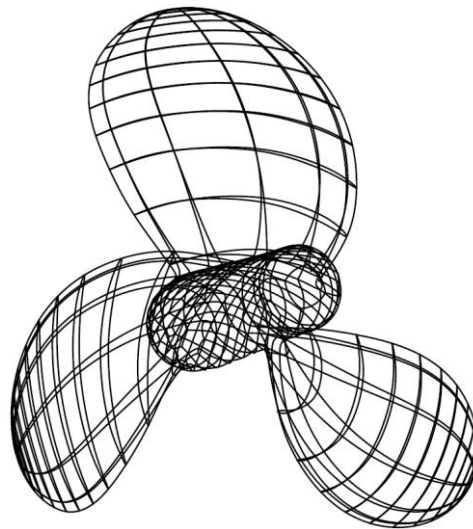




*National Technical University of Athens
School of
Naval Architecture & Marine Engineering
Department of Ship Design & Maritime Transport*

PARAMETRIC DESIGN OF A MARINE PROPELLER USING T-SPLINES



A Diploma Thesis by Arapakopoulos Andreas

Thesis Supervisor: Ass. Prof. A. Ginnis

Committee Member: Prof. G. Politis

Committee Member: Prof. G. Zarafonitis

Athens, February 2019

Ευχαριστίες

Πρώτα απ' όλα, θα ήθελα να εκφράσω τις βαθιές μου ευχαριστίες σε ολόκληρη την οικογένειά μου και στους στενούς μου φίλους. Πολλοί συνάδελφοι, φίλοι και μέλη της οικογένειας με βοήθησαν με την επιμέλεια του κειμένου και με μεταφράσεις. Η στήριξή τους και η θετική τους ενέργεια θα παραμείνουν χαραγμένες στη μνήμη μου και ήταν ένα ανεκτίμητο δώρο για μένα. Επομένως, το ελάχιστο που μπορώ να κάνω είναι να μοιραστώ μαζί τους τη μεγάλη μου χαρά, που ολοκληρώνω αυτή την προσπάθεια.

Μια ακόμη πηγή ενθάρρυνσης, ήταν ο υπεύθυνος της διπλωματικής μου εργασίας, ο Αλέξανδρος Γκίνης. Εκφράζω τη βαθιά μου εκτίμηση προς αυτόν και θα ήθελα πραγματικά να τον ευχαριστήσω για την συνολική του αρωγή και καθοδήγηση. Το δημιουργικό του πνεύμα, η ειλικρίνεια του και η ανταλλαγή ιδεών μας ξεπέρασαν τα όρια μιας τυπικής σχέσης μεταξύ ενός καθηγητή και του φοιτητή του.

Ευχαριστώ επίσης, τον Κωνσταντίνο Κώστα και τους φοιτητές του στο Πανεπιστήμιο Nazarbayev για τη βοήθειά τους σε μια πολύ κρίσιμη φάση αυτής της έρευνας. Η συνεργασία μου μαζί τους χαρακτηρίστηκε από επαγγελματισμό και συνέπεια παρά την χιλιομετρική απόσταση και τη διαφορά στην τοπική ζώνη ώρας. Επιπλέον, πολύ σημαντική ήταν η κριτική του Γεράσιμου Πολίτη πάνω την δουλειά μου, ο οποίος είναι ειδικός στην σχεδίαση της έλικας. Επιπρόσθετα, θα ήθελα να αναφερθώ στον Παναγιώτη Κακλή, του οποίου ο ρόλος στην επιλογή του θέματος αυτής της διπλωματικής ήταν καθοριστικός.

Τελευταίο αλλά εξίσου σημαντικό είναι, ότι αναγνωρίζω με ευγνωμοσύνη το υπόβαθρο, τους πόρους και τις θαυμάσιες ευκαιρίες που μου παρείχε το Εθνικό Μετσόβιο Πολυτεχνείο (ΕΜΠ).

Acknowledgements

First and foremost, I would like to express my deep thanks to my entire family and to my close friends. Many co-students, friends and family members have helped me with proof-reading and translations. Their support and their positive energy will remain engraved in my memory and were an invaluable gift for me. So, in terms of the minimum I can do, I share with them the great joy of finalizing this effort.

Another source of encouragement, was my thesis supervisor, Alexandros Ginnis. My deep appreciation is directed towards to him and I would really like to thank him for his universal help and guidance. His creative spirit, openness and our exchange of ideas have exceeded the limits of a formal relationship between a professor and his student.

Many thanks also Konstantinos Kostas and his students in Nazarbayev University, for their help in a very crucial phase of the research. My collaboration with them was characterized by professionalism and consistency despite the kilometric distance and the difference in the local time zone. Furthermore, very important was the criticism of Gerasimos Politis on my work, who is an expertise in marine propeller design. In addition, I would like to mention Panagiotis Kaklis, whose role in the choice of the topic of this diploma thesis was determinant.

Last but not least, I gratefully acknowledge the framework, funds and wonderful opportunities provided to me by the National Technical University of Athens (NTUA).

Σύνοψη

Αυτή η διπλωματική εργασία επικεντρώνεται στην ανάπτυξη ενός παραμετρικού μοντέλου, που χρησιμοποιείται για την παραγωγή γεωμετρικών μοντέλων ναυτικών έλικων, που αναπαριστώνται από μία ενιαία υδατοστεγή T-spline επιφάνεια. Η διαδικασία σχεδιασμού του πλαισίου της γεωμετρίας και οι αλγόριθμοι που παρουσιάζονται προγραμματίστηκαν κάνοντας χρήση της γλώσσας C # μέσα στο Grasshopper, το οποίο αποτελεί ένα εργαλείο παραμετρικής μοντελοποίησης και οπτικού προγραμματισμού του Rhinoceros 3D.

Μία από τις κύριες λειτουργίες αυτού του παραμετρικού μοντέλου έγκειται στην ικανότητά του να παράγει αυτόματα και γρήγορα έγκυρες γεωμετρικές αναπαραστάσεις προπελών (πτερύγια και πλήμνη) με βάση ένα μικρό σύνολο σημαντικών παραμέτρων, που έχουν γεωμετρική και φυσική υπόσταση. Αυτή η λειτουργικότητα αποτελεί βασική προϋπόθεση όταν αντιμετωπίζουμε το πρόβλημα της βελτιστοποίησης του σχεδιασμού της επιφάνειας διαφόρων σχημάτων, όπως τα πτερύγια ναυτικών έλικων και ανεμογεννητριών, που έχουν πολύπλοκη γεωμετρία και η απόδοσή τους επηρεάζεται αισθητά από τη γεωμετρία αυτή.

Επιπροσθέτως, χρησιμοποιώντας T-splines, ο μοντελοποιητής δημιουργεί ομαλά και κατάλληλα για ανάλυση μοντέλα, τα οποία είναι σε θέση να εξαλείψουν την χρονοβόρα, απαιτητική και δαπανηρή διαδικασία της μετατροπής ενός CAD μοντέλου σε ένα κατάλληλο Computer Aided Engineering (CAE). Συνήθως η διαδικασία αυτή αποτελείται από μία προσεγγιστική παραγωγή πλέγματος του προαναφερθέντος μοντέλου. Αυτός ο στόχος μπορεί να επιτευχθεί μέσω της ισογεωμετρικής ανάλυσης (IGA), η οποία παρέχει μια άμεση και στενή σχέση μεταξύ CAD και CAE.

Η τελική απαίτηση, που τίθεται σε αυτή τη διπλωματική εργασία, είναι η ικανότητα του εξής παραμετρικού μοντέλου να αναπαράγει και να αναπαριστά με ακρίβεια υφιστάμενα μοντέλα ελίκων. Για αυτό το λόγο, μια προσέγγιση της Wageningen Σειράς B έχει επιτευχθεί μέσω αυτού του παραμετρικού μοντέλου. Επιπρόσθετα στην παραπάνω προσέγγιση, ένα τρισδιάστατο μοντέλο έλικας σχεδιασμένο στον OpenProp συγκρίνεται λεπτομερώς σε σχέση με τις απαιτήσεις του T-spline παραμετρικού μοντέλου, προκειμένου να επιτευχθούν στόχοι συγκριτικής αξιολόγησης.

Λέξεις Κλειδιά: Παραμετρική Μοντελοποίηση, Αναπαράσταση Ναυτικών Ελίκων, T-splines, B-series, OpenProp, Rhino 3D, Grasshopper

Abstract

This diploma thesis is focused on the development of a parametric model used in the generation of marine propeller geometrical model instances represented by one single watertight T-spline surface. The designing process of the wireframe geometry and the presented algorithms have been implemented in C# inside Grasshopper, a parametric modeling and visual programming plug-in tool of Rhinoceros 3D.

One of the main functions of this parametric modeler lies in its ability to automatically and quickly produce valid geometric representations of marine propellers (blades and hub) based on a small set of geometrically and physically meaningful parameters. This functionality is a major prerequisite, when dealing with the problem of design/shape optimization of functional surfaces, such as marine propeller & wind turbine blades, that possess complex geometry and geometrically-sensitive performance.

Further to this, by using T-splines, the modeler generates smooth, analysis-suitable instances, which will eliminate the time-consuming, labor-intensive and costly overhead of transforming a Computer Aided Design (CAD) model into an appropriate Computer Aided Engineering (CAE) model, commonly through an approximate mesh-model generation. This aim can be achieved by appealing to IsoGeometric Analysis (IGA), which provides a direct and tight link between CAD and CAE.

The final requirement posed on this diploma thesis, is the capacity of the provided parametric model in accurately reconstructing and representing existing propeller models. Therefore, an approximation of Wageningen B-series has been accomplished through the aforementioned parametric model. Additionally to the above approximation, a prototype marine propeller 3D model is obtained from OpenProp and is compared in detail with respect to the T-spline parametric model requirements, in order to achieve benchmarking purposes.

Keywords: Parametric Modeling, Marine Propeller Representation, T-splines, B-series, OpenProp, Rhino 3D, Grasshopper

Table of Contents

Ευχαριστίες.....	2
Acknowledgements	3
Σύνοψη	4
Abstract	5
Table of Contents	6
List of Figures.....	8
List of Tables.....	10
Nomenclature.....	11
I Introduction.....	12
A. The Significance of Marine Propellers.....	12
B. CAD History.....	13
C. Parametric Models	16
D. Software	17
E. Motivational Background and Objectives	19
II Geometric Tools for the Model.....	22
A. B-splines and NURBS	22
A.i. Knot vectors and the B-spline basis	22
A.ii. B-Spline Curves.....	24
A.iii. NURBS curves	25
A.iv. NURBS surfaces and solids	26
A.v. The index, parameter, and physical spaces	27
B. T-Splines	29
C. NURBS vs T-Splines.....	32
III Conceptual Marine Propeller Design	37
A. Marine Propeller Geometry	37
A.i. Blade Description	37
A.ii. Pitch.....	40
A.iii. Skew & Rake	41
A.iv. Propeller Outlines and Area	43
A.v. Section Geometry and Definition.....	45

B. Hydrofoil Mother Concept	46
IV Construction Steps	50
A. Working Interface.....	50
B. Step 1. Parametric Model of a Generic 2-D Hydrofoil.....	51
C. Step 2. Basic Parameters	53
D. Step 3. Radial Parameters	55
E. Step4. Blade Cage	58
F. Step 5. Hub Cage	62
G. Step 6. Control Cage of the Marine Propeller	64
V Model Instances	68
A. Wageningen B-series	68
B. OpenProp.....	72
C. Special Designs of Marine Propellers	76
VI Conclusions and Future Work	78
Appendix A: Propeller Model Utilized	79
Appendix B: Wageningen Screw B-series	82
References	85

List of Figures

FIGURE I.1: THE HISTORY OF CAD SINCE 1957	15
FIGURE I.2: COMPLETE BREAKDOWN OF ALL RHINO'S DEVELOPER TOOLS	18
FIGURE I.3: ESTIMATION OF THE RELATIVE TIME COSTS OF EACH TASK AT SANDIA NATIONAL LABORATORIES	21
FIGURE I.4: INCREASING COMPLEXITY OF ENGINEERING DESIGNS	21
FIGURE II.1: QUADRATIC BASIS FUNCTIONS	24
FIGURE II.2: QUADRATIC B-SPLINE CURVE WITH CONTROL POINTS AND CONTROL POLYGON.....	25
FIGURE II.3: THE PHYSICAL MESH AND THE CONTROL MESH OF A BIQUADRATIC B-SPLINE SURFACE	27
FIGURE II.4: A SUBSET OF THE PARAMETER SPACE.....	28
FIGURE II.5: THE INDEX SPACE	28
FIGURE II.6: PRE-IMAGE OF A T-MESH.....	29
P II.7: POSSIBLE AMBIGUITY	30
FIGURE II.8: KNOT LINES FOR BLENDING FUNCTION $Bi(s, t)$	31
FIGURE II.9: A NURBS SURFACE WITH A SET OF CONTROL POINTS	32
FIGURE II.10: QUADRIC PRIMITIVES	32
FIGURE II.11: MODEL OF A HAND COMPRISED OF MULTIPLE NURBS PATCHES	33
FIGURE II.12: HEAD MODELED (A) AS A NURBS AND (B) AS A T-SPLINE	34
FIGURE II.13: A GAP BETWEEN TWO NURBS SURFACES, FIXED WITH A T-SPLINE.....	34
FIGURE II.14: WATERCRAFT MODELED (A) AS A NURBS AND (B) AS A T-SPLINE	35
FIGURE III.1: BLADE REFERENCE LINES	37
FIGURE III.2: MARINE PROPELLER PARTS.....	38
FIGURE III.3: CYLINDRICAL BLADE SECTION DEFINITION	39
FIGURE III.4: GEOMETRICAL REPRESENTATION OF PITCH	40
FIGURE III.5: SKEW IN ANGULAR FORM DEFINITION.....	41
FIGURE III.6: TIP RAKE DEFINITION.....	42
FIGURE III.7: DEFINITION OF TOTAL RAKE	42
FIGURE III.8: OUTLINE DEFINITION	44
FIGURE III.9: AIRFOIL TERMINOLOGY	45
FIGURE III.10: HYDROFOIL SECTIONS POSITIONED AND SIZED	46
FIGURE III.11: HYDROFOIL SECTIONS ROTATED VIA PITCH ANGLE.....	48
FIGURE III.12: HYDROFOIL SECTIONS WARPED AND DISPLACED VIA RAKE AND SKEW	49
FIGURE III.13: SCHEMATIC REPRESENTATION OF A CONTROL POINT PROJECTED ON A CYLINDER.....	49
FIGURE IV.1: GRASSHOPPER INTERFACE OF THE PARAMETRIC MODEL	50
FIGURE IV.2: BUILDING THE PARAMETRIC MODEL FOR A GENERIC 2-D HYDROFOIL INSIDE GRASSHOPPER	51
FIGURE IV.3: 2-D HYDROFOIL MODEL	52
FIGURE IV.4: BASIC PARAMETERS WITH NUMBER SLIDERS INSIDE GRASSHOPPER	54
FIGURE IV.5: GENERATING DISTRIBUTIONS OF RADIAL PARAMETERS WITH GRAPH MAPPERS	56
FIGURE IV.6: THREE OPTIONS TO CREATE THE DISTRIBUTIONS OF THE RADIAL PARAMETERS.....	57
FIGURE IV.7: DECISION MAKING.....	57
FIGURE IV.8: CONTROL POINTS MAPPED INTO THE 3-DIMENSIONAL SPACE	58
FIGURE IV.9: BOUNDING HYDROFOIL MODEL.....	59
FIGURE IV.10: TWO NEARBY SECTIONS FOR THE GENERATION OF THE FILLET AREA	59
FIGURE IV.11: ADJUSTMENTS AT THE CONTROL POINTS OF THE FIRST SECTION	60
FIGURE IV.12: LAST SECTION OF THE BLADE CAGE	61
FIGURE IV.13: THREE BLADE CAGES OF A MARINE PROPELLER.....	61

FIGURE IV.14: CONSTRUCTION OF THE HUB CAGE USING A SET OF POLYGONS IN A ROW	63
FIGURE IV.15: PROCESS OF DESIGNING A MARINE PROPELLER WITH T-SPLINES INSIDE GRASSHOPPER	64
FIGURE IV.16: STAR POINTS ON THE T-MESH OF A MARINE PROPELLER	65
FIGURE IV.17: ENTIRE CONTROL CAGE OF MARINE PROPELLER.....	66
FIGURE IV.18: MARINE PROPELLER CURVATURE ANALYSIS USING GAUSSIAN SURFACE	67
FIGURE IV.19: EVALUATION OF MARINE PROPELLER SURFACE SMOOTHNESS AND CONTINUITY USING ZEBRA	67
FIGURE V.1: GENERATING THE CLOUD OF THE B-SERIES PROPELLER POINTS INSIDE GRASSHOPPER.....	68
FIGURE V.2: A CLOUD OF A B-SERIES PROPELLER POINTS.....	69
FIGURE V.3: COMPARISON OF B-SERIES SECTIONS WITH A SET OF APPROXIMATED SECTIONS	70
FIGURE V.4: GENERATING A B-SERIES HYDROFOIL USING GALAPAGOS GENERIC PLATFORM	70
FIGURE V.5: FOUR GEOMETRIC MODEL INSTANCES OF B-SERIES PROPELLER	71
FIGURE V.6: DEFAULT GEOMETRY-INPUT PARAMETER VALUES FOR OPENPROP	72
FIGURE V.7: DEFAULT OPENPROP 3D PROPELLER MODEL.....	72
FIGURE V.8: DEFAULT OPENPROP RADIAL DISTRIBUTIONS OF BASIC GEOMETRICAL QUANTITIES	73
FIGURE V.9: POINT-SURFACE DEVIATIONS BETWEEN OPENPROP GENERATED POINT SET AND THE CORRESPONDING T-SPLINE BLADE SURFACE	74
FIGURE V.10: FOUR GENERIC PROPELLER MODEL INSTANCES.....	77
FIGURE A.1: B3-60 PROPELLER INSIDE RHINO.....	79
FIGURE A.2: HYDROFOIL SECTION MOTHER FOR B3-60 PROPELLER	80
FIGURE A.3: GRAPHS OF RADIAL PARAMETERS FOR B3-60 PROPELLER	81
FIGURE B.1: DEFINITION OF GEOMETRIC BLADE SECTION PARAMETERS OF WAGENINGEN B-SERIES PROPELLERS	83

List of Tables

TABLE II.1: A COMPARISON OF C^0 FINITE ELEMENTS, NURBS, AND T-SPLINES	36
TABLE III.1: PART OF C# CODE FOR THE 1 ST TRANSFORMATION	47
TABLE III.2: PART OF C# CODE FOR THE 2 ND TRANSFORMATION.....	47
TABLE IV.1: PARAMETERS FOR THE CONSTRUCTION OF A GENERIC 2-D HYDROFOIL.....	51
TABLE IV.2: BASIC PARAMETERS FOR THE CONSTRUCTION OF A GENERIC MARINE PROPELLER	53
TABLE IV.3: THE CATEGORIZED RADIAL PARAMETERS	55
TABLE V.1: SUMMARY OF THE WAGENINGEN B-SCREW SERIES.....	71
TABLE V.2: COMPARISON OF OPENPROP SECTIONS WITH T-SPLINE PARAMETRIC MODEL.....	75
TABLE V.3: PROPERTIES OF THE FOUR GENERIC PROPELLER MODELS OF FIGURE V.10	76
TABLE A.1: BASIC PARAMETERS FOR B-60 PROPELLER	79
TABLE A.2: HYDROFOIL MOTHER PARAMETERS WITH THEIR VALUES FOR B3-60 PROPELLER	80
TABLE B.1: DIMENSIONS OF WAGENINGEN B-PROPELLER SERIES	82
TABLE B.2: VALUES OF V_1 AND V_2 FOR USE IN EQUATIONS (B.1) AND (B.2)	84

Nomenclature

Symbol	SI unit	Definition
<i>CAD</i>		<i>Computer Aided Design</i>
<i>CAM</i>		<i>Computer Aided Manufacturing</i>
<i>CAE</i>		<i>Computer Aided Engineering</i>
<i>ADAM</i>		<i>Automated Drafting And Machining</i>
<i>NURBS</i>		<i>Non-Uniform Rational B-Spline</i>
<i>FEM</i>		<i>Finite Element Method</i>
<i>FEA</i>		<i>Finite Element Analysis</i>
<i>DTA</i>		<i>Design through Analysis</i>
<i>IGA</i>		<i>Isogeometric Analysis</i>
<i>IGES</i>		<i>Initial Graphics Exchange Specification</i>
<i>ITTC</i>		<i>International Towing Tank Conference</i>
<i>D</i>	<i>[m]</i>	<i>Diameter of the Propeller</i>
<i>Z</i>		<i>Number of Blades</i>
<i>rh</i>		<i>Hub Diameter divided by Propeller Diameter</i>
<i>R_i</i>	<i>[m]</i>	<i>Radius of <i>i</i>th Section</i>
<i>p_i</i>	<i>[m]</i>	<i>Pitch of <i>i</i>th Section</i>
<i>φ_i</i>	<i>[rad]</i>	<i>Pitch Angle of <i>i</i>th Section</i>
<i>Sk_i</i>	<i>[m]</i>	<i>Skew of <i>i</i>th Section</i>
<i>∂s_i</i>	<i>[rad]</i>	<i>Skew Angle of <i>i</i>th Section</i>
<i>ip</i>	<i>[m]</i>	<i>Rake</i>
<i>∂ip</i>	<i>[rad]</i>	<i>Rake angle</i>
<i>A_P</i>	<i>[m²]</i>	<i>Projected Area of the Blade</i>
<i>A_E</i>	<i>[m²]</i>	<i>Expanded Area of the Blade</i>
<i>A_D</i>	<i>[m²]</i>	<i>Developed Area of the Blade</i>
<i>A₀</i>	<i>[m²]</i>	<i>Propeller Disc Area</i>

I Introduction

A. The Significance of Marine Propellers

Marine propellers lie submerged in water aft of the ship, they form an integral part of a ship and have a significant impact on ship performance [1]. A ship propeller is a mechanical device that transmits power by converting rotational motion into thrust. It consumes the power produced by the engine and puts this power into thrust power [2]. The power transfer is enhanced by the particular geometry of the propeller. The propeller consists of a revolving shaft with several blades attached to it. The propeller shaft connects the engine with the marine propeller. As part of the power transmission, the propeller shaft rotates due to a torque produced by the engine. The geometry [3] of the propeller blades is designed in such a way that, when a propeller rotates, a force is produced, which pushes the ship through the water. This force is known as thrust and it comes from a pressure difference between the forward and rear surface of the airfoil-shaped blade, and the water accelerated behind the blade. Most marine propellers are screw propellers with fixed helical blades rotating around a horizontal axis or propeller shaft. In consequence, the hydrodynamic performance of a marine propeller is tightly linked to the geometric characteristics and the shape of its blades.

Designers and manufacturers of propellers define the efficiency of those as the ratio of the power produced by the propeller (thrust) to the power consumed by the engine. Propeller efficiency indicates how much energy is lost in the power transmission, which is described above. Naturally, efficient propellers are desirable due to the relationship between propeller efficiency and fuel consumption, where the fuel consumption is the real indicator of ship performance. As stated at [4], fuel costs represent as much as 50–60% of total ship operating cost, depending on the type of ship and service. Taking into account the continuous increase in bunker price, improving the propeller efficiency seems to be an unavoidable task for the global shipping industry.

Therefore, propeller design, it is one of the most basic scientific research topics in Naval Engineering. Since the first use of marine propellers in 1850 much has been said and published on the development of those, but there is much more to be done. Unquestionably in today's times, the demand for higher efficiency is more intense than ever and therefore propulsor designers are forced to adopt more complex geometries. Fortunately, the propeller design process has been in explosive growth during the last two decades due to the rapid evolution in computational power, which can lead to higher accuracy models with better performance.

B. CAD History

The use of computer systems to aid in the creation, modification, analysis, or optimization of a design is called Computer-aided design (CAD) [5]. A CAD system is a combination of hardware and software, that is used for making 2-D and 3-D models of almost anything from jewelry to airplanes. The purpose of developing this kind of systems is to increase the productivity of the designer, improve the quality of design, improve communications through documentation, and to create a database for manufacturing. CAD software is often referred to as CAD CAM software ('CAM' is the acronym for Computer Aided Manufacturing). The usage of CAM software is to control machine tools and related ones in the manufacturing of workpieces [6]. Its primary purpose is to create a faster production process and tools with more precise dimensions and material consistency. CAD-CAM systems exist today for all of the major computer platforms, including Windows, Linux, Unix and Mac OS X and they are widely accepted and used throughout the industry. These systems moved from costly workstations based mainly on UNIX to off-the-shelf PCs. The increasing power of computers, and especially the introduction of lower cost minicomputers with optimized Fortran compilers and graphics capable terminals, made CAD software more accessible to engineers. The Computer Software used to analyze CAD geometry tools that have been developed to support these activities are considered CAE tools [7]. CAE tools are being used, for instance, to analyze the robustness and performance of components and assemblies.

The beginnings of CAD [8] [9] can be traced to the year 1957, when the first commercial numerical-control programming system named PRONTO developed by Dr. Patrick J. Hanratty. As a result of the very high cost of early computers, large aerospace and automotive companies were the earliest commercial users of CAD software. In 1960 Sketchpad, was developed by Ivan Sutherland and it was the first to ever use a total graphic user interface, users wrote with a light pen on an x-y pointer display.

2D drafting applications were typically the first-generation CAD software systems. Much of the early pioneering research in 2D CAD software was performed at MIT's Mathematical Laboratory. In 1965 at Cambridge University's Computing Laboratory began serious research into 3D modeling CAD software. In the meantime, French researchers were doing also pioneering work into complex 3D curve and surface geometry computation. Paul de Casteljaou made fundamental strides in computing complex 3D curve geometry and Bezier published his breakthrough research, incorporating some of de Casteljaou's algorithms, in the late 1960s.

In the mathematical field of numerical analysis, De Casteljaou's algorithm [10] is a recursive method to evaluate polynomials in Bernstein form or Bézier curves [11]. As it is known, recursive functions can be easily implemented particularly when it comes to a programming. So that is one of the main reasons that B-Spline curves [12] is purely

evolved out of the availability of computers. The work of both Bezier and de Casteljau continues to be one of the foundations of 3D CAD software to the present time.

3D wireframe features were developed in the beginning of the sixties, and in 1972 MAGI (Mathematics Application Group, Inc.) released the first commercially available 3D solid modeler Syntha Vision. In 1971, one of the most influential events in the development of CAD was the founding of MCS (Manufacturing and Consulting Services Inc.) by Patrick J. Hanratty, who wrote the system ADAM (Automated Drafting And Machining). This interactive graphic design, drafting and manufacturing system was written in Fortran and designed to work on virtually every machine, a huge hit that went on to be updated to work on 16 and 32-bit computers. In today's times, 80% of CAD programs can be traced back to the roots of ADAM.

Solid modeling further enhanced the 3D capabilities of CAD systems. Throughout the 1980s, the CAD software market was inevitably shifting the CAD software market to 3D and solid modeling due to the new generation of powerful UNIX workstations and emerging 3D rendering. CAD software vendors had begun as a collection of fast-growing companies benefiting from rapid advances in computer hardware and a potential market that was expanding as falling computer prices and maintenance costs made CAD software available to more users. NURBS, mathematical representation of freeform surfaces, appeared firstly on Silicon Graphics workstations in 1989. Also, in 1989 T-FLEX and later Pro/ENGINEER introduced CADs based on parametric engines.

Parametric modeling is a method based on algorithmic thinking and means that the model is defined by parameters. It enables the expression of parameters to determine the relationships between the design elements in order to define a set of multiple solutions. A change of dimension values in one place also changes other dimensions to preserve relation of all elements in the resulting design. In this regard, parametric design can significantly improve the field of landscape architecture by providing new tools of design investigation. So, the architects can set up more effectively the entire design process.

THE HISTORY OF CAD

FATHER OF CAD

PRONTO
By: Dr. Patrick Hanratty



PRONTO was the first commercial numerical control programming system, spawned everything that is CAD. Known as the building blocks of everything CAD.

Dr. Patrick Hanratty, an American aerospace engineer, passed on the 'father of CAD and CAM'.

CADD
By: McDermott-Chapman

Based for parts layouts and geometry work, continued to be improved upon and customized for specific uses.

Digigraphics
By: Itek

First commercial CAD system, \$500,000.00 per system, only sold 6 copies.



ADAM
By: Patrick Hanratty



Unigraphics
By: Siemens NX

High end easy to use software used by many corporations that set a new gold standard for CAD software at this time.

Autodesk AutoCAD
First CAD software made by PCs instead of the mainframe computers.

CADENAS Founded
Founded originally as an engineering firm but realized the potential of the engineering IT age.



STEP
Took over from IGES as the new format to use when transferring 3D models from one to another. 1994 was the initial release of STEP and made it an international standard for models, still the most used format.

SolidWorks 95
By: Dassault Systemes

Another software that succeeded in ease of use, allowed more engineers than ever to take advantage of 3D CAD technology.

CATIA Conferencing Groupware
By: Dassault Systemes

The first to move online, allowing users to review and annotate CATIA models with others over the internet, quickly followed by others. Unigraphics' (MAN web author and Co-Creator's) OpenSpace 98.

Autodesk 360
Moved to the cloud, others followed.



The Future
Focus on Virtual Reality



1957



1960
Sketchpad
By: Ivan Sutherland

First to ever use a total graphic user interface, users write with a light pen on an xy grid display, set users constrain properties as a drawing, created the use of "objects" and "instances".

1966
PDGS
By: Ford

Ford and other manufacturers started releasing internally developed CAD/CAM systems.



1970
SynthaVision
By: MAGI

First commercially available 3D solid modeling program.

1971
ComputerVision
By: Dr. Kenneth Vespinic

Rational B-spline geometry added to CAD.

1977
CADAM
Used by Lockheed, introduced CAD to aerospace design.



1980
MiniCAD
Best selling CAD software for Mac computers.

1981
GEOMOD
Featuring NURBS

1982
Pro/Engineer (PTC Creo)
First mainstream CAD program that took the ideas of Sketchpad and made it come to life, based on solid models, history based features, and the use of constraints, this was a huge turn in CAD history.



1984
Autodesk AutoCAD
Release 13

Made the Autodesk program 3D.

1995
eCATALOG
By: CADENAS

CADENAS unites the native 3D CAD model market with eCATALOG solutions, digital product catalogs that featured multiple Native CAD formats for the first time.



1995
Solid Edge
By: Siemens

Made as a PLM software, functions on Windows, provides solid modeling, assembly modeling, and 2D orthographic view, response to the success of SolidWorks.

1999
Autodesk Inventor
Autodesk's new direction, tried to be more intuitive and simple, also allowed the creation of complex assemblies in record time, still in use, really opened the game in the CAD world.



2012
3D CAD App
First APP for 3D CAD manufacturer models by CADENAS

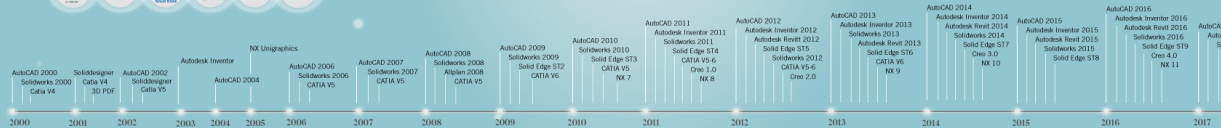


2015
Onshape
Completely cloud based CAD program

2017
PARTsolutions
Helping manufacturers "future proof" their catalog by keeping current with future feature formats, versions and releases.



POPULAR CAD FORMATS



Popular CAD Formats and Versions Added to the CADENAS eCATALOGsolutions platform

Figure I.1: The History of CAD since 1957

C. Parametric Models

The need for parametric models that will facilitate the automatic generation of valid propeller model instances and enable the design optimization of blades has been identified by several researchers, including Pritchard (1985) [13], Gräsel et al (2004) [14] and Koini et al. [15] for turbomachinery blades, Epps et al. (2009) [16] Calcani et al. (2010) [17] Mirjalili et al. (2015) [18] and Herath et al. (2015) [19] for marine propellers. However, the number of parametric models specifically designed and used for design optimization of marine propellers are rather limited and are commonly build around the computational package used for the assessment of the propeller model and/or the specific characteristic the researchers are trying to optimize. Furthermore, the parameters employed in the parametric models found in literature, e.g., [20] and [21], are mainly appealing to coordinates of control points and there are limited cases where all, or most, parameters have a physical meaning.

At the same time, various commercial, academic and open-source software packages exist for the design and/or design optimization propellers. The commercial tools include PSP and QPROP from MARIN (Maritime Research Institute Netherlands) [22], PropCAD and PropElements from Hydrocomp Inc. [23] and many others. The list of academic/open-source software tools includes XROTOR [24], JavaProp [25], JBlade [26] [27], OpenProp [28] [29] [30] and PROP DESIGN [31].

XROTOR is an interactive program for the design and analysis of ducted and free-tip propellers and windmills, JavaProp is a relatively simple piece of code based on the blade element theory, JBLADE is an open-source propeller design and analysis code based on David Marten's QBLADE and André Deperrois' XFLR5 software tools, OpenProp is a free software for the design and analysis of marine propellers and horizontal-axis turbines based on Matlab [32] and PROP DESIGN is an open source, public domain, aircraft propeller design software allowing the design of ducted or unducted aircraft propellers with straight or swept blades and constant or elliptical chord distributions. In all of above cases, propeller models are tightly linked to the computational procedure employed (or assumed by the corresponding software tool) and in no case these propeller models are the type of analysis-suitable models required in the IGA context (see: "*Motivational Background and Objectives*"), i.e., smooth & accurate geometric representations of propeller blades that can be directly used in analysis.

D. Software

In this thesis, the software utilized is Rhinoceros. Rhinoceros [33] is a commercial CAD application software and 3D computer graphics by Robert McNeel & Associates. The geometry of Rhinoceros is based on the NURBS mathematical model, which focuses on producing mathematically precise representation of curves and freeform surfaces in computer graphics. Rhinoceros is mainly used in processes of computer-aided design (CAD), computer-aided manufacturing (CAM), rapid prototyping, 3D printing and reverse engineering in industries including architecture, industrial design (e.g. automotive design, watercraft design), product design (e.g. jewelry design) as well as for multimedia and graphic design [34].

The capabilities of Rhinoceros can be complemented and expanded in specific fields like rendering and animation, architecture, marine, jewelry, engineering, prototyping, and others [35] by plug-ins, which are available from both McNeel and other software companies. One of the most well known McNeel plug-ins is a parametric modeling and visual programming tool, called Grasshopper [36], that runs within the Rhinoceros 3D computer-aided design (CAD) application and takes advantage of Rhino's existing tools. This graphical algorithm editor has attracted many architects to Rhinoceros due to its ease of use and ability to create complex algorithmic structures. Grasshopper was developed by David Rutten at Robert McNeel & Associates and offers new ways to expand and control the 3D design and modeling processes, including automating repetitive processes.

It provides the users of Grasshopper the ability to use mathematical functions to control or generate shapes and enables them to quickly change fundamental attributes of a complicated model, such as a marine propeller, where a user might want even more flexibility. Some of this flexibility can be accessed also from within Rhino by using the built-in support for scripting languages like Rhinoscript or Python. These scripting tools (*Figure 1.2*) offer powerful control over Rhino's modeling commands, including some that are not available through the graphic interface of Rhino. Using scripting languages, though, requires a fairly in-depth knowledge of computer programming techniques, which, of course, many users don't have. Grasshopper combines the graphical approach of working in Rhino with the powerful algorithmic techniques found in scripting.

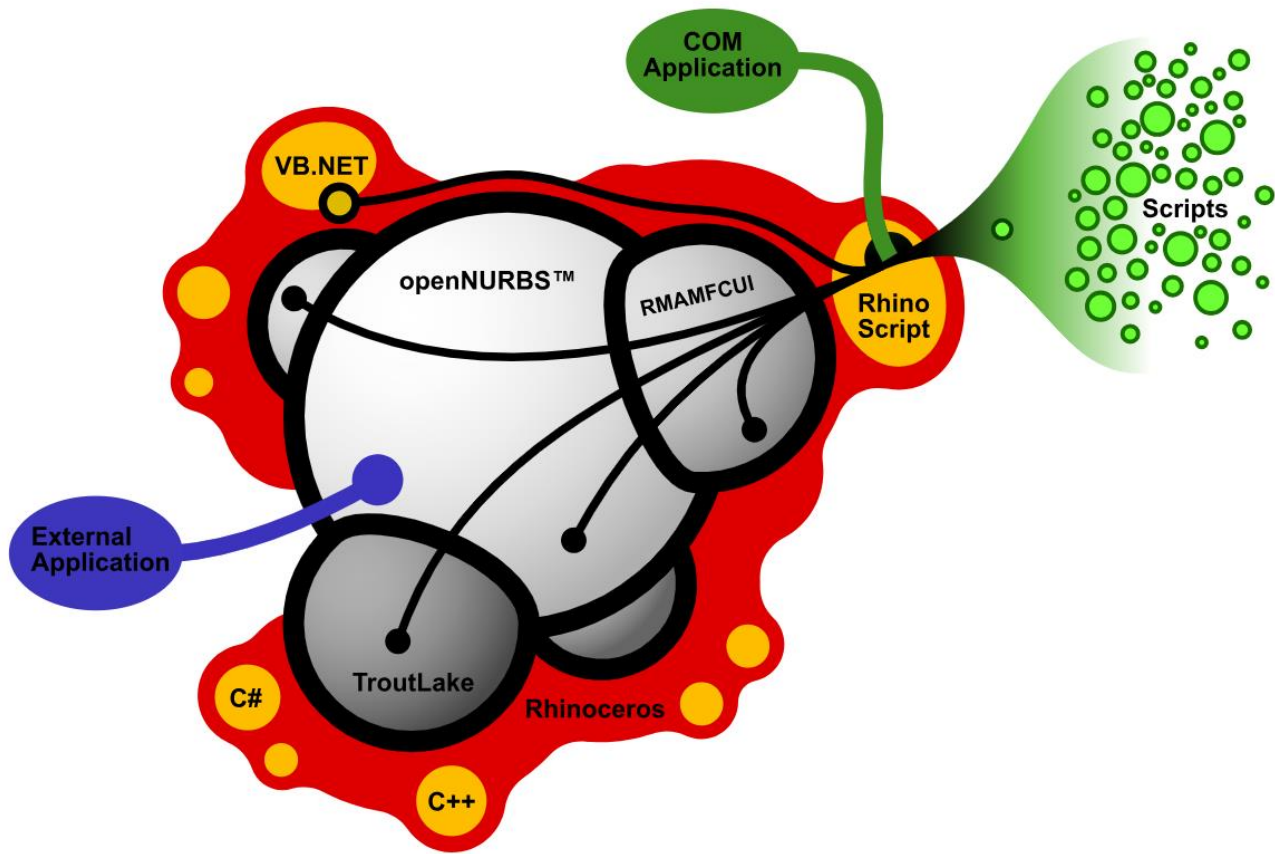


Figure 1.2: Complete Breakdown of all Rhino's Developer Tools

E. Motivational Background and Objectives

In Computer Aided Engineering fields an iterative process [37] to find an optimal design is often required. This process consists of a modeling phase followed by a numerical simulation and an analysis phase. The classical finite element method (FEM), in which the finite elements are conforming with the physical boundaries of the model, is a common choice for the second phase. The design-through-analysis (DTA) tools are fundamental to simulate complex physical phenomena and systems, such as a marine propeller. At Sandia National Laboratories, the anatomy of DTA the process has been studied by Ted Blacker and more specifically the percentage of time devoted to each task has been calculated and is summarized in *Figure 1.4*. This research has shown, that for complex engineering designs (*Figure 1.3* shows a typical automobile consists of about 3,000 parts, a fighter jet over 30,000, the Boeing 777 over 100,000, and a modern nuclear submarine over 1,000,000) the transition from the geometric model to the simulation model causes more than 80 % of the engineering effort. This approximation, 80/20 modeling/analysis ratio, seems to be a very common ratio in industrial area and there is a strong desire to reverse it. For the purpose of overcoming the difficulties involved in this transition process, various methodologies have been developed.

The most prominent method is Isogeometric Analysis (IGA) as proposed by Hughes at [38] and later described in detail in [39]. IGA aims at bridging the gap between the CAD model and computational analysis by utilizing the same shape functions in CAD and FEA. This functionality allows to eliminate the troublesome & time-consuming meshing process involved in the common engineering analysis paradigm and at the same time preserve the exact geometry (no approximation via meshing is required), while rendering possible the automatic data/model transfer between CAD & CAE packages. This approach has been already successfully applied in shape optimization of hydrofoils [40] [41] and ship hulls [42] [43] [44]. A DTA framework can be created using combined Isogeometric Analysis (IGA) and a smooth geometry of a model. The smoothness of the geometric model offers essential computational advantages over the de-facto finite element method (FEM), in which the finite elements are conforming with the physical boundaries of the model. By the term smooth geometry, we usually mean that interelement continuity may be C^n , $0 < n < p$, where p is the polynomial degree of the basis. Isogeometric analysis has emerged as an important alternative to traditional engineering design and analysis methodologies. There are numerous candidate computational geometry technologies that may be used in Isogeometric Analysis.

The most widely used in engineering design are NURBS and they are ubiquitous in CAD systems, representing billions of dollars in development investment. For instance, NURBS is the only free-form surface type, that is supported in the IGES [45] file format, the most popular format for data exchange between CAD software. Most of the early developments in IGA focused on establishing the behavior of the smooth bivariate and trivariate NURBS basis in analysis since NURBS geometry underpins almost all major design packages.

While the smoothness of NURBS make them useful in isogeometric analysis, NURBS are severely limited by the fact that they are four sided. In traditional NURBS-based design, modeling a complicated engineering design often requires hundreds, if not thousands, of NURBS patches, which are usually discontinuous across patch boundaries and it very is difficult to join multiple these patches into a single watertight model. Also, almost all NURBS models use trimming curves. So, designing a marine propeller in a CAD system based on NURBS, is usually not suitable as a basis for analysis. Therefore, development of technology meets the demands of both design and analysis.

T-splines [46] are a new mathematical formulation for surfaces and are a superior alternative to NURBS, the current geometry standard in computer-aided design systems. T-Splines models are mathematically watertight and are not limited to rectangular domains. Thus, the designers are able to represent entire models with a single surface via T-spline technology, which makes the models easier to analyze and optimize, and can minimize the likelihood of mistakes in downstream processes. Commercial bicubic T-spline surface modeling capabilities have been recently introduced in Maya [47] and Rhino [48] two NURBS-based design systems. This facilitates their utility as a design tool.

In this thesis, an IGA-based parametric design of a marine propeller using T-splines technology framework implemented using the Grasshopper algorithmic modeling interface for Rhinoceros 3D. Combining the increasing interest in designing the optimal propeller shape, an automated optimization can fill the design space through a parametric modeler with numerous designs that gravitate, guided by the optimization algorithm, towards an optimal design.

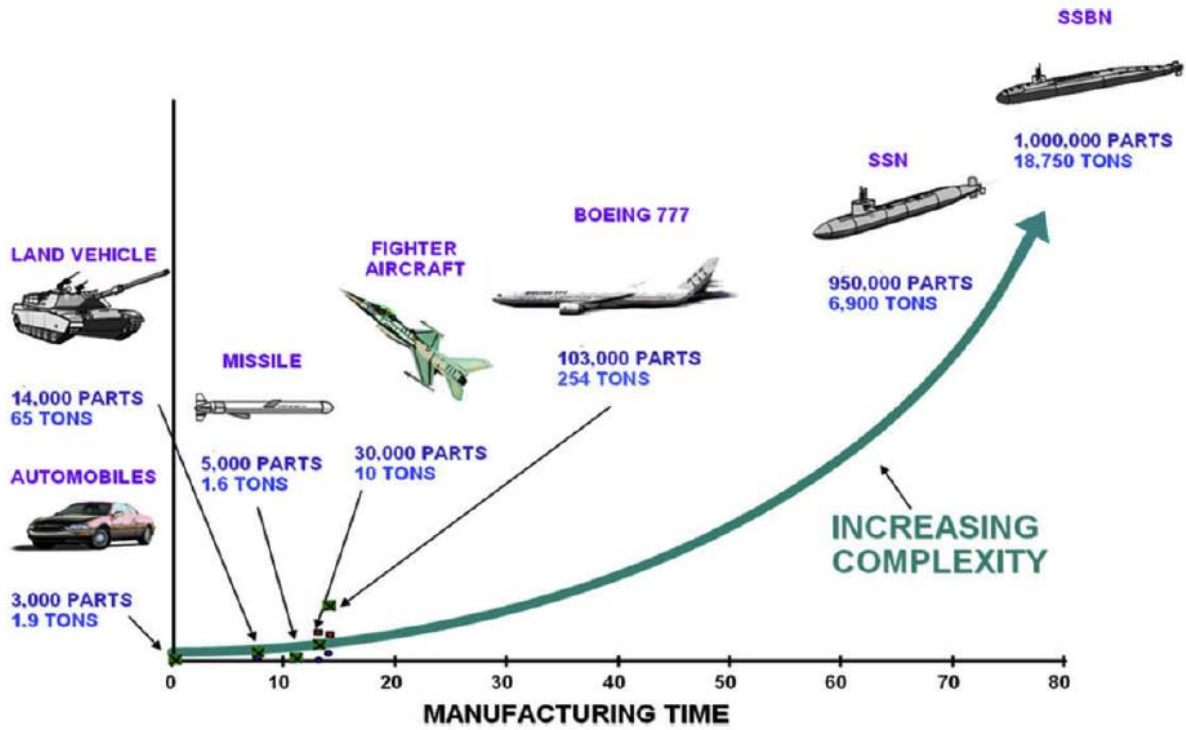


Figure I.4: Increasing Complexity of Engineering Designs

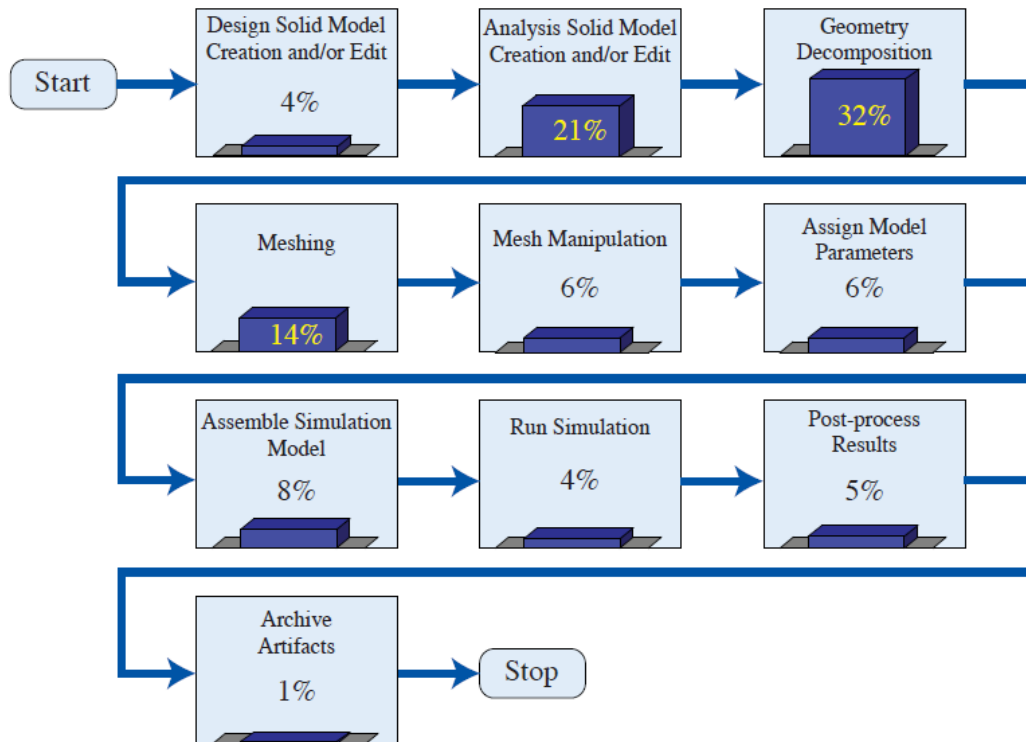


Figure I.3: Estimation of the relative time costs of each task at Sandia National Laboratories

II Geometric Tools for the Model

A. B-splines and NURBS

As background for our discussion of T-splines, a brief overview of B-spline and NURBS curves, surfaces, and solids is provided focusing on those features, which are important for understanding the generalization to T-splines. The cases considered here consist of a single NURBS patch. However, the generalization to the multi-patch NURBS case, it just involves a transformation between the control point indices of each patch and the corresponding global control points.

A.i. Knot vectors and the B-spline basis

A knot vector is a non-decreasing sequence of real numbers that indicate parameter values, denoted $\Xi = \{\xi_1, \xi_2, \dots, \xi_{n+p+1}\}$, where $\xi_i \in \mathbb{R}$ is the i^{th} knot, p is the polynomial degree of the B-spline basis functions, and n is the number of basis functions, which comprise the B-spline. B-spline basis functions for a given degree, p , are defined recursively in the parameter space by the Cox-de Boor recursion formula as follows:

$$N_{A,0}(\xi) = \begin{cases} 1 & \xi_A \leq \xi < \xi_{A+1} \\ 0 & \text{otherwise} \end{cases} \quad (II.1)$$

$$N_{A,p}(\xi) = \frac{\xi - \xi_A}{\xi_{A+p} - \xi_A} N_{A,p-1}(\xi) + \frac{\xi_{A+p+1} - \xi}{\xi_{A+p+1} - \xi_{A+1}} N_{A,p-1}(\xi) \quad (II.2)$$

The de Boor algorithm [49] provides a standard method and is perhaps the most famous of the existing algorithms for evaluation of B-spline basis functions and their derivatives.

From (II.1) and (II.2), one can verify that B-spline basis functions possess the following properties:

i. *Partition of unity:*

$$\sum_{A=1}^n N_{A,p}(\xi) = 1, \xi \in [\xi_1, \xi_{n+p+1}] \quad (II.3)$$

ii. *Pointwise nonnegativity:*

$$N_{(A,p)}(\xi) \geq 0, A = 1, 2, \dots, n \quad (II.4)$$

iii. *Linear independence:*

$$\sum_{A=1}^n c_A N_{A,p}(\xi) = 0 \Leftrightarrow c_B = 0, B = 1, 2, \dots, n \quad (II.5)$$

iv. *Compact support:*

$$\{\xi \mid N_{A,p}(\xi) > 0\} \subset [\xi_A, \xi_{A+p+1}] \quad (II.6)$$

v. *Control of continuity:*

If a knot value has multiplicity k (i. e., $\xi_i = \xi_{i+1} = \dots = \xi_{i+k-1}$), then the basis functions are C^{p-k} -continuous at ξ_i . When $k = p$, the basis is C^0 and interpolatory at that location.

The first four properties ensure a well-conditioned and sparse matrix, while the fifth allows continuity to be reduced to better resolve steep gradients [50].

An example of a quadratic B-spline basis for $\mathcal{E} = \{0, 0, 0, 1, 2, 3, 4, 4, 5, 5, 5\}$ is shown in *Figure II.1*. The basis is interpolatory at $\xi = 0, 4, 5$. If the first and last knot values are repeated at least p times the knot vector is called open vector due to the use of an open knot vector. Otherwise, the basis is $C^{p-1} = C^1$ across element boundaries.

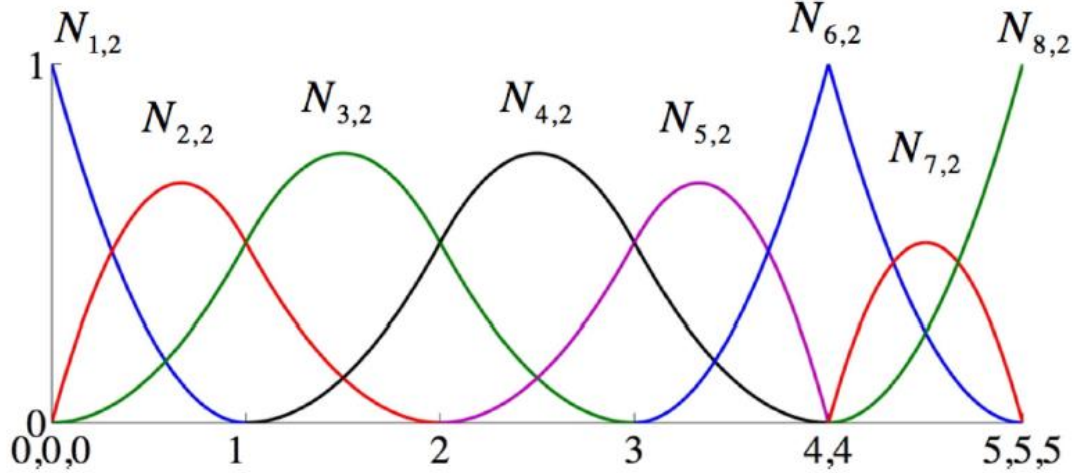


Figure II.1: Quadratic basis functions

A.ii. B-Spline Curves

A B-spline curve of degree p in \mathbb{R}^{d_s} is defined by a set of B-spline basis functions, $N = \{N_{A,p}\}_{A=1}^n$ and control points $P = \{P\}_{A=1}^n$ as follows:

$$C(\xi) = \sum_{A=1}^n P_A N_{A,p}(\xi) \quad (II.7)$$

Important properties of B-spline curves are:

- i. *Affine Covariance*: An affine transformation of a B-spline curve is obtained by applying the transformation to its control points.
- ii. *Convex Hull*: A B-spline curve lies within the union of all convex hulls formed by $p + 1$ contiguous control points (see [51] for the relationship between the convex hull and the polynomial order of the curve).
- iii. *Variation Diminishing*: A B-spline curve in \mathbb{R}^{d_s} cannot cross an affine hyperplane of codimension 1 (e.g., a line in \mathbb{R}^2 , plane in \mathbb{R}^3) more times than does its control polygon [52].

In addition, B-spline curves inherit all of the continuity properties of their underlying bases. This is illustrated in *Figure II.2*, where a B-spline curve from the basis shown in *Figure II.1*. At the spatial location corresponding to parameter value $\xi = 4$, the B-spline curve is only continuous. The B-spline curve interpolates the control point \mathbf{P}_6 at this location. The use of open knot vectors ensures that the first and last control points, \mathbf{P}_1 and \mathbf{P}_8 , are interpolated as well.

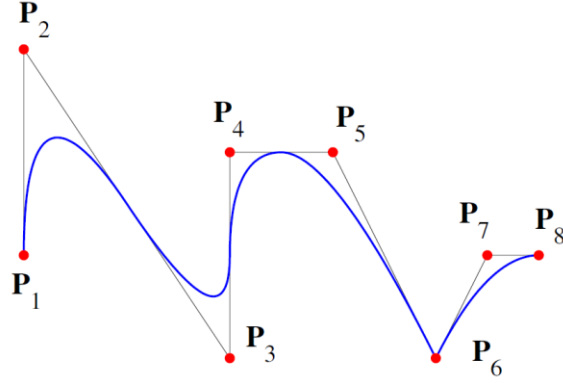


Figure 11.2: Quadratic B-Spline curve with control points and control polygon

A.iii. NURBS curves

A NURBS (Non-Uniform Rational B-Spline) is defined by a knot vector $\mathcal{E} = \{\xi_1, \xi_2, \dots, \xi_{n+p+1}\}$, a set of rational basis functions $R = \{R_{A,p}\}_{A=1}^n$, and a set of control points $P = \{P_A\}_{A=1}^n$ as:

$$C(\xi) = \sum_{A=1}^n P_A R_{A,p}(\xi) \quad (11.8)$$

The NURBS basis functions are defined as:

$$R_{A,p}(\xi) = \frac{w_A N_{A,p}(\xi)}{W(\xi)} \quad (11.9)$$

Where $W(\xi)$ is the weight function and w_B is the weight corresponding the B^{th} NURBS basis function.

$$W(\xi) = \sum_{B=1}^n w_B N_{B,p}(\xi) \quad (11.10)$$

For more efficient computation, a rational curve in \mathbb{R}^n can be represented by a polynomial curve in the projective space \mathbb{P}^n . Working in the projective coordinate system allows the algorithms which operate on B-splines to be applied to NURBS. As an example, if P_A is a control point of a NURBS curve then the corresponding homogeneous control point in projective space is $\tilde{P}_A = \{w_A P_A, w_A\}^T$. Thus, given a NURBS curve defined in \mathbb{R}^n by (11.8) the corresponding B-spline curve defined in \mathbb{P}^n is:

$$C(\xi) = \sum_{A=1}^n \tilde{P}_A N_{A,p}(\xi) \quad (11.11)$$

A.iv. NURBS surfaces and solids

To maintain single-index notation, which is standard for T-splines, we introduce a mapping, \tilde{A} , between the tensor product space and the global indexing of the basis functions and control points. Let $i = 1, 2, \dots, n$, $j = 1, 2, \dots, m$, and $k = 1, 2, \dots, l$ then in two dimensions and in three dimensions we define:

$$\tilde{A} = m(i - 1) + j \quad (11.12)$$

$$\tilde{A}(i, j, k) = (l \times m)(i - 1) + l(j - 1) + k \quad (11.13)$$

NURBS basis functions for surfaces and volumes are defined by the tensor product of univariate B-spline basis functions. Let $N_{i,p}(\xi)$, $M_{j,q}(\eta)$, and $L_{l,r}(\zeta)$ be univariate B-spline basis functions associated with the knot vectors $\Xi^1 = \{\xi_1, \xi_2, \dots, \xi_{n+p+1}\}$, $\Xi^2 = \{\eta_1, \eta_2, \dots, \eta_{m+q+1}\}$, and $\Xi^3 = \{\zeta_1, \zeta_2, \dots, \zeta_{l+r+1}\}$. In two dimensions, with $A = \tilde{A}(i, j)$ and $\hat{A} = \tilde{A}(i, j)$

$$R_A^{p,q}(\xi, \eta) = \frac{N_{i,p}(\xi)M_{j,q}(\eta)w_A}{\sum_{i=1}^n \sum_{j=1}^m N_{i,p}(\xi)M_{j,q}(\eta)w_{\hat{A}}} \quad (11.14)$$

Where $R_A^{p,q}(\xi, \eta)$ are the surface NURBS basis functions. Similarly, in three dimensions, with $A = \tilde{A}(i, j, k)$ and $\hat{A} = \tilde{A}(i, j, k)$

$$R_A^{p,q,r}(\xi, \eta, \zeta) = \frac{N_{i,p}(\xi)M_{j,q}(\eta)L_{l,r}(\zeta)w_A}{\sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^l N_{i,p}(\xi)M_{j,q}(\eta)L_{l,r}(\zeta)w_{\hat{A}}} \quad (11.15)$$

Where $R_A^{p,q,r}(\xi, \eta, \zeta)$ are the volume NURBS basis functions.

Given a control mesh $\{P_A\}$, where $A = 1, 2, \dots, (n \times m)$ for surfaces, and $A = 1, 2, \dots, (n \times m \times l)$ for volumes, we can define a NURBS surface and a NURBS volume as it follows:

$$S(\xi, \eta) = \sum_{A=1}^{n \times m} R_A^{p,q}(\xi, \eta)P_A \quad (11.16)$$

$$V(\xi, \eta) = \sum_{A=1}^{n \times m \times l} R_A^{p,q,r}(\xi, \eta, \zeta)P_A \quad (11.17)$$

NURBS inherit all of the important properties from their piecewise polynomial counterparts and are shown below.

- i. *Partition of unity*
- ii. *Pointwise nonnegative*
- iii. *Affine covariance*
- iv. *The convex hull property*

The continuity of a NURBS object follows from that of the basis in exactly the same manner as for B-splines. A B-spline object that does not use weights is called a polynomial B-spline to distinguish it from a rational B-spline. Note that if all the weights are set equal, a rational B-spline is reduced to a polynomial B-spline.

A.v. The index, parameter, and physical spaces

Figure II.3 shows both the physical mesh and control mesh in the physical space for a biquadratic NURBS surface generated from $\mathcal{E}^1 = \{0,0,0,1,2,2,2\}$ and $\mathcal{E}^2 = \{0,0,0,1,1,1\}$. Note that the control points are not generally interpolated by the surface except of the corner control points, which are lying on the surface.

The physical mesh is comprised of the image of the parent domain under the geometrical mapping. The curves in the physical mesh are the images of the knot lines. In this case, we have two elements. The control mesh is comprised of the red control points, the black lines connecting the control points, and the images of the knot spans under a bilinear mapping.

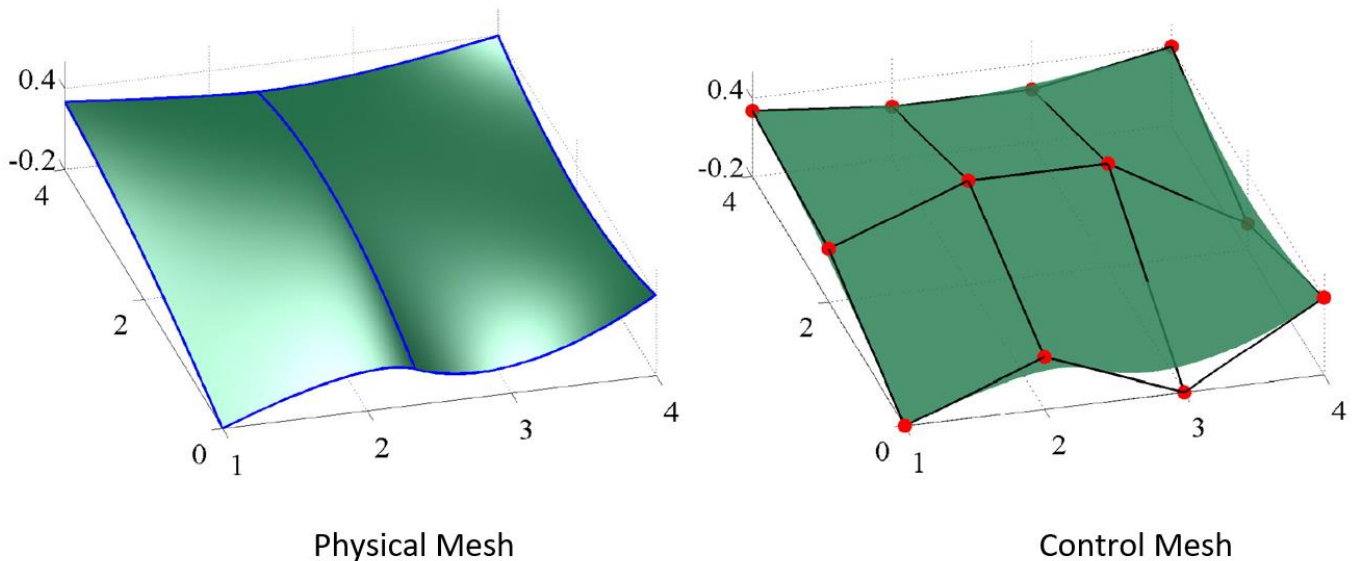


Figure II.3: The physical mesh and the control mesh of a biquadratic B-spline surface

Figure II.4 shows the domain of the mesh, a subset of the parameter space, corresponding to the biquadratic NURBS surface, seen in Figure II.3. The domain of the mesh is simply the pre-image of the physical mesh. The NURBS mapping takes each point in the parameter space to a point in the physical space, and the images of the knot lines under the NURBS mapping create the boundaries of the physical elements.

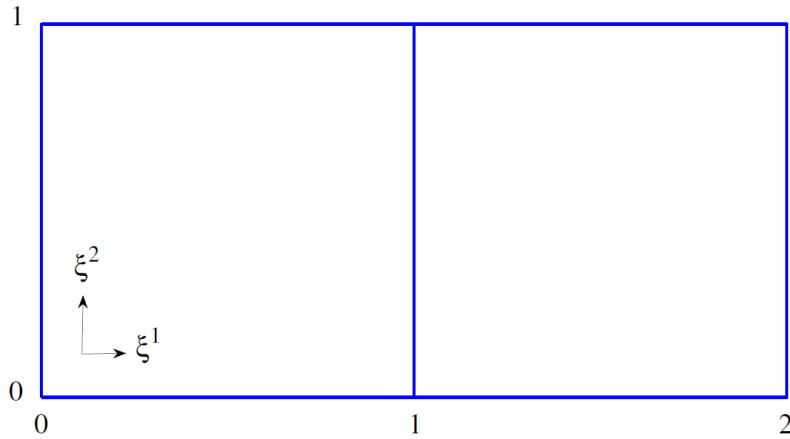


Figure II.4: A subset of the parameter space

The index space corresponding to the biquadratic NURBS surface, seen in Figure II.3, is shown in Figure II.5. This point of view is very useful for developing algorithms, as well as for building intuition about T-splines. In the index space, it is easy to spot the knot lines at which the support of any given function will begin or end, as well as which functions have support within any given element.

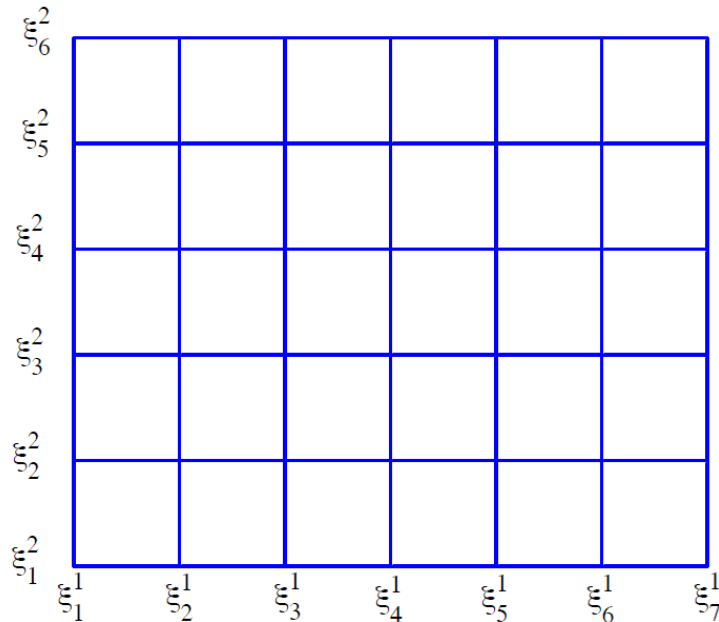


Figure II.5: The index space

B. T-Splines

This section gives a brief review of T-splines as presented in [46]. The notion of T-splines extends to any degree, in this section the discussion is restricted to cubic T-splines. Cubic T-splines are C^2 in the absence of multiple knots. A control grid for a T-spline surface is named T-mesh. When a T-mesh forms a rectangular grid, the T-spline reduces to a tensor product B-spline surface. Knot information for T-splines is expressed using knot intervals [53], non-negative numbers assigned to each edge of a T-spline control grid, that indicate the difference between two knots. *Figure II.6* [54] shows the pre-image of a portion of a T-mesh in (s, t) parameter space.

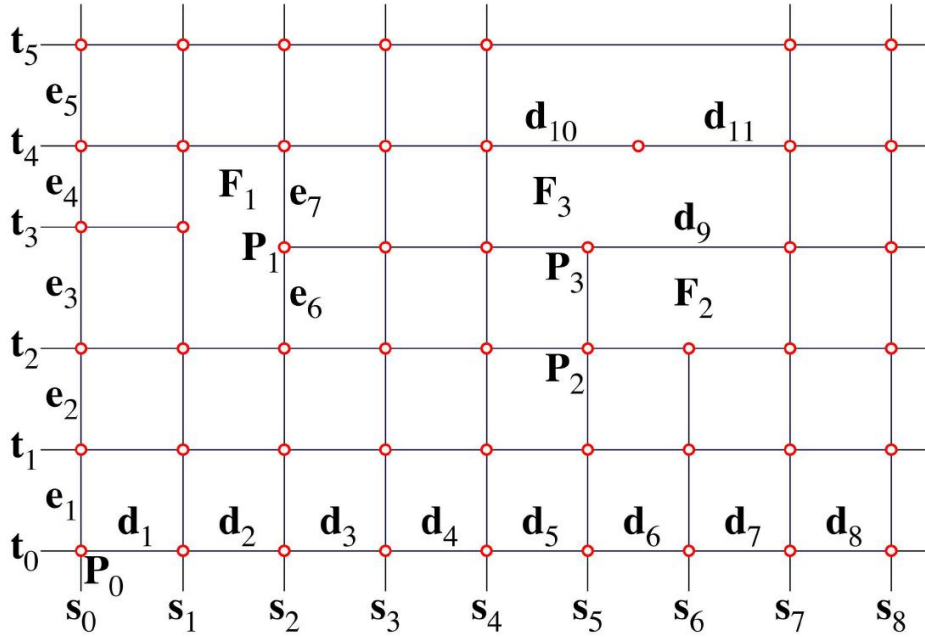


Figure II.6: Pre-image of a T-mesh

The d_i and e_i denote knot intervals, with red edges containing boundary-condition knot intervals. A T-mesh is basically a rectangular grid that allows T-junctions, hence the name T-Splines. A T-junction is a vertex shared by one t-edge and two s-edges, or by one s-edge and two t-edges. Knot intervals are constrained by the by the following rules:

Rule 1: The sum of all knot intervals along one side of any face must equal the sum of the knot intervals on the opposing side. Thus, on face F_1 in *Figure II.6*, $e_3 + e_4 = e_6 + e_7$, and on face F_2 , $d_6 + d_7 = d_9$.

Rule 2: If a T-junction on one edge of a face can legally be connected to a T-junction on an opposing edge of the face (thereby splitting the face into two faces), that edge must be included in the T-mesh without violating Rule 1. Legally connected means, that the sum of knot vectors on opposing sides of each face must always be equal. For example, a horizontal line would need to split face F_1 if and only if $e_3 = e_6$ and $e_4 = e_7$.

The motivation for Rule 2 is shown in *Figure II.7*. In this case, is legal to have an edge connecting A with P, and is also legal to have an edge connecting B with P. However, these two choices will create different t knot vectors for P, so Rule 2 resolves such ambiguity.

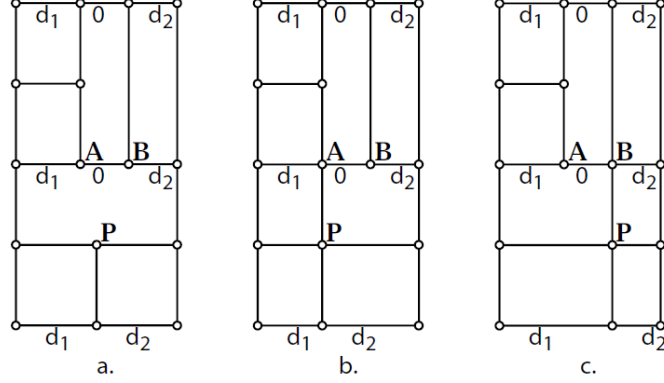


Figure II.7: Possible ambiguity

The knot coordinate system for a T-spline surface is used in writing an explicit formula:

$$P(s, t) = (x(s, t), y(s, t), z(s, t), w(s, t)) = \sum_{i=1}^n P_i B_i(s, t) \quad (II.18)$$

Where $P_i(x_i, y_i, z_i, w_i)$ are control points in P^4 , whose weights are w_i and whose Cartesian coordinates are $\frac{1}{w_i}(x_i, y_i, z_i)$. The Cartesian coordinates of points on the surface are given by the following equation:

$$\frac{\sum_{i=1}^n (x_i, y_i, z_i) B_i(s, t)}{\sum_{i=1}^n w_i B_i(s, t)} \quad (II.19)$$

In (II.18) $B_i(s, t)$ are the blending functions and are defined as follows:

$$B_i(s, t) = N[s_{i0}, s_{i1}, s_{i2}, s_{i3}, s_{i4}](s) N[t_{i0}, t_{i1}, t_{i2}, t_{i3}, t_{i4}](t) \quad (II.20)$$

Where $N[s_{i0}, s_{i1}, s_{i2}, s_{i3}, s_{i4}](s)$ is the cubic B-spline basis function associated with the knot vector

$$s_i = [s_{i0}, s_{i1}, s_{i2}, s_{i3}, s_{i4}] \quad (II.21)$$

and $N[t_{i0}, t_{i1}, t_{i2}, t_{i3}, t_{i4}](t)$ is the cubic B-spline basis function associated with the knot vector

$$t_i = [t_{i0}, t_{i1}, t_{i2}, t_{i3}, t_{i4}] \quad (II.22)$$

The knot lines for blending functions are illustrated in *Figure II.8*.

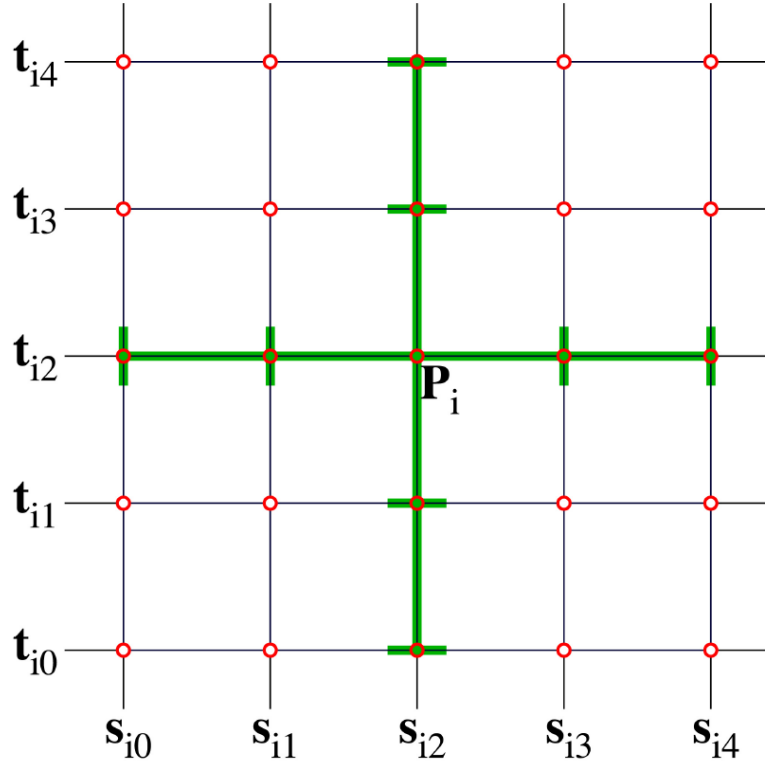


Figure II.8: Knot lines for blending function $B_i(s, t)$

The T-spline equation is very similar to the equation for a rational B-spline surface. The difference between those equations is in how the knot vectors \mathbf{s}_i (II.21) and \mathbf{t}_i (II.22) are determined for each blending function $B_i(s, t)$. The explanation of how we inferred these knot vectors from T-mesh neighborhood of \mathbf{P}_i is provided by the Rule 3:

Rule 3: The knot coordinates of P_i are (s_{i2}, t_{i2}) and the knots s_{i3} and s_{i4} are found by considering a ray in parameter space $R(\alpha) = (s_{i2} + \alpha, t_{i2})$ and are the s coordinates of the first two s -edges intersected by the ray (not including the initial (s_{i2}, t_{i2})). The other knots in \mathbf{s}_i and \mathbf{t}_i are found in like manner.

We illustrate the above rule by a few examples. The knot vectors for \mathbf{P}_1 in *Figure II.6* are $\mathbf{s}_1 = [s_0, s_1, s_2, s_3, s_4]$ and $\mathbf{t}_1 = [t_1, t_2, t_2 + e_6, t_4, t_5]$. For \mathbf{P}_2 , $\mathbf{s}_2 = [s_3, s_4, s_5, s_6, s_7]$ and $\mathbf{t}_2 = [t_0, t_1, t_2, t_2 + e_6, t_4]$. For \mathbf{P}_3 , $\mathbf{s}_3 = [s_3, s_4, s_5, s_7, s_8]$ and $\mathbf{t}_3 = [t_1, t_2, t_2 + e_6, t_4, t_5]$. Once these knot vectors are determined for each blending function $B_i(s, t)$, the T-spline is defined using (II.18) and (II.20).

C. NURBS vs T-Splines

A NURBS surface is defined using a set of control points (Figure II.9), which lie topologically in a rectangular grid. Therefore, one of the major strengths of NURBS surfaces is that they can exactly represent all quadric surfaces, such as cylinder, spheres, ellipsoids, etc., as illustrated in Figure II.10 [55]. Additionally, they possess useful mathematical properties, such as the ability to be refined through knot insertion and C^{p-1} continuity for degree p NURBS. They also are convenient for freeform surface modeling and there exist many efficient and numerically stable algorithms to generate NURBS objects.

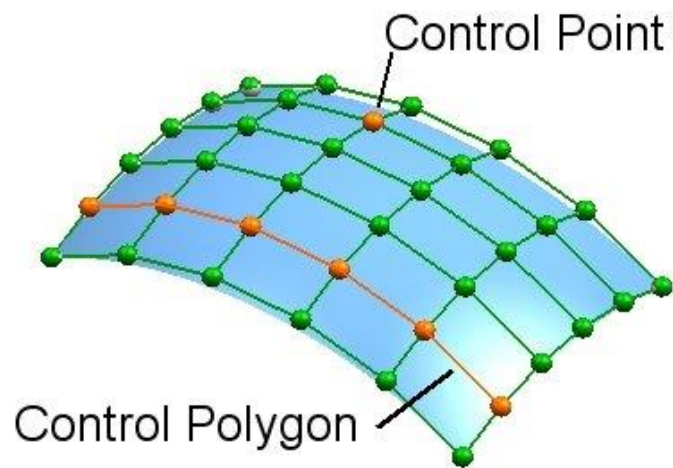


Figure II.9: A NURBS surface with a set of control points

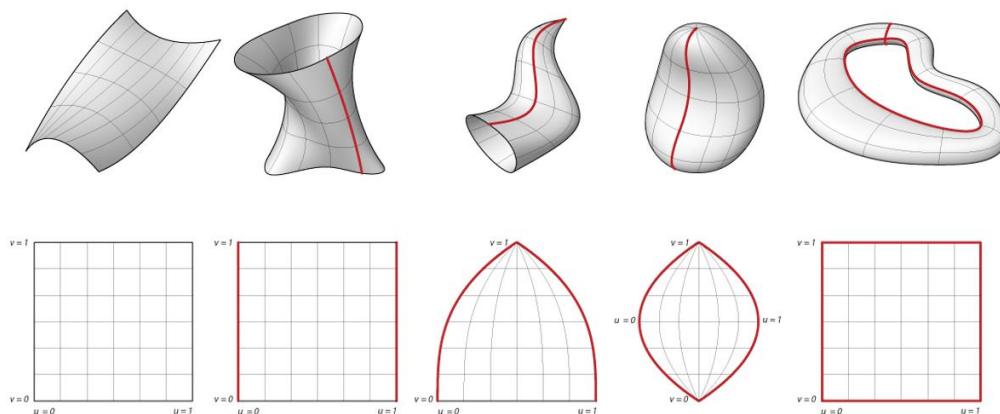


Figure II.10: Quadric primitives

Certainly, in the case of a marine propeller, which is a more complex model than the quadric primitives, using a typical NURBS surface for the representation of its geometry will create a large percentage of superfluous control points, which they will contain no significant geometric information, but merely are needed to satisfy the topological constraints. Superfluous control points are a serious nuisance for marine designers [56], because they require the designer to deal with more data and because they make the model more difficult to fair. This deficiency of NURBS will cause gaps and overlaps at intersections of surfaces, that cannot be avoided, leading to generation of a mesh. As an example, in the *Figure II.11* the hand geometry is modeled with multiple NURBS patches and the location where patches intersect introduces gaps and overlaps.



Figure II.11: Model of a hand comprised of multiple NURBS patches

Another serious problem with NURBS is that it is mathematically impossible for a trimmed NURBS to accurately represent the intersection of two NURBS surfaces without introducing gaps in the model. Additionally, it is impossible to represent most shapes (including a propeller) using a single watertight NURBS surface.

On the other hand, T-splines address the fundamental limitations of NURBS and can model complicated engineering designs, such as a marine propeller, as a single, watertight geometry. Furthermore, NURBS are a special case of T-splines, so the existing technology based on NURBS extends to T-splines. T-Splines models are a solution to the gap/overlap problem of NURBS surfaces and significantly reduce the number of superfluous control points. With fewer control points, such models are easier to fair. In the *Figure II.12* a head model is shown, the NURBS model is defined by 4712 control points and the T-Spline model by 1109. The points with the red mark in the NURBS model are superfluous.

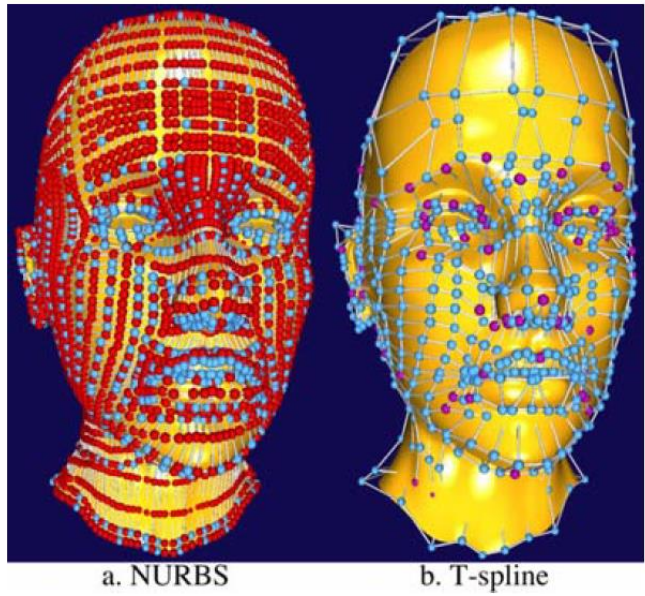


Figure II.12: Head modeled (a) as a NURBS and (b) as a T-Spline

In the *Figure II.13*, the T-Spline model of the hand, introduced in *Figure II.11* eliminates gaps and overlaps and is a single watertight geometry.

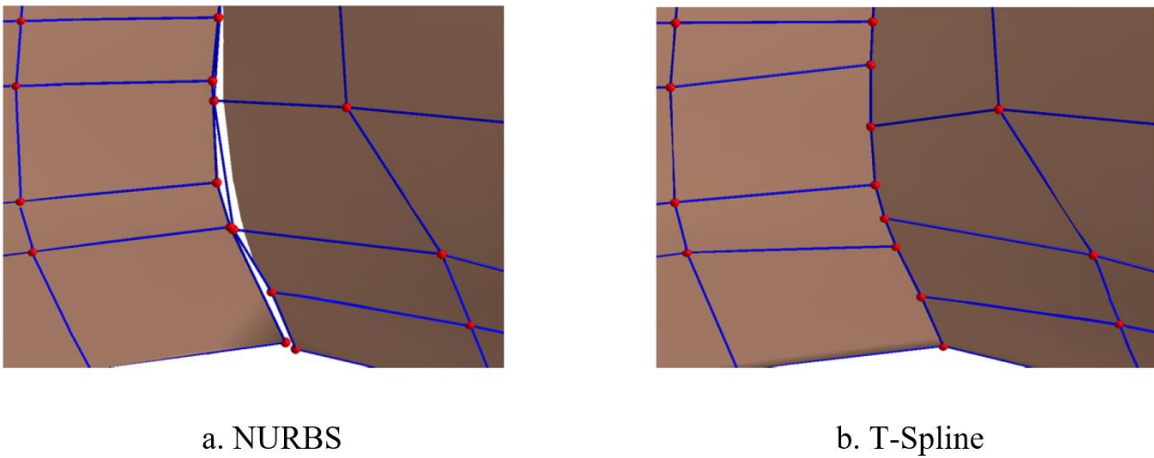


Figure II.13: A gap between two NURBS surfaces, fixed with a T-Spline

A major difference between a T-spline control mesh (Tmesh) and a NURBS control mesh is that T-splines allow a row of control points to terminate. The final control point in a partial row is called a T-junction, hence the name T-splines. T-junctions enable T-splines to be locally refined, by contrast NURBS refinement requires the insertion of an entire row of control points. Refinement is the process of adding new control points to a control mesh without changing the surface and it is an important basic operation used by designers. T-mesh also, permits control points of valence other than four, as illustrated in *Figure II.6*. Control points with valence other than four are called star points in some communities and extraordinary points in others. Star points give T-splines the power to model any surface using a single T-mesh. A personal watercraft is depicted in *Figure II.14* modeled as a NURBS (13 surfaces) and as a T-spline (1 surface). The yellow control points are star points. As mentioned before, a basic limitation of NURBS is that the trimmed NURBS models are not mathematically watertight. However, using T-splines it is possible to overcome this trimming problem. Any trimmed NURBS model can be represented by a watertight trimless T-spline [57] and multiple NURBS patches can be merged into a single watertight T-spline surface [58].

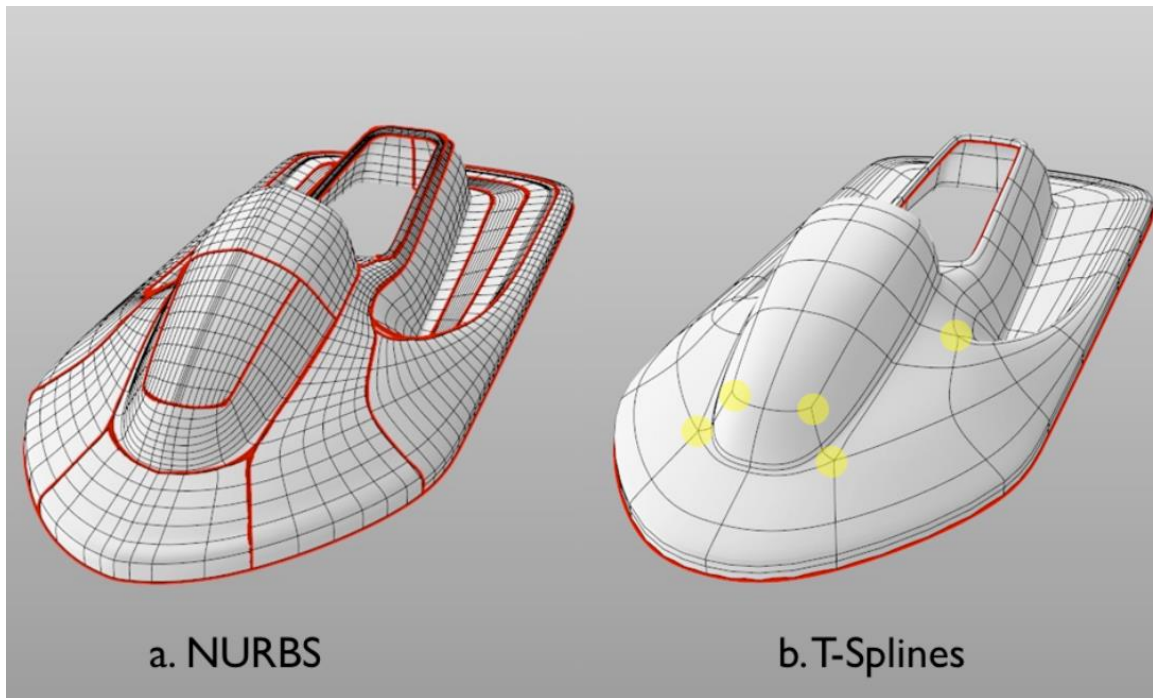


Figure II.14: Watercraft modeled (a) as a NURBS and (b) as a T-Spline

These properties make T-splines an ideal discretization technology for isogeometric analysis. *Table II.1* [59] lists several of the most critical and desirable properties of a design-through-analysis discretization technology. It is interesting to note that the largest group of properties are required to both design and analysis.

Table II.1: A comparison of C^0 finite elements, NURBS, and T-splines

	C^0 FE	NURBS	T-SPLINES
WATER TIGHT	✓	x	✓
LINEARLY INDEPENDENT	✓	✓	✓
PARTITION OF UNITY PROPERTY	✓	✓	✓
AFFINE COVARIANCE	✓	✓	✓
PASS STANDARD PATCH TESTS	✓	✓	✓
LOCALLY REFINEABLE	✓	x	✓
ACCOMMODATE EXTRAORDINARY POINTS	✓	x	✓
TRIMLESS OPTION	x	x	✓
SIMPLY IMPLEMENTED IN FEA CODES	✓	✓	✓
USED IN DESIGN	x	✓	✓
FORWARDS AND BACKWARDS COMPATIBLE WITH NURBS	x	✓	✓
EXACT REPRESENTATION OF CONIC SECTIONS	x	✓	✓
HIGHER-ORDER SMOOTHNESS	x	✓	✓

III Conceptual Marine Propeller Design

A. Marine Propeller Geometry

A.i. Blade Description

A brief discussion on propeller terminology and definitions is necessary to be provided, before the description of the parametric model. This will allow the reader to appreciate the complexity of the propeller geometry and follow later with more ease the description of the parametric model construction approaches

Detailed descriptions of propellers & blade's geometry can be found in several publications, e.g., [60] [61] [62], but here are only presented the basic definitions, which will facilitate the reader's understanding of parametric models' presentation in "Construction Steps". The coordinate system, which is adopted as the global reference frame for this parametric model, was proposed by the ITTC in 1978 [63] and it is shown in the *Figure III.1*. The X-axis is positive, forward and coincident with the shaft axis. the Y-axis is positive to starboard and the Z-axis is positive in the vertically downward direction.

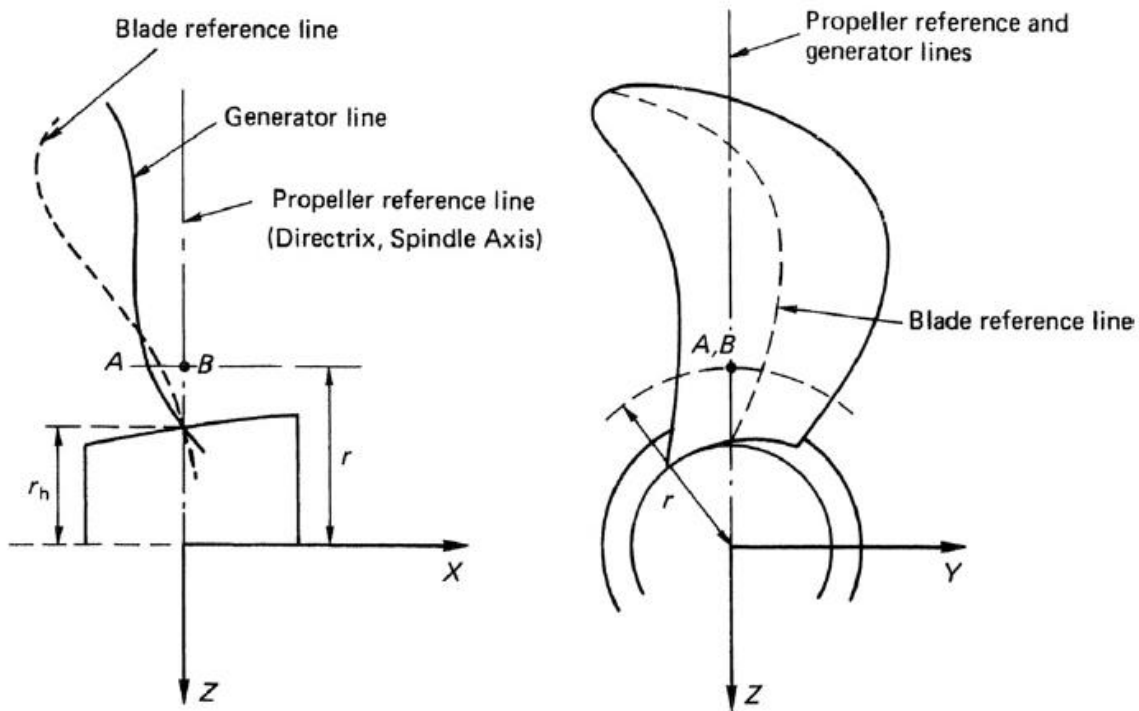


Figure III.1: Blade Reference Lines

The propeller blade is defined about a line normal to the shaft axis called “propeller reference line” or the “directrix”. Frequently synonymous with the propeller reference line or the “directrix is the term ‘spindle axis’, as seen in *Figure III.1*, the point A is depicted, where this helix intersects the plane defined by the directrix and the X-axis. Thus, the locus of all such points between the tip and root of the blade compose this generator line.

A common three-bladed marine propeller, along with its major constituent parts, is shown in *Figure III.2*. A propeller blade has two main hydrodynamic surfaces. The face (or pressure side) of the propeller is that part of the propeller seen, when viewed from astern and along the shaft axis and corresponds to the surface area of the blade that experiences high pressure and faces the flow downstream. On the other hand, the back of the blade is the surface, which faces forward, faces the flow upstream and experiences low pressure. The root of the blade comprises the area of the blade that attaches to propeller’s hub (a filleted blade area that guarantees structural strength and smooth transition between the hub and the blade), while the farthest point of the propeller along the span-wise/ radial direction is called the blade tip. Propellers are divided into right-handed and left-handed ones. When viewing the propeller from stern, a right-handed propeller rotates clockwise and the leading edge of each blade will be furthest edge along the hub. The edge to the left is called trailing edge. Similarly, the left-handed propellers rotate in a counterclockwise direction.

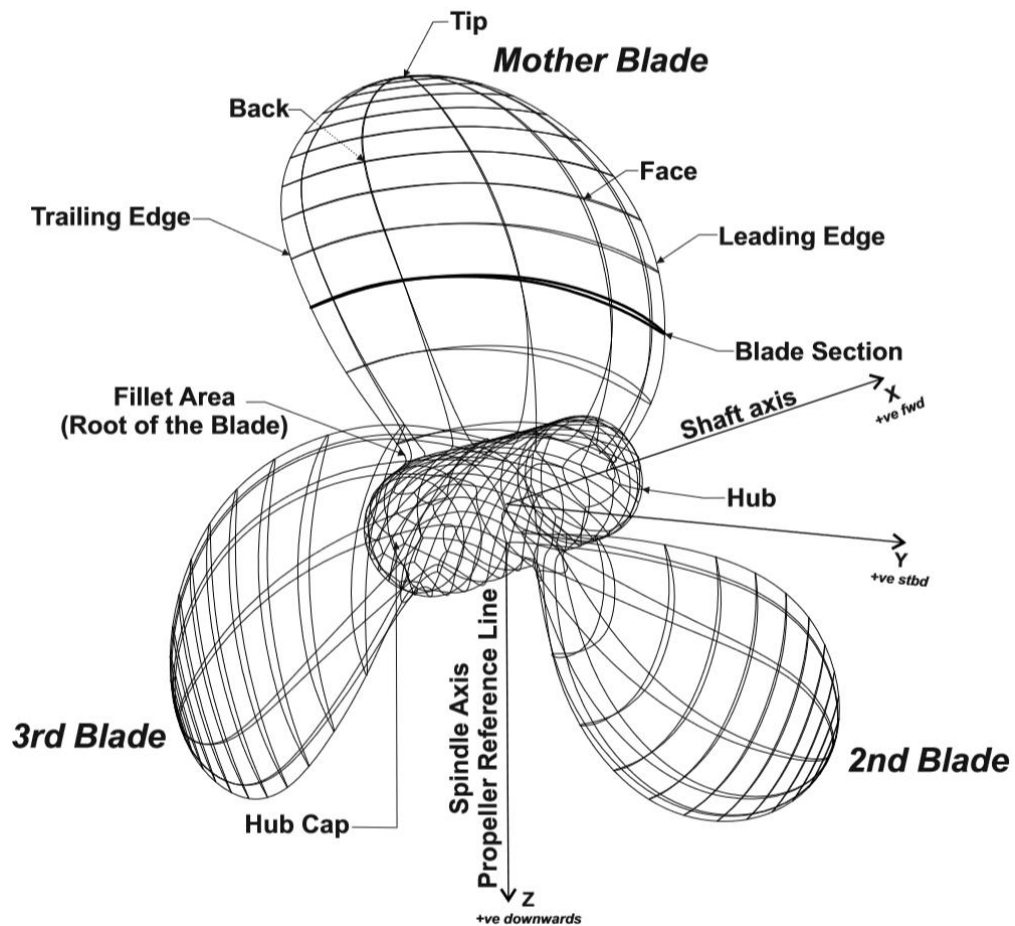


Figure III.2: Marine propeller parts

A series of hydrofoil sections, such as the one shown in *Figure III.9*, are commonly used to generate the blade surfaces of a propeller (*Figure III.3*) [64]. The blade may be generated via a single hydrofoil, appropriately transformed in the span-wise direction, or via multiple, differently-shaped hydrofoil profiles for different areas of the blade, e.g., root, mid and tip areas. Each hydrofoil section forming the blade is located at a corresponding radius, $R_i \leq R$, where R is the propeller's radius, and lies on a cylinder with the same R_i radius and axis coincident to the propeller's shaft axis. Further to this, hydrofoils are commonly rotated/twisted about the reference line by the so-called pitch angle and translated according to the skew and rake of the propeller. The terms pitch, skew and rake are described below. Finally, as it is shown in *Figure III.2*, the blades of a marine propeller are identical to the mother blade.

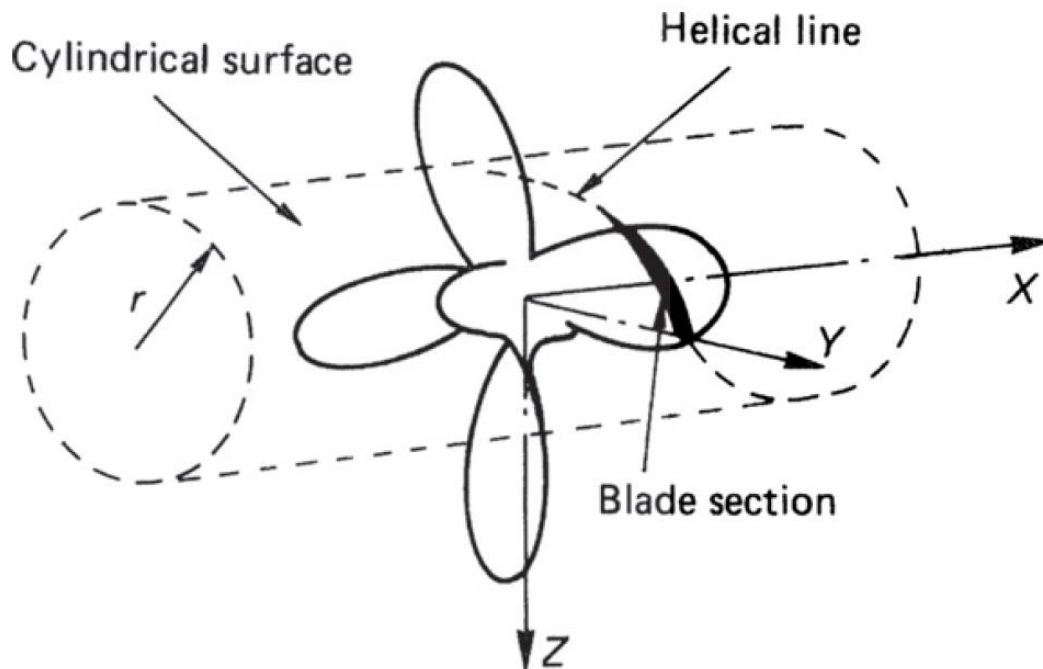


Figure III.3: Cylindrical Blade Section Definition

A.ii. Pitch

As described previously, a blade section is located at specified radius R_i placed, so that its chord line is rotated via a φ_i angle about the reference line (Z-axis); see *Figure III.4*. In the *Figure III.4* is also shown, a surface of a cylinder of radius R_i rolled out to its fully developed length $2\pi R_i$.

Thus, when this surface is wrapped around a cylinder of radius R_i , the chord line forms part of a helix on the cylinder in the same manner as a screw thread. This angle φ_i is called pitch angle or helix angle and is given by the following equation.

$$\varphi_i = \tan^{-1}\left(\frac{p_i}{2\pi R_i}\right) \quad (III.1)$$

In one revolution of the cylinder, a point on the helix travels p_i distance, measured normally on OX direction, and is named pitch of the section. Frequently, the pitch is supplied as dimensionless values p_i/D called pitch to diameter ratios. Each section of the blade can have a different pitch angle φ_i and therefore the pitch at $\frac{R_i}{R} = 0.7$, where R is the propeller radius, is commonly used as a representative value (the nominal pitch).

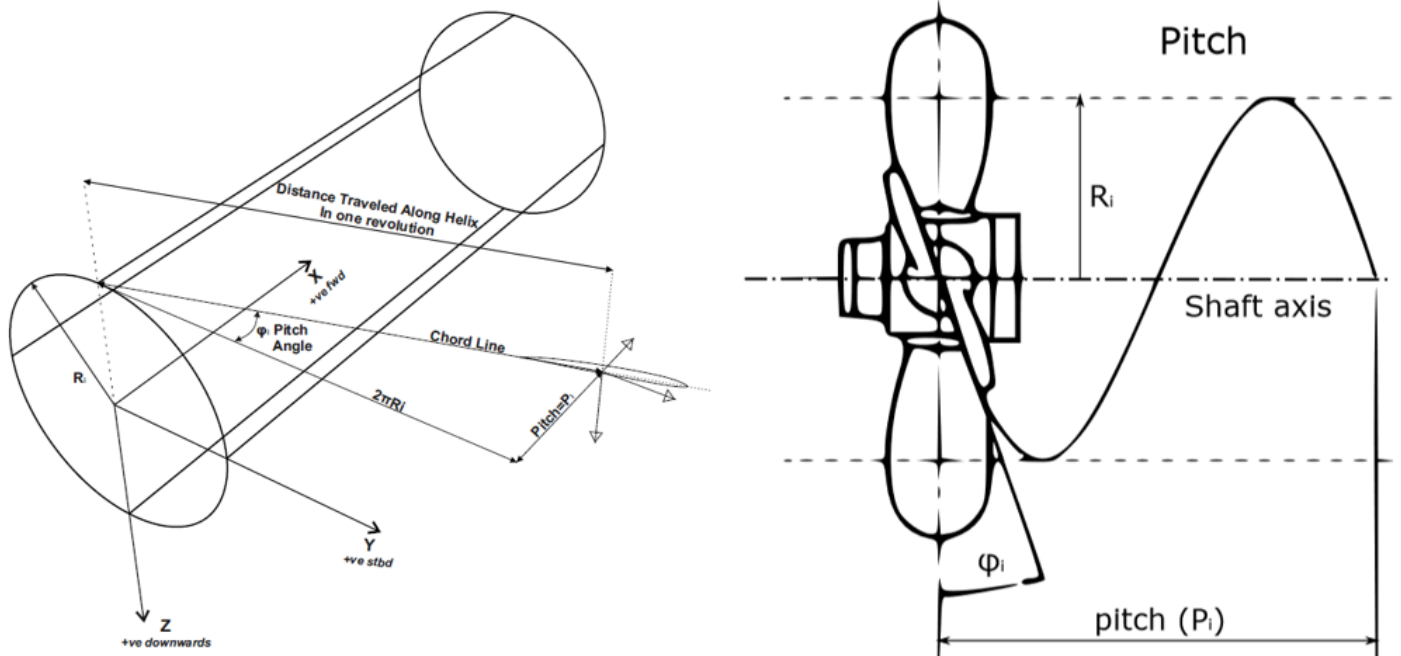


Figure III.4: Geometrical Representation of Pitch

A.iii. Skew & Rake

Skew is defined to be the displacement of a section along its chord line direction. Skew is also specified in angular form, called the skew angle. The skew angle θ_{si} of a particular section at R_i is shown in *Figure III.5* and is the angle between the reference line and the midchord point of the section on the transverse plane on the YZ plane. In *Figure III.5*, the two different types of propeller skew: balanced and biased skew designs, are also depicted.

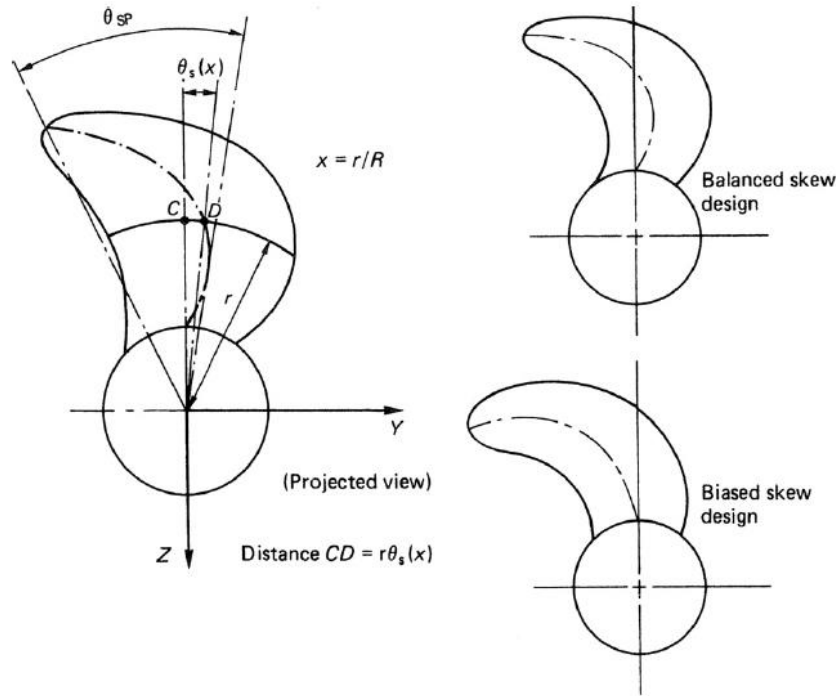


Figure III.5: Skew in Angular Form Definition

Rake corresponds to an aft or forward displacement (*Figure III.6*) of the section along the shaft-axis. The terms rake and skew have a cross-coupling component due to the helical nature of blade sections. Propeller rake is induced when the sections are displaced along the pitch helix and is divided into two components: generator line rake (i_G) and skew induced rake (i_S). The total rake of the i^{th} section with respect to directrix (i_T) is given by:

$$i_{Ti} = i_{Si} + i_{Gi} \quad (III.2)$$

In cases, when the generator line is a linear function of radius, it is meaningful to talk about propeller rake (i_p) or a propeller rake angle (θ_{ip}). These terms are measured at the propeller tip as shown in *Figure III.6*, where the propeller rake angle is given by:

$$\theta_{ip} = \tan^{-1} \left(\frac{i_p}{R} \right) \quad (III.3)$$

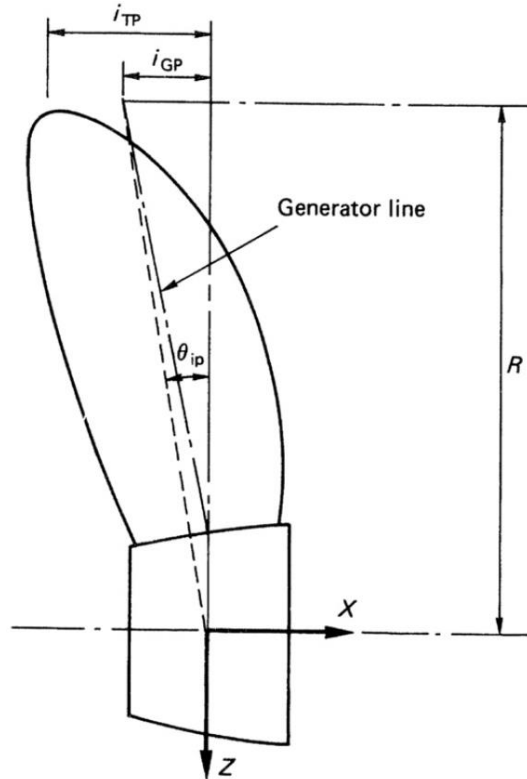


Figure III.6: Tip Rake Definition

Figure III.7 is presented in order to clarify the meaning of the skew induced rake consider of a section. In this Figure, an unwrapping cylindrical section at some radius R_i between the tip and root of the blade is illustrated. Skew induced rake is the component, measured in the X-direction, of the helical distance around the cylinder from the mid-chord point of the section to the projection of the directrix when viewed normally to the YZ plane. The following Equation (III.4) gives the skew induced rake of i^{th} section.

$$i_{si} = r\theta_{si} \tan(\varphi_i) \quad (III.4)$$

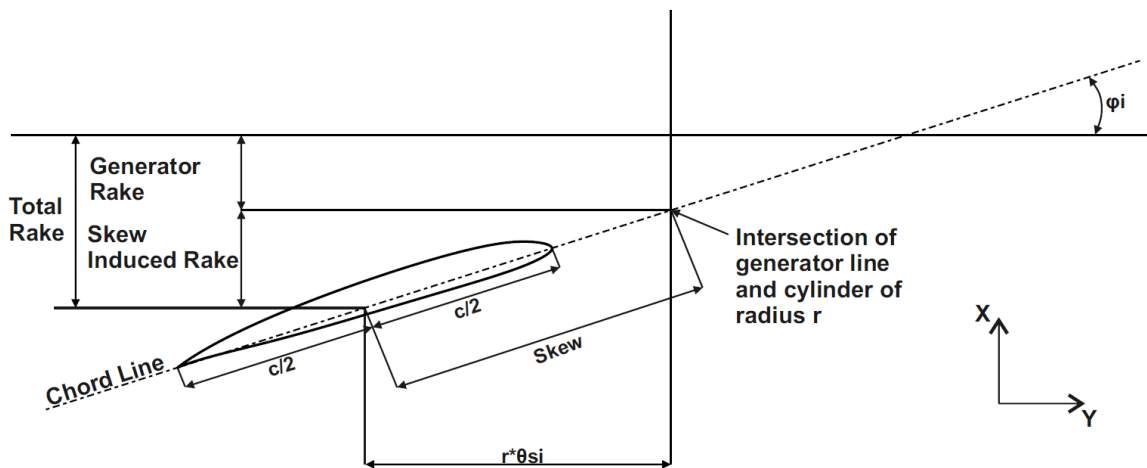


Figure III.7: Definition of Total Rake

A.iv. Propeller Outlines and Area

There are four basic outlines, which describe the propeller blade shape:

1. The projected outline.
2. The developed outline.
3. The expanded outline.
4. The swept outline.

The projected outline is the view of the propeller blade, when the propeller is viewed along the shaft center line. The helical sections in this view, are defined in their appropriate pitch angles and the sections are seen to lie along circular arcs whose center is the shaft axis. This view together with the developed and expanded views is illustrated in *Figure III.8*. From this Figure it is clear that the projected area A_p is given by

$$A_p = Z \int_{r_h}^R (\theta_{TE} - \theta_{LE}) r dr \quad (III.5)$$

The developed outline is related to the mentioned above projected outline. It is a helically based view, but the pitch of the sections has been reduced to zero and is used to give an appreciation of the true form of the blade and the distribution of chord lengths. In *Figure III.8* is shown this view in relation to the projected outline. It is necessary to integrate the area under the developed profile curve numerically if a precise value is required. For most purposes, to calculate the developed area it is sufficient to use the approximation for the developed area A_D as being, where A_E is the expanded area of the blade.

$$A_D \approx A_E \quad (III.6)$$

However, several researchers have developed empirical relationships for the estimation of the developed area. A relationship, like this, proposed by Burrill [65] for non-skewed forms, is

$$A_D \approx \frac{A_p}{(1.067 - 0.229P/D)} \quad (III.7)$$

In general, the developed area is greater than the projected area and slightly less than the expanded area.

The expanded outline is not really an outline, it could more correctly be termed a plotting of the chord lengths at their correct radial stations about the directrix. In this outline, the pitch angle of each section is reduced to zero. However, this view is used to give an idea of the blade section forms used, as these are frequently plotted on the chord lengths, as seen in *Figure III.8*. The expanded area is the simplest of the areas that can be calculated and is given by following the relationship:

$$A_E = Z \int_{r_h}^R c \, dr \quad (III.8)$$

Blade area ratio is either the projected, developed or expanded blade area, depending on the context, divided by the propeller disc area A_0 :

$$\begin{cases} \frac{A_p}{A_0} = \frac{4A_p}{\pi D^2} \\ \frac{A_D}{A_0} = \frac{4A_D}{\pi D^2} \\ \frac{A_E}{A_0} = \frac{4A_E}{\pi D^2} \end{cases} \quad (III.9)$$

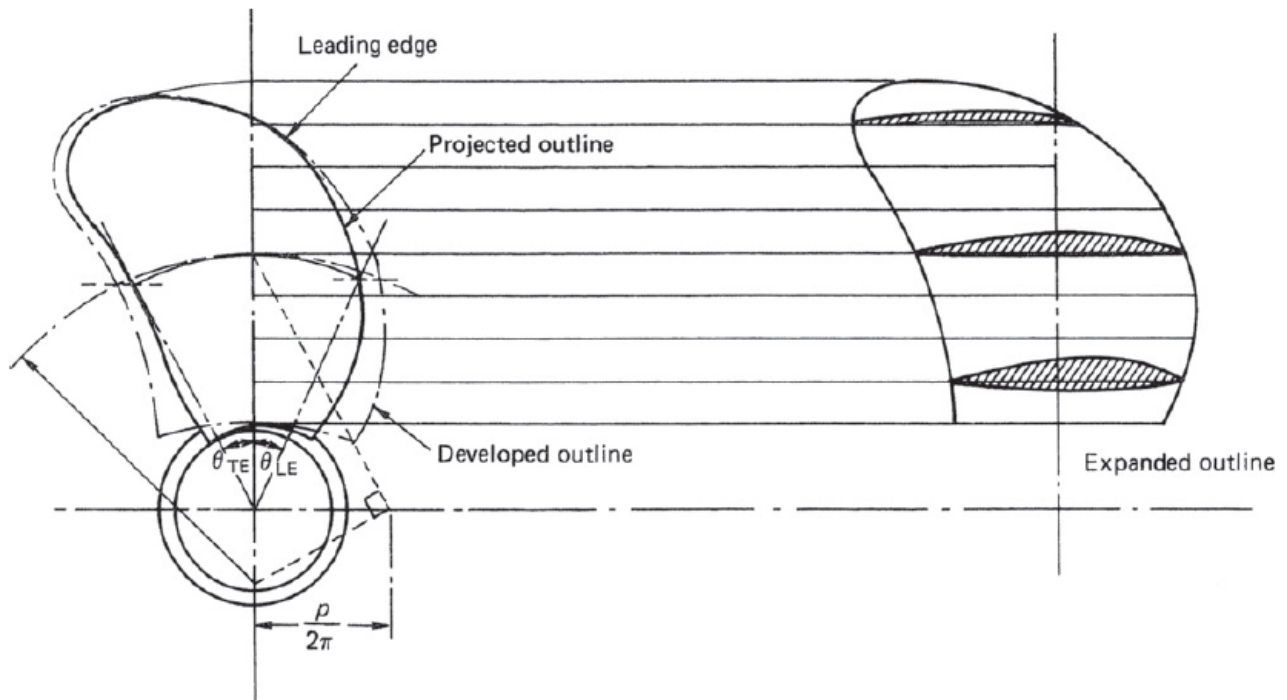


Figure III.8: Outline Definition

A.v. Section Geometry and Definition

A hydrofoil is a specialized version of the airfoil [66] that is manufactured to work in water. The shape of the hydrofoil plays a vital role for the construction of a marine propeller and *Figure III.9* shows a typical airfoil and illustrates various items of airfoil terminology [67].

The mean line or camber line of a hydrofoil comprises the locus of equidistant points between the upper and lower sides. The points where the camber line meets the hydrofoil coincide with the leading and trailing edge points of the hydrofoil and the straight line connecting these two edge points, is called the chord line or simply chord (c). The maximum distance between the mean/camber line and the chord line is called camber, while the hydrofoil thickness is the maximum distance between its upper and lower side.

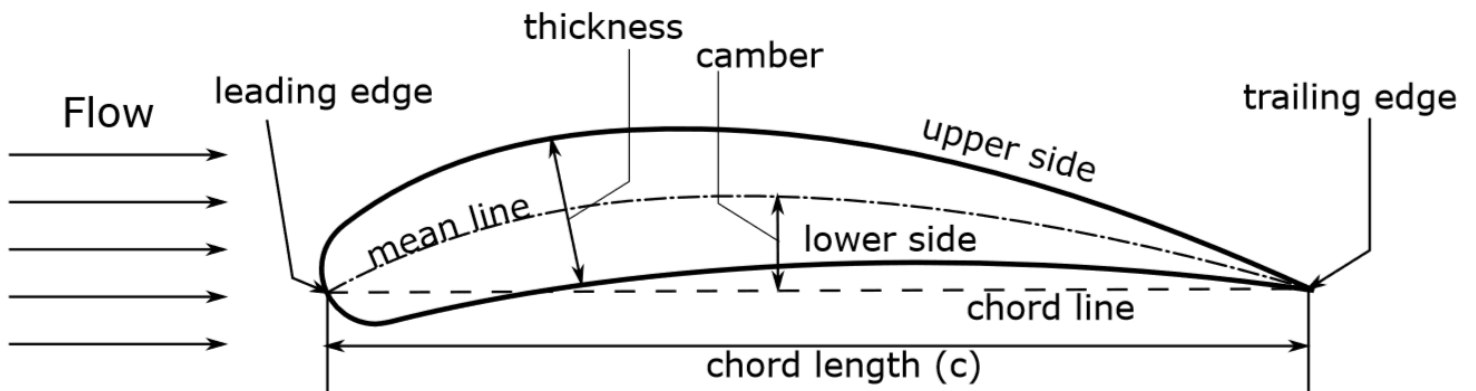


Figure III.9: Airfoil Terminology

B. Hydrofoil Mother Concept

In this model, for the generation of the blade the “*Hydrofoil Mother*” concept has been adopted, and it is referring to a hydrofoil, that functions as a geometrical basis for all hydrofoil sections. The process, that is followed to design the “*Hydrofoil Mother*” or the initial hydrofoil is described in “*Step 1. Parametric Model of a Generic 2-D Hydrofoil*” and leads to a B-spline shaped-foil. The control points of the initial hydrofoil are transformed into the 3D-space (see “*Step 4. Blade Cage*”) in order to create the T-spline blade cage.

The first transformation is related with the sizing of the hydrofoils and their radial positioning along the wingspan and *Figure III.10* illustrates this transformation of 30 B-spline curves (sections). The part of C# code, which is used for the first transformation of the sections is shown on *Table III.1*.

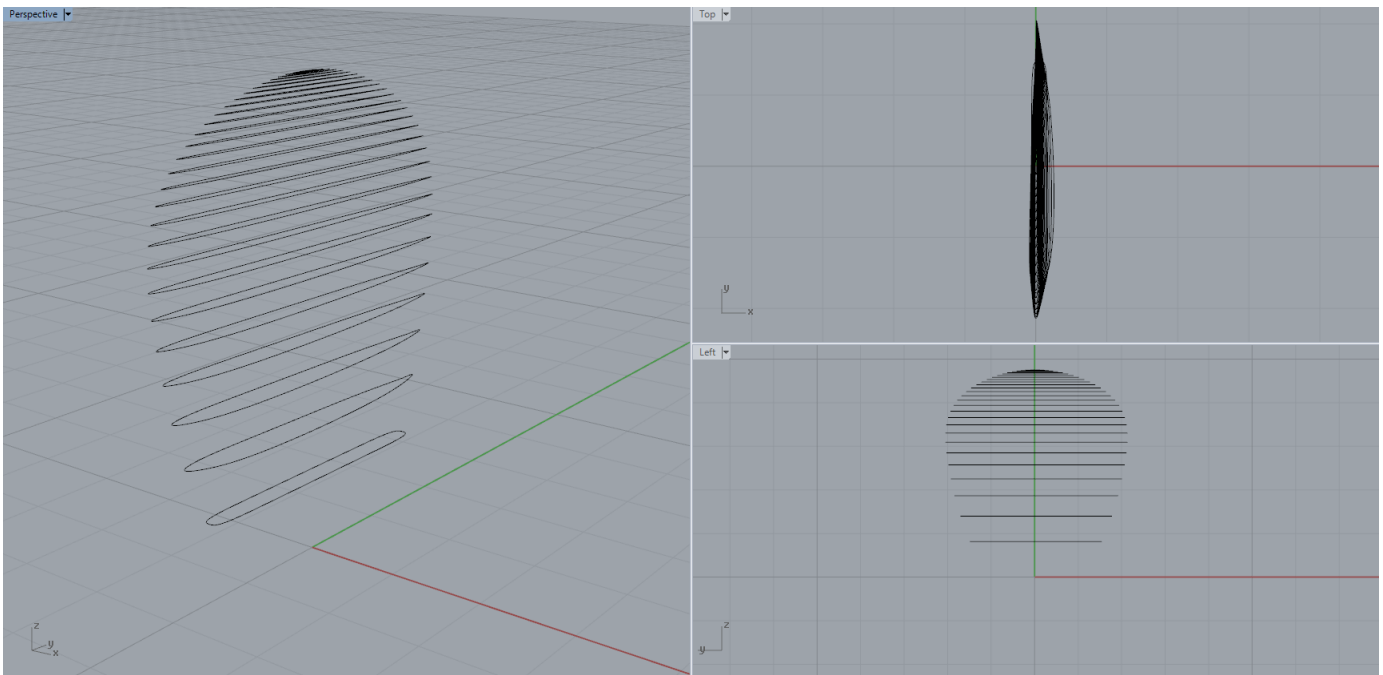


Figure III.10: Hydrofoil Sections Positioned and Sized

Table III.1: Part of C# code for the 1st Transformation

```

//sizing airfoils of every section
List<Point3d> Profile1 = new List<Point3d>();
for (j = 0; j <= Profile_F.Count - 1; j++)
{
    Point3d pt = new Point3d();
    //first section(connection with hub)
    //Profile_F bounding rectangle airfoil
    if (i == 0&& i == 1)
    {
        pt.X = -1.0 * RotationSign * Profile_F[j].Y / Thickness_mother * ThicknessRatio[i] * Diameter * 1.2;
        pt.Y = -1.0 * RotationSign * (Profile_F[j].X / ProfileLength -0.5) * ChordLengthRatio[i] * Diameter * Chord_mother *1.2;
        pt.Z = 0.0;
    }
    //sections except from the last and the first one
    else if ((i < NumberSections - 1) && (i>1))
    {
        pt.X = -1.0 * RotationSign * Profile[j].Y / Thickness_mother * ThicknessRatio[i] * Diameter;
        pt.Y = -1.0 * RotationSign * (Profile[j].X / ProfileLength -0.5) * ChordLengthRatio[i] * Diameter * Chord_mother;
        pt.Z = 0.0;
    }
    else
    {
        pt.X = -1.0 * RotationSign * Profile_F[j].Y / Thickness_mother * ThicknessRatio[i] * Diameter*0.5;
        pt.Y = -1.0 * RotationSign * (Profile_F[j].X / ProfileLength -0.5) * ChordLengthRatio[i] * Diameter * Chord_mother;
        pt.Z = 0.0;
    }
    Profile1.Add(pt);
}

```

Subsequently in the second transformation, the control points of the profiles are rotated via the pitch angle of each section around the Z axis, as seen in *Figure III.11* The following basic rotation matrix rotate vectors by an angle θ about the Z-axis, in three dimensions, using the right-hand rule and is used in C# code on *TableIII.2*.

$$R_z(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (III.10)$$

Table III.2: Part of C# code for the 2nd Transformation

```

//rotation of the blade (pitch)
List<Point3d> Profile2 = new List<Point3d>();
for (j = 0; j <= Profile1.Count - 1; j++)
{
    Point3d pt = new Point3d();
    pt.X = Profile1[j].X * Math.Cos(Angle) - Profile1[j].Y * Math.Sin(Angle); //rotation matrix
    pt.Y = Profile1[j].Y * Math.Cos(Angle) + Profile1[j].X * Math.Sin(Angle); //rotation matrix
    pt.Z = Radius + Profile1[j].Z; //Profile1 values only last section
    Profile2.Add(pt);
}

```

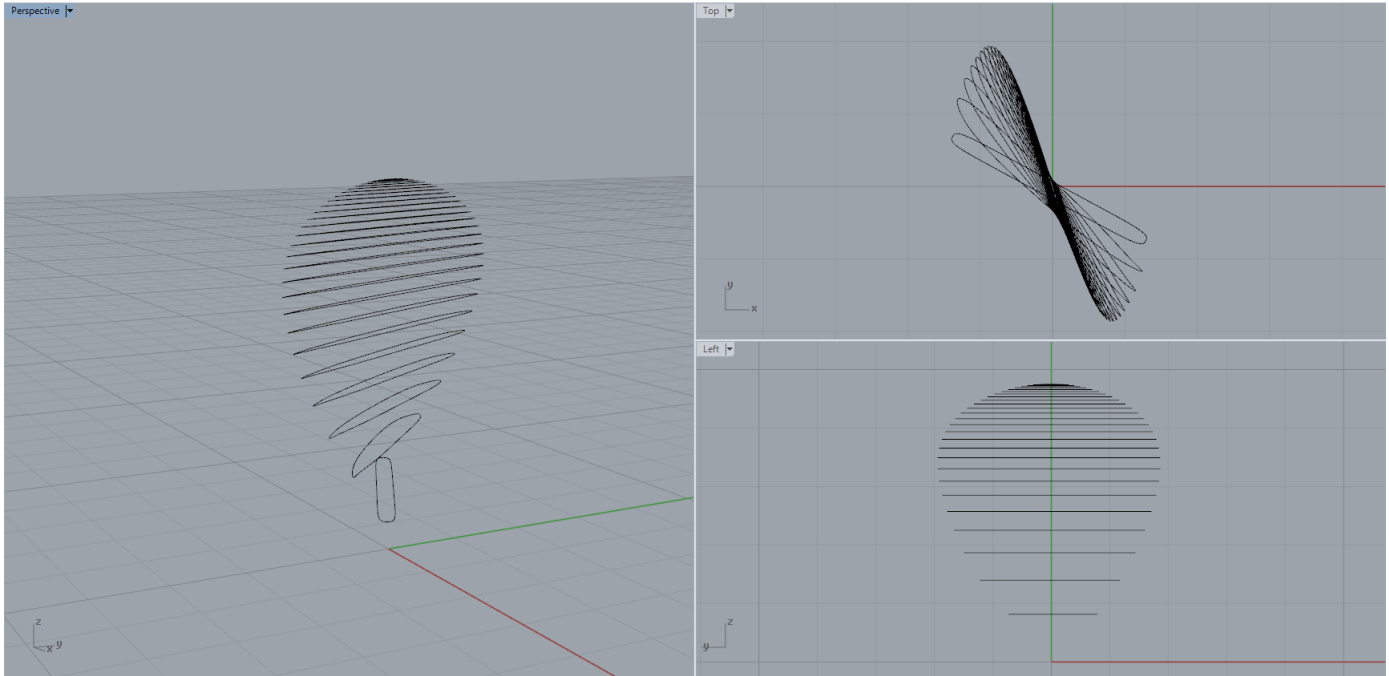


Figure III.11: Hydrofoil Sections Rotated via Pitch Angle

Finally, the control points of the hydrofoils are displaced due to the rake and the skew of each section. In the third transformation (*Table III.3*), parallel to this displacement, the sections are also warped, because they are lying on the surface of cylinders, as seen in *Figure III.3*. The impact of the third transformation on the control points is shown in *Figure III.12*.

Table IV.6: Part of C# code for the 3rd Transformation

```
//Projection of the Profile2 on a cylinder with axe on X and radius = Radius (of every
section)
List<Point3d> Profile3 = new List<Point3d>();
for (j = 0; j <= Profile2.Count - 1; j++)
{
    Point3d pt = new Point3d();
    Theta = Profile2[j].Y / Radius - RotationSign * Skew[i] + GeneralRotation;
    pt.X = Profile2[j].X - RakeRatio[i] * Diameter;
    pt.Y = Profile2[j].Z * Math.Sin(Theta);
    pt.Z = Profile2[j].Z * Math.Cos(Theta);
    Profile3.Add(pt);
}
```

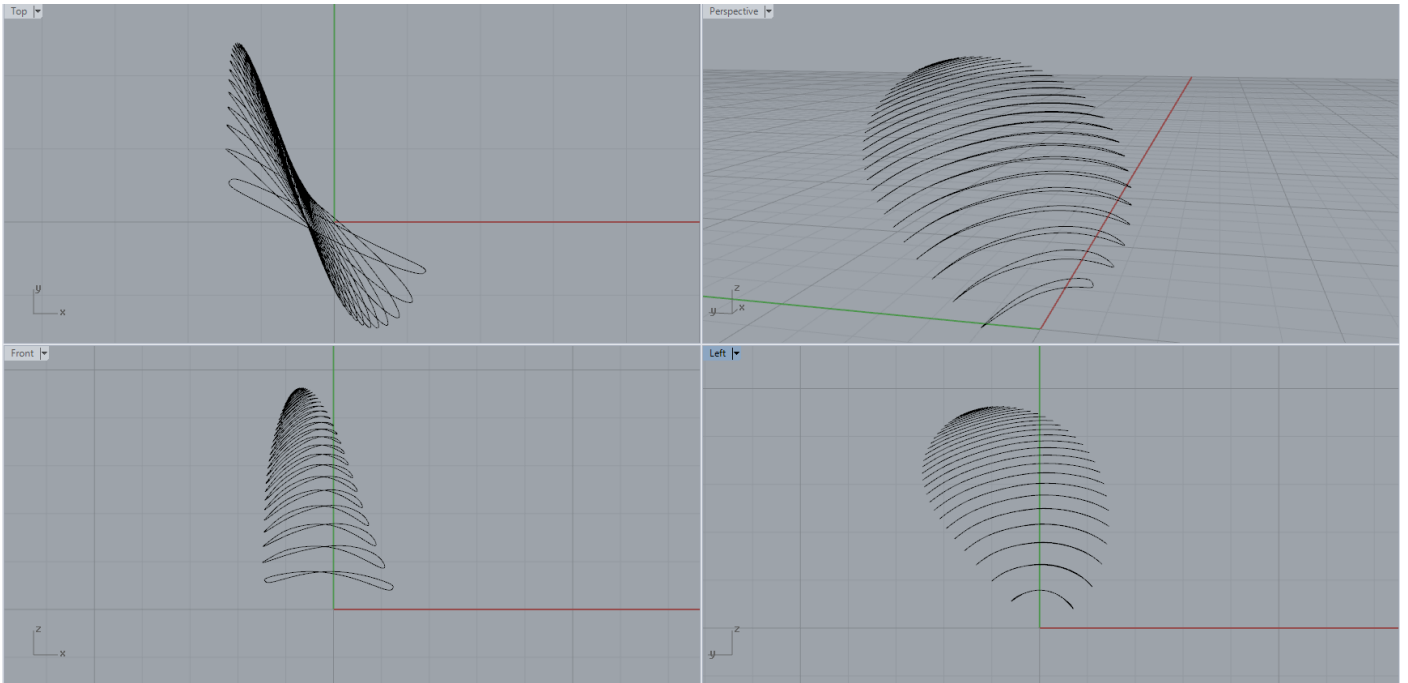



Figure III.12: Hydrofoil Sections Warped and Displaced via Rake and Skew

The way, that the control points of the sections projected on the cylinders with radius equal with their corresponding section radius is depicted in *Figure III.13*. The terms shown in the below Figure are defined in the above parts of C# codes.

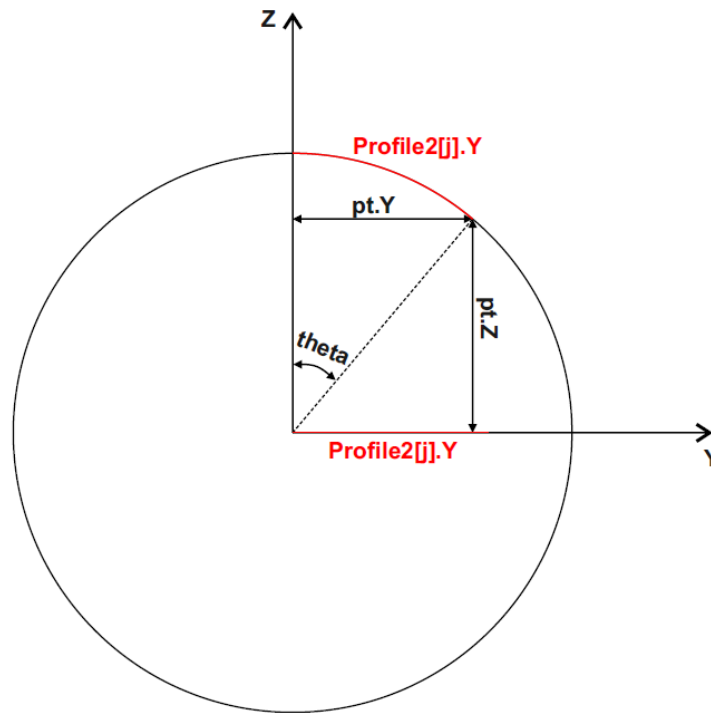


Figure III.13: Schematic Representation of a Control Point Projected on a Cylinder

IV Construction Steps

A. Working Interface

The working interface [68] of the parametric modeler is shown in the figure below. The main window is called canvas and an algorithm is developed by dragging components (functional blocks) and connecting them with wires. The program flows from the left to the right and the components are divided into groups depending on their function in the model. Finally, the generation of a marine propeller T-spline control cage is a 6-step process and each construction step is described in detail below.

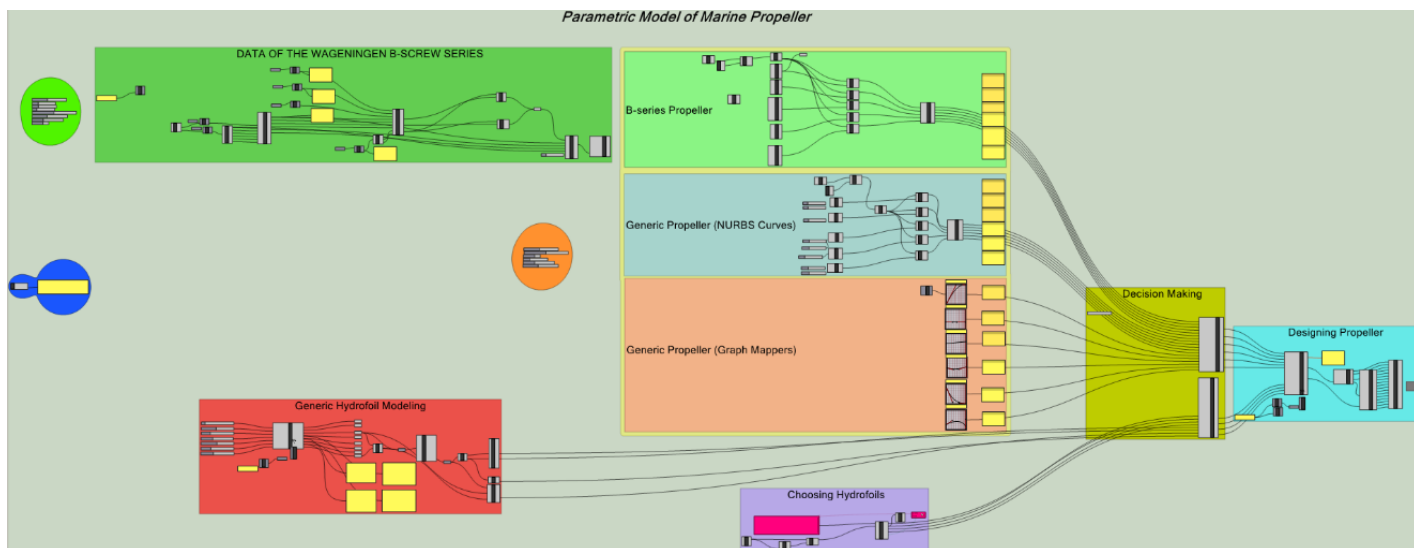


Figure IV.1: Grasshopper Interface of the parametric model

B. Step 1. Parametric Model of a Generic 2-D Hydrofoil

The designing process of a marine propeller begins with a parametric model for a generic 2-D hydrofoil [41], which is shown in the red group of components (*Figure IV.2*).

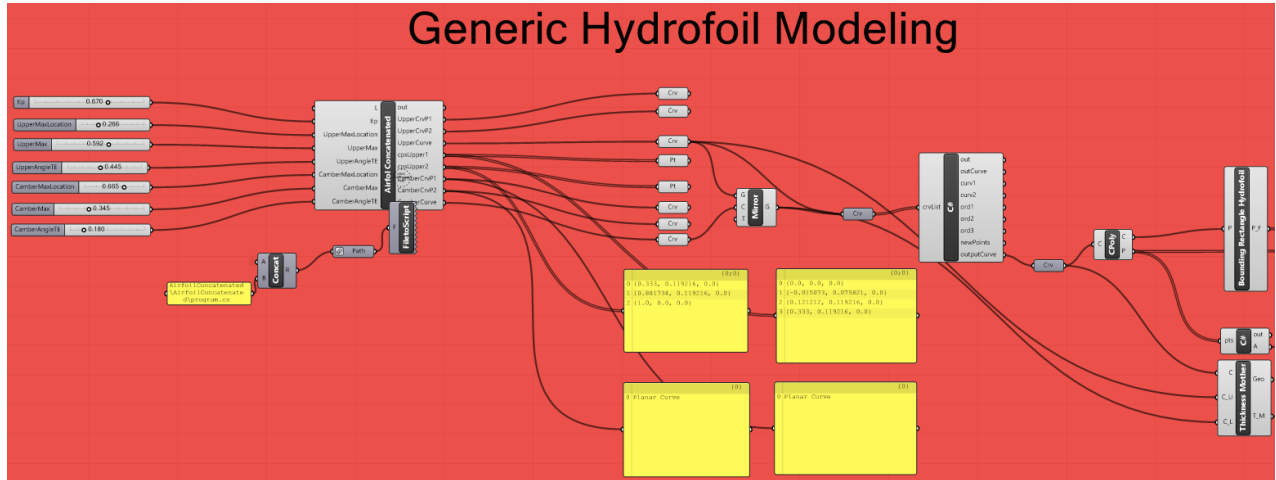


Figure IV.2: Building the Parametric Model for a Generic 2-D hydrofoil inside Grasshopper

This model creates a closed cubic B-Spline curve that represents a hydrofoil, using a set of 8 parameters (*Table IV.1*). Except of the chord's length parameter (L), all the others are defined using nondimensional ratios, so that their values range always from 0 to 1. The chord length is a free parameter and functions as a scale for the hydrofoil geometry.

Table IV.1: Parameters for the Construction of a Generic 2-D Hydrofoil

Nr.	Name	Description	Symbol	Actual Range
1	Length	Length of hydrofoil's chord	L	free
2	Max width	Maximum width of suction side wrt chord	max_z	$\left[\frac{L}{500}, \frac{L}{5} \right]$
3	Camber width	Camber maximum width wrt chord	max_c	$[0, 0.091max_z]$
4	Max-width position	Longitudinal position of suction side's max width	x_{z_max}	$\left[\frac{L}{5}, \frac{7L}{10} \right]$
5	Max-camber-width position	Longitudinal position of camber's max width	x_{c_max}	$\left[0, \frac{3L}{10} \right] + \frac{7x_{z_max}}{10}$
6	Suction-side angle	Suction's side angle at trailing edge wrt chord	a_b	$\left[\arctan\left(\frac{z_{max}}{L - x_{z_max}}\right), 89 \right]$
7	Camber angle	Camber angle at trailing edge wrt chord	a_{b_p}	$[0, a_b]$
8	Tip	Leading edge form factor	tip	$[0.1, 0.9]$

The construction of the parametric model initiates with the definition of four simple Bezier curves, that are employed in the generation of the final cubic B-Spline curve. The upper part of the hydrofoil is consisted of two parts: a simple cubic Bezier curve and a simple quadratic one for the fore and aft part respectively. The first control point of the cubic curve is located at (0,0) while the last one is at (x_{z_max}, z_{max}) . The positions of the two interior control points are controlled by the tip form factor parameter. Specifically, the first point lies on the line segment defined by $(0, 0)$ and $(0, z_{max})$, while the second on the segment defined by $(0, z_{max})$ and (x_{z_max}, z_{max}) .

The camber curve is designed in a similar way using two quadratic simple Bezier curves. The construction of these two curves is equivalent to the curves referring to the hydrofoil suction side with x_{z_max} , z_{max} and a_b being replaced by x_{c_max} , c_{max} and a_{b_p} . A merging algorithm [69] of two B-spline curves into one B-spline curve has been developed. So, the suction side and the camber curve can be represented by one B-spline curve respectively. Thereupon, the pressure side of the hydrofoil is constructed by mirroring the suction side about the camber curve.

Finally, the two curves corresponding to upper and lower part of the hydrofoil have also been merged and as a result a closed B-spline curve is generated, which represents the 2-D hydrofoil model. This method leads us to a cubic B-spline curve without superfluous control points, which is consisted of 12 control points (*Figure IV.3*). The geometrical definition of the 8 parameters is depicted in the figure below.

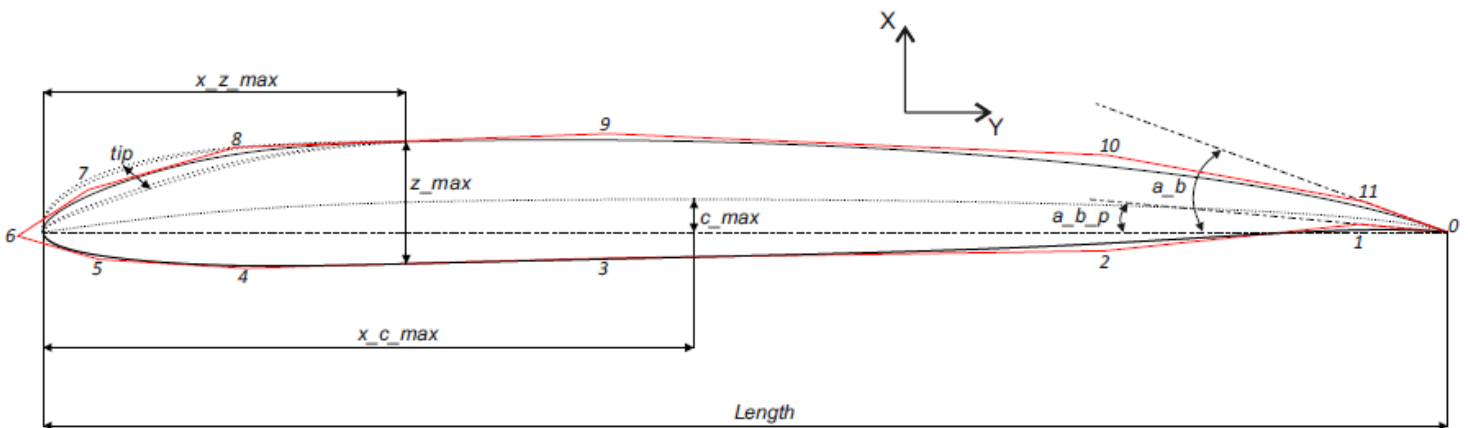


Figure IV.3: 2-D Hydrofoil Model

C. Step 2. Basic Parameters

Thereafter, the user chooses the basic parameters for the construction of a marine propeller, which are described in *Table IV.2*. A different set of basic parameters is used in order to approximate a B-series propeller model.

Number Sliders are used to import numerical values to the parameters. A number slider is a special interface object that allows quick changing of individual numeric values.

Table IV.2: Basic Parameters for the Construction of a Generic Marine Propeller

Basic Parameters					
Name	Description	Generic		B-series	
		Usage	Range	Usage	Range
<i>Diameter</i>	<i>Diameter of the Propeller</i>	✓	<i>Non-negative number</i>	✓	<i>Non-negative number</i>
<i>Blades</i>	<i>Number of Propeller Blades</i>	✓	[2, 7]	✓	[3, 7]
<i>Hub Diameter Ratio</i>	<i>Hub Diameter divided by Propeller Diameter</i>	✓	[0.130, 0.200]	✓	<i>about 0.167</i>
<i>Number of sections</i>	<i>Number of Hydrofoil Sections composing the Blade</i>	✓	[7, 30]	✓	[7, 15]
<i>Rotation X (°)</i>	<i>Angle of Rotation around X-axis</i>	✓	[-90, 90]	✓	[-90, 90]
<i>Right Handed</i>	<i>Direction of Propeller Rotation</i>	✓	<i>True or False</i>	✓	<i>True or False</i>
A_E/A_0	<i>Expanded Area divided by Disc Area (Blade Area Ratio)</i>	✗	✗	✓	[0.3, 1.05]
P/D	<i>Pitch to Diameter Ratio</i>	✗	✗	✓	[0.6, 1.4]

In the *Figure IV.4* the basic parameters, which mentioned above, with their number sliders inside Grasshopper.



Figure IV.4: Basic Parameters with Number Sliders inside Grasshopper

D. Step 3. Radial Parameters

In the third step, a set of radial parameters is defined (*Table IV.3*), which are affecting the blade's geometry along the wingspan. Each parameter is obtained as a ratio of quantities and these are separated in three categories.

Table IV.3: The Categorized Radial Parameters

RADIAL PARAMETERS

Posiotioning Sections		Hydrofoil Shape Changing		Mapping Sectios Control Points	
Name	Description	Name	Description	Name	Description
<i>Sections_Ratio</i>	<i>Defining the Distributuon of the Sections along wingspan</i>	<i>Chord_Ratio</i>	<i>Determing Chord Length of the Section</i>	<i>Pitch_Ratio</i>	<i>Rotation via Pitch Angle</i>
		<i>Thickness_Ratio</i>	<i>Determing Thickness of the Section</i>	<i>Skew_Ratio</i>	<i>Rotation via Skew Angle</i>
				<i>Rake_Ratio</i>	<i>Movement via Rake</i>

In the first category, only the *Sections_Ratio* parameter is contained and is utilized in order to have a desirable section spacing along the wingspan. This parameter is a driver for the rest of the Radial Parameters, because the radial position of the sections follows the distribution of the first graph. The second category of parameters is defining the chord length and the thickness of each hydrofoil section. Using the parameters of the third category, namely the skew angle, the rake and the pitch of the propeller; the hydrofoil sections are mapped into the 3-D space (see *Hydrofoil Mother Concept*).

There are three options (*Figure IV.6*) to create the radial distributions of those parameters. According to the first one (light red group in *Figure IV.6*), the distributions are created with the help of Graph Mappers components. This Grasshopper tool allows the user to remap a set of numbers, which lie on X-axis, using the curve onto the Y-axis. The red vertical lines in *Figure IV.5* are representing the numerical values on X-axis and their number is defined on the previous step with the Number Sections parameter. In our model we choose, as a graphic type, a cubic Bezier curve and six Graph Mappers components are used, which are depicted in the *Figure IV.5*. In this way, different propeller shapes can be generated interactively, by changing the positions of the control points (vide the little circles in the *Figure IV.5*). The problem with this method is that the control points can change only by mouse movements, so by designing the propeller in this way, the parametric model cannot be a part of an optimization process. Therefore, the parametric model has been enhanced with a second option.

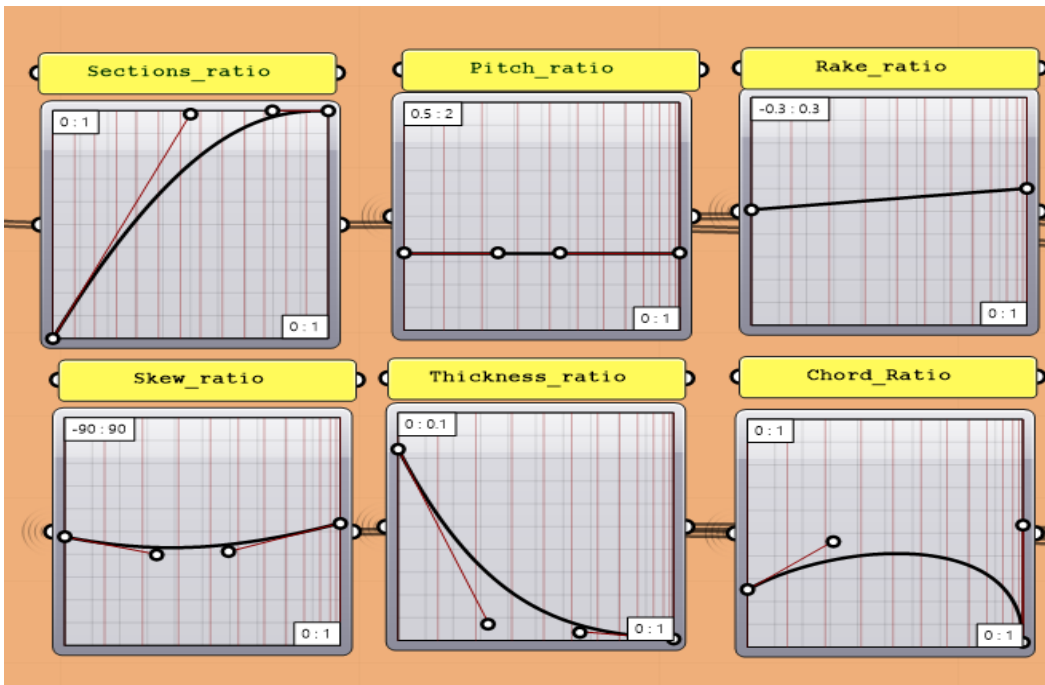


Figure IV.5: Generating Distributions of Radial Parameters with Graph Mappers

In the same manner with the Graph Mappers methodology, a set of cubic Bezier curves has been utilized for the generation of the radial distributions mentioned above. The difference between these options, is that in the second one, the graphs have been constructed with a set of in-house C# codes (light blue group in *Figure IV.6*) and the positions of the control points can be changed by a set of number sliders. The purpose of the code is to create the radial distributions of the parameters to follow a propeller logic. In the same time, the code can offer a great deal of variety in the resulting shapes, which is necessary in an optimization algorithm.

The third option (light green group in *Figure IV.6*) is related with the creation of Wageningen B-series Graphs, which will be discussed in the next Chapter. The path, that the algorithm will follow depends on the user's decision (*Figure IV.7*).

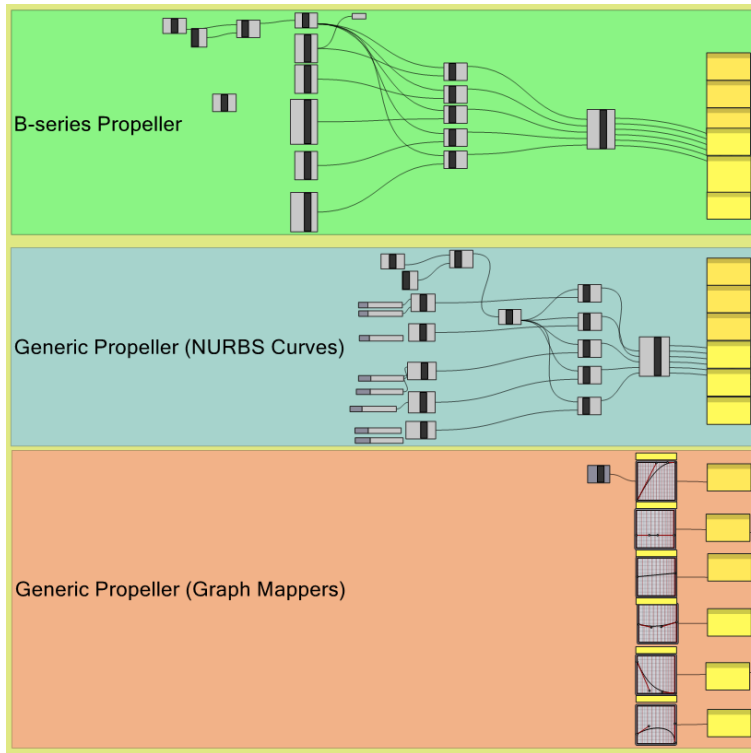


Figure IV.6: Three options to create the Distributions of the Radial Parameters

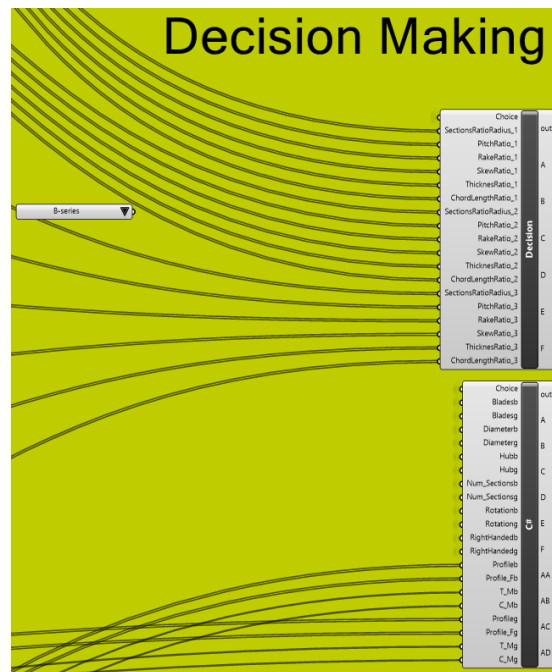


Figure IV.7: Decision Making

E. Step4. Blade Cage

A set of in-house C# codes components have been developed aiming to build the control cage of the blade. In our model, the “*Hydrofoil Mother Concept*” has been adopted. In order to create the control cage of the blade, a set of control points has to be mapped via the radial parameters (*Table IV.3*). The control points of all sections have been generated from the control points of the initial hydrofoil, hence the name “*mother*”.

As mentioned before, each section is formed by a control polygon contained by 12 control points. The control points of each section should be transformed as described in “*Hydrofoil Mother Concept*”. In the *Figure IV.8* the control points are shown with red mark and the blue lines connecting these points define the blade cage. The order of the sections is from the root of the blade to its tip. So, the first section is referring to the section at the root of the blade and the last section is located at the tip of it.

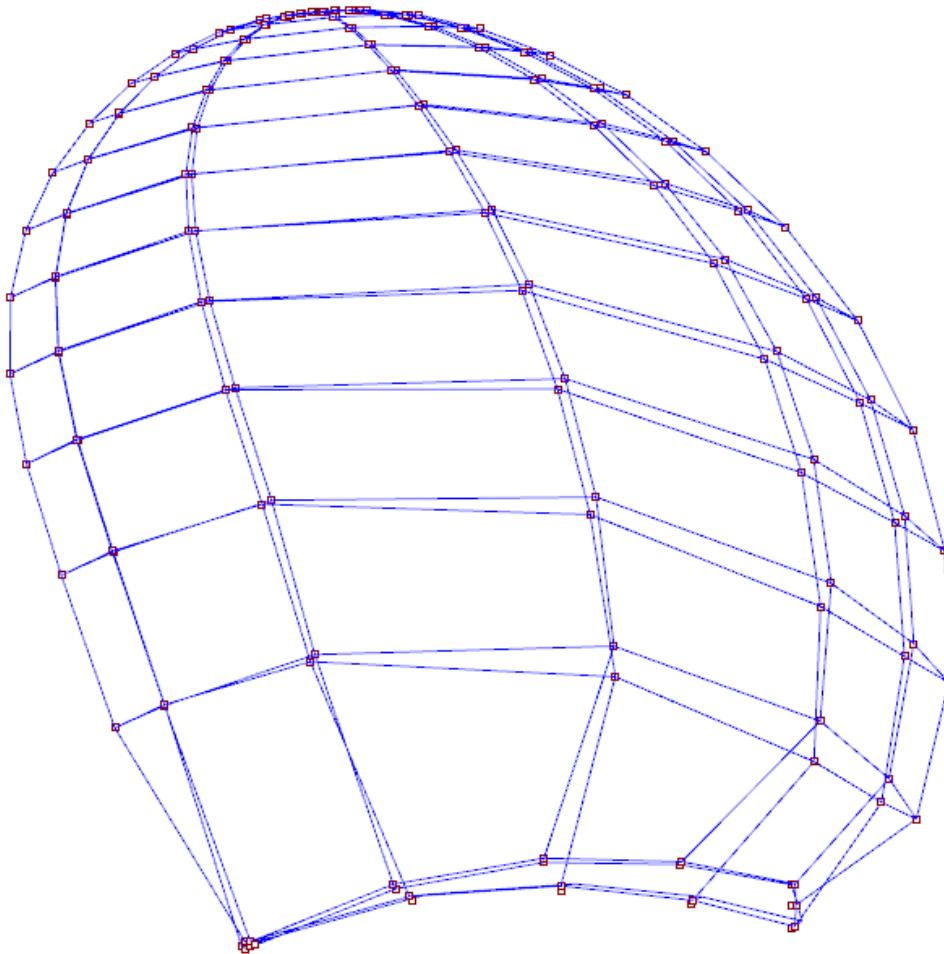


Figure IV.8: Control Points Mapped into the 3-Dimensional Space

For the generation of the blade cage an additional shape of Hydrofoil is utilized and it is also generated from the “Hydrofoil Mother”. It is called “Bounding Hydrofoil” because of its shape (*Figure IV.9*).

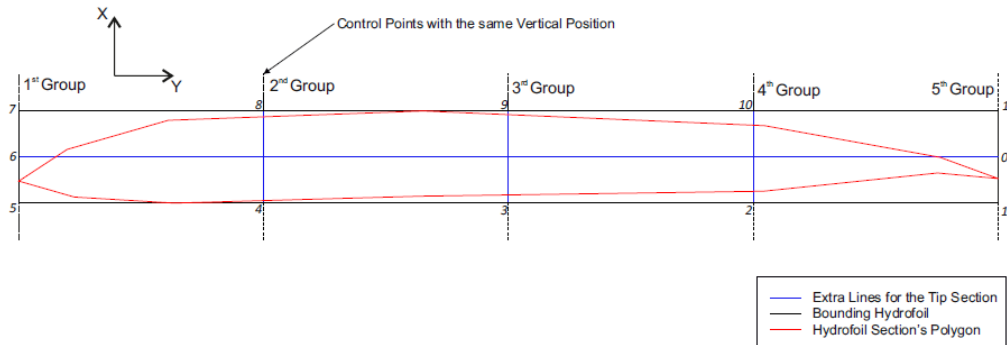


Figure IV.9: Bounding Hydrofoil Model

This hydrofoil is used on the tip and on the root of the blade. These areas were an issue for our designing process due to the following reasons.

- Achieve the proper smoothness of the fillet area through the control cage
- The blade and the hub of a propeller are extended in different directions
- Closing of the control cage at the tip, but not with a single point

The fillet area of the propeller implemented by using two, close to each other, adapted bounding hydrofoils sections (*Figure IV.10*). The only purpose of this area is to connect the hub with the blades by maintaining the mechanical strength of the propeller, resulting an hydrodynamically inactive zone. Therefore, thicker sections (rectangle) are employed in the generation of the fillet. The fillet radius is also a parameter of the model and it can change from the radial distance between the first and the second section.

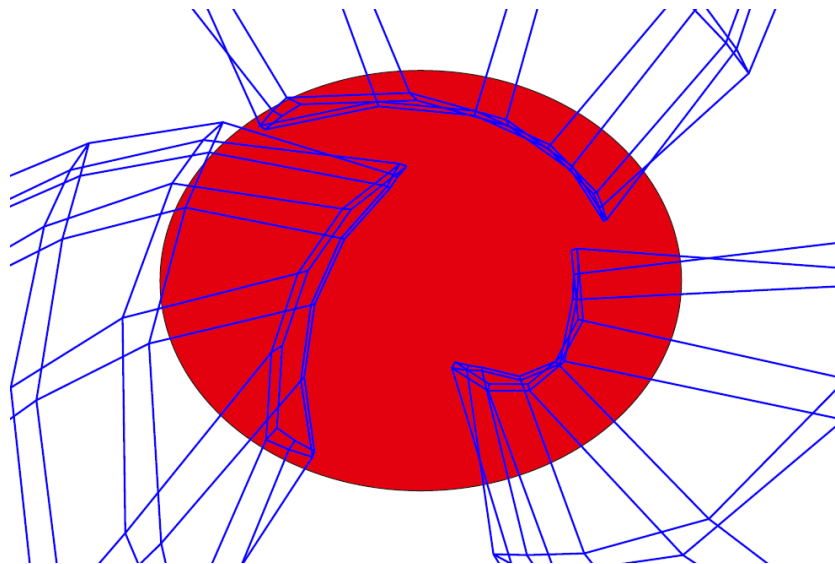


Figure IV.10: Two Nearby Sections for the Generation of the Fillet Area

The construction of a single watertight T-spline surface of a propeller is a difficult task, because as mentioned before the blade and the hub of a propeller are extended in different directions. In consequence, the control points of the first section must be located in proper positions, so the hub can receipt the blades of the propeller. The way that the root section points adjusted is explained below. These 12 control points are separated into 5 groups depending on their vertical position on the expanded view of the section (*Figure IV.9*). For each group of points, one circle is defined, in which one point of the group is lying in the circle mentioned above. The rest of the points, which don't belong to the topology of the created circles, are being moved, in order each group of points to belong to their adjacent circle (*Figure IV.11*).

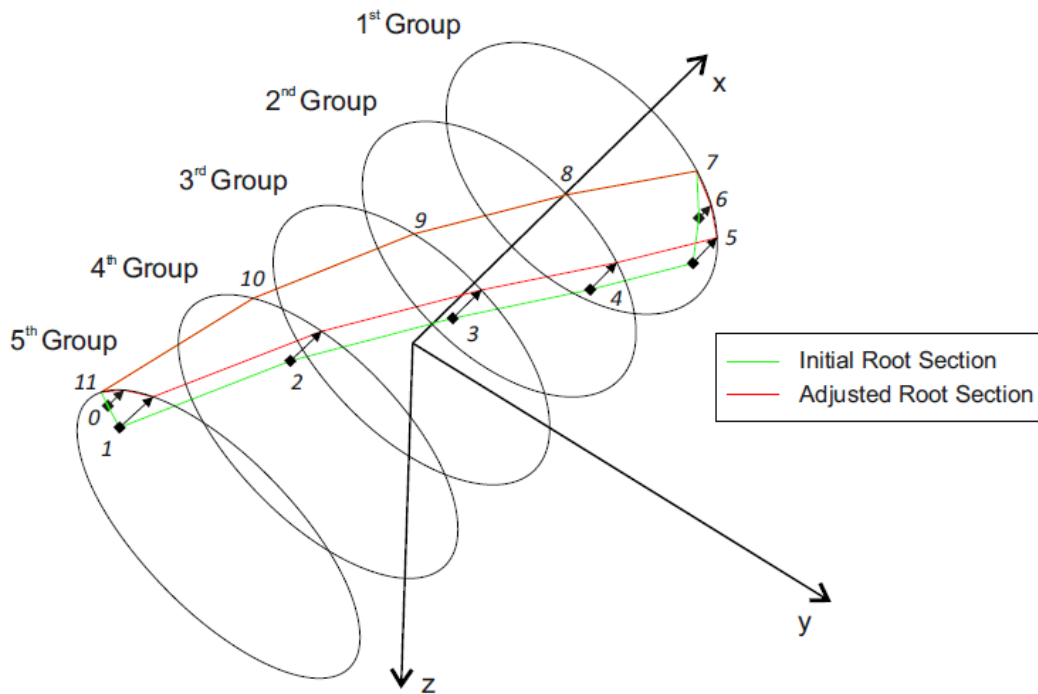


Figure IV.11: Adjustments at the Control Points of the First Section

The closing of the blade's tip was always a trouble for the propulsor designers. In our model, we use an adapted bounding hydrofoil (see: *Figure IV.12*, red section) for the representation of the tip, because it's not desirable, the cage to close on a single point. For this adaptation some extra lines are used as illustrated in *Figure IV.9*. This set of control points are also lying topologically on a rectangular grid. Additionally, the last section should be very thin and long in order to approximate the tip of the blade.

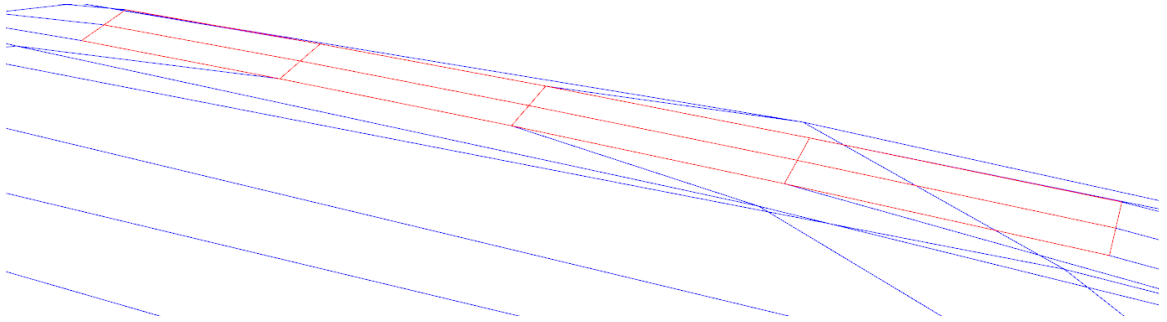


Figure IV.12: Last Section of the Blade Cage

Once, the Blade cage is constructed, the rest blades of the propeller are generated by rotating the initial cage via the following angle.

$$\varphi_i = \frac{2\pi(i-1)}{Z}, i = 1, 2, \dots, Z \quad (IV.1)$$

Where Z is the number of Blades and i is representing the serial number of the rest blades. The Figure IV.13 shows three blade cages of a marine propeller.

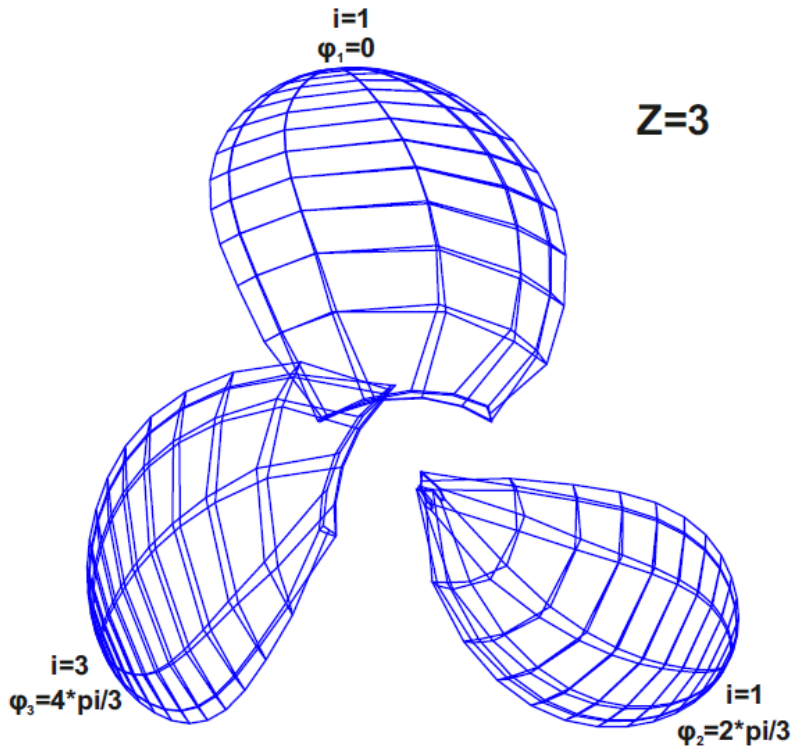


Figure IV.13: Three Blade Cages of a Marine Propeller

F. Step 5. Hub Cage

Subsequently, the parametric model creates a proper hub, which will be connected with the blade cage. The shape of the hub is in a simplicity a cylinder that closes on the front with a cap. The approximation of the hub is attained by using a set of polygons in a row, which are connected to each other. The closing of the hub (cap) is created by decreasing the radius and the sides of the polygons. The polygons that are utilized, are divided in two categories.

The polygons of the first category are illustrated in the *Figure IV.14* with black mark and they are called initial polygons. The construction process begins with these polygons, that are five, as many as the groups of the control points at the root section. In each of these groups corresponds a circle (*Figure IV.14*). The radius of those circles is equivalent with the radius of the polygons, that initially produced and their centers are located in the same position. So, the root section points are also part of the initial polygons. The longer the length that the blades are covering over the perimeter of the circle, the smallest the number of polygon sides and conversely.

Therefore, the initial number of polygons is declared exclusively from the thickness of the root section and from the number of blades. In our model, the number of the initial polygon sides can receive only specific values, which are the following: 8, 12, 16, 24, and 32. Numbers greater than 32 were rejected, because they make the propeller model more complex. Those values have been chosen, because they are multiple of number 4 and in this model (last section-polygon should be a 4-sided polygon.)

The second category of polygons is shown in *Figure IV.14* with red color. When viewed from astern and along the shaft axis, five of those polygons are in front of the blades and another five are behind. The order of construction in each side of the second category polygons is from the main body of the hub to its edge. These polygons are regular and the first section-polygon has as many sides as the initial polygons and the last one is a 4-sided polygon. The distances of the polygons and the reduction of their sides is accomplished in a way, that the symmetry of the hub surface is kept, hence the chosen numbers multiple of four.

In order to achieve C^2 continuity and a smooth surface, the helical line of the blade should also continue in the hub. This is achievable by maintaining the value of the pitch angle at the root and by rotating the regular polygons via this pitch angle.

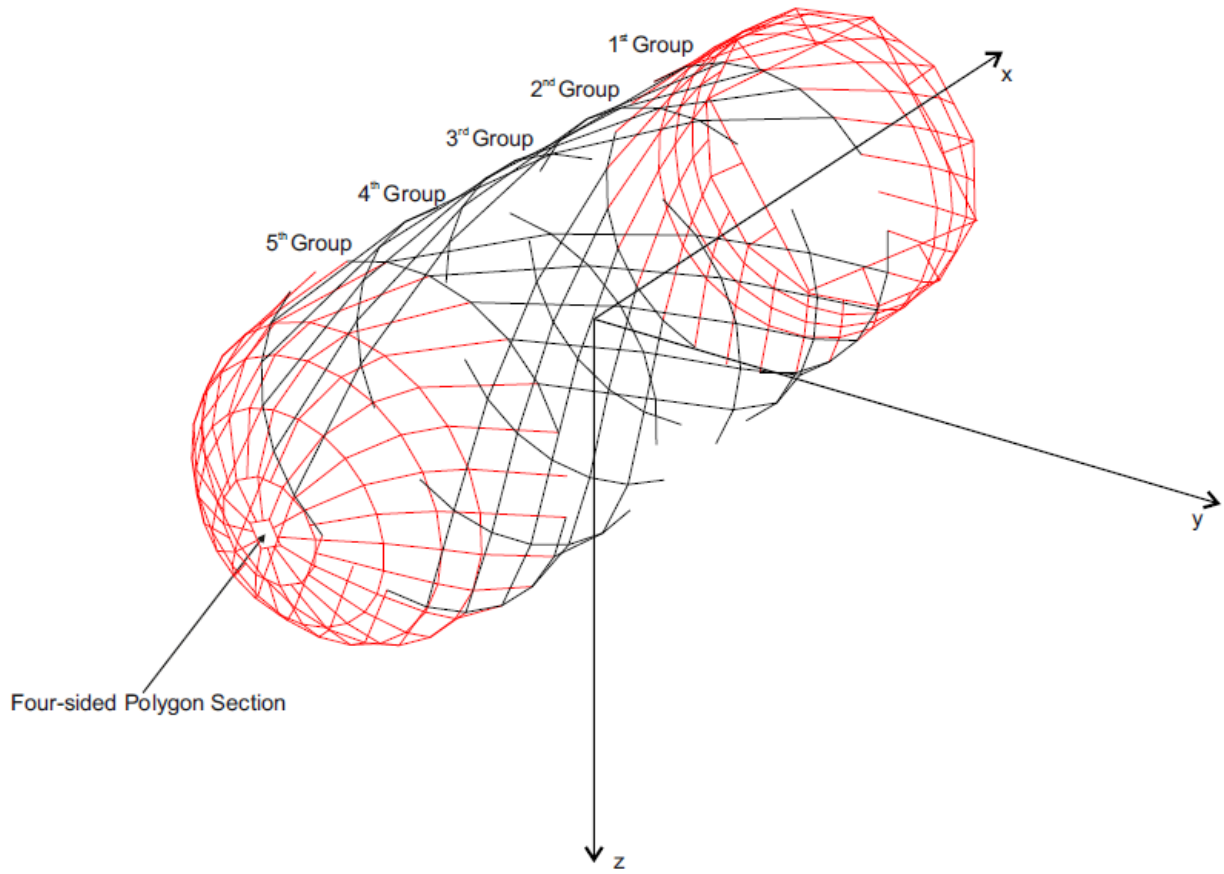


Figure IV.14: Construction of the Hub Cage Using a set of Polygons in a Row

G. Step 6. Control Cage of the Marine Propeller

The process of generation of a marine propeller T-spline control cage inside Grasshopper is shown in the *Figure IV.15*.

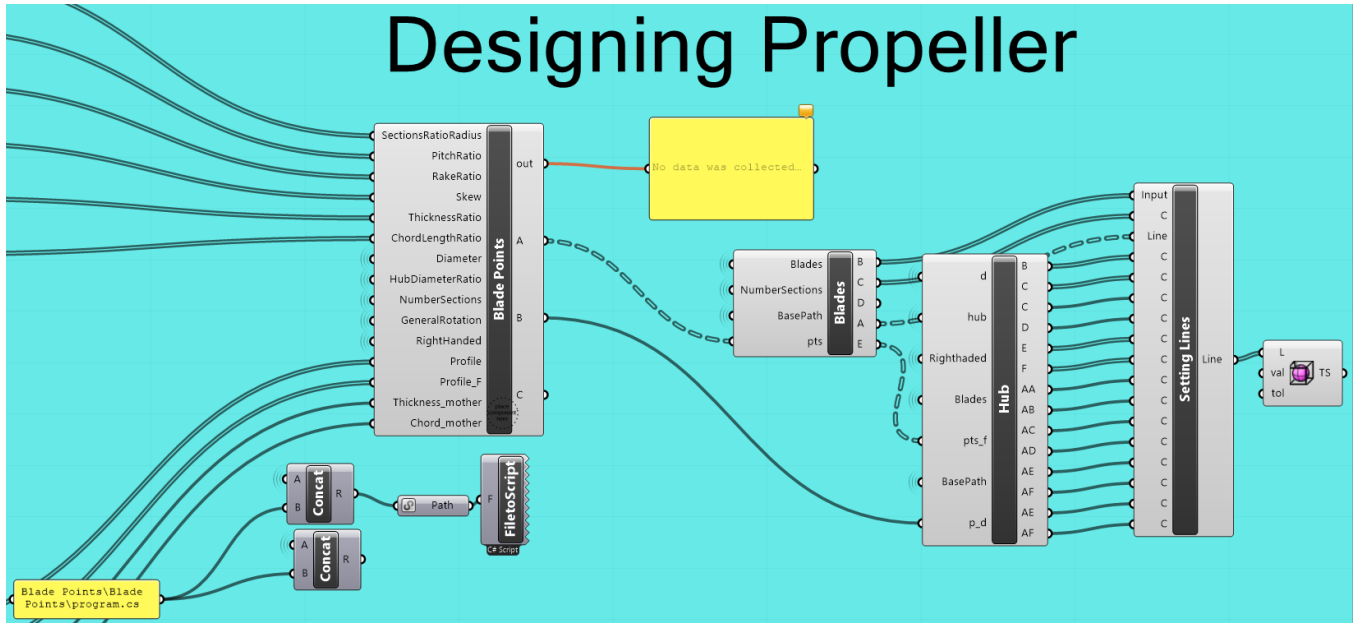


Figure IV.15: Process of Designing a Marine Propeller with T-splines inside Grasshopper

The T-spline surface is constructed in the last component of the above figure, which uses the command of Rhino: `tsFromLines` [70]. The `tsFromLines` command provides a way to create complex surfaces, that are closer to the desired final shape than the default primitives. The workflow of creating a T-spline surface from lines is the following:

- a. Creating the control polygon
- b. Connecting control polygon edges correctly (see: *Figure IV.17*)
- c. Running the `tsFromLines` command

The *Figure IV.16* shows the star points of the T-spline mesh of the propeller and they are located at the hub, tip and the fillet area. The star points or the extraordinary points are interior T-mesh vertices that are not T-junctions and whose valence do not equal four. In last two figures of the section, a curvature analysis has been accomplished inside Rhino using a Gaussian surface [71] [72] and an evaluation of marine propeller surface smoothness and continuity using the Zebra [73] command.

As a last remark, we need to mention here that the processing time, i.e., the time required for building the control cage of a model instance, requires around 1-2 secs on an entry level standard PC. Further to this, an additional time of up to 30 secs might be required for the T-spline surface construction, especially if a smooth tessellation is required for visualization purposes.

The superiority in smoothness for a T-spline surface model is also extended to the basis used for its representation. Specifically, the single cubic T-Spline surface modeling the propeller geometry exhibits overall C^2 continuity, limited to G^1 in the vicinity of extraordinary points (illustrated in *Figure IV.16*).

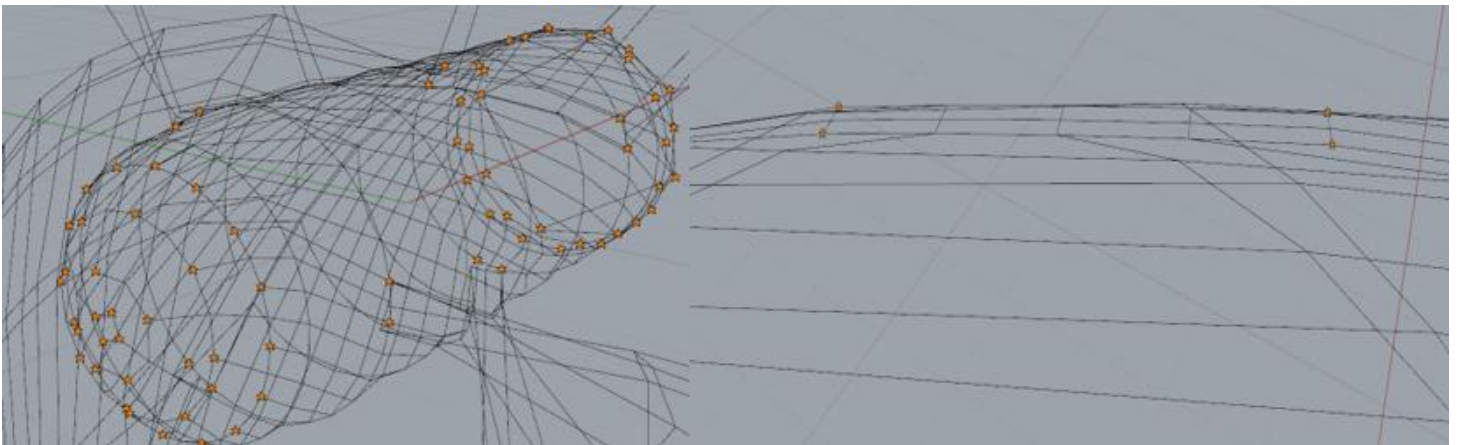


Figure IV.16: Star points on the T-mesh of a Marine Propeller

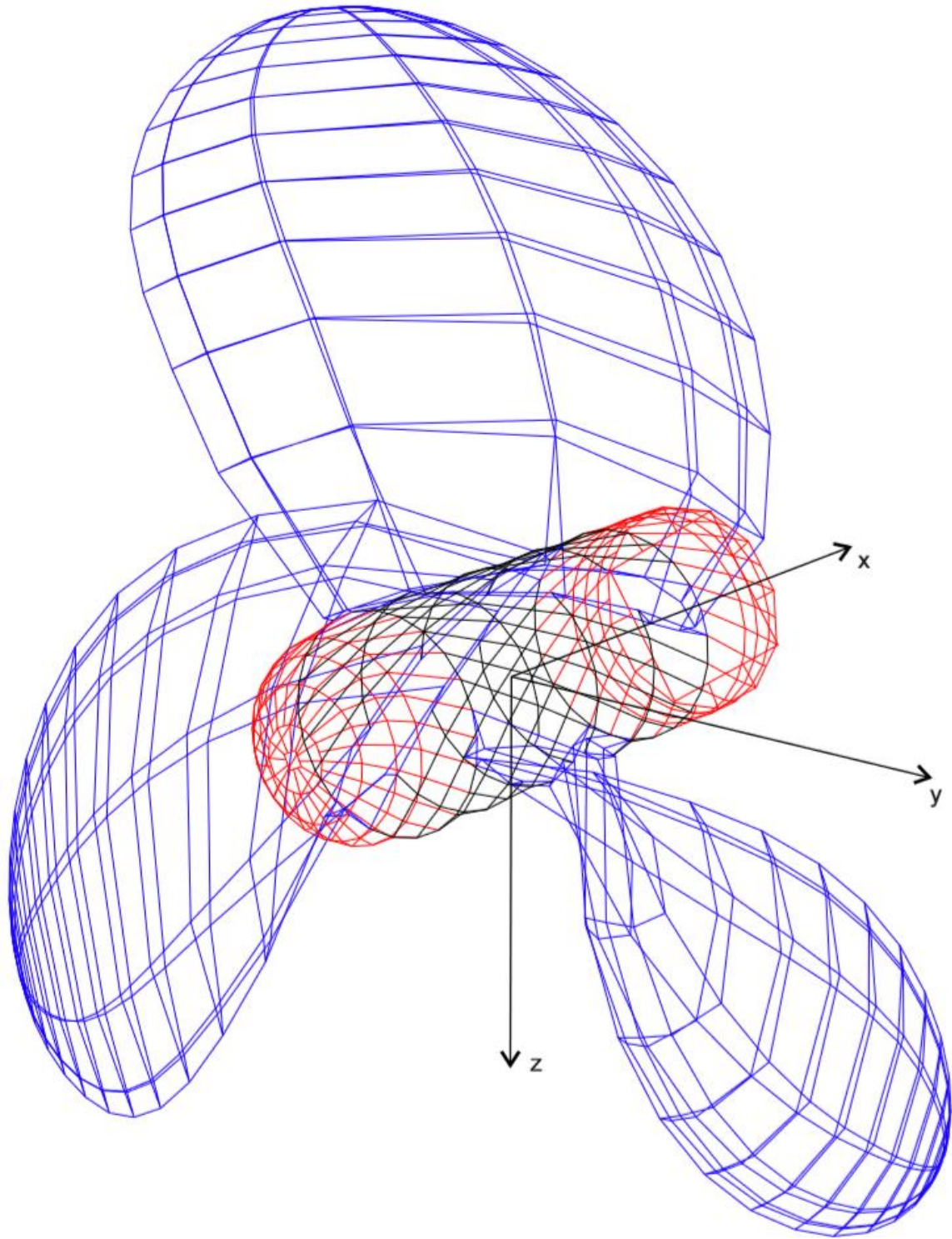


Figure IV.17: Entire Control Cage of Marine Propeller

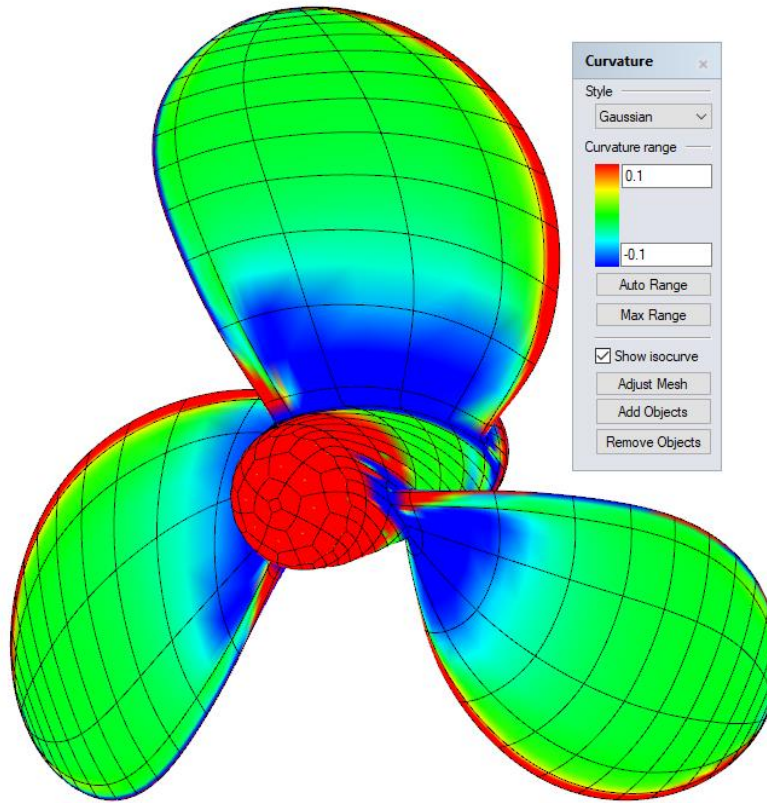


Figure IV.18: Marine Propeller Curvature Analysis using Gaussian Surface

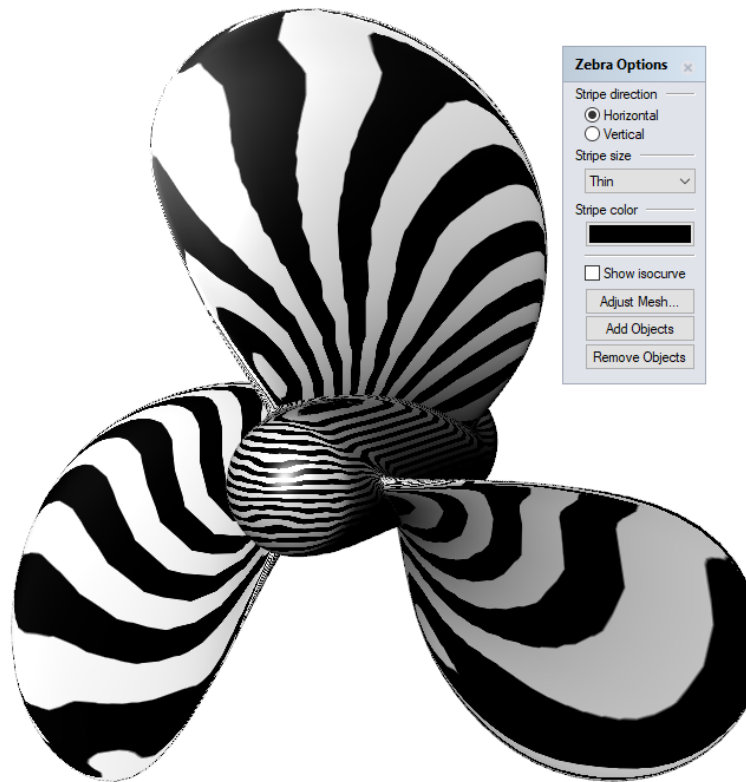


Figure IV.19: Evaluation of Marine Propeller Surface Smoothness and Continuity using Zebra

V Model Instances

A. Wageningen B-series

Various generic shapes are produced by this parametric model. This variety is also including real marine propeller models and it is feasible to a user of the modeler to receive any propeller he desires, by applying to the algorithm the proper data input. As part of this parametric model, an approximation on the geometry of the Wageningen B-Screw Series [74] using T-Splines technology has been carried out.

Wageningen B-Screw Series [75] is the most extensive and one of the oldest propeller series, due to its simplicity is also one of the most widely used. The B-series was originally presented in a set of papers presented by Troost [76] referred to as the “Troost series”. The number of blades of B-Series propellers ranging from 3 to 7, blade area ratios ranging from 0.30 to 1.05 and the pitch to diameter ratio values are between 0.6 to 1.4. The main feature of B-series is, that the propellers have constant pitch with the exception of 4-bladed propellers. The XY-ZZ notation is used to denote the propeller series, where (X=B) denoting the “B-series”, Y is the number of blades and ZZ is the blade area ratio (A_E/A_0).

Table IV.2 shows the parameters, that are utilized for the construction of a B-series propeller, which is defining completely the geometry of the B-series model. The challenging part of building a B-series blade was not only the creation of the appropriate radial functions of the parameters (Table IV.3), but also the drawing of a hydrofoil section, which corresponds to the B-series definition.

The below Figure depicts the process of generating the cloud of the B-series propeller points inside Grasshopper, as shown in Figure V.2.

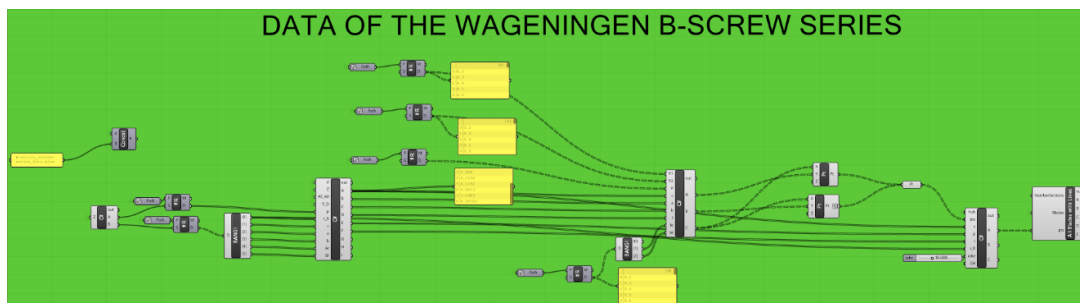


Figure V.1: Generating the Cloud of the B-series Propeller Points inside Grasshopper

The overall geometric properties of the original Wageningen B-series are known from [77]. So, calculating the formulas given by a cloud of points lying on the surface of the required blade is generated. The resulting points are part of 8 sections located at $0.2 r/R$, $0.3 r/R$, ..., $0.9 r/R$ (*Figure V.2*), and each section consists 40 points.

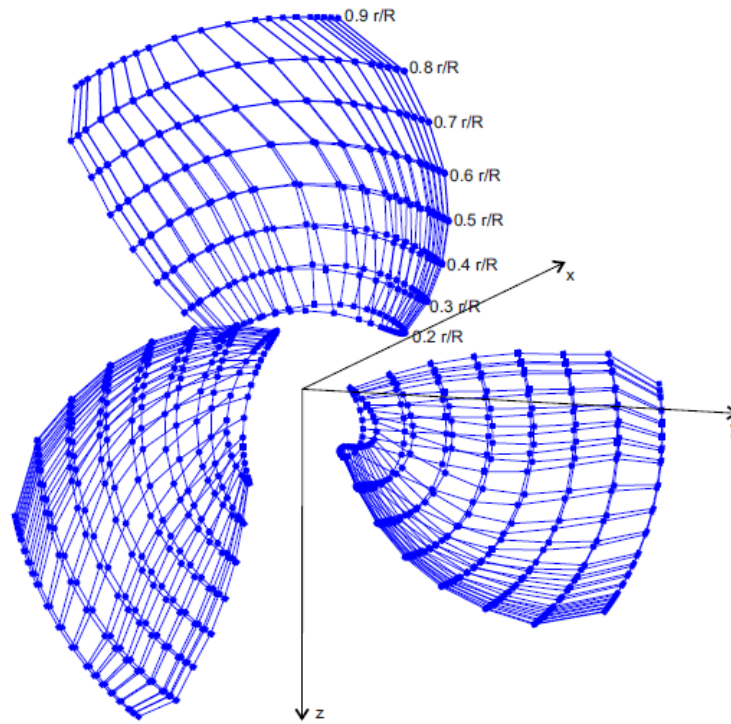


Figure V.2: A Cloud of a B-series Propeller Points

In consequence, depending on the position of the resulting points, the distributions of the radial parameters are set up. In order to approximate the 2-D profiles of a B-series blade, we develop an evolutionary algorithm [78] by utilizing the parametric model for the generic hydrofoil. The outcoming sections are very close to the B-series definitions (*Figure V.3*). A generic platform for the application of Evolutionary Algorithms is provided within Grasshopper, as shown in *Figure V.4*, and it is named Galapagos [79]. This Grasshopper component can optimize a shape, so that it best achieves a user defined goal. The parameters (*Table IV.1*), except of the chord length of the hydrofoil, are the genes of the algorithm. Galapagos can try these 7 numerical values in various combinations, and each combination, or genome, produces a unique hydrofoil shape. Counting the distances between the B-series points and the outcoming hydrofoil shapes (curves in *Figure V.3*), the maximum value of these distances is saved. The goal or fitness function of the algorithm is to minimize the aforementioned maximum distance.

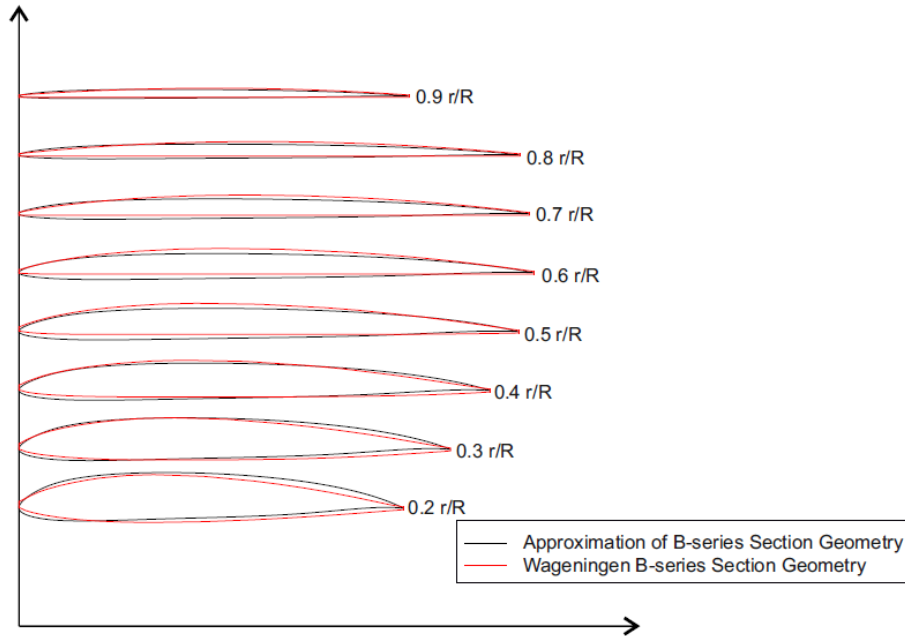


Figure V.3: Comparison of B-series sections with a set of Approximated Sections

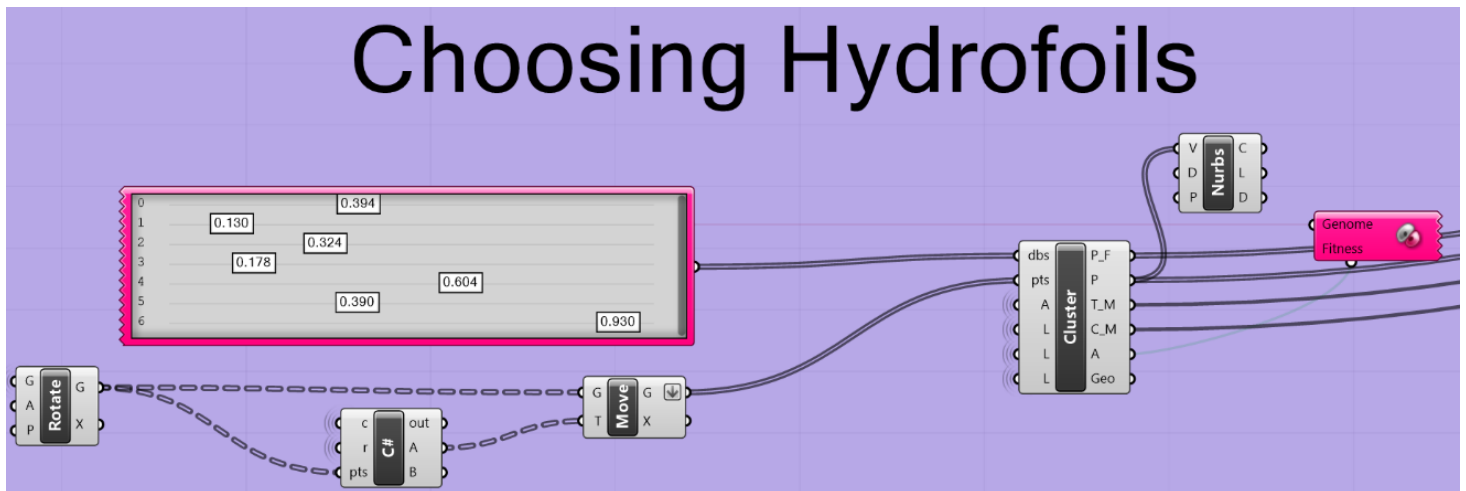


Figure V.4: Generating a B-series Hydrofoil Using Galapagos Generic Platform

In this way, by applying an evolutionary algorithm, an appropriate “*Hydrofoil Mother*” is produced. Thereafter, all the required data are set for the continuation of the designing process. Following the same design strategy with the construction of a generic propeller cage, which is described in the previous chapter, the creation of a B-series propeller has been accomplished.

Table V.1 shows the extent of the series in terms of a blade number versus blade area ratio matrix. The 20 blade area-blade number configurations, which are shown below, gives a summary of the series. For completeness purposes Figure V.5 illustrates 4 geometric model instances of B-series propellers.

Table V.1: Summary of the Wageningen B-Screw Series

Blade Number (Z)	Blade Area Ratio A_E/A_0								
2	0.30								
3	0.35		0.50		0.65		0.80		
4	0.40		0.55		0.70		0.85 1.00		
5	0.45			0.60		0.75			1.05
6	0.50			0.65		0.80			
7	0.55			0.70		0.85			

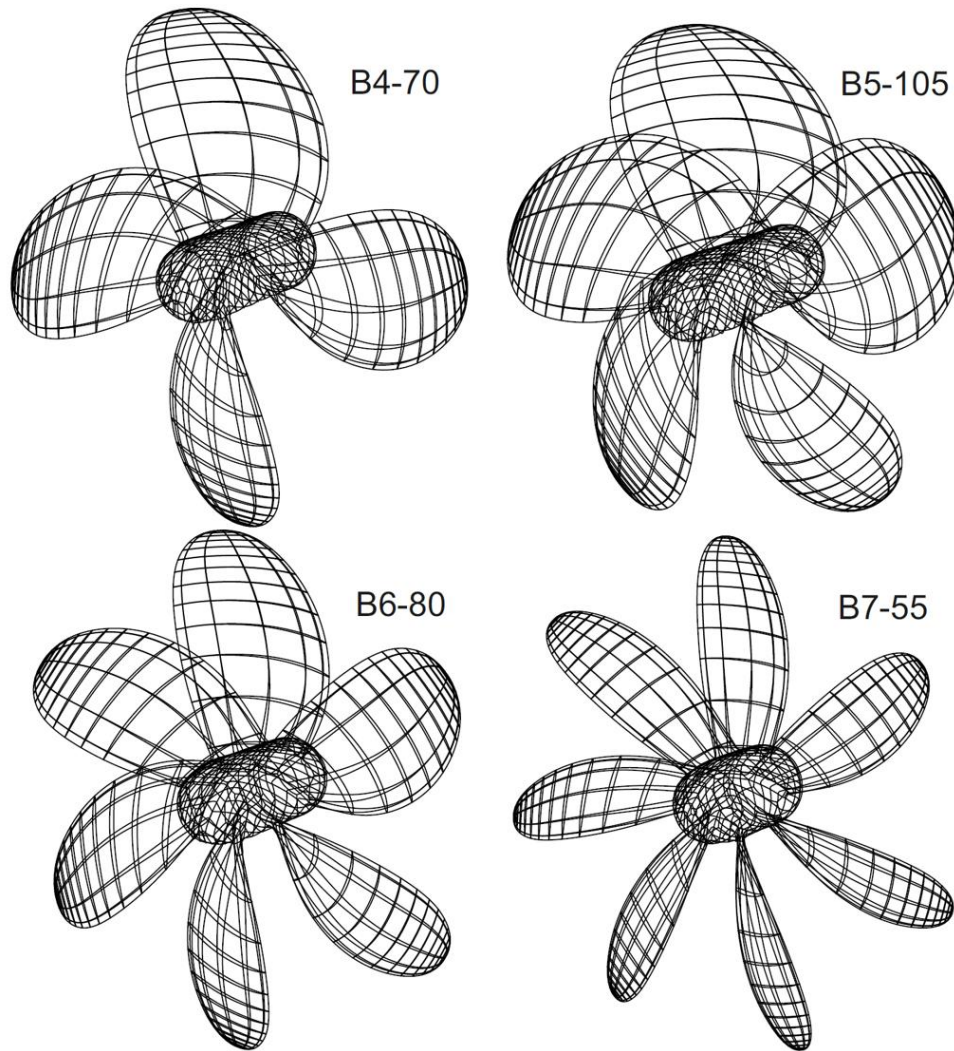


Figure V.5: Four Geometric Model Instances of B-series propeller

B. OpenProp

For benchmarking purposes, a prototype marine propeller 3D model, obtained from OpenProp software package, was used. Openprop is an open-source software package used for the design and analysis of marine propellers and horizontal axis turbines. The standard required input for generating a 3D marine propeller model in OpenProp includes the number of blades, the propeller & hub diameter, the template hydrofoil’s mean line and thickness, along with distributions for chord length, blade thickness, pitch, skew and rake along the radial direction; see default OpenProp geometry input parameters in *Figure V.6*. The OpenProp 3D model corresponding to this default set of parameters is depicted in *Figure V.7*.

The screenshot shows the OpenProp v3.3.4 software interface. It is divided into several sections:

- Specifications:**
 - Number of blades: 3
 - Rotation speed: 200
 - Rotor diameter (m): 2
 - Required thrust (N): 25000
 - Ship speed (m/s): 5
 - Hub diameter (m): 0.4
 - Fluid density: 1000
 - # radial panels: 20
 - # chordwise: 20
- Blade Design Values:**

r/R	c/D	Cd	t/D	Skew	Xs/D
0.2	0.16	0.008	0.0329	0	0
0.3	0.1812	0.008	0.0281	0	0
0.4	0.2024	0.008	0.0239	0	0
0.5	0.2196	0.008	0.0198	0	0
0.6	0.2305	0.008	0.016	0	0
0.7	0.2311	0.008	0.0125	0	0
0.8	0.2173	0.008	0.0091	0	0
0.9	0.1807	0.008	0.006	0	0
0.95	0.1388	0.008	0.0045	0	0
1	0.001	0.008	0	0	0
- Inflow Profile Values:**

r	Va/Vs	Vt/Vs
- Options:**
 - Propeller
 - Turbine
 - Hub
 - Chord optimiza...
 - Viscous forces
 - Optimization pl...
 - Geometry plots
 - Performance c...
- Airfoil type:**
 - Meanline type: NACA a=0.8
 - Thickness type: NACA 65A010
- Ducted Propeller:**
 - Thrust Ratio: 1
 - Duct section drag: 0.008
 - duct D / prop D: 1
- Non-dimensional Parameters:**
 - J = V/nD = 0.75
 - L = omega*R/V = 4.18879
 - CT = 0.63662
 - KT = T/(rho*n^2*D^4) = 0.140625
- Tools:**
 - Filename: DefaultPropeller
 - Buttons: Load, Save, Run OpenPr...

Figure V.6: Default Geometry-Input Parameter Values for OpenProp

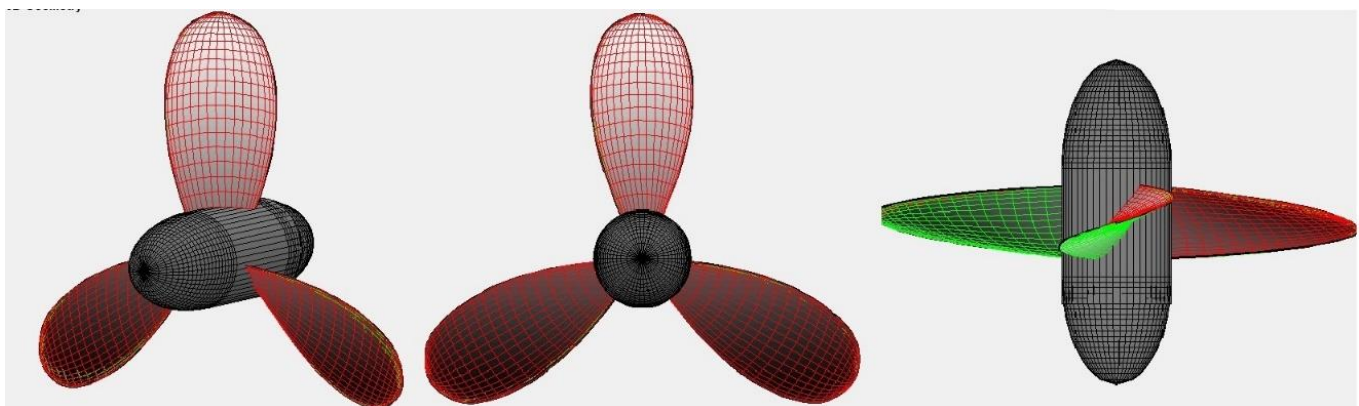


Figure V.7: Default OpenProp 3D Propeller Model

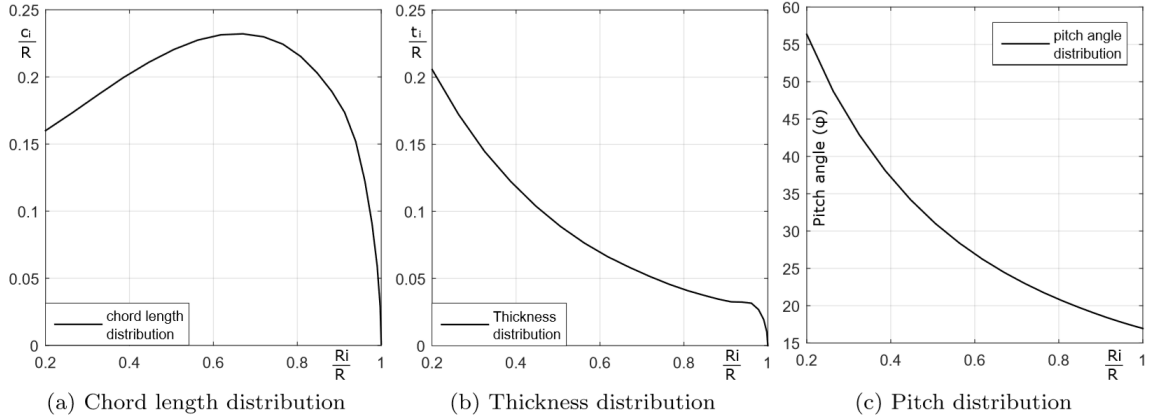


Figure V.8: Default OpenProp Radial Distributions of Basic Geometrical Quantities

In sequel, the use of the parametric model in approximating the default OpenProp 3D propeller model is demonstrated. The comparison is limited to the propeller blade, which is the most complex and decisive geometrical element for the propeller’s performance and the process has as follows:

- i. The basic parameter values, Diameter (D), number of blades and the number of sections are set to 2, 3 and 16 respectively.
- ii. The parameters described in *Table IV.1* are employed in approximating the NACA65A010 hydrofoil profile.
- iii. The distributions for chord length, thickness and pitch angle depicted in *Figure V.8* are approximated using the corresponding mechanisms built in the parametric model (see: “*Step 3. Radial Parameters*”).
- iv. The T-spline corresponding 3D propeller model is then automatically generated; see *Figure V.9* and consists of 884 control points (including the hub component).

The first thing that is assessed, is the approximation level achieved by the parametric model when compared to the OpenProp prototype surface blades. To this end, points from 20 sections belonging to one OpenProp blade are generated and compared them against the corresponding sections of this thesis model. In *Figure V.9*, the corresponding results for the T-spline generated blade is presented. It is easy to see that the overall deviation is larger and that there is a region (red colored region), in with the surface deviations exceed 1cm. This average deviation is around 3.8×10^{-3} m. A detailed comparison of deviations in all generated sections are included in *Table V.2*.

Specifically, for each section we present the average and max deviations achieved by each parametric model when compared to OpenProp’s prototype blade. Deviations are presented both as distance values (Av. Dev., Max. Dev.) and percentages of the section’s chord length (Av. Per, Max. Per). Finally, each section’s chord length, as approximated by each model, is also included in the comparison.

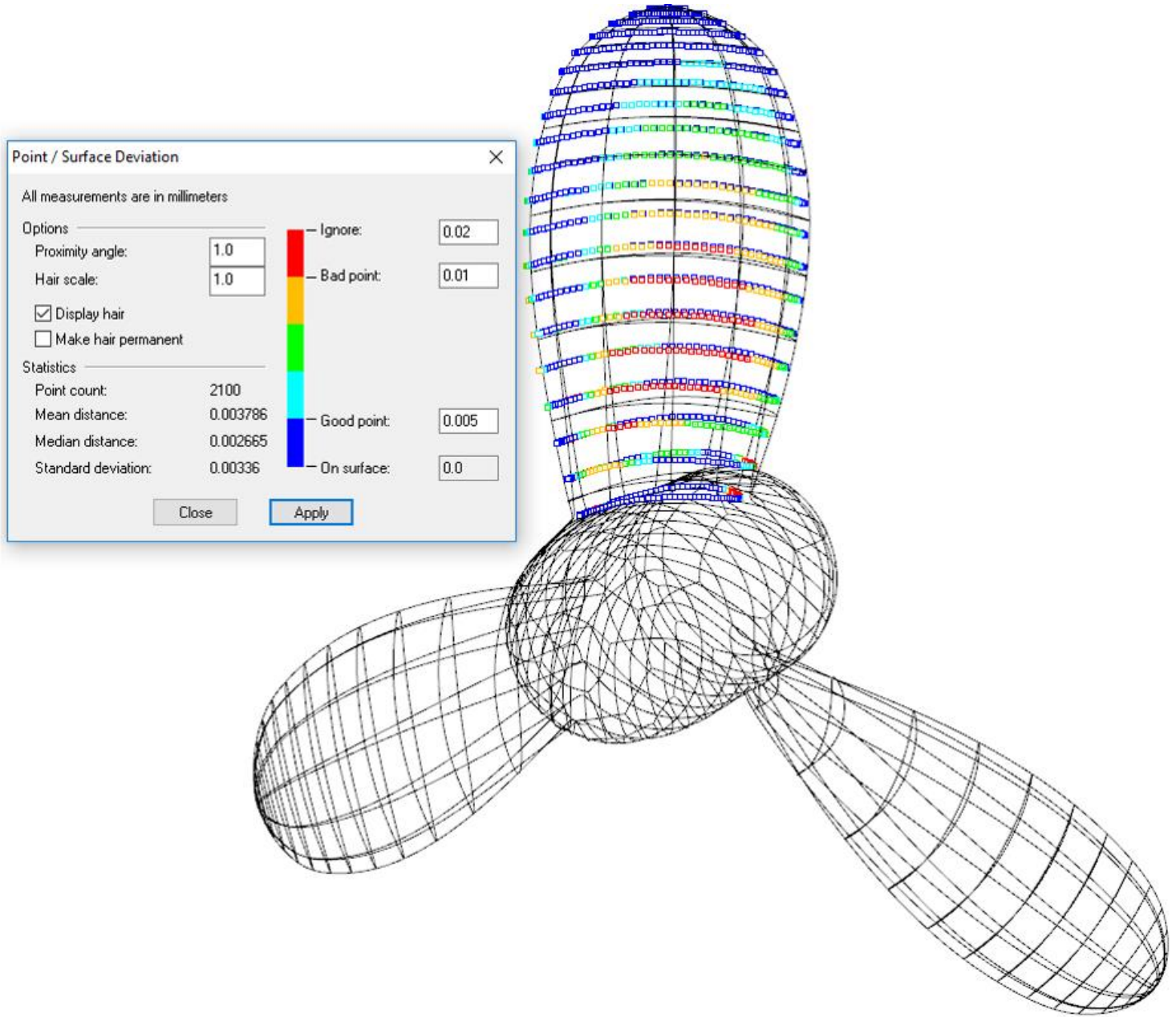


Figure V.9: Point-surface Deviations between OpenProp Generated Point Set and the corresponding T-spline Blade Surface

Table V.2: Comparison of OpenProp sections with T-spline Parametric Model

Section #0 (0.200 r_R)					Section #1 (0.263 r_R)				
Av. Dev.	Max. Dev.	Chord (L)	Av. Per.	Max. Per.	Av. Dev.	Max. Dev.	Chord (L)	Av. Per.	Max. Per.
0.00702	0.01978	0.34671	2.03%	5.70%	0.00563	0.01332	0.34086	1.65%	3.91%
Section #2 (0.325 r_R)					Section #3 (0.387 r_R)				
Av. Dev.	Max. Dev.	Chord (L)	Av. Per.	Max. Per.	Av. Dev.	Max. Dev.	Chord (L)	Av. Per.	Max. Per.
0.00522	0.01212	0.36299	1.44%	3.34%	0.00575	0.01236	0.385176	1.49%	3.21%
Section #4 (0.447 r_R)					Section #5 (0.506 r_R)				
Av. Dev.	Max. Dev.	Chord (L)	Av. Per.	Max. Per.	Av. Dev.	Max. Dev.	Chord (L)	Av. Per.	Max. Per.
0.00574	0.01236	0.40535	1.42%	3.05%	0.00537	0.01206	0.422928	1.27%	2.85%
Section #6 (0.563 r_R)					Section #7 (0.618 r_R)				
Av. Dev.	Max. Dev.	Chord (L)	Av. Per.	Max. Per.	Av. Dev.	Max. Dev.	Chord (L)	Av. Per.	Max. Per.
0.00496	0.01246	0.436891	1.13%	2.85%	0.00453	0.01216	0.446082	1.02%	2.72%
Section #8 (0.670 r_R)					Section #9 (0.720 r_R)				
Av. Dev.	Max. Dev.	Chord (L)	Av. Per.	Max. Per.	Av. Dev.	Max. Dev.	Chord (L)	Av. Per.	Max. Per.
0.00403	0.011452	0.449488	0.91%	2.55%	0.00366	0.01107	0.446361	0.82%	2.48%
Section #10 (0.766 r_R)					Section #11 (0.808 r_R)				
Av. Dev.	Max. Dev.	Chord (L)	Av. Per.	Max. Per.	Av. Dev.	Max. Dev.	Chord (L)	Av. Per.	Max. Per.
0.00336	0.01081	0.43633	0.77%	2.48%	0.0031	0.0100	0.41953	0.74%	2.41%
Section #12 (0.847 r_R)					Section #13 (0.882 r_R)				
Av. Dev.	Max. Dev.	Chord (L)	Av. Per.	Max. Per.	Av. Dev.	Max. Dev.	Chord (L)	Av. Per.	Max. Per.
0.00282	0.00917	0.39728	0.71%	2.31%	0.0024	0.0087	0.37045	0.69%	2.34%
Section #14 (0.913 r_R)					Section #15 (0.939 r_R)				
Av. Dev.	Max. Dev.	Chord (L)	Av. Per.	Max. Per.	Av. Dev.	Max. Dev.	Chord (L)	Av. Per.	Max. Per.
0.00231	0.00896	0.337891	0.68%	2.65%	0.00213	0.00883	0.294328	0.71%	3.00%
Section #16 (0.961 r_R)					Section #17 (0.978 r_R)				
Av. Dev.	Max. Dev.	Chord (L)	Av. Per.	Max. Per.	Av. Dev.	Max. Dev.	Chord (L)	Av. Per.	Max. Per.
0.00177	0.00707	0.23812	0.74%	2.97%	0.00194	0.00581	0.175485	1.11%	3.31%
Section #18 (0.990 r_R)					Section #19 (0.998 r_R)				
Av. Dev.	Max. Dev.	Chord (L)	Av. Per.	Max. Per.	Av. Dev.	Max. Dev.	Chord (L)	Av. Per.	Max. Per.
0.0028	0.00499	0.112106	2.03%	4.45%	0.002658	0.005091	0.04909	5.42%	10.39%

The approximating accuracy, at this point of development, is not reaching the desirable accuracy, but the smoothness exhibited by the parametric model with respect to both its representation bases and surface quality, makes it certainly attractive for a propeller-design optimization environment powered by an IsoGeometric Analysis Boundary Element Method solver for the assessment of blades' performance. Furthermore, the small number of control points, which normally leads to a small number of degrees of freedom (number of unknowns) for the linear system appearing in the context of the BEM formulation, is an additional and equally important benefit.

C. Special Designs of Marine Propellers

This section shows the ability of the parametric model in producing a big variety of generic propeller shapes. *Figure V.10* illustrates four marine propeller models and the basic parameters for the construction of each model are shown in *Table V.3*.

These four models were selected, because of their properties. The properties, that each propeller model possess are described, as follows.

- i. The first model is a two-bladed propeller. Two-bladed propellers are not usually used in marine propeller engineering. Furthermore, the hub diameter of this model is taking the maximum value of the parameter, as shown in *Table IV.2*.
- ii. The Model B is a highly skewed propeller with three blades. The advantages and disadvantages of highly skewed propellers compared with conventional propellers are examined at [80].
- iii. The Model C is a very thin propeller with four blades. Due to the propeller's thickness, a few sides of polygons are employed to build the hub as explained in "*Step 5. Hub Cage*". Also, the number of the sections of the blade cage is 7, hence the small number of control points (448), that define the T-spline surface. Using few blade sections leads normally to a small number of control points and to a small number of degrees of freedom (number of unknowns) for the linear system appearing in the context of the BEM formulation.
- iv. The last model is a left-handed propeller, which is consisted of 1940 control points. Further to this, 30 sections (maximum value of this parameter) are employed for the generation of the blade cage. A big number of blade sections can lead to a better approximation of an existing propeller model.

Table V.3: Properties of the Four Generic Propeller Models of Figure V.10

	Model A	Model B	Model C	Model D
Diameter (m)	1	1	1	1
Hub Diameter Ratio	0.250	0.200	0.167	0.167
RightHanded	True	True	True	False
Number of Blades	2	3	4	5
Number of Sections	11	15	7	30
Number of Control Points	584	848	448	1940

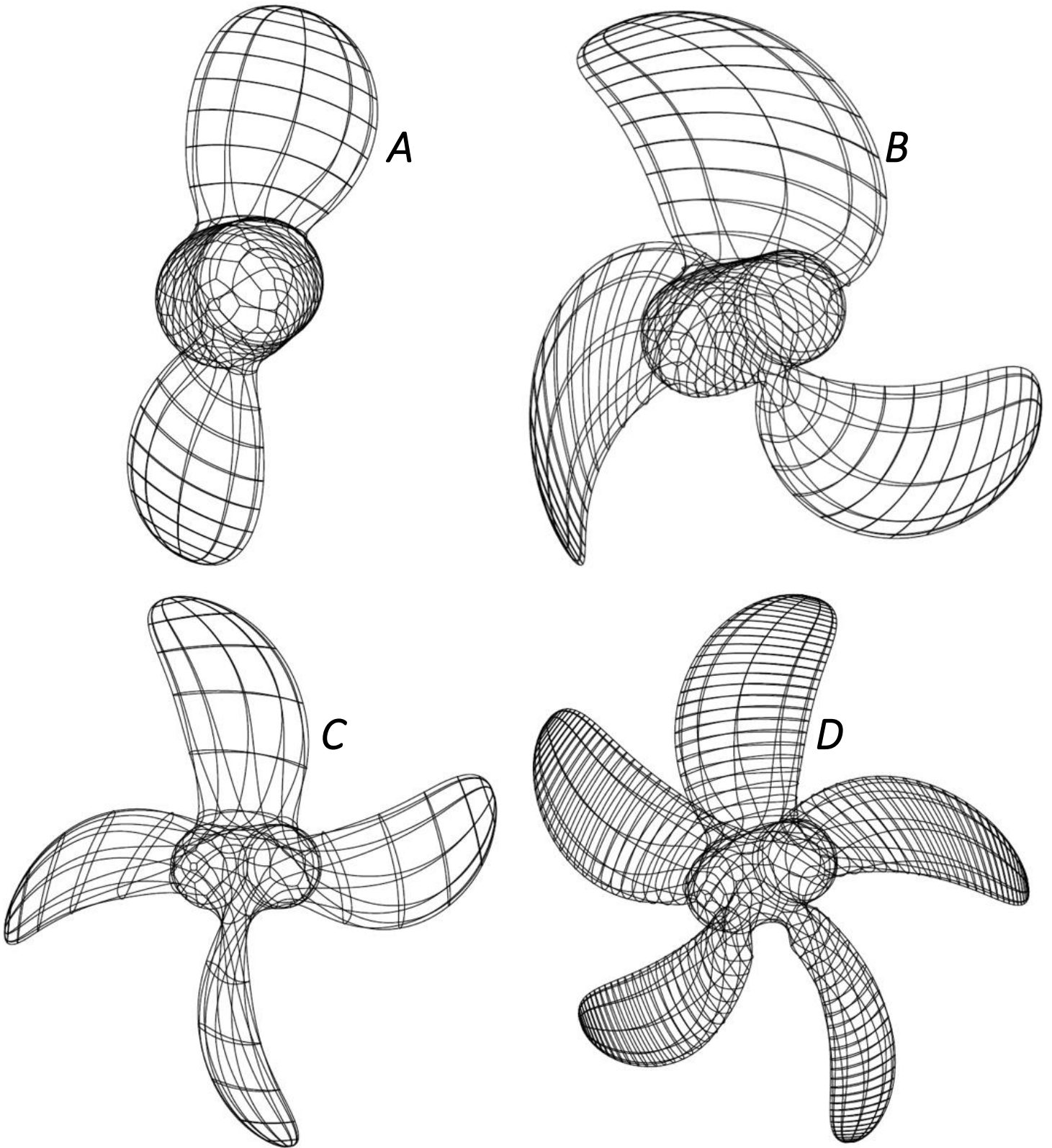


Figure V.10: Four Generic Propeller Model Instances

VI Conclusions and Future Work

This work is focused on the creation of a geometric parametric model, capable of generating various marine propellers model instances represented by a single watertight T-spline surface, which are suitable for engineering analysis following the IsoGeometric Analysis paradigm.

Further to this, it has been demonstrated that the parametric model can quickly and automatically produce valid geometric representations of marine propellers, based on a small set of geometrically and physically meaningful parameters. A subset of these parameters is firstly used in the construction of a template hydrofoil profile, which is later copied and transformed in the span-wise direction on the basis of a series of distribution laws defined by the remaining part of parameters. These distribution laws describe the scaling, rotation and translation of the template profile and specifically correspond to chord length, thickness, pitch, rake and skew propeller distributions.

Also in this thesis, the parametric model's approximation capacity in reconstructing existing marine propellers have been tested. Initially, an approximation on the geometry of Wageningen B-Screw Series, one of the most widely used propeller series has been accomplished. Following this approximation, a benchmark case has been examined using the OpenProp's default propeller model in comparison with the presented T-spline model. The quantitative results, that came up from this comparison are encouraging, but still ways of enhancing the model's approximating accuracy needs to be investigated more.

So, combining the increasing interest in designing the optimal propeller shape, this parametric model seems to possess the required properties, i.e., robustness, automatic and quick instance generation, for their introduction to an automatic design optimization framework.

Finally, the aforementioned parametric model constitutes part of the work carried out in a research project entitled "Shape Optimization of Lift and Thrust generating surfaces with the aid of IsoGeometric Analysis" and the ultimate test for the model will be performed via the project-specific isogeometric analysis propeller solver, which is currently under development. This computational solver will investigate in depth the appropriateness of the model for engineering analysis, in an isogeometric context, and simultaneously will further check its performance in covering the feasible design space in the context of shape optimization.

Appendix A: Propeller Model Utilized

In all chapters of this thesis except of the chapter “*Model Instances*”, a specific model of a marine propeller is illustrated, in order for the reader to understand better the modeling process and not to be confused with a big variety of shapes. This specific model is a B3-60 propeller and is illustrated inside Rhino in *Figure A.1* and the values of the Basic Parameters, that assigned in the model are shown in *Table A.1*.

Table A.1: Basic Parameters for B-60 Propeller

Diameter (mm)	250
AE_A0	0.6
Blades	3
P_D	1
Hub Diameter Ratio (unitless)	0.167
Number of sections	15
Rotation X (°)	11
Right Handed	True

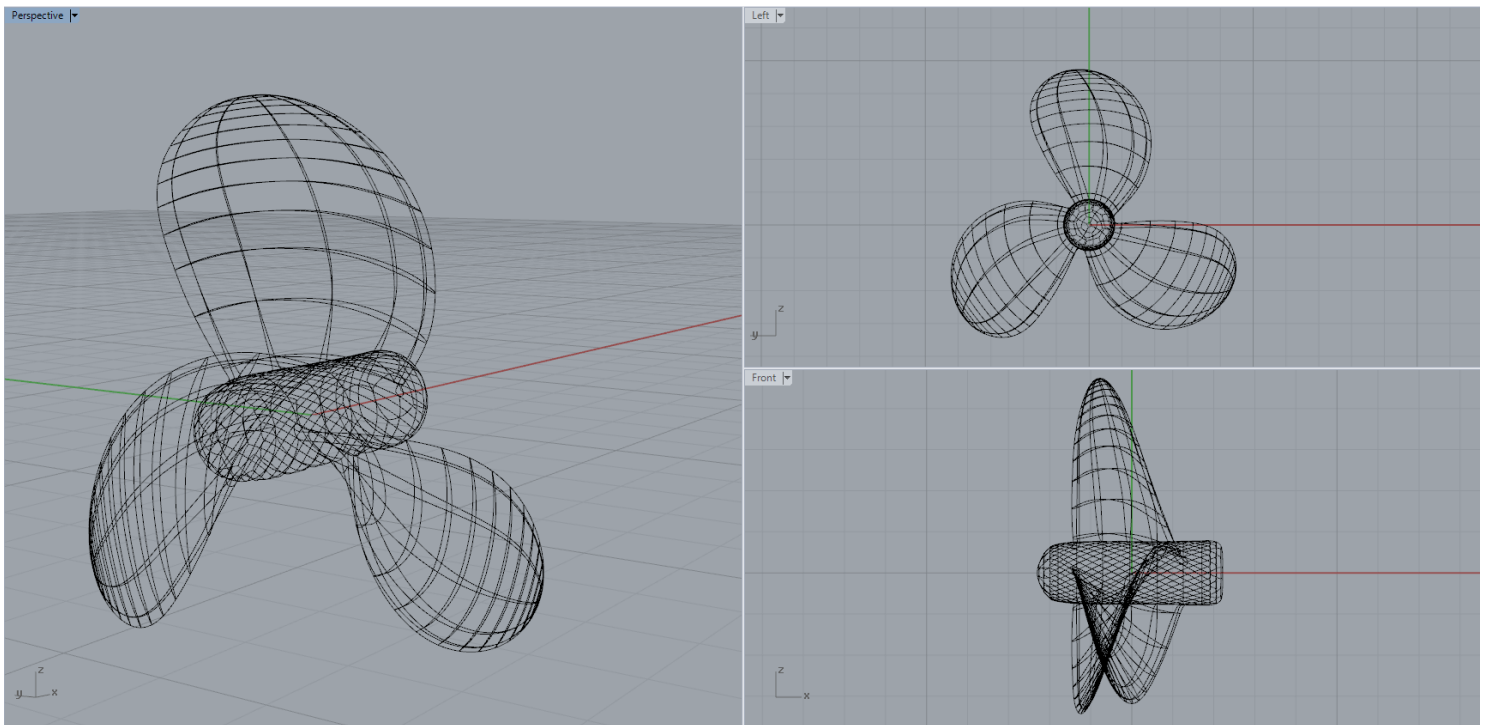


Figure A.1: B3-60 Propeller inside Rhino

The hydrofoil section mother, that is used for the construction of the above propeller is shown in *Figure A.2* and the parameters of this section with their values are shown in the *Table A.2*. These values were exported by the evolutionary algorithm, as described in “*Wageningen B-series*” section.

Table A.2: Hydrofoil Mother Parameters with their Values for B3-60 Propeller

<i>max_z</i>	<i>0.130</i>
<i>max_c</i>	<i>0.324</i>
<i>x_z_max</i>	<i>0.178</i>
<i>x_c_max</i>	<i>0.604</i>
<i>a_b</i>	<i>0.390</i>
<i>a_b_p</i>	<i>0.930</i>
<i>tip</i>	<i>0.394</i>



Figure A.2: Hydrofoil Section Mother for B3-60 Propeller

In *Figure A.3*, the distributions of the radial parameters of the B3-60 propeller are depicted. In the horizontal axis the values are ranging from 0 to 1 and in the vertical axis the parameters are taking their real-unit values. Therefore, the graph of skew angle (measured in degrees) differs in length, from the others.

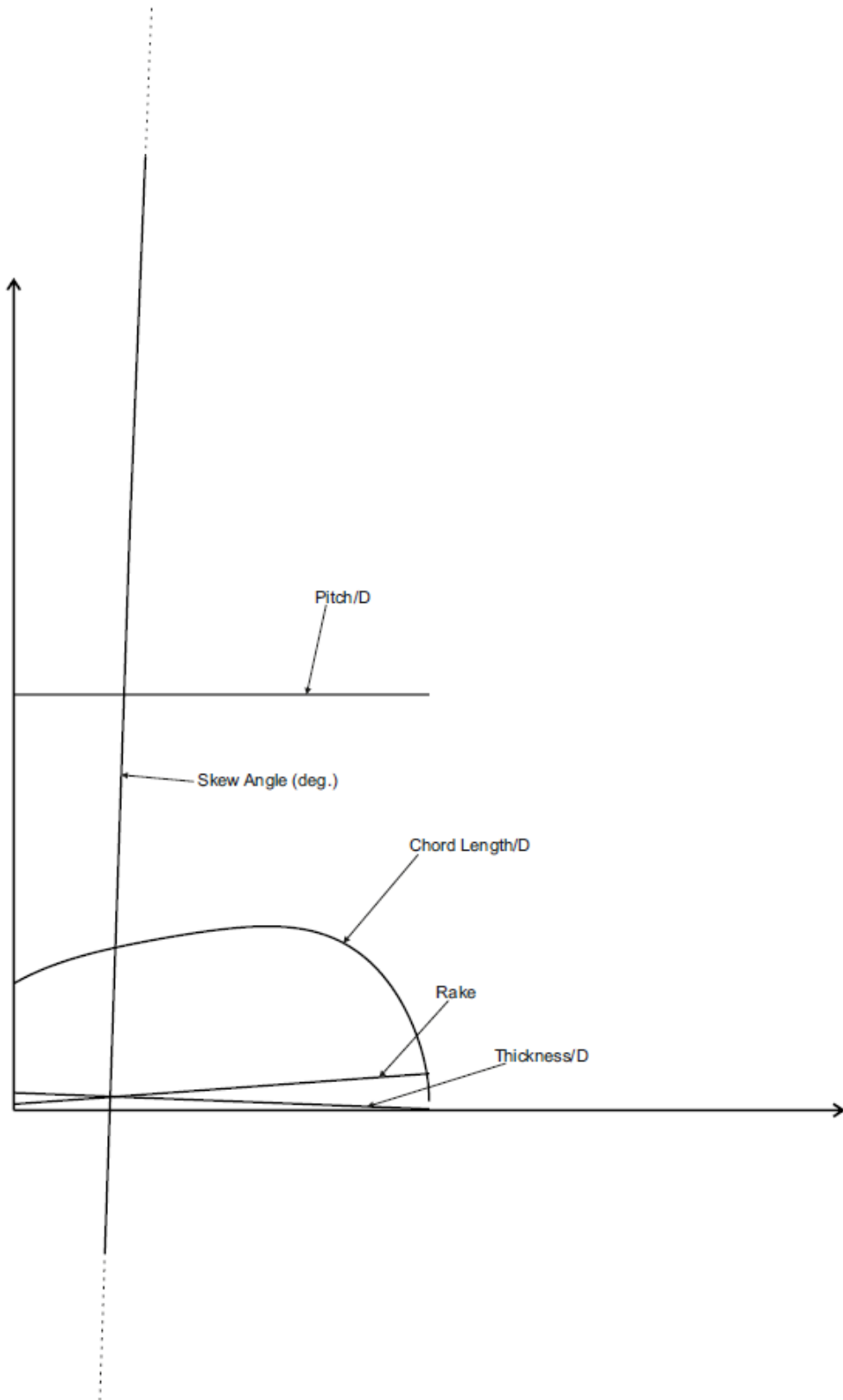


Figure A.3: Graphs of Radial Parameters for B3-60 Propeller

Appendix B: Wageningen Screw B-series

The overall geometric properties of the original Wageningen B-series are shown in *Table B.1*.

Table B.1: Dimensions of Wageningen B-propeller series

Dimensions of four-, five-, six- and seven bladed B-screw series					
r/R	$\frac{c_r}{D} \frac{Z}{A_E/A_0}$	a_r/c_r	b_r/c_r	$\frac{S_r}{D} = A_r - B_r Z$	
				A_r	B_r
0.2	1.662	0.617	0.350	0.0526	0.0040
0.3	1.882	0.613	0.350	0.0464	0.0035
0.4	2.050	0.601	0.351	0.0402	0.0030
0.5	2.152	0.586	0.355	0.0340	0.0025
0.6	2.187	0.561	0.389	0.0278	0.0020
0.7	2.144	0.524	0.443	0.0216	0.0015
0.8	1.970	0.463	0.479	0.0154	0.0010
0.9	1.582	0.351	0.500	0.0092	0.0005
1.0	0.000	0.000	0.000	0.0030	0.0000
Dimensions of three-bladed B-screw series					
r/R	$\frac{c_r}{D} \frac{Z}{A_E/A_0}$	a_r/c_r	b_r/c_r	$\frac{S_r}{D} = A_r - B_r Z$	
				A_r	B_r
0.2	1.663	0.616	0.350	0.0526	0.0040
0.3	1.832	0.611	0.350	0.0464	0.0035
0.4	2.000	0.599	0.350	0.0402	0.0030
0.5	2.120	0.583	0.355	0.0340	0.0025
0.6	2.186	0.558	0.389	0.0278	0.0020
0.7	2.168	0.526	0.442	0.0216	0.0015
0.8	2.127	0.481	0.478	0.0154	0.0010
0.9	1.657	0.400	0.500	0.0092	0.0005
1.0	0.000	0.000	0.000	0.0030	0.0000
$A_r, B_r = \text{constants in equation for } S_r/D$ $a_r = \text{distance between leading edge and generation line at } r$ $b_r = \text{distance between leading edge and location of maximum thickness}$ $c_r = \text{chord length of blade section thickness at radius } r$ $s_r = \text{maximum blade section thickness at radius } r$					

In the *Figure B.1*, the geometric blade section parameters of a Wageningen B-series propeller are depicted.

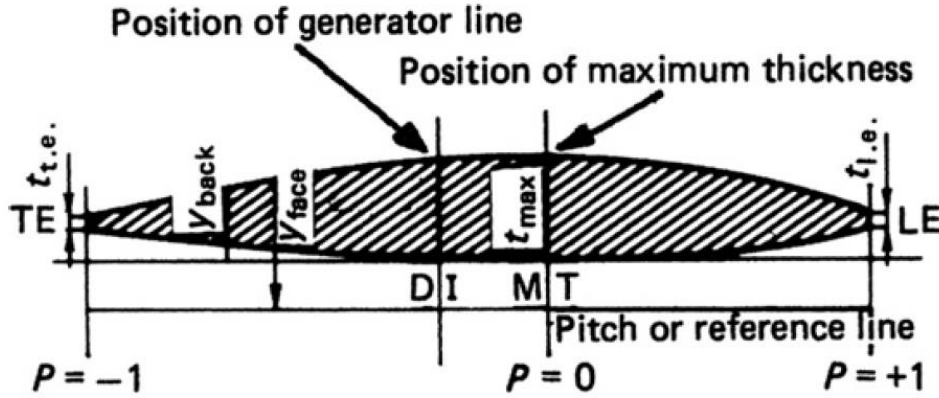


Figure B.1: Definition of geometric blade section parameters of Wageningen B-series propellers

Referring to the Figure B.1, note the following:

LE = leading edge

TE = trailing edge

MT = location of maximum thickness

DI = location of directrix

y_{face}, y_{back} = vertical ordinate of a point on a blade section on the face and on the back with the respect to the pitch line

t_{max} = maximum thickness of blade section

$t_{t.e.}, t_{l.e.}$ = extrapolated blade section thickness at the trailing and leading edges

V_1, V_2 = tabulated functions dependent on r/R and P

P = non – dimensional coordinate along pitch line from position of maximum thickness to leading edge , (where $P = 1$) and from position of maximum thickness to trailing edge (where $P = -1$)

The required coordinates of the profiles can be calculated by means of formulas, as shown in the equations (B.1) and (B.2), analogous to the formulas given by Van Gent and Van Oossanen [81] and Van Oossanen [82].

$$\text{for } P < 0 \begin{cases} y_{face} = V_1(t_{max} - t_{t.e.}) \\ y_{back} = (V_1 + V_2)(t_{max} - t_{t.e.}) + t_{t.e.} \end{cases} \quad (B.1)$$

$$\text{for } P \geq 0 \begin{cases} y_{face} = V_1(t_{max} - t_{l.e.}) \\ y_{back} = (V_1 + V_2)(t_{max} - t_{l.e.}) + t_{l.e.} \end{cases} \quad (B.2)$$

Values of tabulated functions V_1 and V_2 are given in Table B.2. Also, the values of the $t_{l.e.}$ and $t_{t.e.}$ are usually chosen in accordance with rules laid down by classification societies or in accordance with manufacturing requirements.

Table B.2: Values of V_1 and V_2 for use in equations (B.1) and (B.2)

Values of V_1 for Use in the Equations												
r/R	P	-1.0	-0.95	-0.9	-0.8	-0.7	-0.6	-0.5	-0.4	-0.2	0	
0.7-1.0		0	0	0	0	0	0	0	0	0	0	
0.6		0.0382	0.0169	0.0067	0.0022	0.0006	0	0	0	0	0	
0.5		0.0522	0.0420	0.0330	0.0190	0.0100	0.0040	0.0012	0	0	0	
0.4		0.1467	0.1200	0.0972	0.0630	0.0395	0.0214	0.0116	0.0044	0	0	
0.3		0.2306	0.2040	0.1790	0.1333	0.0943	0.0623	0.0376	0.0202	0.0033	0	
0.25		0.2598	0.2372	0.2115	0.1651	0.1246	0.0899	0.0579	0.0350	0.0084	0	
0.2		0.2826	0.2631	0.2400	0.1967	0.1570	0.1207	0.0880	0.0592	0.0172	0	
0.15		0.3000	0.2824	0.265	0.2300	0.1950	0.1610	0.1280	0.0955	0.0365	0	
		+1.0	+0.95	+0.9	+0.85	+0.8	+0.7	+0.6	+0.5	+0.4	+0.2	0
0.7-1.0		0	0	0	0	0	0	0	0	0	0	0
0.6		0.0382	0.0169	0.0067	0.0022	0.0006	0	0	0	0	0	0
0.5		0.1278	0.0778	0.0500	0.0328	0.0211	0.0085	0.0034	0.0008	0	0	0
0.4		0.2181	0.1467	0.1088	0.0833	0.0637	0.0357	0.0189	0.0090	0.0033	0	0
0.3		0.2923	0.2186	0.1760	0.1445	0.1191	0.0790	0.0503	0.0300	0.0148	0.0027	0
0.25		0.3256	0.2513	0.2068	0.1747	0.1465	0.1008	0.0669	0.0417	0.0224	0.0031	0
0.2		0.3560	0.2821	0.2353	0.2000	0.1685	0.1180	0.0804	0.0520	0.0304	0.0049	0
0.15		0.3860	0.3150	0.2642	0.223	0.1870	0.1320	0.0920	0.0615	0.0384	0.0096	0
Values of V_2 for Use in the Equations												
r/R	P	-1.0	-0.95	-0.9	-0.8	-0.7	-0.6	-0.5	-0.4	-0.2	0	
0.9-1.0		0	0.0975	0.1900	0.3600	0.5100	0.6400	0.7500	0.8400	0.9600	1	
0.85		0	0.0975	0.1900	0.3600	0.5100	0.6400	0.7500	0.8400	0.9600	1	
0.8		0	0.0975	0.1900	0.3600	0.5100	0.6400	0.7500	0.8400	0.9600	1	
0.7		0	0.0975	0.1900	0.3600	0.5100	0.6400	0.7500	0.8400	0.9600	1	
0.6		0	0.0965	0.1885	0.3585	0.5110	0.6415	0.7530	0.8426	0.9613	1	
0.5		0	0.0950	0.1865	0.3569	0.5140	0.6439	0.7580	0.8456	0.9639	1	
0.4		0	0.0905	0.1810	0.3500	0.5040	0.6353	0.7525	0.8415	0.9645	1	
0.3		0	0.0800	0.1670	0.3360	0.4885	0.6195	0.7335	0.8265	0.9583	1	
0.25		0	0.0725	0.1567	0.3228	0.4740	0.6050	0.7184	0.8139	0.9519	1	
0.2		0	0.0640	0.1455	0.3060	0.4535	0.5842	0.6995	0.7984	0.9446	1	
0.15		0	0.0540	0.1325	0.287	0.428	0.5585	0.6770	0.7805	0.9360	1	
r/R	P	+1.0	+0.95	+0.9	+0.85	+0.8	+0.7	+0.6	+0.5	+0.4	+0.2	0
0.9-1.0		0	0.0975	0.1900	0.2775	0.3600	0.5100	0.6400	0.7500	0.8400	0.9600	1
0.85		0	0.1000	0.1950	0.2830	0.3660	0.5160	0.6455	0.7550	0.8450	0.9615	1
0.8		0	0.1050	0.2028	0.2925	0.3765	0.5265	0.6545	0.7635	0.8520	0.9635	1
0.7		0	0.1240	0.2337	0.3300	0.4140	0.5615	0.6840	0.7850	0.8660	0.9675	1
0.6		0	0.1485	0.2720	0.3775	0.4620	0.6060	0.7200	0.8090	0.8790	0.9690	1
0.5		0	0.1750	0.3056	0.4135	0.5039	0.6430	0.7478	0.8275	0.8880	0.9710	1
0.4		0	0.1935	0.3235	0.4335	0.5220	0.6590	0.7593	0.8345	0.8933	0.9725	1
0.3		0	0.1890	0.3197	0.4265	0.5130	0.6505	0.7520	0.8315	0.8920	0.9750	1
0.25		0	0.1758	0.3042	0.4108	0.4982	0.6359	0.7415	0.8259	0.8899	0.9751	1
0.2		0	0.1560	0.2840	0.3905	0.4777	0.6190	0.7277	0.8170	0.8875	0.9750	1
0.15		0	0.1300	0.2600	0.3665	0.4520	0.5995	0.7105	0.8055	0.8825	0.9760	1

References

- [1] J. Carlton, *Marine Propellers and Propulsion*, Butterworth-Heinemann, 2012.
- [2] "WIKIPEDIA," [Online]. Available: <https://en.wikipedia.org/wiki/Propeller>.
- [3] C. P. Holst, *The Geometry of the Screw-propeller*, Brill, 1924.
- [4] "World Shipping Council," 2008.
- [5] M.M.M. SARCAR, K. MALLIKARJUNA RAO, K. LALIT NARAYAN, *Computer Aided Design and Manufacturing*, PHI Learning Pvt. Ltd, 2008.
- [6] U. S. C. O. o. T. Assessment, *Computerized Manufacturing Automation: Employment, Education, and the Workplace, Volume 1*, Congress of the U.S., Office of Technology Assessment, 1984.
- [7] "WIKIPEDIA," [Online]. Available: https://en.wikipedia.org/wiki/Computer-aided_engineering.
- [8] M. Bozdoc, "i-MB," [Online]. Available: <http://mbinfo.mbdesign.net/CAD-History.htm>.
- [9] "CADAZZ," [Online]. Available: <http://www.cadazz.com/index.htm>.
- [10] D. H. Gerald Farin, *The Essentials of CAGD*, Taylor & Francis, 2000.
- [11] M. E. Mortenson, *Mathematics for Computer Graphics Applications*, Industrial Press Inc, 1999.
- [12] J. H. G. Jean Gallier, "Curves and Surfaces in Geometric Modeling," Morgan Kaufmann, 2000, pp. 171-236.
- [13] L. J. Pritchard, "An Eleven Parameter Axial Turbine Airfoil Geometry Model," in *ASME 1985 International Gas Turbine Conference and Exhibit*, 1985.
- [14] Jürgen Gräsel, Akin Keskin, Marius Swoboda, Hans Przewozny, André Saxer, "A Full Parametric Model for Turbomachinery Blade Design and Optimisation," in *ASME 2004 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2004.
- [15] Georgia N. Koini, Sotirios S. Sarakinos, Ioannis K. Nikolos, "A software tool for parametric design of turbomachinery blades," *Advances in Engineering Softwar*, vol. 40, no. 1, pp. 41-51, 2009.
- [16] B. Epps, J. Chalfant, R. Kimball, A. Techet, K. Flood, C. Chryssostomidis, "OpenProp: an open-source parametric design and analysis tool for propellers," in *Proceedings of the 2009 Grand Challenges in Modeling & Simulation Conference*, 2009.

- [17] D. Calcagni, F. Salvatore, G. Bernardini, M. Miozzi, "Automated Marine Propeller Design Combining Hydrodynamics Models and Neural Networks," in *Proceedings of the First International Symposium on Fishing Vessel Energy Efficiency*, 2010.
- [18] S. Mirjalili, A. Lewis, S. A. M. Mirjalili, "Multi-objective Optimisation of Marine Propellers," *Procedia Computer Science*, vol. 51, pp. 2247-2256, 2015.
- [19] M. T. Herath, S. Natarajan, B. G. Prusty, N. S. John, "Isogeometric analysis and Genetic Algorithm for shape-adaptive composite marine propellers," *Computer Methods in Applied Mechanics and Engineering*, vol. 284, pp. 835-860, 2015.
- [20] Y. Kenji, I. Osamu, "Applications of genetic algorithm to aerodynamic shape optimization," in *12th Computational Fluid Dynamics Conference*, 1995.
- [21] S. Pierret, Designing turbomachinery blades by means of the function approximation concept, Faculte Polytechnique de Mons, Von Karman Institute for Fluid Dynamics, 1999.
- [22] "Maritime Research Institute Netherlands, Propeller design & analysis," [Online]. Available: <http://www.marin.nl/web/show>.
- [23] "Hydrocomp Inc., Propeller design," [Online]. Available: <http://hydrocompinc.com/tasks/propeller-design>.
- [24] "Xrotor: an interactive program for the design and analysis of ducted and free-tip propellers and windmills (2011)," [Online]. Available: <http://web.mit.edu/drela/Public/web/xrotor/>.
- [25] H. Martin, "Javaprop - design and analysis of propellers," [Online]. Available: <http://www.mh-aerotoools.de/airfoils/javaprop.htm>.
- [26] J. Morgado, Development of an open source software tool for propeller design in the maat, 2016.
- [27] "JBLADE: a Propeller Design and Analysis Code," in *2013 International Powered Lift Conference, American Institute of Aeronautics and Astro*, 2013.
- [28] B. Epps, "On the rotor lifting line wake model," *Journal of Ship Production and Design*, vol. 32, no. 3, pp. 1-15, 2016.
- [29] B.P. Epps, R.W. Kimball, "Unified Rotor Lifting Line Theory," *Journal of Ship Research*, vol. 57, no. 4, pp. 1-21, 2013.
- [30] B.P. Epps, R.W. Kimball, "Open-source software for the design and analysis of marine propellers and horizontal-axis turbines," OpenProp v3, [Online]. Available: <http://engineering.dartmouth.edu/epps/openprop>.
- [31] F. Anthony, "PROP DESIGN," [Online]. Available: <https://propdesign.jimdo.com/>.

- [32] "MATLAB - MathWorks - MATLAB & Simulink," [Online]. Available: <https://www.mathworks.com/products/matlab.html>.
- [33] "WIKIPEDIA," [Online]. Available: https://en.wikipedia.org/wiki/Rhinoceros_3D.
- [34] "Rhinoceros," Robert McNeel & Associates, [Online]. Available: <https://www.rhino3d.com/resources/>.
- [35] "Novedge," LLC, Novedge, [Online]. Available: <https://novedge.com/products/2217>.
- [36] "Grasshopper," Robert McNeel & Associates, [Online]. Available: <https://www.grasshopper3d.com>.
- [37] B. Wassermann, T. Bog, S. Kollmannsberger and E. Rank, "A design-through-analysis approach using the Finite Cell Method," *Eccomas Proceedia*, 2016.
- [38] T.J.R. Hughes, J.A. Cottrell, Y. Bazilevs, "Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement," *Computer Methods in Applied Mechanics and Engineering*, 2005.
- [39] T. J. R. H. Y. B. J. Austin Cottrell, *Isogeometric Analysis: Toward Integration of CAD and FEA*, Wiley, 2009.
- [40] C. G. Politis, A. Papagiannopoulos, K. A. Belibassakis, P. D. Kaklis, K. V. Kostas, A. I. Ginnis, T. P. Gerostathis, "An isogeometric BEM for exterior potential-flow problems around lifting bodies," in *11th World Congress on Computational Mechanics, WCCM 2014, 5th European Conference on Computational Mechanics, ECCM 2014 and 6th European Conference on Computational Fluid Dynamics, ECFD 2014*, 2014.
- [41] A. C. P. K.V.Kostas, "Shape-optimization of 2D hydrofoils using an Isogeometric BEM solver," *Computer-Aided Design*, January 2017.
- [42] A. I. Ginnis, R. Duvigneau, C. Politis, K. V. Kostas, K. Belibassakis, T. Gerostathis, P. D. Kaklis, "A Multi-Objective Optimization Environment for Ship-Hull Design based on a BEM-Isogeometric Solver," in *The fifth Conference on Computational Methods in Marine Engi*, Hamburg, Germany, 2013.
- [43] K.V.Kostas, A.I.Ginnis, C.G.Politis, P.D.Kaklis, "Ship-hull shape optimization with a T-spline based BEM–isogeometric solver," *Computer Methods in Applied Mechanics and Engineering*, vol. 284, pp. 611-622, 2015.
- [44] P. D. Kaklis, C. G. Politis, K. A. Belibassakis, A. I. Ginnis, K. V. Kostas, G. P. Gerostathis, "Boundary-Element Methods and Wave Loading on Ships," in *Encyclopedia of Computational Mechanics Second Edition, Vol. Fluids–Part 2*, Wiley, 2017, pp. 1-35.
- [45] WIKIPEDIA, [Online]. Available: <https://en.wikipedia.org/wiki/IGES>.

- [46] T. W. Sederberg, J. Zheng, A. Bakenov, A. Nasri, "T-splines and T-NURCCS," *ACM Transactions on Graphics*, pp. 477-484.
- [47] "T-Splines, Inc.," [Online]. Available: <http://www.tsplines.com/maya/>.
- [48] "T-Splines, Inc.," [Online]. Available: <http://www.tsplines.com/rhino/>.
- [49] G. Farin, *NURBS for Curve & Surface Design: From Projective Geometry to Practical Use*, Taylor & Francis, 1999.
- [50] J.A. Cottrell , T.J.R. Hughes, A. Reali, "Studies of refinement and continuity in isogeometric structural analysis," *Computer Methods in Applied Mechanics and Engineering*, September 2007.
- [51] D. F. Rogers, *An Introduction to NURBS: With Historical Perspective*, Morgan Kaufmann, 2001.
- [52] Les Piegl, Wayne Tiller, *The NURBS Book*, Springer Science & Business Media, 1996.
- [53] Thomas W. Sederberg, Jianmin Zheng, Xiaowen Song, "Knot Intervals and Multi-Degree Splines," *Computer Aided Geometric Design*, pp. 455-468, May 2003.
- [54] Thomas W. Sederberg, David L. Cardon, G. Thomas Finnigan, Nicholas S. North, Jianmin Zheng, Tom Lyche , "T-spline simplification and local refinement," *ACM Transactions on Graphics* , pp. 276-283 , August 2004.
- [55] RAAMAC, [Online]. Available: <http://raamac.cee.illinois.edu/nurbs-fitting/>.
- [56] Matthew T. Sederberg, Thomas W. Sederberg, "T-Splines: A Technology for Marine Design with Minimal Control Points," 2010.
- [57] Thomas W. Sederberg, G. Thomas Finnigan, Xin Li, Hongwei Lin, Heather Ipson, "Watertight Trimmed NURBS," *ACM Transactions on Graphics*, August 2008.
- [58] H. Ipson, "T-spline merging," Master's thesis, Brigham Young University, 2005.
- [59] M. A. Scott, "T-splines as a design-through-analysis technology," The University of Texas at Austin, 2011.
- [60] J. Carlton, in *Marine Propellers and Propulsion*, London, Elsevier, 2012, pp. 29-46.
- [61] D. R. Smith and J. E. Slater, "THE GEOMETRY OF MARINE PROPELLERS," 1988.
- [62] J. E. Kerwin, J. B. Hadler, *Principles of Naval Architecture Series: Propulsion*, SNAME, 2010.
- [63] H. Lackenby, "ITTC Dictionary of Ship Hydrodynamics Volume 6 of Maritime technology monograph," in *Royal institution of naval architects*, 1978.
- [64] C. P. Holst, "THE GEOMETRY OF THE SCREW-PROPELLER," Brill, Delft, 1924.

- [65] L.C. Burrill, A. Emerson, "Propeller cavitation: Further tests on 16in. propeller models in the King's College cavitation tunnel," *International Shipbuilding Progress*, pp. 119-131, April 1963.
- [66] I. H. Abbott and A. E. von Doenhoff, *Theory of Wing Sections: Including a Summary of Airfoil Data*, New York: Courier Corporation, 1959.
- [67] J. H. H. Hurt, in *AERODYNAMICS FOR NAVAL AVIATORS*, Createspace Independent Pub, 1965, pp. 20-22.
- [68] A. O. Payne, "The Grasshopper Primer," Mode Lab, [Online]. Available: <http://grasshopperprimer.com/en/index.html>.
- [69] Chiew-Lan Tai; Shi-Min Hu; Qi-Xing Huang, "Approximate merging of B-spline curves via knot adjustment and constrained optimization," *Computer-Aided Design*, vol. 35, no. 10, pp. 893-899, 2003.
- [70] M. Sederberg, "T-Splines 3 User Manual," T-Splines, Inc., 2011.
- [71] Robert McNeel & Associates, [Online]. Available: <http://docs.mcneel.com/rhino/5/help/en-us/commands/curvatureanalysis.htm>.
- [72] WIKIPEDIA, [Online]. Available: https://en.wikipedia.org/wiki/Gaussian_surface.
- [73] Robert McNeel & Associates, [Online]. Available: <http://docs.mcneel.com/rhino/5/help/en-us/commands/zebra.htm>.
- [74] W. P. A. van Lammeren, M. W. C. Oosterveld, *The Wageningen B-screw Series*, Netherlands Ship Model Basin, 1969.
- [75] WPA van Lammeren, JD van Manen , MWC Oosterveld , "The Wageningen B-Screw Series," *Trans. SNAME*, 1969.
- [76] L. Troost, *Open-water tests with modern propeller forms*, North East Coast Inst. of Engineers and Shipbuilders, 1938, 1940 and 1951.
- [77] M. O. a. P. V. OOSSANEN, "Futher Computer-Analyzed Data of The Wageningen B-screw series," *International Shipbuilding*, 1 July 1975.
- [78] P. A. Vikhar, "Evolutionary Algorithms: A Critical Review and its Future Prospects," in *International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICCC)*, 2016.
- [79] Y. Krymsky, "Responsive Skins," 31 January 2014. [Online]. Available: <https://yazdanistudioresearch.wordpress.com/2014/01/31/optimization-with-galapagos/>.

- [80] R A Cumming, W B Morgan, R J Boswell, "HIGHLY SKEWED PROPELLERS," in *Society of Naval Architects and Marine Engineers*, 1972.
- [81] van Gent, W. and van Oossanen, P., "Influence of wake on propeller loading and cavitation," *International Shipbuilding Progress*, pp. 279-321, August 1973.
- [82] P. Van Oossanen, Calculation of performance and cavitation characteristics of propellers including effects of non-uniform flow and viscosity, Delft University of Technology: Netherlands Ship Model Basin, Publication No. 457, 1974.