.9936	0.0009
0.6522	0.0890	0.2540	0.1062	0.0014	0.0050	0.1973	-0.0467	0.6067	-0.0266	1.0000	0.0000
0.6371	0.0924	0.2393	0.1040	0.0006	0.0032	0.2124	-0.0479	0.6224	-0.0242		
0.6221	0.0957	0.2247	0.1015	0.0002	0.0016	0.2276	-0.0490	0.6380	-0.0219		

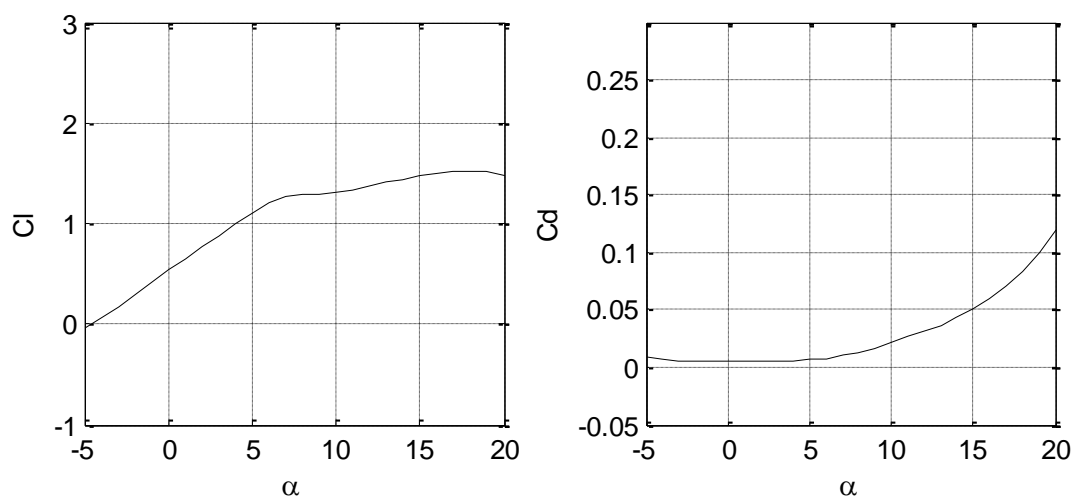


Figure A.18 Lift and drag coefficients of E1098-0.170 ( $Rn = 2.0 \times 10^6$ ).

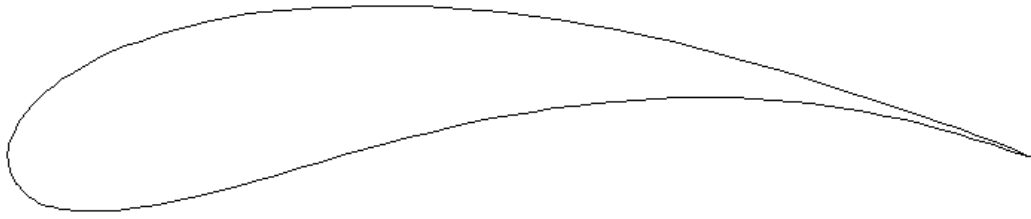


Figure A.19 Profile of G652-0.170.

Table A.10 Coordinates of G652-0.170.

x/c	y/c	x/c	y/c	x/c	y/c	x/c	y/c	x/c	y/c	x/c	y/c
1.0000	0.0000	0.4630	0.1454	0.0646	0.0848	0.0007	-0.0056	0.0975	-0.0520	0.5740	0.0540
0.9913	0.0038	0.4433	0.1471	0.0582	0.0807	0.0013	-0.0083	0.1060	-0.0514	0.5939	0.0559
0.9773	0.0097	0.4240	0.1484	0.0523	0.0767	0.0021	-0.0111	0.1151	-0.0506	0.6136	0.0575
0.9622	0.0158	0.4049	0.1493	0.0469	0.0728	0.0031	-0.0139	0.1249	-0.0494	0.6331	0.0587
0.9459	0.0222	0.3857	0.1497	0.0419	0.0689	0.0042	-0.0167	0.1355	-0.0479	0.6524	0.0595
0.9282	0.0287	0.3662	0.1498	0.0374	0.0651	0.0056	-0.0195	0.1469	-0.0460	0.6715	0.0599
0.9090	0.0355	0.3463	0.1495	0.0331	0.0613	0.0071	-0.0224	0.1592	-0.0436	0.6904	0.0599
0.8887	0.0426	0.3261	0.1488	0.0293	0.0576	0.0089	-0.0252	0.1726	-0.0407	0.7091	0.0594
0.8672	0.0499	0.3060	0.1479	0.0257	0.0539	0.0109	-0.0279	0.1873	-0.0372	0.7278	0.0586
0.8452	0.0572	0.2865	0.1467	0.0224	0.0503	0.0130	-0.0307	0.2035	-0.0331	0.7464	0.0573
0.8228	0.0646	0.2681	0.1453	0.0194	0.0468	0.0154	-0.0333	0.2213	-0.0284	0.7650	0.0557
0.8004	0.0717	0.2509	0.1437	0.0166	0.0432	0.0179	-0.0358	0.2409	-0.0230	0.7836	0.0536
0.7781	0.0787	0.2348	0.1417	0.0140	0.0398	0.0207	-0.0382	0.2618	-0.0170	0.8021	0.0512
0.7560	0.0855	0.2195	0.1394	0.0117	0.0363	0.0236	-0.0404	0.2837	-0.0106	0.8205	0.0483
0.7342	0.0919	0.2049	0.1367	0.0096	0.0329	0.0269	-0.0424	0.3059	-0.0040	0.8387	0.0451
0.7126	0.0981	0.1907	0.1336	0.0078	0.0296	0.0304	-0.0442	0.3281	0.0025	0.8568	0.0415
0.6912	0.1039	0.1767	0.1301	0.0061	0.0264	0.0341	-0.0457	0.3498	0.0088	0.8748	0.0375
0.6701	0.1093	0.1629	0.1262	0.0047	0.0232	0.0382	-0.0471	0.3711	0.0148	0.8925	0.0332
0.6491	0.1144	0.1495	0.1222	0.0035	0.0201	0.0426	-0.0483	0.3919	0.0204	0.9101	0.0285
0.6284	0.1192	0.1368	0.1181	0.0024	0.0171	0.0473	-0.0493	0.4123	0.0255	0.9275	0.0235
0.6077	0.1236	0.1252	0.1141	0.0016	0.0142	0.0524	-0.0502	0.4326	0.0303	0.9445	0.0183
0.5870	0.1277	0.1144	0.1101	0.0009	0.0113	0.0577	-0.0510	0.4528	0.0347	0.9610	0.0131
0.5662	0.1314	0.1046	0.1060	0.0004	0.0084	0.0633	-0.0516	0.4730	0.0388	0.9765	0.0080
0.5454	0.1349	0.0954	0.1018	0.0001	0.0056	0.0693	-0.0521	0.4932	0.0425	0.9910	0.0031
0.5245	0.1380	0.0869	0.0975	0.0000	0.0028	0.0756	-0.0524	0.5135	0.0459	1.0000	0.0000
0.5037	0.1408	0.0789	0.0932	0.0001	0.0000	0.0824	-0.0524	0.5337	0.0490		
0.4831	0.1433	0.0715	0.0890	0.0003	-0.0028	0.0897	-0.0523	0.5539	0.0517		

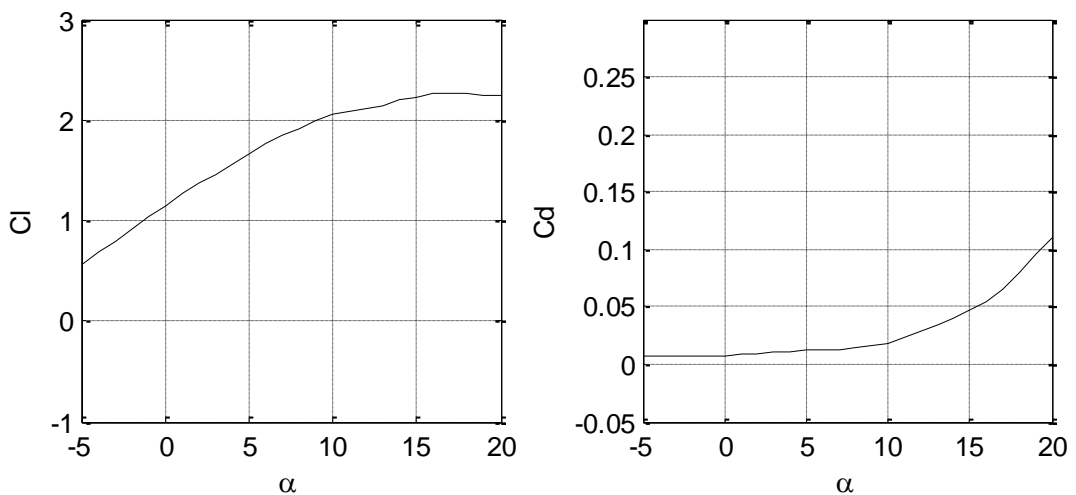


Figure A.20 Lift and drag coefficients of G652-0.170 ( $Rn = 2.0 \times 10^6$ ).



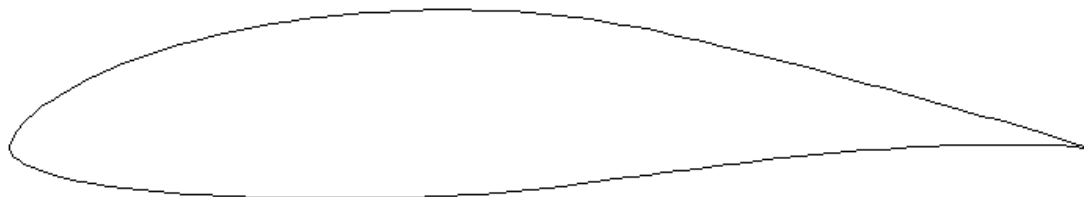


Figure A.21 Profile of M0.70-0.170.

Table A.11 Coordinates of M0.70-0.170.

x/c	y/c	x/c	y/c	x/c	y/c	x/c	y/c	x/c	y/c	x/c	y/c
1.0000	0.0000	0.6103	0.1039	0.2132	0.1077	0.0000	0.0007	0.2401	-0.0447	0.6519	-0.0160
0.9940	0.0017	0.5954	0.1070	0.1989	0.1047	0.0001	-0.0010	0.2553	-0.0450	0.6675	-0.0141
0.9832	0.0051	0.5806	0.1099	0.1848	0.1015	0.0006	-0.0028	0.2705	-0.0453	0.6830	-0.0123
0.9707	0.0090	0.5658	0.1126	0.1707	0.0980	0.0015	-0.0046	0.2858	-0.0454	0.6984	-0.0106
0.9572	0.0132	0.5513	0.1151	0.1569	0.0943	0.0028	-0.0064	0.3010	-0.0455	0.7139	-0.0089
0.9430	0.0174	0.5368	0.1173	0.1432	0.0903	0.0045	-0.0082	0.3162	-0.0454	0.7293	-0.0072
0.9283	0.0215	0.5223	0.1193	0.1297	0.0860	0.0067	-0.0100	0.3314	-0.0453	0.7447	-0.0057
0.9133	0.0257	0.5079	0.1209	0.1164	0.0814	0.0092	-0.0118	0.3466	-0.0451	0.7601	-0.0043
0.8982	0.0299	0.4933	0.1223	0.1034	0.0765	0.0123	-0.0136	0.3617	-0.0448	0.7755	-0.0029
0.8831	0.0341	0.4787	0.1235	0.0907	0.0714	0.0161	-0.0155	0.3767	-0.0444	0.7909	-0.0017
0.8679	0.0383	0.4640	0.1244	0.0784	0.0660	0.0207	-0.0175	0.3917	-0.0439	0.8062	-0.0005
0.8527	0.0424	0.4491	0.1251	0.0666	0.0603	0.0264	-0.0198	0.4066	-0.0432	0.8215	0.0005
0.8376	0.0466	0.4342	0.1256	0.0556	0.0545	0.0335	-0.0222	0.4215	-0.0424	0.8368	0.0014
0.8224	0.0507	0.4193	0.1259	0.0455	0.0486	0.0421	-0.0248	0.4364	-0.0414	0.8520	0.0022
0.8072	0.0548	0.4044	0.1260	0.0366	0.0429	0.0523	-0.0273	0.4513	-0.0402	0.8672	0.0028
0.7919	0.0589	0.3895	0.1259	0.0288	0.0373	0.0641	-0.0298	0.4663	-0.0389	0.8824	0.0033
0.7767	0.0630	0.3746	0.1255	0.0224	0.0322	0.0770	-0.0321	0.4814	-0.0374	0.8975	0.0037
0.7615	0.0670	0.3598	0.1250	0.0173	0.0276	0.0907	-0.0342	0.4965	-0.0357	0.9125	0.0038
0.7462	0.0710	0.3450	0.1242	0.0132	0.0237	0.1048	-0.0360	0.5119	-0.0339	0.9273	0.0038
0.7310	0.0750	0.3302	0.1233	0.0101	0.0202	0.1193	-0.0376	0.5273	-0.0320	0.9420	0.0036
0.7159	0.0789	0.3154	0.1221	0.0075	0.0170	0.1341	-0.0390	0.5428	-0.0300	0.9564	0.0032
0.7007	0.0828	0.3007	0.1207	0.0054	0.0141	0.1490	-0.0403	0.5583	-0.0280	0.9701	0.0025
0.6856	0.0865	0.2860	0.1191	0.0037	0.0114	0.1641	-0.0413	0.5739	-0.0260	0.9828	0.0017
0.6705	0.0902	0.2713	0.1172	0.0024	0.0089	0.1792	-0.0422	0.5896	-0.0240	0.9938	0.0007
0.6554	0.0938	0.2567	0.1152	0.0014	0.0066	0.1944	-0.0430	0.6052	-0.0219	1.0000	0.0000
0.6403	0.0973	0.2421	0.1129	0.0006	0.0045	0.2096	-0.0437	0.6208	-0.0199		
0.6253	0.1007	0.2276	0.1104	0.0002	0.0026	0.2249	-0.0442	0.6364	-0.0180		

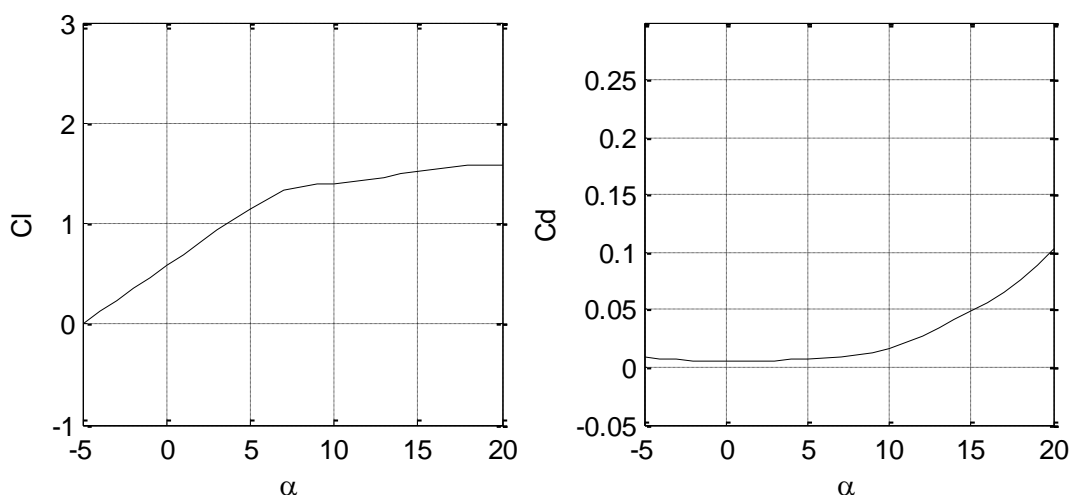


Figure A.22 Lift and drag coefficients of M0.70-0.170 ( $Rn = 2.0 \times 10^6$ ).

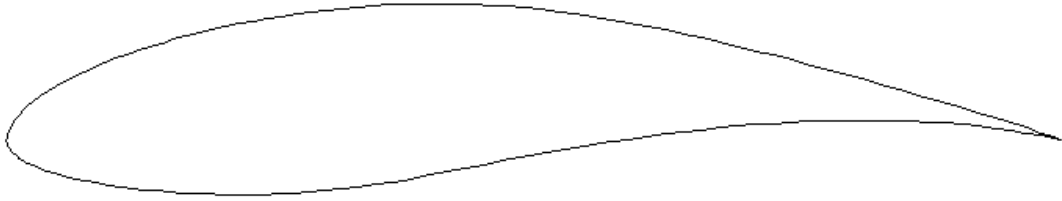


Figure A.23 Profile of M0.0\_0.65\_0.35-0.170.

Table A.12 Coordinates of M0.0\_0.65\_0.35-0.170.

x/c	y/c	x/c	y/c	x/c	y/c	x/c	y/c	x/c	y/c	x/c	y/c
1.0000	0.0000	0.5567	0.1157	0.1592	0.0945	0.0003	-0.0018	0.1919	-0.0512	0.6257	0.0054
0.9929	0.0024	0.5401	0.1183	0.1477	0.0913	0.0006	-0.0038	0.2048	-0.0515	0.6428	0.0075
0.9809	0.0065	0.5238	0.1207	0.1363	0.0879	0.0012	-0.0058	0.2179	-0.0517	0.6598	0.0095
0.9675	0.0112	0.5076	0.1227	0.1253	0.0843	0.0021	-0.0077	0.2312	-0.0516	0.6768	0.0113
0.9530	0.0159	0.4915	0.1244	0.1145	0.0807	0.0034	-0.0097	0.2448	-0.0514	0.6936	0.0129
0.9376	0.0207	0.4754	0.1257	0.1040	0.0769	0.0050	-0.0116	0.2587	-0.0509	0.7103	0.0143
0.9214	0.0256	0.4592	0.1268	0.0938	0.0729	0.0071	-0.0136	0.2729	-0.0502	0.7269	0.0155
0.9044	0.0306	0.4429	0.1275	0.0839	0.0688	0.0094	-0.0155	0.2875	-0.0492	0.7435	0.0165
0.8869	0.0357	0.4266	0.1279	0.0742	0.0645	0.0122	-0.0176	0.3025	-0.0479	0.7600	0.0173
0.8692	0.0408	0.4104	0.1281	0.0649	0.0601	0.0155	-0.0197	0.3179	-0.0462	0.7765	0.0179
0.8514	0.0459	0.3944	0.1280	0.0559	0.0555	0.0195	-0.0219	0.3339	-0.0443	0.7929	0.0183
0.8336	0.0510	0.3788	0.1277	0.0475	0.0508	0.0244	-0.0242	0.3503	-0.0420	0.8093	0.0184
0.8158	0.0560	0.3635	0.1271	0.0396	0.0461	0.0303	-0.0267	0.3672	-0.0393	0.8257	0.0183
0.7980	0.0609	0.3485	0.1263	0.0325	0.0413	0.0373	-0.0292	0.3843	-0.0364	0.8420	0.0180
0.7804	0.0657	0.3337	0.1252	0.0262	0.0367	0.0454	-0.0317	0.4016	-0.0333	0.8582	0.0174
0.7628	0.0704	0.3190	0.1239	0.0208	0.0322	0.0545	-0.0342	0.4190	-0.0300	0.8743	0.0167
0.7453	0.0750	0.3044	0.1223	0.0161	0.0279	0.0644	-0.0365	0.4363	-0.0266	0.8903	0.0156
0.7279	0.0795	0.2898	0.1204	0.0124	0.0240	0.0747	-0.0387	0.4535	-0.0233	0.9062	0.0143
0.7106	0.0838	0.2754	0.1184	0.0095	0.0205	0.0855	-0.0407	0.4707	-0.0200	0.9219	0.0128
0.6934	0.0880	0.2613	0.1163	0.0072	0.0174	0.0965	-0.0425	0.4878	-0.0167	0.9373	0.0109
0.6762	0.0920	0.2475	0.1141	0.0053	0.0145	0.1078	-0.0442	0.5050	-0.0136	0.9524	0.0088
0.6590	0.0959	0.2340	0.1117	0.0038	0.0117	0.1192	-0.0456	0.5222	-0.0105	0.9669	0.0066
0.6418	0.0997	0.2209	0.1092	0.0025	0.0090	0.1309	-0.0470	0.5395	-0.0075	0.9804	0.0041
0.6247	0.1032	0.2081	0.1065	0.0016	0.0065	0.1427	-0.0481	0.5568	-0.0047	0.9927	0.0017
0.6075	0.1066	0.1955	0.1037	0.0009	0.0042	0.1547	-0.0491	0.5741	-0.0020	1.0000	0.0000
0.5904	0.1099	0.1832	0.1008	0.0004	0.0021	0.1669	-0.0500	0.5913	0.0006		
0.5735	0.1129	0.1711	0.0977	0.0002	0.0002	0.1793	-0.0507	0.6086	0.0031		

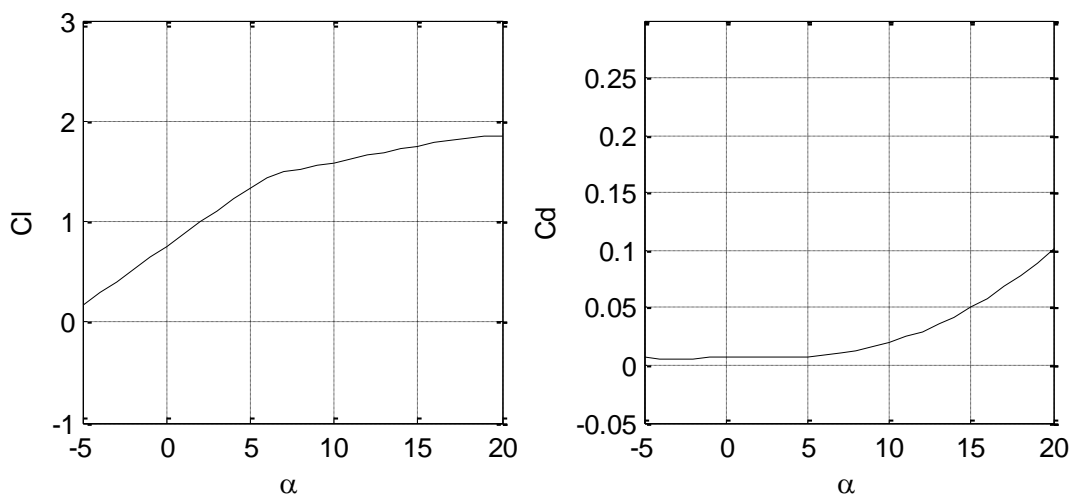


Figure A.24 Lift and drag coefficients of M0.0\_0.65\_0.35-0.170 ( $Rn = 2.0 \times 10^6$ ).  
 Ju Young Kang, University of Strathclyde, Jan 2011. Page 150

**A.3 Targeting S814 (40% radius)**

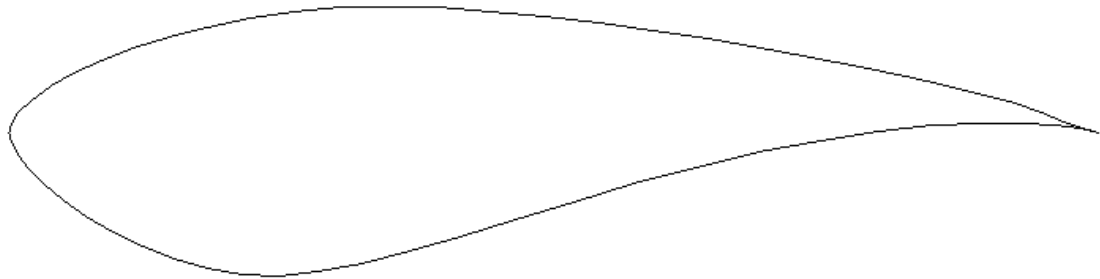


Figure A.25 Profile of S814.

Table A.13 Coordinates of S814.

x/c	y/c	x/c	y/c	x/c	y/c	x/c	y/c
1.00000	0.00000	0.39435	0.11565	0.00025	-0.00255	0.41459	-0.09525
0.99628	0.00108	0.34826	0.11696	0.00068	-0.00458	0.46526	-0.07966
0.98568	0.00464	0.30438	0.11563	0.00093	-0.00549	0.51881	-0.06324
0.96943	0.01069	0.26198	0.11161	0.00638	-0.01743	0.57457	-0.04689
0.94857	0.01853	0.22134	0.10563	0.01679	-0.03158	0.63164	-0.03152
0.92363	0.02716	0.18290	0.09800	0.03137	-0.04649	0.68891	-0.01790
0.89451	0.03574	0.14711	0.08897	0.04964	-0.06166	0.74513	-0.00669
0.86085	0.04418	0.11437	0.07876	0.07124	-0.07671	0.79891	0.00170
0.82302	0.05275	0.08504	0.06762	0.09561	-0.09109	0.84883	0.00707
0.78159	0.06142	0.05948	0.05577	0.12244	-0.10431	0.89349	0.00952
0.73713	0.07011	0.03801	0.04346	0.15120	-0.11573	0.93161	0.00940
0.69027	0.07866	0.02095	0.03095	0.18167	-0.12458	0.96209	0.00728
0.64165	0.08690	0.00858	0.01856	0.21367	-0.13024	0.98382	0.00389
0.59191	0.09463	0.00143	0.00667	0.24714	-0.13131	0.99613	0.00101
0.54169	0.10163	0.00112	0.00580	0.28394	-0.12776	1.00000	0.00000
0.49163	0.10765	0.00034	0.00296	0.32378	-0.12029		
0.44232	0.11242	0.00000	0.00022	0.36733	-0.10909		

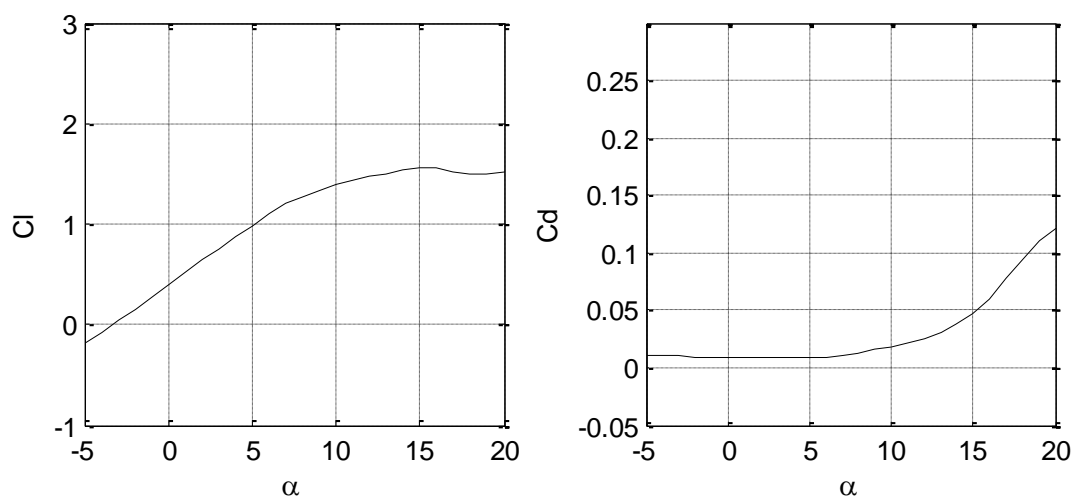


Figure A.26 Lift and drag coefficients of S814 ( $R_n = 1.5 \times 10^6$ ).

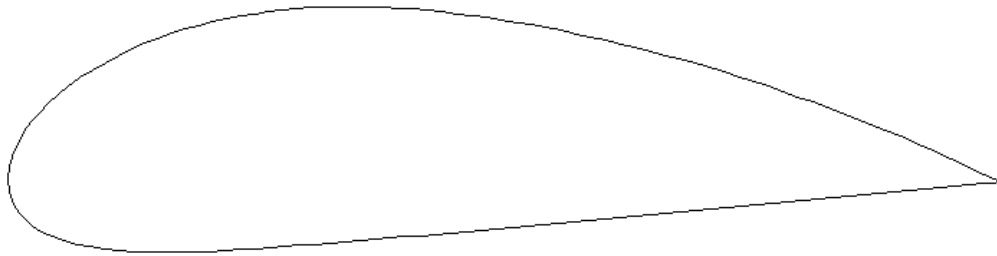


Figure A.27 Profile of N6409-0.240.

Table A.14 Coordinates of N6409-0.240.

x/c	y/c	x/c	y/c	x/c	y/c	x/c	y/c	x/c	y/c	x/c	y/c
1.0000	0.0000	0.6177	0.1402	0.2202	0.1654	0.0000	0.0011	0.2339	-0.0686	0.6480	-0.0327
0.9945	0.0027	0.6027	0.1437	0.2061	0.1623	0.0001	-0.0011	0.2491	-0.0676	0.6635	-0.0312
0.9841	0.0078	0.5877	0.1471	0.1922	0.1588	0.0005	-0.0051	0.2644	-0.0665	0.6789	-0.0297
0.9716	0.0138	0.5727	0.1504	0.1782	0.1548	0.0013	-0.0102	0.2796	-0.0653	0.6943	-0.0282
0.9579	0.0203	0.5577	0.1535	0.1644	0.1505	0.0025	-0.0147	0.2947	-0.0642	0.7097	-0.0267
0.9436	0.0269	0.5427	0.1564	0.1507	0.1456	0.0040	-0.0192	0.3099	-0.0629	0.7251	-0.0252
0.9291	0.0334	0.5276	0.1592	0.1372	0.1403	0.0059	-0.0234	0.3250	-0.0617	0.7405	-0.0237
0.9145	0.0398	0.5126	0.1618	0.1238	0.1345	0.0081	-0.0274	0.3401	-0.0604	0.7559	-0.0223
0.8999	0.0461	0.4976	0.1642	0.1106	0.1282	0.0109	-0.0314	0.3551	-0.0592	0.7713	-0.0208
0.8853	0.0523	0.4826	0.1665	0.0976	0.1213	0.0142	-0.0353	0.3702	-0.0580	0.7867	-0.0194
0.8706	0.0583	0.4675	0.1686	0.0848	0.1139	0.0181	-0.0392	0.3853	-0.0568	0.8021	-0.0180
0.8559	0.0642	0.4524	0.1705	0.0726	0.1059	0.0229	-0.0432	0.4005	-0.0556	0.8175	-0.0166
0.8411	0.0700	0.4374	0.1722	0.0609	0.0974	0.0288	-0.0473	0.4158	-0.0544	0.8329	-0.0152
0.8263	0.0756	0.4224	0.1738	0.0499	0.0885	0.0361	-0.0515	0.4313	-0.0531	0.8483	-0.0138
0.8115	0.0811	0.4076	0.1751	0.0399	0.0793	0.0451	-0.0556	0.4468	-0.0518	0.8637	-0.0124
0.7967	0.0865	0.3928	0.1761	0.0314	0.0705	0.0560	-0.0595	0.4624	-0.0505	0.8791	-0.0110
0.7819	0.0917	0.3782	0.1769	0.0244	0.0624	0.0684	-0.0629	0.4780	-0.0491	0.8945	-0.0096
0.7671	0.0968	0.3637	0.1775	0.0188	0.0549	0.0819	-0.0657	0.4935	-0.0477	0.9099	-0.0082
0.7522	0.1017	0.3492	0.1777	0.0143	0.0480	0.0961	-0.0678	0.5090	-0.0462	0.9252	-0.0068
0.7373	0.1066	0.3348	0.1776	0.0107	0.0417	0.1109	-0.0694	0.5244	-0.0447	0.9405	-0.0054
0.7224	0.1112	0.3203	0.1773	0.0078	0.0359	0.1259	-0.0704	0.5399	-0.0433	0.9556	-0.0041
0.7075	0.1158	0.3059	0.1766	0.0056	0.0306	0.1412	-0.0710	0.5554	-0.0418	0.9701	-0.0028
0.6926	0.1202	0.2915	0.1756	0.0038	0.0257	0.1567	-0.0712	0.5708	-0.0403	0.9834	-0.0015
0.6776	0.1245	0.2772	0.1742	0.0024	0.0210	0.1722	-0.0712	0.5863	-0.0387	0.9942	-0.0005
0.6627	0.1286	0.2628	0.1726	0.0014	0.0163	0.1877	-0.0708	0.6017	-0.0372	1.0000	0.0000
0.6477	0.1326	0.2486	0.1706	0.0007	0.0113	0.2032	-0.0702	0.6172	-0.0357		
0.6327	0.1365	0.2343	0.1682	0.0002	0.0069	0.2186	-0.0695	0.6326	-0.0342		

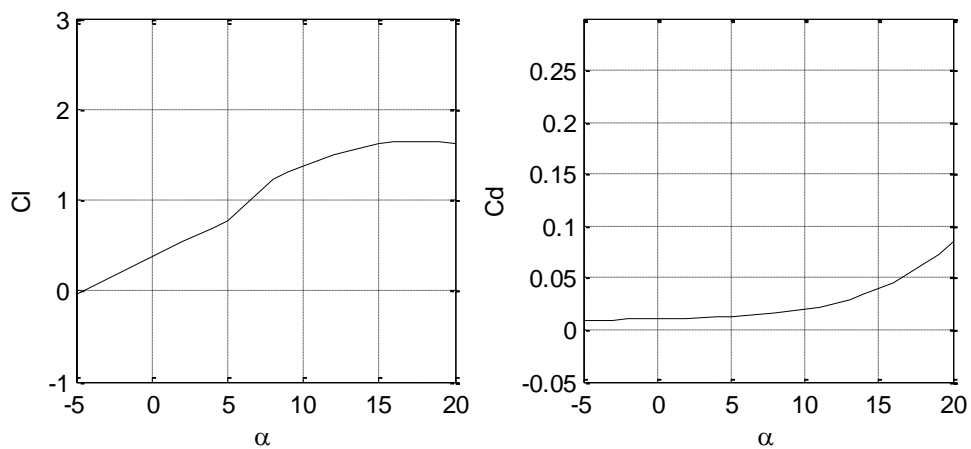


Figure A.28 Lift and drag coefficients of N6409-0.240 ( $Rn = 1.5 \times 10^6$ ).

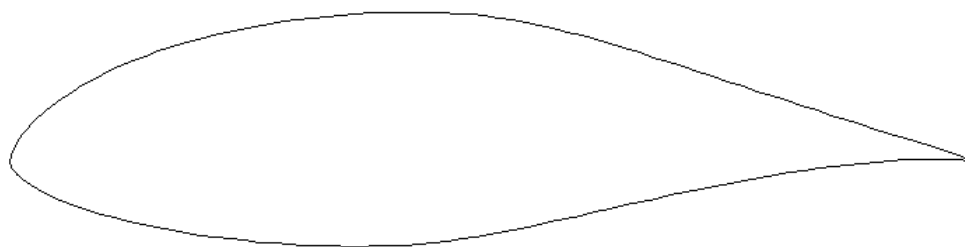


Figure A.29 Profile of E1098- 0.240 (M1.00, M0.0\_1.0\_0.0).

Table A.15 Coordinates of E1098- 0.240 (M1.00, M0.0\_1.0\_0.0).

x/c	y/c	x/c	y/c	x/c	y/c	x/c	y/c	x/c	y/c	x/c	y/c
1.0000	0.0000	0.6071	0.1247	0.2102	0.1291	0.0000	-0.0002	0.2428	-0.0818	0.6536	-0.0420
0.9938	0.0018	0.5922	0.1289	0.1958	0.1254	0.0001	-0.0020	0.2580	-0.0832	0.6692	-0.0386
0.9829	0.0054	0.5775	0.1328	0.1816	0.1215	0.0006	-0.0042	0.2732	-0.0845	0.6847	-0.0352
0.9703	0.0097	0.5629	0.1364	0.1675	0.1173	0.0016	-0.0066	0.2884	-0.0855	0.7002	-0.0319
0.9569	0.0142	0.5485	0.1396	0.1536	0.1127	0.0029	-0.0090	0.3037	-0.0864	0.7157	-0.0287
0.9427	0.0187	0.5342	0.1426	0.1399	0.1078	0.0047	-0.0114	0.3189	-0.0870	0.7311	-0.0256
0.9280	0.0233	0.5200	0.1451	0.1265	0.1026	0.0070	-0.0139	0.3341	-0.0874	0.7466	-0.0226
0.9128	0.0279	0.5058	0.1473	0.1132	0.0971	0.0097	-0.0164	0.3493	-0.0877	0.7619	-0.0197
0.8975	0.0327	0.4915	0.1491	0.1003	0.0912	0.0130	-0.0191	0.3645	-0.0877	0.7773	-0.0170
0.8821	0.0374	0.4771	0.1505	0.0877	0.0849	0.0169	-0.0219	0.3795	-0.0875	0.7926	-0.0143
0.8668	0.0422	0.4625	0.1516	0.0756	0.0784	0.0218	-0.0251	0.3945	-0.0870	0.8079	-0.0119
0.8514	0.0471	0.4477	0.1524	0.0641	0.0715	0.0279	-0.0287	0.4092	-0.0862	0.8232	-0.0095
0.8360	0.0520	0.4329	0.1530	0.0533	0.0645	0.0355	-0.0327	0.4239	-0.0852	0.8384	-0.0074
0.8207	0.0570	0.4180	0.1532	0.0436	0.0575	0.0447	-0.0370	0.4385	-0.0838	0.8536	-0.0054
0.8053	0.0619	0.4030	0.1531	0.0351	0.0507	0.0554	-0.0415	0.4532	-0.0821	0.8687	-0.0036
0.7899	0.0670	0.3881	0.1528	0.0278	0.0441	0.0676	-0.0461	0.4679	-0.0801	0.8838	-0.0020
0.7745	0.0720	0.3731	0.1522	0.0215	0.0378	0.0807	-0.0505	0.4828	-0.0778	0.8987	-0.0006
0.7591	0.0770	0.3581	0.1514	0.0166	0.0323	0.0944	-0.0546	0.4978	-0.0752	0.9136	0.0005
0.7437	0.0821	0.3432	0.1503	0.0128	0.0275	0.1085	-0.0585	0.5131	-0.0724	0.9282	0.0013
0.7283	0.0871	0.3282	0.1489	0.0098	0.0234	0.1230	-0.0621	0.5285	-0.0693	0.9427	0.0018
0.7130	0.0921	0.3133	0.1473	0.0074	0.0196	0.1376	-0.0654	0.5440	-0.0661	0.9567	0.0021
0.6978	0.0970	0.2984	0.1455	0.0053	0.0160	0.1524	-0.0685	0.5596	-0.0628	0.9701	0.0020
0.6826	0.1019	0.2836	0.1434	0.0037	0.0127	0.1673	-0.0713	0.5753	-0.0594	0.9826	0.0015
0.6674	0.1067	0.2688	0.1411	0.0024	0.0096	0.1822	-0.0738	0.5910	-0.0559	0.9936	0.0007
0.6522	0.1114	0.2540	0.1385	0.0014	0.0068	0.1973	-0.0761	0.6067	-0.0524	1.0000	0.0000
0.6371	0.1160	0.2393	0.1356	0.0006	0.0043	0.2124	-0.0782	0.6224	-0.0489		
0.6221	0.1204	0.2247	0.1325	0.0002	0.0021	0.2276	-0.0801	0.6380	-0.0454		

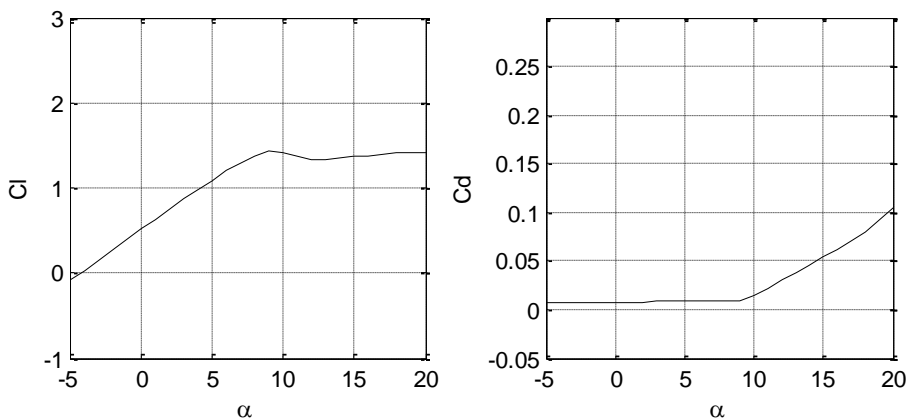


Figure A.30 Lift and drag coefficients of E1098- 0.240 (M1.00, M0.0\_1.0\_0.0) ( $R_n = 1.5 \times 10^6$ ).

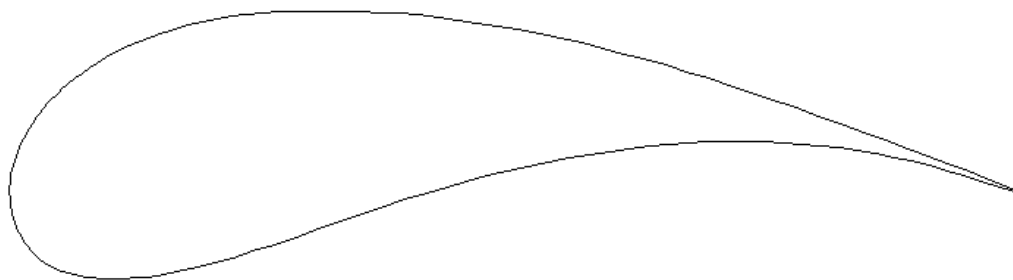


Figure A.31 Profile of G652-0.240.

Table A.16 Coordinates of G652-0.240.

x/c	y/c	x/c	y/c	x/c	y/c	x/c	y/c	x/c	y/c	x/c	y/c
1.0000	0.0000	0.4630	0.1677	0.0646	0.1129	0.0007	-0.0083	0.0975	-0.0839	0.5740	0.0383
0.9913	0.0040	0.4433	0.1706	0.0582	0.1079	0.0013	-0.0127	0.1060	-0.0840	0.5939	0.0414
0.9773	0.0101	0.4240	0.1731	0.0523	0.1029	0.0021	-0.0167	0.1151	-0.0837	0.6136	0.0441
0.9622	0.0165	0.4049	0.1751	0.0469	0.0979	0.0031	-0.0207	0.1249	-0.0831	0.6331	0.0464
0.9459	0.0230	0.3857	0.1767	0.0419	0.0930	0.0042	-0.0247	0.1355	-0.0820	0.6524	0.0483
0.9282	0.0298	0.3662	0.1779	0.0374	0.0881	0.0056	-0.0287	0.1469	-0.0805	0.6715	0.0498
0.9090	0.0369	0.3463	0.1787	0.0331	0.0832	0.0071	-0.0328	0.1592	-0.0784	0.6904	0.0508
0.8887	0.0444	0.3261	0.1791	0.0293	0.0784	0.0089	-0.0369	0.1726	-0.0757	0.7091	0.0513
0.8672	0.0521	0.3060	0.1792	0.0257	0.0736	0.0109	-0.0409	0.1873	-0.0722	0.7278	0.0514
0.8452	0.0600	0.2865	0.1789	0.0224	0.0688	0.0130	-0.0449	0.2035	-0.0680	0.7464	0.0510
0.8228	0.0680	0.2681	0.1784	0.0194	0.0640	0.0154	-0.0487	0.2213	-0.0630	0.7650	0.0501
0.8004	0.0759	0.2509	0.1774	0.0166	0.0593	0.0179	-0.0525	0.2409	-0.0570	0.7836	0.0488
0.7781	0.0838	0.2348	0.1760	0.0140	0.0545	0.0207	-0.0560	0.2618	-0.0503	0.8021	0.0470
0.7560	0.0914	0.2195	0.1741	0.0117	0.0498	0.0236	-0.0594	0.2837	-0.0429	0.8205	0.0448
0.7342	0.0989	0.2049	0.1716	0.0096	0.0451	0.0269	-0.0625	0.3059	-0.0353	0.8387	0.0421
0.7126	0.1061	0.1907	0.1686	0.0078	0.0405	0.0304	-0.0653	0.3281	-0.0276	0.8568	0.0390
0.6912	0.1129	0.1767	0.1650	0.0061	0.0360	0.0341	-0.0680	0.3498	-0.0201	0.8748	0.0355
0.6701	0.1195	0.1629	0.1610	0.0047	0.0316	0.0382	-0.0703	0.3711	-0.0130	0.8925	0.0315
0.6491	0.1258	0.1495	0.1567	0.0035	0.0273	0.0426	-0.0725	0.3919	-0.0062	0.9101	0.0271
0.6284	0.1317	0.1368	0.1523	0.0024	0.0231	0.0473	-0.0745	0.4123	0.0001	0.9275	0.0224
0.6077	0.1373	0.1252	0.1478	0.0016	0.0189	0.0524	-0.0764	0.4326	0.0061	0.9445	0.0174
0.5870	0.1426	0.1144	0.1432	0.0009	0.0148	0.0577	-0.0780	0.4528	0.0118	0.9610	0.0125
0.5662	0.1476	0.1046	0.1384	0.0004	0.0101	0.0633	-0.0795	0.4730	0.0171	0.9765	0.0076
0.5454	0.1523	0.0954	0.1335	0.0001	0.0061	0.0693	-0.0808	0.4932	0.0220	0.9910	0.0030
0.5245	0.1566	0.0869	0.1284	0.0000	0.0028	0.0756	-0.0819	0.5135	0.0267	1.0000	0.0000
0.5037	0.1607	0.0789	0.1232	0.0001	-0.0002	0.0824	-0.0828	0.5337	0.0309		
0.4831	0.1644	0.0715	0.1180	0.0003	-0.0039	0.0897	-0.0835	0.5539	0.0348		

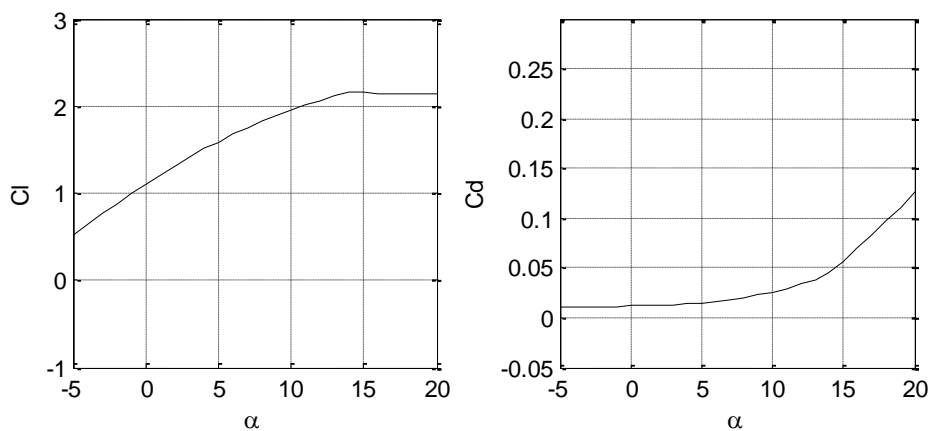


Figure A.32 Lift and drag coefficients of G652-0.240 ( $Rn = 1.5 \times 10^6$ ).

**A.4 Targeting S815 (30% radius)**

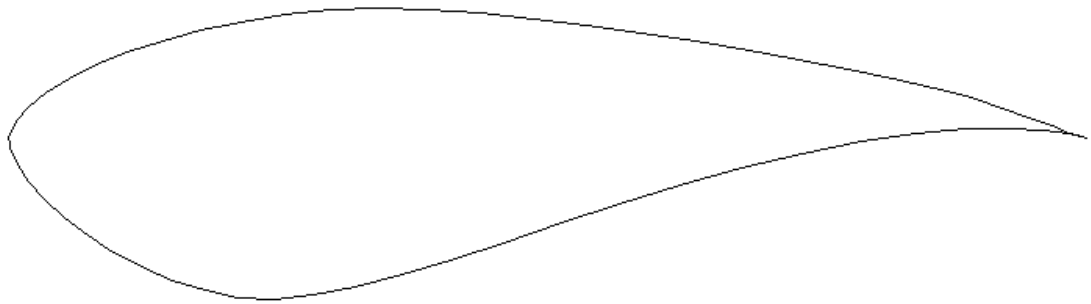


Figure A.33 Profile of S815.

Table A.17 Coordinates of S815.

x/c	y/c	x/c	y/c	x/c	y/c	x/c	y/c
1.0000	0.0000	0.3889	0.1189	0.0000	-0.0013	0.4106	-0.1125
0.9962	0.0011	0.3428	0.1202	0.0004	-0.0046	0.4601	-0.0963
0.9853	0.0049	0.2990	0.1186	0.0015	-0.0090	0.5124	-0.0788
0.9687	0.0112	0.2568	0.1143	0.0078	-0.0238	0.5671	-0.0608
0.9476	0.0194	0.2163	0.1079	0.0186	-0.0398	0.6234	-0.0434
0.9223	0.0284	0.1780	0.0998	0.0332	-0.0564	0.6804	-0.0274
0.8929	0.0373	0.1424	0.0903	0.0514	-0.0731	0.7369	-0.0138
0.8589	0.0460	0.1100	0.0795	0.0728	-0.0895	0.7915	-0.0030
0.8206	0.0548	0.0810	0.0678	0.0968	-0.1050	0.8426	0.0043
0.7788	0.0637	0.0558	0.0553	0.1233	-0.1192	0.8888	0.0083
0.7339	0.0726	0.0347	0.0423	0.1516	-0.1315	0.9285	0.0092
0.6866	0.0813	0.0182	0.0292	0.1816	-0.1410	0.9604	0.0075
0.6375	0.0898	0.0065	0.0162	0.2131	-0.1471	0.9832	0.0041
0.5874	0.0976	0.0017	0.0078	0.2459	-0.1485	0.9960	0.0011
0.5368	0.1048	0.0007	0.0047	0.2821	-0.1450	1.0000	0.0000
0.4865	0.1109	0.0004	0.0038	0.3216	-0.1376		
0.4370	0.1157	0.0001	0.0017	0.3644	-0.1265		

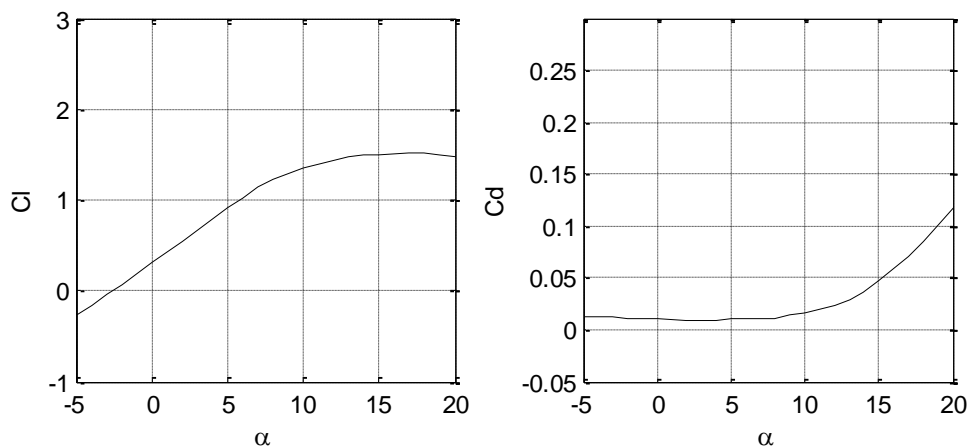


Figure A.34 Lift and drag coefficients of S815 ( $R_n = 1.2 \times 10^6$ ).

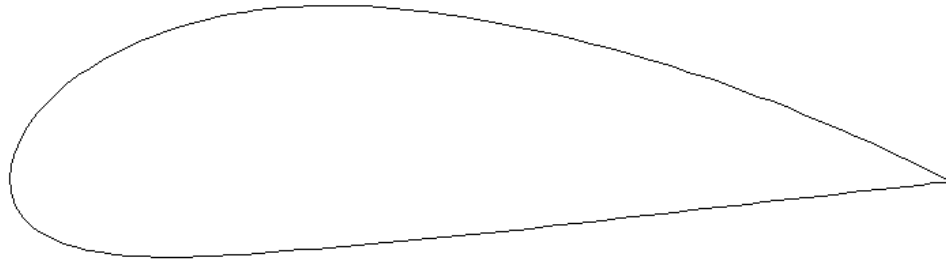


Figure A.35 Profile of N6409-0.260 (M1.00).

Table A.18 Coordinates of N6409-0.260 (M1.00).

x/c	y/c	x/c	y/c	x/c	y/c	x/c	y/c	x/c	y/c	x/c	y/c
1.0000	0.0000	0.6177	0.1475	0.2202	0.1752	0.0000	0.0010	0.2339	-0.0784	0.6480	-0.0395
0.9945	0.0029	0.6027	0.1513	0.2061	0.1720	0.0001	-0.0014	0.2491	-0.0775	0.6635	-0.0378
0.9841	0.0082	0.5877	0.1549	0.1922	0.1684	0.0005	-0.0057	0.2644	-0.0765	0.6789	-0.0361
0.9716	0.0145	0.5727	0.1583	0.1782	0.1643	0.0013	-0.0113	0.2796	-0.0753	0.6943	-0.0343
0.9579	0.0213	0.5577	0.1616	0.1644	0.1597	0.0025	-0.0162	0.2947	-0.0742	0.7097	-0.0326
0.9436	0.0282	0.5427	0.1647	0.1507	0.1547	0.0040	-0.0211	0.3099	-0.0729	0.7251	-0.0308
0.9291	0.0351	0.5276	0.1677	0.1372	0.1491	0.0059	-0.0256	0.3250	-0.0716	0.7405	-0.0291
0.9145	0.0418	0.5126	0.1705	0.1238	0.1431	0.0081	-0.0301	0.3401	-0.0704	0.7559	-0.0274
0.8999	0.0484	0.4976	0.1731	0.1106	0.1364	0.0109	-0.0344	0.3551	-0.0691	0.7713	-0.0257
0.8853	0.0549	0.4826	0.1755	0.0976	0.1292	0.0142	-0.0387	0.3702	-0.0678	0.7867	-0.0240
0.8706	0.0613	0.4675	0.1777	0.0848	0.1214	0.0181	-0.0430	0.3853	-0.0665	0.8021	-0.0222
0.8559	0.0674	0.4524	0.1798	0.0726	0.1130	0.0229	-0.0475	0.4005	-0.0652	0.8175	-0.0205
0.8411	0.0735	0.4374	0.1816	0.0609	0.1040	0.0288	-0.0521	0.4158	-0.0639	0.8329	-0.0188
0.8263	0.0794	0.4224	0.1833	0.0499	0.0946	0.0361	-0.0567	0.4313	-0.0625	0.8483	-0.0171
0.8115	0.0852	0.4076	0.1847	0.0399	0.0848	0.0451	-0.0614	0.4468	-0.0611	0.8637	-0.0154
0.7967	0.0908	0.3928	0.1858	0.0314	0.0755	0.0560	-0.0658	0.4624	-0.0596	0.8791	-0.0137
0.7819	0.0963	0.3782	0.1867	0.0244	0.0668	0.0684	-0.0698	0.4780	-0.0581	0.8945	-0.0120
0.7671	0.1017	0.3637	0.1873	0.0188	0.0588	0.0819	-0.0731	0.4935	-0.0565	0.9099	-0.0103
0.7522	0.1069	0.3492	0.1876	0.0143	0.0514	0.0961	-0.0757	0.5090	-0.0549	0.9252	-0.0086
0.7373	0.1120	0.3348	0.1876	0.0107	0.0447	0.1109	-0.0776	0.5244	-0.0533	0.9405	-0.0068
0.7224	0.1169	0.3203	0.1872	0.0078	0.0385	0.1259	-0.0790	0.5399	-0.0516	0.9556	-0.0051
0.7075	0.1217	0.3059	0.1866	0.0056	0.0329	0.1412	-0.0799	0.5554	-0.0499	0.9701	-0.0035
0.6926	0.1264	0.2915	0.1856	0.0038	0.0276	0.1567	-0.0804	0.5708	-0.0482	0.9834	-0.0019
0.6776	0.1309	0.2772	0.1842	0.0024	0.0224	0.1722	-0.0805	0.5863	-0.0465	0.9942	-0.0007
0.6627	0.1353	0.2628	0.1825	0.0014	0.0174	0.1877	-0.0803	0.6017	-0.0448	1.0000	0.0000
0.6477	0.1395	0.2486	0.1805	0.0007	0.0120	0.2032	-0.0799	0.6172	-0.0430		
0.6327	0.1436	0.2343	0.1780	0.0002	0.0072	0.2186	-0.0792	0.6326	-0.0413		

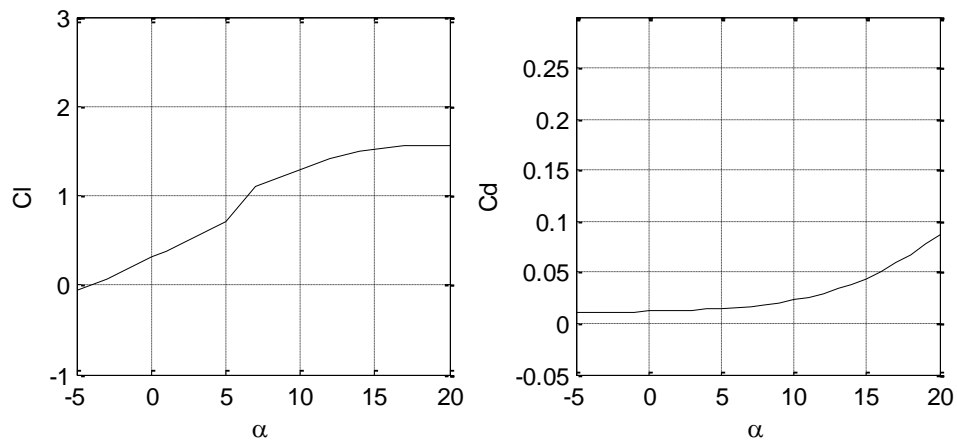


Figure A.36 Lift and drag coefficients of N6409-0.260 (M1.00) ( $R_n = 1.2 \times 10^6$ ).



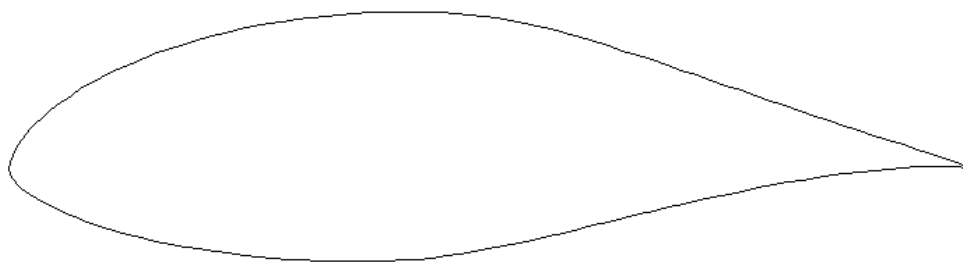


Figure A.37 Profile of E1098-0.260.

Table A.19 Coordinates of E1098-0.260 (M1.00).

x/c	y/c	x/c	y/c	x/c	y/c	x/c	y/c	x/c	y/c	x/c	y/c
1.0000	0.0000	0.6071	0.1321	0.2102	0.1377	0.0000	-0.0003	0.2428	-0.0908	0.6536	-0.0483
0.9938	0.0018	0.5922	0.1366	0.1958	0.1338	0.0001	-0.0022	0.2580	-0.0925	0.6692	-0.0446
0.9829	0.0055	0.5775	0.1408	0.1816	0.1296	0.0006	-0.0045	0.2732	-0.0939	0.6847	-0.0409
0.9703	0.0100	0.5629	0.1446	0.1675	0.1251	0.0016	-0.0072	0.2884	-0.0951	0.7002	-0.0373
0.9569	0.0147	0.5485	0.1482	0.1536	0.1203	0.0029	-0.0098	0.3037	-0.0960	0.7157	-0.0337
0.9427	0.0194	0.5342	0.1513	0.1399	0.1151	0.0047	-0.0125	0.3189	-0.0968	0.7311	-0.0303
0.9280	0.0242	0.5200	0.1541	0.1265	0.1095	0.0070	-0.0152	0.3341	-0.0973	0.7466	-0.0270
0.9128	0.0291	0.5058	0.1565	0.1132	0.1036	0.0097	-0.0181	0.3493	-0.0976	0.7619	-0.0237
0.8975	0.0341	0.4915	0.1585	0.1003	0.0973	0.0130	-0.0210	0.3645	-0.0977	0.7773	-0.0206
0.8821	0.0391	0.4771	0.1601	0.0877	0.0906	0.0169	-0.0242	0.3795	-0.0975	0.7926	-0.0177
0.8668	0.0442	0.4625	0.1613	0.0756	0.0837	0.0218	-0.0277	0.3945	-0.0970	0.8079	-0.0149
0.8514	0.0493	0.4477	0.1622	0.0641	0.0764	0.0279	-0.0317	0.4092	-0.0962	0.8232	-0.0123
0.8360	0.0545	0.4329	0.1628	0.0533	0.0689	0.0355	-0.0362	0.4239	-0.0951	0.8384	-0.0098
0.8207	0.0597	0.4180	0.1631	0.0436	0.0614	0.0447	-0.0410	0.4385	-0.0937	0.8536	-0.0075
0.8053	0.0650	0.4030	0.1631	0.0351	0.0541	0.0554	-0.0460	0.4532	-0.0919	0.8687	-0.0055
0.7899	0.0704	0.3881	0.1628	0.0278	0.0471	0.0676	-0.0510	0.4679	-0.0898	0.8838	-0.0036
0.7745	0.0757	0.3731	0.1622	0.0215	0.0404	0.0807	-0.0560	0.4828	-0.0873	0.8987	-0.0020
0.7591	0.0811	0.3581	0.1613	0.0166	0.0345	0.0944	-0.0606	0.4978	-0.0845	0.9136	-0.0007
0.7437	0.0865	0.3432	0.1602	0.0128	0.0295	0.1085	-0.0649	0.5131	-0.0815	0.9282	0.0004
0.7283	0.0918	0.3282	0.1588	0.0098	0.0250	0.1230	-0.0689	0.5285	-0.0782	0.9427	0.0011
0.7130	0.0971	0.3133	0.1571	0.0074	0.0210	0.1376	-0.0726	0.5440	-0.0747	0.9567	0.0016
0.6978	0.1024	0.2984	0.1551	0.0053	0.0172	0.1524	-0.0760	0.5596	-0.0711	0.9701	0.0016
0.6826	0.1076	0.2836	0.1529	0.0037	0.0136	0.1673	-0.0791	0.5753	-0.0674	0.9826	0.0013
0.6674	0.1127	0.2688	0.1504	0.0024	0.0103	0.1822	-0.0820	0.5910	-0.0636	0.9936	0.0007
0.6522	0.1178	0.2540	0.1477	0.0014	0.0074	0.1973	-0.0845	0.6067	-0.0598	1.0000	0.0000
0.6371	0.1227	0.2393	0.1446	0.0006	0.0046	0.2124	-0.0869	0.6224	-0.0560		
0.6221	0.1275	0.2247	0.1413	0.0002	0.0023	0.2276	-0.0890	0.6380	-0.0521		

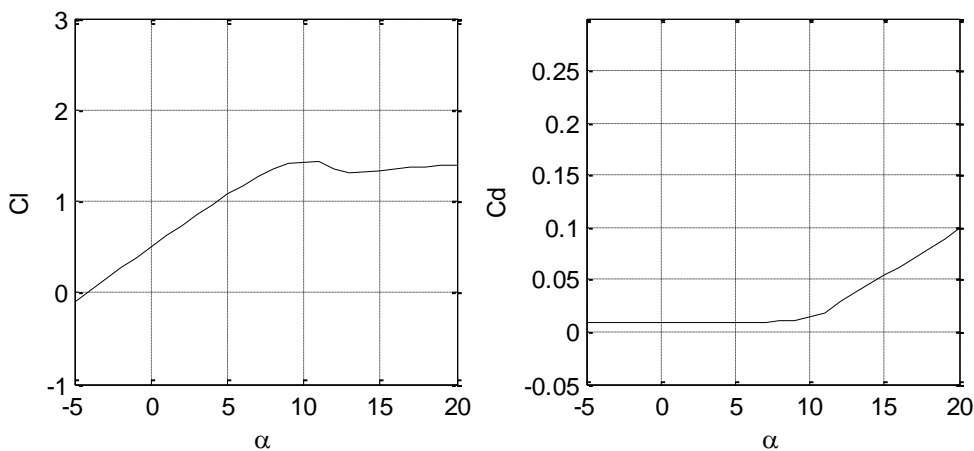


Figure A.38 Lift and drag coefficients of E1098-0.260 (M1.00) ( $R_n = 1.2 \times 10^6$ ).

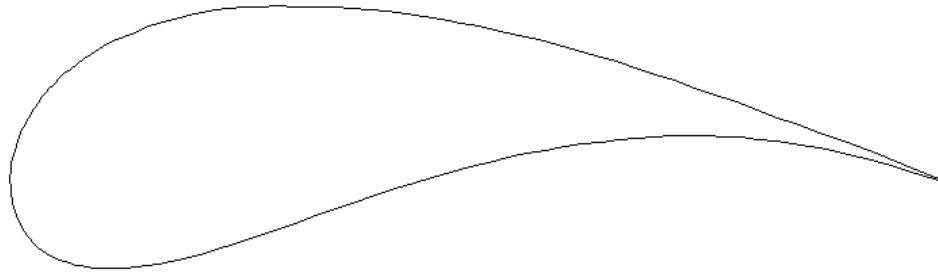


Figure A.39 Profile of G652-0.260.

Table A.20 Coordinates of G652-0.260.

x/c	y/c	x/c	y/c	x/c	y/c	x/c	y/c	x/c	y/c	x/c	y/c
1.0000	0.0000	0.4630	0.1741	0.0646	0.1209	0.0007	-0.0091	0.0975	-0.0930	0.5740	0.0338
0.9913	0.0040	0.4433	0.1774	0.0582	0.1156	0.0013	-0.0139	0.1060	-0.0933	0.5939	0.0373
0.9773	0.0102	0.4240	0.1802	0.0523	0.1103	0.0021	-0.0183	0.1151	-0.0932	0.6136	0.0403
0.9622	0.0167	0.4049	0.1825	0.0469	0.1051	0.0031	-0.0226	0.1249	-0.0927	0.6331	0.0430
0.9459	0.0233	0.3857	0.1844	0.0419	0.0999	0.0042	-0.0270	0.1355	-0.0918	0.6524	0.0451
0.9282	0.0302	0.3662	0.1859	0.0374	0.0947	0.0056	-0.0314	0.1469	-0.0903	0.6715	0.0469
0.9090	0.0373	0.3463	0.1870	0.0331	0.0895	0.0071	-0.0358	0.1592	-0.0883	0.6904	0.0482
0.8887	0.0449	0.3261	0.1877	0.0293	0.0844	0.0089	-0.0402	0.1726	-0.0856	0.7091	0.0490
0.8672	0.0527	0.3060	0.1881	0.0257	0.0792	0.0109	-0.0446	0.1873	-0.0822	0.7278	0.0493
0.8452	0.0608	0.2865	0.1881	0.0224	0.0741	0.0130	-0.0489	0.2035	-0.0780	0.7464	0.0491
0.8228	0.0690	0.2681	0.1878	0.0194	0.0690	0.0154	-0.0531	0.2213	-0.0729	0.7650	0.0485
0.8004	0.0771	0.2509	0.1870	0.0166	0.0638	0.0179	-0.0572	0.2409	-0.0668	0.7836	0.0474
0.7781	0.0852	0.2348	0.1858	0.0140	0.0587	0.0207	-0.0611	0.2618	-0.0598	0.8021	0.0458
0.7560	0.0931	0.2195	0.1840	0.0117	0.0536	0.0236	-0.0648	0.2837	-0.0522	0.8205	0.0438
0.7342	0.1009	0.2049	0.1816	0.0096	0.0486	0.0269	-0.0682	0.3059	-0.0442	0.8387	0.0413
0.7126	0.1083	0.1907	0.1786	0.0078	0.0436	0.0304	-0.0714	0.3281	-0.0362	0.8568	0.0383
0.6912	0.1155	0.1767	0.1750	0.0061	0.0388	0.0341	-0.0743	0.3498	-0.0284	0.8748	0.0349
0.6701	0.1224	0.1629	0.1709	0.0047	0.0340	0.0382	-0.0770	0.3711	-0.0209	0.8925	0.0310
0.6491	0.1290	0.1495	0.1665	0.0035	0.0294	0.0426	-0.0795	0.3919	-0.0138	0.9101	0.0267
0.6284	0.1353	0.1368	0.1620	0.0024	0.0248	0.0473	-0.0817	0.4123	-0.0071	0.9275	0.0220
0.6077	0.1412	0.1252	0.1574	0.0016	0.0203	0.0524	-0.0838	0.4326	-0.0008	0.9445	0.0172
0.5870	0.1469	0.1144	0.1526	0.0009	0.0158	0.0577	-0.0858	0.4528	0.0052	0.9610	0.0123
0.5662	0.1522	0.1046	0.1477	0.0004	0.0106	0.0633	-0.0875	0.4730	0.0109	0.9765	0.0075
0.5454	0.1572	0.0954	0.1425	0.0001	0.0062	0.0693	-0.0891	0.4932	0.0162	0.9910	0.0029
0.5245	0.1620	0.0869	0.1372	0.0000	0.0028	0.0756	-0.0904	0.5135	0.0212	1.0000	0.0000
0.5037	0.1664	0.0789	0.1317	0.0001	-0.0003	0.0824	-0.0915	0.5337	0.0258		
0.4831	0.1704	0.0715	0.1263	0.0003	-0.0042	0.0897	-0.0924	0.5539	0.0300		

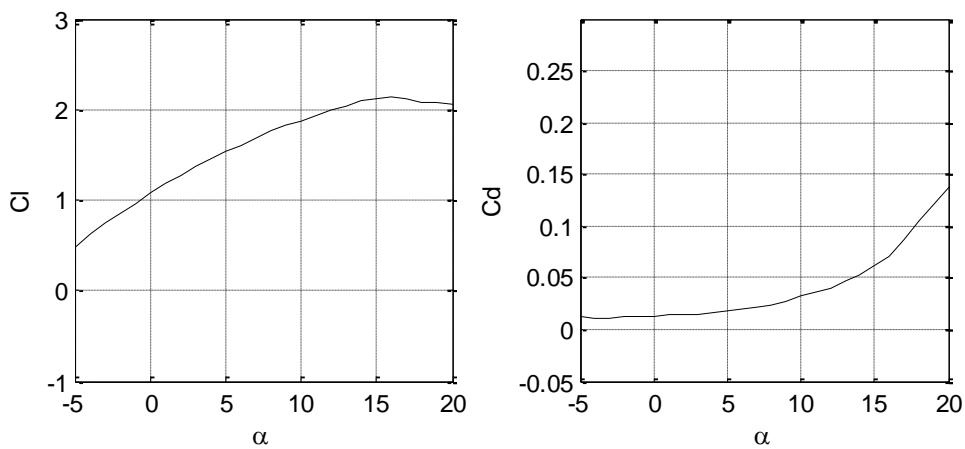


Figure A.40 Lift and drag coefficients of G652-0.260 ( $R_n = 1.2 \times 10^6$ ).

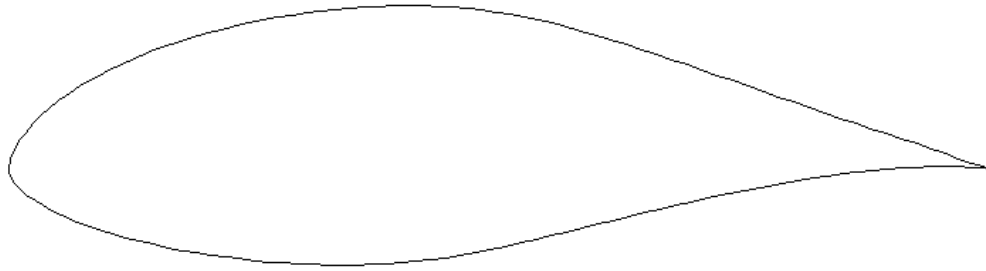


Figure A.41 Profile of M0.0\_0.95\_0.05-0.260.

Table A.21 Coordinates of M0.0\_0.95\_0.05-0.260.

x/c	y/c	x/c	y/c	x/c	y/c	x/c	y/c	x/c	y/c	x/c	y/c
1.0000	0.0000	0.5999	0.1344	0.2029	0.1371	0.0000	-0.0004	0.2355	-0.0910	0.6496	-0.0444
0.9936	0.0019	0.5848	0.1388	0.1890	0.1331	0.0002	-0.0024	0.2504	-0.0926	0.6654	-0.0407
0.9826	0.0058	0.5698	0.1429	0.1751	0.1288	0.0007	-0.0050	0.2653	-0.0940	0.6812	-0.0370
0.9699	0.0104	0.5550	0.1467	0.1615	0.1243	0.0016	-0.0078	0.2802	-0.0951	0.6969	-0.0335
0.9563	0.0152	0.5404	0.1502	0.1480	0.1194	0.0030	-0.0105	0.2953	-0.0959	0.7125	-0.0300
0.9420	0.0200	0.5258	0.1532	0.1348	0.1142	0.0048	-0.0133	0.3103	-0.0966	0.7282	-0.0266
0.9270	0.0249	0.5113	0.1559	0.1218	0.1087	0.0070	-0.0161	0.3254	-0.0970	0.7438	-0.0234
0.9116	0.0299	0.4968	0.1582	0.1090	0.1028	0.0097	-0.0190	0.3405	-0.0971	0.7593	-0.0203
0.8960	0.0351	0.4822	0.1601	0.0966	0.0966	0.0128	-0.0221	0.3556	-0.0970	0.7748	-0.0173
0.8803	0.0403	0.4675	0.1616	0.0845	0.0900	0.0167	-0.0253	0.3707	-0.0966	0.7903	-0.0145
0.8646	0.0455	0.4528	0.1627	0.0728	0.0831	0.0215	-0.0289	0.3858	-0.0959	0.8058	-0.0119
0.8489	0.0508	0.4379	0.1636	0.0617	0.0759	0.0274	-0.0329	0.4008	-0.0949	0.8212	-0.0094
0.8331	0.0562	0.4230	0.1641	0.0514	0.0685	0.0348	-0.0373	0.4158	-0.0935	0.8366	-0.0071
0.8174	0.0616	0.4080	0.1643	0.0420	0.0612	0.0436	-0.0421	0.4308	-0.0918	0.8519	-0.0051
0.8017	0.0670	0.3931	0.1641	0.0339	0.0540	0.0540	-0.0470	0.4458	-0.0897	0.8672	-0.0032
0.7860	0.0724	0.3782	0.1637	0.0268	0.0470	0.0657	-0.0520	0.4609	-0.0873	0.8824	-0.0016
0.7703	0.0779	0.3633	0.1630	0.0208	0.0404	0.0784	-0.0568	0.4761	-0.0845	0.8975	-0.0002
0.7546	0.0833	0.3484	0.1619	0.0160	0.0346	0.0916	-0.0614	0.4915	-0.0815	0.9125	0.0009
0.7390	0.0888	0.3335	0.1606	0.0123	0.0296	0.1052	-0.0656	0.5070	-0.0783	0.9273	0.0017
0.7233	0.0942	0.3187	0.1591	0.0094	0.0251	0.1192	-0.0696	0.5227	-0.0748	0.9419	0.0022
0.7078	0.0995	0.3039	0.1572	0.0071	0.0211	0.1333	-0.0732	0.5384	-0.0712	0.9561	0.0023
0.6922	0.1048	0.2892	0.1552	0.0051	0.0173	0.1476	-0.0766	0.5543	-0.0675	0.9696	0.0022
0.6767	0.1100	0.2746	0.1528	0.0035	0.0136	0.1621	-0.0796	0.5702	-0.0637	0.9823	0.0016
0.6613	0.1152	0.2601	0.1502	0.0022	0.0103	0.1766	-0.0824	0.5861	-0.0598	0.9935	0.0008
0.6458	0.1202	0.2457	0.1473	0.0013	0.0073	0.1912	-0.0849	0.6020	-0.0560	1.0000	0.0000
0.6305	0.1251	0.2313	0.1442	0.0006	0.0044	0.2059	-0.0872	0.6179	-0.0521		
0.6151	0.1298	0.2171	0.1407	0.0002	0.0019	0.2207	-0.0892	0.6338	-0.0482		

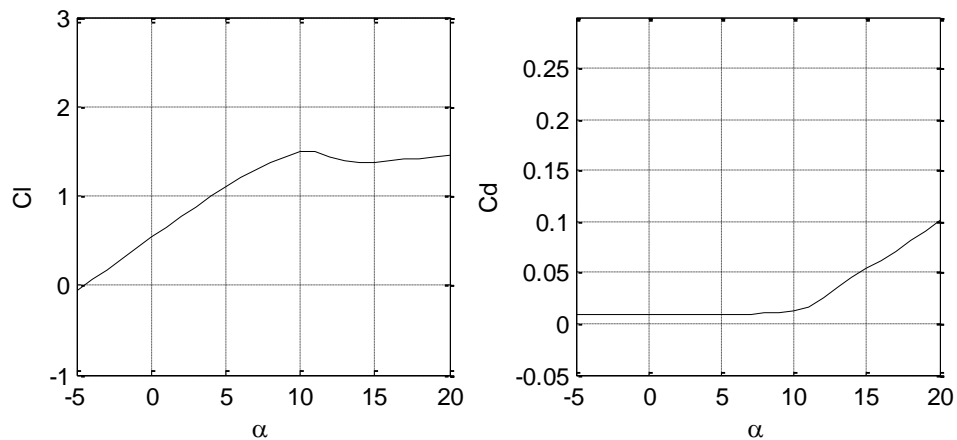


Figure A.42 Lift and drag coefficients of M0.0\_0.95\_0.05-0.260 ( $Rn = 1.2 \times 10^6$ ).

## APPENDIX B. CAVITATION OF CURRENT TURBINE AIRFOIL

As shown in Figure B.1, cavitation is the formation of vapour bubbles within a liquid in a region where the local pressure of the liquid falls to, or below, the vapour pressure of the fluid.

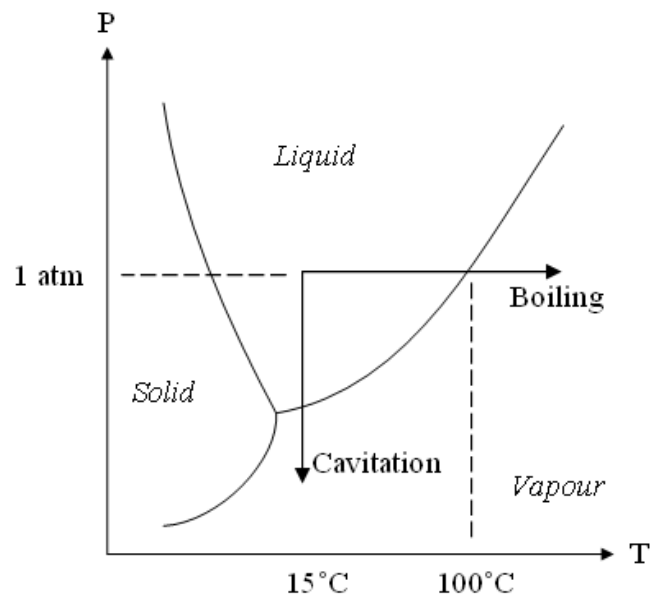


Figure B.1 Generation of Cavitation.

It may occur in places where the water has been accelerated to high velocities in the operation of current turbine. Cavitation causes a significant reduction of efficiency, because it distorts the flow pattern, and produces noise and vibration. In addition, cavitation contributes to erosion of the turbine blades.

The inception of cavitation can be predicted from the pressure distribution of the airfoil [122]. Let us consider NACA 4415 airfoil at 4 degrees angle of attack in a 2D, steady and non-viscous flow as shown in Figure B.2.

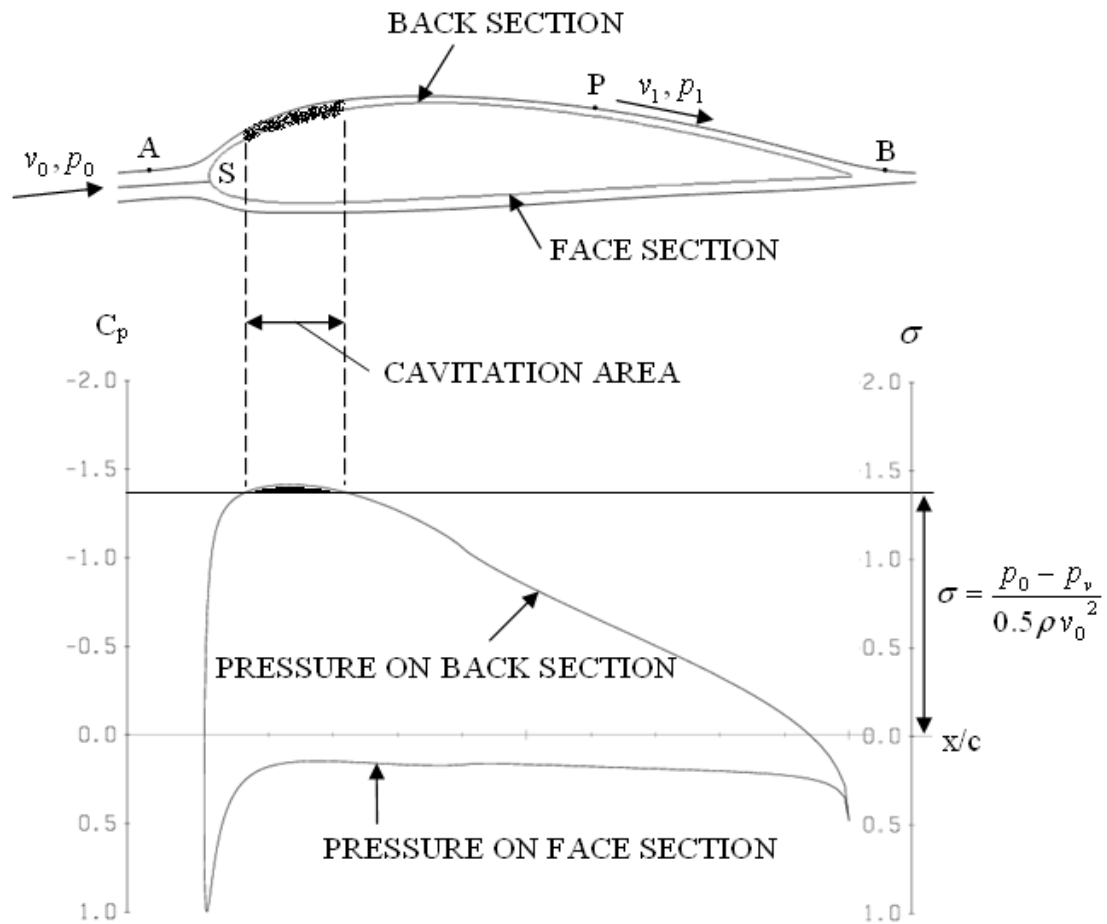


Figure B.2 Flow and pressure around an airfoil.

When the uniform steady velocity far ahead of the airfoil and the corresponding pressure are  $v_0$  and  $p_0$  respectively, and the velocity and pressure at a any point P on the streamline AB as shown in Figure B.2, the following equation can be given from Bernoulli's theorem:

$$p_1 + \frac{1}{2} \rho v_1^2 = p_0 + \frac{1}{2} \rho v_0^2 \quad (\text{B-1})$$

$$p_1 - p_0 = \frac{1}{2} \rho (v_0^2 - v_1^2) \quad (\text{B-2})$$

and the change in pressure is:

$$\Delta p = p_1 - p_0 \quad (\text{B-3})$$

and it can be rewritten as:

$$p_1 = \Delta p + p_0 \quad (\text{B-4})$$

At a point S near the nose of the airfoil section, the flow is splitting up and it loses all velocity in the direction of motion along the streamline; this point S is called a stagnation point, and the change of the pressure will be obtained from the Equation (B-3) as follow:

$$\Delta p = p_1 - p_0 = \frac{1}{2} \rho (v_0^2 - v_1^2) = \frac{1}{2} \rho v_0^2 \quad (\text{B-5})$$

This value is called a dynamic or stagnation flow pressure. Since cavitation will occur when the local pressure  $p_1$  has fallen to, or below, the vapour pressure of water,  $p_v$ , the following criterion equation can be obtained according to Equation (B-4):

$$p_v = \Delta p + p_0 \quad (\text{B-6})$$

Cavitation will occur when the following equations are satisfied.

$$p_v \geq \Delta p + p_0 \quad (\text{B-7})$$

$$-\Delta p \geq p_0 - p_v \quad (\text{B-8})$$

Dividing  $\Delta p$  and  $p_0 - p_v$  by the dynamic pressure  $0.5\rho v_0^2$ , a pressure coefficient  $C_p$  and a cavitation number  $\sigma$  are produced as follows:

$$C_p = \frac{\Delta p}{0.5\rho v_0^2} = \frac{p_1 - p_0}{0.5\rho v_0^2} \quad (\text{B-9})$$

$$\sigma = \frac{P_0 - P_v}{0.5\rho v_0^2} \quad (\text{B-10})$$

According to Equations (B-8), (B-9) and (B-10), a simple criterion based on the cavitation number  $\sigma$  and pressure coefficient  $C_p$  can be obtained:

$$\sigma \leq -C_p : \text{cavitation occurs}$$

$$\sigma > -C_p : \text{cavitation does not occur}$$

A cavitation inception for a particular airfoil section can be described by a cavitation bucket diagram, as shown in Figure B.3. The diagram is plotted as a function of the cavitation number to angle of attack. The cavitation number  $\sigma$  can be obtained by substituting the negative minimum pressure coefficient  $-C_p$  since the cavitation will begin to occur when the cavitation number is equivalent to the negative minimum pressure coefficient ( $\sigma = -C_p$ ).

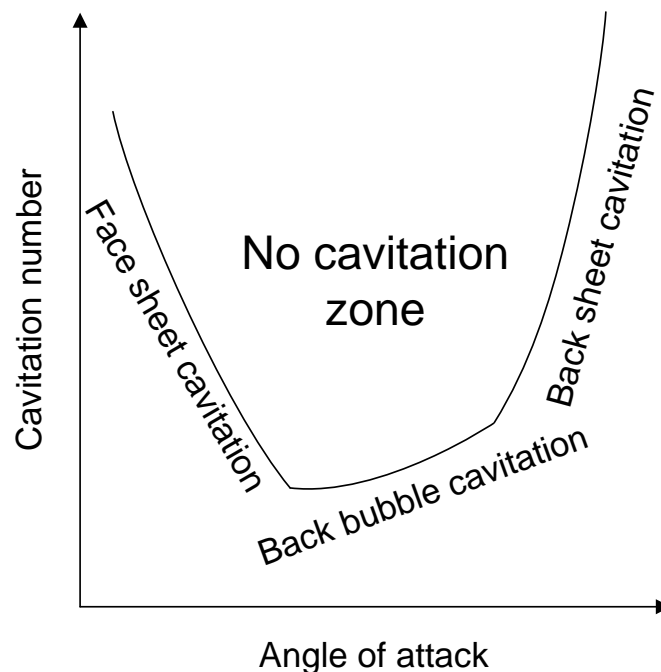


Figure B.3 Cavitation bucket diagram.

No cavitation will occur as long as the design operates in the bucket. The lower the range of cavitation number (the higher minimum pressure coefficient) at given angle of attack, the wider the bucket and cavitation free zone is, as shown in Figure B.3. When the airfoil sections are working at negative angles of attack, sheet cavitation may appear on the face of the blade. Conversely, if the sections are operating at positive angles of attack, the sheet cavitation may occur on the back of the blade. The sheet cavitations have often a very stable character although there are cases where instability may occur when working in a wake. When the sections are operating at a small angle of attack, bubble cavitation may occur on the back of the blade, and this cavitation appears as large individual bubbles, grows and contracts rapidly.