

**Novel Representations for Modelling Free-form
Objects under Constraints:
A Parametric Modelling Tool for
Computer-Aided Ship Design.**

Theofanis Katsoulis

Naval Architecture, Ocean & Marine Engineering
University of Strathclyde, Glasgow

January 14, 2021

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

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

Acknowledgements

Every journey in life -good or bad- comes to its fulfilment. We can weep for the ending and remain attached to the past, or board the next wagon and accept what it comes. In any case, life is about experiencing, isn't it? My PhD-journey has come to an end, but before the final curtain goes down I would like to thank all those contributed in their own way:

- A) **R.S.**;
- B) My supervisor Dr. **Panagiotis D. Kaklis**, for providing me this great opportunity and being always available to answer my questions;
- Γ) My colleagues and PhD fellows **Sotirios Chouliaras** and **Shahroz Khan**, for exchanging knowledge, questions, and concerns, not limited to our work;
- Δ) PhD student **Xinning Wang**, for the collaboration we had on our co-authored published article;
- E) Dr. **A.I. Ginnis**, for the collaboration we had during my secondments in the National Technical University of Athens;
- Z) Dr. **K.V. Kostas** and Prof. **C.G. Politis**, for their immediate replies every time I needed their insight;
- H) All members of Rhino's supporting team *Robert McNeel & Associates* and especially **Dale Fugier**, **Bob Mcneel**, **Luis Fraguada**, and **Carlos Pérez Albà**;

-
- Θ) Autodesk's supporting team and especially **Radha Patil** from the Education Support Team, for providing me and extending the necessary T-splines plug-in license, at a very crucial period for my research;
- I) Dr. **Iraklis Lazakis**, for trusting in me;
- K) **Carol Georges**, Departmental Administrator of the NAOME department of Strathclyde University for helping me from the very first day of my PhD-journey. R.I.P. Carol;
- Λ) **Susan Pawson**, Administrative Assistant of the NAOME department of Strathclyde University, for being kind and helping myself and every other student in the NAOME department;
- M) **Lynne Samson** Financial Administrator of the NAOME department of Strathclyde University for her guidance on financial aspects;
- N) All the staff of the Administration Department of NAOME;
- Ξ) All **ARCADES** members;
- O) My **Father**, Christos Katsoulis for EVERYTHING he is doing for me;
- Π) My **Mother**, Andriana Tsementzi for EVERYTHING she is doing for me;
- P) My **Sister**, Maria-Olympia Katsouli, for EVERYTHING she is doing for me and being my best friend;
- Ω) **BBJ G.S.D.**

Contents

| | |
|--|-------------|
| Acknowledgements | ii |
| Acronyms | xvii |
| Introduction | 1 |
| Novelties | 6 |
| 1 Theoretical Background | 9 |
| 1.1 From B-splines and NURBS to T-splines | 9 |
| 1.2 Parametric modelling | 15 |
| 2 A T-splines-based Parametric Modelling Tool for ship-hull design | 19 |
| 2.1 Introduction | 19 |
| 2.2 Parameters: introduction and handling | 21 |
| 2.3 From parameters to the control cage and the output surface | 25 |
| 3 Geometric Validity of TshipPM and Data Analysis | 39 |
| 3.1 Introduction | 39 |
| 3.2 Definitions and geometric validity evaluation criteria | 40 |
| 3.3 Constraints | 44 |
| 3.3.1 Transition curve constraints | 44 |
| 3.3.2 Other constraints and conditions | 47 |
| 3.4 Behaviour of TshipPM against strong parameter values variation | 50 |
| 3.5 Data analysis and design space | 53 |

| | | |
|----------|--|------------|
| 4 | A Method to Remodel a Parent Ship-hull using TshipPM | 61 |
| 4.1 | Introduction | 61 |
| 4.2 | Geometric and ship-design criteria for remodelling evaluation | 62 |
| 4.3 | Control cage construction | 62 |
| 4.3.1 | Extraction of physical parameters | 63 |
| 4.3.2 | Assigning non-dimensional parameter values with respect to physical parameters | 66 |
| 4.3.3 | Midship control cage construction | 67 |
| 4.3.4 | Bow control cage construction | 73 |
| 4.3.5 | Stern control cage construction | 78 |
| 4.4 | Evaluation and optimisation | 81 |
| 4.4.1 | Pre-optimisation process and results | 82 |
| 4.4.2 | Optimisation for a more accurate remodelling of the parent ship-hull | 88 |
| 4.4.3 | Post-optimisation evaluation: results and comparison with parent hull | 92 |
| 5 | TshipPM vs CAESES: A comparative study | 100 |
| 5.1 | Introduction | 100 |
| 5.2 | CAESES parametric modeller | 101 |
| 5.3 | Common parameters | 104 |
| 5.4 | Moments comparison | 105 |
| 5.5 | SAC comparison | 109 |
| 5.6 | Gaussian-curvature comparison | 110 |
| 5.7 | Sectional-curvature comparison | 115 |
| 5.8 | Hausdorff distance comparison | 116 |
| 6 | A shape-optimisation tool integrating a MO-TLBO algorithm and TshipPM | 119 |
| 6.1 | Introduction | 119 |
| 6.2 | Introduction to TLBO | 120 |

| | | |
|-----|--|------------|
| 6.3 | TshipPM MO-TLBO: objective functions | 121 |
| 6.4 | TshipPM MO-TLBO: method and features | 123 |
| 6.5 | Results | 130 |
| | Summary | 142 |
| | Bibliography | 145 |

List of Figures

| | | |
|-----|---|----|
| 1.1 | T-splines single patch (top) and NURBS multi-patch (bottom) ship-hulls. | 10 |
| 1.2 | Trimmed NURBS produce non-watertight models with gaps (top), while T-splines solve this problem taking advantage of T-junctions (bottom) [24]. | 11 |
| 1.3 | NURBS (left) knot lines lie on a global rectangular grid, while T-splines (right) can form T-junctions due to locally defined knot vectors. Extraneous knot lines and control points in NURBS are depicted in black border-style lines. Examples of parametric faces, depicted in red diagonal stripe pattern: one face (rightmost) with 5 vertices (four corner vertices and one T-junction vertex) and two faces with four corner vertices. Note that all faces, in both NURBS and T-splines meshes, are rectangular. | 13 |
| 1.4 | Constructing the T-splines basis function corresponding to the vertex \mathbf{V}_0 of a valid T-mesh. The extents of the constituting univariate knot vectors are colored in green, while the boundary of the support of the resulting basis function is depicted as dashed polygonal line, colored in red. | 14 |

| | | |
|------|--|----|
| 1.5 | Parametric modelling: a simple example. Two different, topologically equivalent valid objects (left and middle) with $d_1 > d'_1$ and $d_0 > d'_0$. Invalid shape due to wrong parameter value assignment and lack of constraint(s) imposition. | 16 |
| 2.1 | Steps to create a TshipPM instance: we first attribute values to $P_0 = Lwl$, $P_1 = B$, $P_2 = T$ and the non-dimensional parameters $\hat{P}_i \in (0, 1)$, $i = 3, \dots, 26$, which, alongside the internal parameters, determine the control cage of the ship-hull; TshipPM plug-in is then called to produce the smooth surface of the final instance. | 22 |
| 2.2 | Relation between physical (dimensional) and non-dimensional parameters. | 25 |
| 2.3 | TshipPM input parameters for Fwd_Part and Aft_Part. | 26 |
| 2.4 | Bow and Stern Parameters. | 26 |
| 2.5 | Typical segmentation of a ship-hull. Main parts are midship, bow, and stern. | 26 |
| 2.6 | Segmentation of midship, consisting of the cylindrical (or parallel) middle part, forward and afterward Parts. | 27 |
| 2.7 | Examples of control cage paths of transition type (left: from flat of side to forward part, depicted in magenta) or boundary type (middle: stern boundary, right: bow boundary, depicted in red). | 27 |
| 2.8 | Control cage path of forward transition curve (magenta) and its defining parameters. | 28 |
| 2.9 | Segmentation of the ship-hull into five segments. | 29 |
| 2.10 | U-shaped polylines and control points of the parallel middle part and their interdependency with physical parameters. | 29 |
| 2.11 | First (FTCA) and second (FTCB) forward transition curves. FTCA is located at the longitudinal position where the flat of side of the forward part ends. | 30 |

| | | |
|------|--|----|
| 2.12 | Third (FTCC) and fourth (FTCD) forward transition curves. FTCD is located at the longitudinal position where the flat of bottom of the forward part ends. | 30 |
| 2.13 | First (ATCA) and second (ATCB) afterward transition curves. ATCB is located at the longitudinal position where the flat of side of the afterward part ends. | 31 |
| 2.14 | Third (ATCC) and fourth (ATCD) afterward transition curves. ATCD is located at the longitudinal position where the flat of bottom of the afterward part ends. | 31 |
| 2.15 | Forward perpendicular (FP_Crv) and bow profile (BowProfCrv) curves. | 32 |
| 2.16 | Bulb Curves. | 32 |
| 2.17 | Afterward perpendicular (AP_Crv) and stern profile (SternPrfCrv) curves. | 33 |
| 2.18 | Two more views of the stern. | 34 |
| 2.19 | Control points of the U-shaped polygonal line \mathcal{M}_0 (depicted in black) and the first two Forward Transition Curves, FTC_0 (depicted in red) and FTC_1 (depicted in green). | 35 |
| 2.20 | An example of a valence-5 extraordinary point. \mathbf{b}_0 is located on the Forward Perpendicular Transition Curve (depicted in green) and is connected with \mathbf{b}_1 of the first Bulb Curve (depicted in red). | 36 |
| 2.21 | TshipPM output: perspective view from the bow with control points and on-surface mappings of T-mesh edges. | 37 |
| 2.22 | TshipPM output: perspective view from the stern. | 38 |
| 2.23 | TshipPM output: side view. | 38 |
| 3.1 | An invalid geometry consisting of a trimmed surface. | 41 |
| 3.2 | Example of an invalid, closed object with negative volume. The self-intersection is encircled. | 42 |
| 3.3 | Undetected by Rhino invalid geometries consisting of evident self-intersections. | 43 |

| | | |
|------|---|----|
| 3.4 | Undetected by Rhino invalid geometries consisting of non-evident self-intersections. | 43 |
| 3.5 | Misplacement of the second Forward Transition Curve (FTCB), depicted in red, as it is located before the first Forward Transition Curve (FTCA), depicted in green. | 44 |
| 3.6 | Longitudinal position of the forward transition curves. Internal parameters ascertain there are no misplacements. | 46 |
| 3.7 | Longitudinal position of the afterward transition curves. Internal parameters ascertain there are no misplacements. | 46 |
| 3.8 | $P_{2,ATCA}$, $P_{2,ATCB}$, and P'_2 have to be constrained so they are always at the same level with or higher than the keel of the ship-hull. | 47 |
| 3.9 | Invalid surface: intersection of the neighbouring surfaces of stern and tube-main body. | 48 |
| 3.10 | Valid surface: the problem has been addressed by introducing a series of control points along the span of the stern (see Fig. 3.11). | 49 |
| 3.11 | Control cage of the valid object: 4 control points along the span of the stern are needed to ascertain geometric validity at the transition from the main body-tube to stern. | 49 |
| 3.12 | Decreasing \widehat{BilgeR} by 30%. | 52 |
| 3.13 | Increasing \widehat{BilgeR} by 30%. | 52 |
| 3.14 | Decreasing \widehat{BulbH} by 20%. | 52 |
| 3.15 | Increasing \widehat{BulbH} by 20%. | 52 |
| 3.16 | Decreasing \widehat{BulbL} by 20%. | 52 |
| 3.17 | Increasing \widehat{BulbL} by 20%. | 52 |
| 3.18 | Decreasing \widehat{BulbH} by 60%. | 52 |
| 3.19 | Decreasing both \widehat{BulbL} and \widehat{BulbH} by 50%. | 52 |
| 4.1 | Measuring TransomH, TransomB, and BulbL on the KVLCC tanker CAD model. | 65 |

| | | |
|-----|--|----|
| 4.2 | Measuring the length of flat of bottom (FoB_Fwd_L) at the forward part and the breadth of flat of bottom transition (FoB_Trans), by using <i>Gaussian curvature</i> from the <i>curvature analysis</i> feature in Rhino5. . . . | 66 |
| 4.3 | Control points extraction of the largest section at Mid_Pos and their interrelation to external and internal parameters. In red is depicted the extracted curve, while in green and blue are the rebuilt and adjusted curves, respectively, and their control points respectively. c1, c2 are internal parameters. | 69 |
| 4.4 | Construction of the cylindrical middle part: define a) Mid_L/2 from largest section (red) of the parent hull, b) U-shaped polylines (blue), and c) parallel transverse lines (green). With c we denote the internal parameter introduced to determine the x-position of the two inner U-shaped polylines. | 70 |
| 4.5 | Extracted boundary curves at waterplane and flat of bottom. Both are extracted starting from M ₀ , with the former extended up to the tip of the waterplane, and the latter up to the FoB_Fwd_L, consisting of 6 and 5 control points respectively. | 71 |
| 4.6 | Setting the location of FTCs. Extracting control points of the waterplane and bottom plane curves and adjustment. In red are the originally extracted curves and their control polygons, in green are the rebuilt ones, while in blue are the final, adjusted curves. The control points of both boundary curves have been relocated in a manner so they share the same x-coordinate. c1 and c2 are internal parameters with $c1 < c2$ | 72 |
| 4.7 | FP and AP sections in red (top). Adjusted FP (right) and AP (left) curves with their control polygons depicted in blue (bottom). | 73 |

| | | |
|------|---|----|
| 4.8 | Extraction of bow profile curve (BowPrfCrv) and its control points, and adjustment. Depicted in red color is the extracted BowPrfCrv consisting of 45 control points. In black is the degree-3 rebuilt curve with 10 control points. In blue is depicted the final adjusted curve with a more accurate match in comparison with the rebuilt curve. Control points $P_1-P'_1$ and $P_2-P'_2$ are adjusted to be collinear and they define the location of the bulb curves. | 75 |
| 4.9 | Extraction of bulb curve and its control points from parent ship. Depicted in red color is the extracted curve, in blue is the adjusted curve. . | 76 |
| 4.10 | Construction of offset curves: depicted in blue is the bow offset polyline and in red is the overhang offset polyline. In black are the lines used for their interconnection with the bow profile curve. | 77 |
| 4.11 | The control cage of the bulb. BowPrfCrv depicted in red , bulb curves in green , offset polylines in magenta , and interconnecting lines in black color. | 77 |
| 4.12 | Extraction of the control points for Afterward Perpendicular (AP_Crv, middle) and Tube (TubeCrv, right) curves, depicted in blue . In red and green are depicted the extracted curves (left). | 79 |
| 4.13 | Extraction of the part of the Stern Profile Curve (SternPrfCrv) at the waterline level and adjustment. Depicted in red color is the originally extracted SternPrfCrv, while in green is the rebuilt and in blue is the adjusted curve. | 80 |
| 4.14 | Extraction of the stern profile curves and their control points, and adjustment. Depicted in red color are the originally extracted stern profile curves, in green the rebuilt, and in blue the adjusted one. | 80 |
| 4.15 | Control cage of the stern. SternPrfCrv depicted in magenta , Stern_Crv in blue , TubeCrv in green , interconnections depicted in black color. . . . | 81 |
| 4.16 | Diagram of the evaluation and optimisation process of the TshipPM model. | 82 |

| | | |
|------|---|----|
| 4.17 | Smoothness at bow of PM_BO (top) and Parent (bottom). PM_BO's surface is at least G^1 -continuous everywhere, while Parent consists of areas of C^0 -continuity (indicated by red circles). | 83 |
| 4.18 | Smoothness at stern of PM_BO (top) and Parent (bottom). PM_BO's surface is at least G^1 -continuous everywhere, while Parent consists of areas of C^0 -continuity, encircled in red. | 84 |
| 4.19 | Color map indicating the Gaussian curvature of TshipPM and parent ship-hulls; Bow view. | 85 |
| 4.20 | Color map indicating the Gaussian curvature of TshipPM and parent ship-hulls; Stern view. | 85 |
| 4.21 | Segmentation of PM_BO and Parent to calculate volume moments for Fwd_Part, Parallel Middle Part, and Aft_Part. | 86 |
| 4.22 | Comparison of Sectional Area Curves (SAC) between PM_BO and Parent. In green color PM_BO SAC is depicted, Parent ship SAC in red, while their deviation is depicted in blue color. | 90 |
| 4.23 | Longitudinal intervals where analysis using SAC showed critical deviations between the sectional areas of TshipPM and parent models. | 91 |
| 4.24 | Sectional areas of PM_BO (blue) and Parent (red) models of the forward part. | 91 |
| 4.25 | Evaluation-optimisation workflow to attain an acceptable ship-hull instance which lies geometrically in the proximity of the parent hull. | 91 |
| 4.26 | KVLCC's control cage with fewer control curves compared to the version of TshipPM used for the remodelling of the KCS ship-hull. | 92 |
| 4.27 | Bow view: smoothness of TshipPM's (left) and Parent's (right) models. TshipPM's surface is at least G^1 -continuous everywhere, while Parent consists of areas of C^0 -continuity. An example is indicated by the red circle. | 93 |

| | | |
|------|---|-----|
| 4.28 | Stern view: smoothness of TshipPM's (left) and Parent's (right) models. TshipPM's surface is at least G^1 -continuous everywhere, while Parent consists of areas of C^0 -continuity. An example is indicated by the red square-frame. | 93 |
| 4.29 | Color map indicating the Gaussian curvature of the TshipPM model and the Parent; Bow view. | 94 |
| 4.30 | Color map indicating the Gaussian curvature of the TshipPM model and the Parent; Stern view. | 94 |
| 4.31 | Comparison of sectional area curves (SAC) between TshipPM and Parent. TshipPM's SAC is depicted in green color, Parent's in red, while their deviation is depicted in blue color. | 98 |
| 4.32 | Hausdorff distance plot illustrating the geometric deviation between TshipPM's produced model and Parent. | 99 |
| 5.1 | CAESES input parameters. | 102 |
| 5.2 | Coordinate system and models' orientation. | 104 |
| 5.3 | Side view of Gaussian-curvature plot of Fwd_Part (top) and Aft_Part (bottom) of PM1 (left) and PM2 (right). | 111 |
| 5.4 | Bow view of Gaussian-curvature plot of PM1 (left) and PM2 (right). . . | 111 |
| 5.5 | Stern view of Gaussian-curvature plot of PM1 (left) and PM2 (right). . | 111 |
| 5.6 | Bottom view of Gaussian-curvature plot of PM1 (top) and PM2 (bottom).111 | |
| 5.7 | Small-scale extraneous oscillations of the Gaussian-curvature distribution of PM2: bottom view. | 112 |
| 5.8 | Small-scale extraneous oscillations of the Gaussian-curvature distribution of PM2: side (top) and stern view (bottom). | 112 |
| 5.9 | SAC comparison among PM1 (red curve), PM2 (blue curve) and the parent hull (green curve): the magenta curve depicts the deviation $SAV_{PM1} - SAV_{PM2}$ with the absolute maximum deviation for PM1 (9.5m) and PM2 (3m) occurring at $x = 168m$ and $x = 59m$ respectively. | 113 |

| | | |
|------|--|-----|
| 5.10 | Improved SAC for PM1 (red curve) after introducing an additional after-ward transition curve between the parallel middle part and the first after-ward transition curve. The absolute maximum deviation for PM1 (3.8m) occurs at $x = 163m$. | 114 |
| 5.11 | PM1 (left), and PM2 (right) porcupine plots of section at $x = -1m$. | 115 |
| 5.12 | PM1 (left), and PM2 (right) porcupine plots of section at $x = 104.7m$. | 116 |
| 5.13 | PM1 (left), and PM2 (right) porcupine plots of section at $x = 151.7m$. | 116 |
| 5.14 | PM1 (left), and PM2 (right) porcupine plots of section at $x = 205.3m$. | 116 |
| 5.15 | Extraneous inflection points of PM2 for sections at $x = 151.7m$ (left) and $x = 205.3m$ (right). | 116 |
| 5.16 | Hausdorff distance plot of PM1 with respect to the parent ship-hull. Color map range is $[0, 0.69m]$ | 117 |
| 5.17 | Hausdorff distance plot of PM2 with respect to the parent ship-hull. Color map range is $[0, 0.69m]$ | 117 |
| 5.18 | Hausdorff distance plot of PM1 with respect to the parent ship-hull. Color map range is $[0, 0.42m]$. | 118 |
| 5.19 | Hausdorff distance plot of PM1 with respect to the parent ship-hull. Color map range is $[0, 0.69m]$ | 118 |
| 6.1 | Costs distinction per segment. Breaking down the problem for a more time-efficient approach. Parameters $FoS_{\widehat{Trans}}$ and $Fob_{\widehat{Trans}}$ are calculated only for the forward part, and their value is constant for the after-ward part. | 124 |
| 6.2 | TshipPM MO-TLBO flowchart for shape optimisation using as cost functions ship-design criteria, i.e., volume moments up to 2 nd order, namely volume, volume centroid, and moments of inertia with respect to volume centroid. | 129 |
| 6.3 | Optimised models resulted from TshipPM MO-TLBO. Bow and stern view captures are illustrated for each model. | 139 |

| | | |
|-----|--|-----|
| 6.4 | Geometric variation between the parallel middle parts of TshipPM MO-TLBO models and the parent hull, using Hausdorff distance. | 140 |
| 6.5 | Geometric variation between the parallel middle part of TshipPM MO-TLBO models and the parent hull, using stricter stopping criteria. Model_11: max= 0.34m, mean= 0.05m. | 141 |

List of Tables

| | | |
|-----|---|-----|
| 2.1 | TshipPM parameters and groups; P_i 's: physical parameters. | 22 |
| 3.1 | TshipPM _C design space. | 56 |
| 3.2 | TshipPM _T design space. | 57 |
| 3.3 | TshipPM data analysis: Histograms information. | 60 |
| 4.1 | Comparison of moments between KCS parent ship-hull and PM_BO model. (Full model) | 86 |
| 4.2 | Comparison of moments between KCS parent ship-hull and PM_BO model. (Segmentation at $x^1 = 104.7\text{m}$, and $x^2 = 151.7\text{m}$) | 87 |
| 4.3 | Comparison of moments between KVLCC parent ship-hull and TshipPM model. | 95 |
| 4.4 | Comparison of moments between KVLCC parent ship-hull and TshipPM model. (Segmentation at $x^1 = 99.69\text{m}$, and $x^2 = 192.30\text{m}$) | 96 |
| 5.1 | CAESES Parameters and Groups. | 102 |
| 5.2 | TshipPM-CAESES common parameters list. | 105 |
| 5.3 | Comparison of moments among KCS parent ship-hull, PM1 and PM2 models (full model). | 107 |
| 5.4 | Comparison of moments among KCS parent ship-hull, PM1 and PM2 models. (Segmentation at $x^1 = 104.7\text{m}$, and $x^2 = 151.7\text{m}$) | 107 |
| 6.1 | TshipPM MO-TLBO results. | 132 |

Acronyms

ADS Actual Design Space of TshipPM

Aft_Part Afterward Part

AP Afterward Perpendicular

AP_Crv Afterward Perpendicular Curve

ATCA First Afterward Transition Curve

ATCB Second Afterward Transition Curve

ATCC Third Afterward Transition Curve

ATCD Fourth Afterward Transition Curve

BEM Boundary Element Method

BowPrfCrv Bow Profile Curve

BulbCrvA First Bulb Curve

BulbCrvB Second Bulb Curve

CAD Computer-Aided Design

CASD Computer-Aided Ship Design

CFD Computer Fluid Dynamics

DoF Degrees of Freedom

FEM Finite Element Method

FP Forward Perpendicular

FP_Crv Forward Perpendicular Curve

FTCA First Forward Transition Curve

FTCB Second Forward Transition Curve

FTCC Third Forward Transition Curve

FTCD Fourth Forward Transition Curve

Fwd_Part Forward Part

IGA Isogeometric Analysis

IP Initial Population

IPS Initial Population Size

KCS MOERI container ship-hull

KVLCC MOERI tanker ship-hull

LP Learning Phase

Mid_Part Parallel Middle Part

MoI Moments of Inertia

MO-TLBO Multi-objective Teaching-Learning-based Optimisation

NURBS Non-Uniform Rational B-splines

PM Parametric Modeller

PM_BO Parametric Model Before Optimisation: TshipPM's KCS ship-hull, produced in the remodelling process, before its final optimisation

PM1 Parametric Model 1: TshipPM's KCS ship-hull, produced for comparison with the CAESES model

PM2 Parametric Model 2: CAESES's KCS ship-hull, produced for comparison with TshipPM's model

SAC Sectional Area Curve

SAV Sectional Area Values

SC Stopping Criteria

SternPrfCrv Stern Profile Curve

TDS Theoretical Design Space of TshipPM

TLBO Teaching-Learning-based Optimisation

TP Teaching Phase

TshipPM T-splines-based parametric modelling tool for ship design

TshipPM_C TshipPM version for containers

TshipPM_T TshipPM version for tankers

Tube_Crv Tube Profile Curve

VC Volume Centroid

Abstract

Digital design of engineering systems, using *Computer-Aided Design* (CAD) tools for developing and optimising products, has been used in the manufacturing process since 1957, when Patrick Hanratty built the first commercial numerical-control programming system known as *PRONTO*, while he was working for *General Electric*. One year later, the French physicist and mathematician Paul de Casteljaou invented a system [1] based on the use of Bernstein polynomials [2], [3] during his employment with the French automobile manufacturer Citroën. In 1968, another French engineer named Pierre Bézier, launched on behalf of the automobile manufacturer Renault the CAD/CAM UNISURF system [4] for surface design, which was fully in use by 1975. The stepping stone of *Computer-Aided Ship Design* (CASD) was set in 1963, when the Norwegian CAD/CAM software *Autokon* was first used, developed by Trygve Reenskaug. It was 14 years later when *parametric modelling* in *ship design* introduced by Prof. H. Nowacki who pioneered in CASD via coupling form parameters with the then novel technology of B-splines [5]. Since then, Parametric Modellers (PM) play a crucial role in the development and -most importantly- shape optimisation of free-form objects with increased complexity such as ship-hulls, for they have to represent robustly and efficiently every solid object.

The purpose of the current work is three-fold:

- Firstly, it is meant to develop and present the corresponding methodology of two tools:
 - a) an innovative, robust, and cost-efficient parametric modelling tool for container and tanker ship-hulls, which are hull forms of increased complexity. The tool is

named *TshipPM* and is an extension of the work presented in [6],

b) a ship-hull shape optimisation tool based on geometric criteria, namely volume moments up to 2nd order.

- Secondly, to evaluate the performance of TshipPM by comparing it with a well-established, commercial parametric modelling tool, CAESES[®]¹, opting for its NURBS functionality.
- Finally, was to provide an easy to follow "handbook" for developers and users on the development of parametric modelling tools, illustrating as well how a PM can be utilised for the construction of a given ship-hull.

The key findings of the current work which also coincide with the main objectives, are:

- A method of constructing a T-splines-based parametric modelling tool (TshipPM) for complex ship-hull design, the development of which is taking into consideration the detailed characteristics of two different ship-hull types, i.e., tankers and containers.
- A set of geometric and design constraints imposed to TshipPM to tackle the intricate and complex issue of *Geometric Validity*, ascertaining the production of valid models for the whole design space the modeller covers.
- A step-by-step method to remodel (using TshipPM) a great variety of complex hulls with complexity not-higher than that of tankers and containers.
- The exceptional performance of TshipPM. A comparative study is conducted to evaluate the performance of TshipPM against CAESES, a well-established commercial parametric modeller for ship design. This work is published in the *Ocean Engineering*² international journal, titled "*A T-splines-based parametric modeller for computer-aided ship design* [7]".

¹<https://www.caeses.com/>

²<https://www.journals.elsevier.com/ocean-engineering>

-
- A method for ship-hull shape optimisation with respect to geometric criteria and especially volume moments up to 2nd order, utilising TshipPM and a multi-objective Teaching-Learning-based optimisation tool. This part of the work also stresses the insufficiency of 2nd order moments to perform as complex-shape optimisers.

TshipPM is a parametric modelling tool for ship-hull design under geometric and design constraints. It uses as a mathematical representation of surfaces -and to some extent solids- the fairly new technology of T-splines, introduced by Sederberg et al. in 2003 [8], which constitutes a generalisation of NURBS, exhibiting several advantages over the latter. T-splines technology aids towards the development of modellers for complex ship forms, by providing geometrically valid objects with smooth surfaces and increased fairness throughout the model's surface, and much lower complexity in comparison with PMs employing the standard NURBS technology. TshipPM uses 27 *external* (or *input*) parameters to produce the control cage of the ship-hull, 3 of which are global and dimensional, while 24 are *non-dimensional* and of local nature. It employs Autodesk[®] T-splines plug-in[®] v.4.0 for Rhino5[®] 3D to create the final, smooth T-splines surfaces using as input the *control cage* created by TshipPM.

The PhD thesis consists of 6 Chapters, structured in the following way:

The **Novelties** of the current work and its contribution to the field of parametric modelling in CASD.

Chapter 1, after a brief discussion on the historical background of splines, it delivers a comparison between T-splines, the representation underlying TshipPM, and NURBS, the industrial standard in CAD, which is also used in the context of CAESES (§1.1). In addition, it introduces the reader to parametric modelling, reviewing in brief the main advancements in the field (§1.2).

Chapter 2, after the necessary introduction to TshipPM's features and characteristics (§2.1), analyses the structure of TshipPM regarding the parameters involved (§2.2), making the distinction into internal and external, dimensional (or physical) and non-dimensional. §2.3 demonstrates the process of creating a ship-hull model, starting from

building the control cage along with its link to external parameters, which are used as input for TshipPM, up to its conversion into the resulting ship-hull surface with the aid of T-splines plug-in.

Chapter 3, after a brief introduction on the importance of robust PMs (§3.1), defines the constraints encountered in ship-hull design and introduces the reader to the concept of *geometric validity*, illustrating examples of invalid ship-hull models (§3.2). §3.3 presents the *geometric* and *design* constraints imposed to TshipPM to ascertain its robustness as far as the production of geometrically valid models is concerned. In §3.4 we provide an experimental indication of the robustness of TshipPM in its response to strong parameter values variation. Finally, in the last section (§3.5), we present the output of a Monte Carlo sampling of 200,000 ship-hull instances produced by TshipPM for both containers and tankers, and we analyse its output information to determine the design space of the PM, by measuring TshipPM's flexibility with regards to geometric characteristics, namely volume centroid and moments of inertia.

Chapter 4 begins by stressing the need of remodelling a given ship-hull with PMs to use it as basis for optimisation (§4.1), while §4.2 refers to the remodelling evaluation criteria. The remodelling process is thoroughly described in §4.3 using as a case study the MOERI KCS container ship-hull [9], and aims to the reconstruction of a parent ship-hull CAD model; we use TshipPM for the construction of model's control cage and we feed it into Rhino5 to create the corresponding surfaces with the aid of the T-splines plug-in. Finally, in §4.4 the evaluation of the remodelled MOERI KVLCC tanker [10] ship-hull is conducted, under a predefined set of criteria, providing feedback to users and developers for any adjustments and their specific locations. The evaluation of MOERI KCS container hull is conducted in Chapter 5.

In **Chapter 5**, TshipPM is compared against CAESES with regards to their outputs against a parent hull, the KCS container ship-hull [9], which has been extensively used by the research community for CAD and Computational Fluid Dynamics (CFD) benchmarking purposes. The comparison criteria are described in §5.1. §5.2 presents the basic characteristics of the CAESES parametric modelling tool, as well as a description of the

physical (or dimensional) parameters CAESES uses to remodel KCS ship-hull and the corresponding external non-dimensional parameters, while §5.3 refers to the common external parameters both PMs use. Finally, in §§5.4 - 5.8 the comparison of both PMs is conducted.

Lastly, **Chapter 6** presents an in-house, C#-built, shape-optimisation tool using TshipPM and an adjusted to the needs of the current work *Multi-objective Teaching-Learning-based Optimisation* (MO-TLBO) method. *TshipPM MO-TLBO* is optimising the shape of a given hull (MOERI KCS) against a set of ship-design criteria, and especially volume moments up to 2nd order. After the required introduction to the chapter (§6.1), a brief presentation of the TLBO method is conducted (§6.2). The set of the objective functions against which the optimisation is conducted is provided in §6.3, while in §6.4 the method of building TshipPM MO-TLBO, its features and functionality are presented in detail. The chapter concludes with §6.5, illustrating the resulted KCS-ship-hull instances produced by TshipPM MO-TLBO.

The thesis concludes with the **Discussion and Summary** of the current work, highlighting its key points, stressing the delivered novelties and main contributions to the field of parametric modelling in CASD.

Novelties

The work presented herein is about an innovative, robust, and cost-effective parametric modelling tool for ship-hull design, and especially for container and tanker ship-hulls, introduced in [6]. The current work transforms the parametric modelling tool described in [6] along several directions, which constitute the *methodological* and *technological* novelties and contributions to the field of parametric modelling in Computer-Aided Ship Design (CASD), as described below:

- TshipPM has been thoroughly transformed to be capable of producing ship-hull models for both containers and tankers, taking into consideration the detailed characteristics of both ship types. It enables a flexible representation of the ship-hull in geometrically challenging areas, namely bow and stern, as well as in the transition areas from midship towards the forward and afterward perpendiculars, by introducing 9 control curves (19 in total) and 3 external parameters (27 in total). Therefore, it is capable of delivering instances lying in the proximity of a parent container or tanker hull;
- An in depth, step by step method of *remodelling* any given container or tanker ship-hull has been developed. The remodelling process is presented using as a case study the MOERI Container Ship's (KCS) hull instance. Utilising TshipPM, the process consists of the control cage construction and the optimisation of the control cage to more accurately remodel the given ship-hull;
- As far as the cumbersome and complicated issue of *geometric validity* is concerned, after extensive experimentation with the modeller and the instances produced out

of the whole parameter values range $([0.01, 0.99])$, a set of *design constraints* was imposed to ascertain the production of valid objects. A large scale sampling of more than 400,000 instances was derived and examined in Rhino5 to evaluate TshipPM against self-intersections, confirming that TshipPM is capable of producing 100% valid geometries for the total parameter value range;

- An adjusted for the purpose of the current work *Multi-objective Teaching-Learning-based Optimisation* method is introduced, which, integrated with TshipPM, forms *TshipPM MO-TLBO*, a tool for shape-optimisation against a set of ship-design criteria (volume moments up to 2nd order). This is meant to be the foundation towards an automated remodelling process.
- TshipPM is compared with CAESES, with both PMS' outputs compared with the KCS container ship-hull. The employed comparison criteria include the common external parameters, volume moments (volume, volume centroid, moments of inertia), sectional area curve (SAC), Gaussian and sectional curvatures for assessing surface and curve fairness respectively, and Hausdorff distance for measuring the geometric distance between the hulls. This work has been published in November 2019 in the *Ocean Engineering*³ international journal, titled “A *T-splines-based parametric modeller for computer-aided ship design*”.
- We provide the reader with the list of all physical (dimensional) parameters that characterise the geometry of the instances created by TshipPM along with a concise description of each one of them, their classification in global and non-global categories and, finally, their functional interrelation, which also involves the non-dimensional parameters. In addition, we describe and display graphically all control curves used in TshipPM, and we provide the functional relations of their control points with the physical (therefore the non-dimensional as well) parameters; Furthermore, the current work illustrates a key feature of the control cage construction process, namely the mapping of the external parameters to the control points of the T-splines representation, by describing it in detail for the

³<https://www.journals.elsevier.com/ocean-engineering>

forward transition part of the hull up to the end of the forward flat of side;

- Finally, as far as the technological contribution is concerned, TshipPM is developed both in RhinoScript and C#, while TshipPM MO-TLBO is developed in C#. The C# versions of both tools are built using RhinoCommon^{®4}, the cross-platform .NET plug-in SDK for Rhino;

⁴<https://developer.rhino3d.com/guides/rhinocommon/>

Chapter 1

Theoretical Background

1.1 From B-splines and NURBS to T-splines

As early as 1946, the Romanian-American mathematician Isaac Jacob Schoenberg introduced the theory of *basic splines* (B-splines) [11]. He and his colleague B. H. Curry enriched fundamentally the field of *numerical analysis* with their work “*on spline distributions and their limits*” [12]. Nevertheless, Carl de Boor, the author of the fundamental book on splines entitled “*A Practical Guide to Splines*” [13], in his paper-postscript for Curry’s and Schoenberg’s foundational work, where he elaborates on *cardinal B-splines*, he stresses that B-splines appeared even earlier (Favard [14]). On the same note he refers to Schoenberg who always maintained that B-splines were already known to Laplace.

The concept of *Non-Uniform Rational B-Splines* (NURBS) is unclear when was first introduced, the first systematic non-uniform rational B-splines treatment, though, is attributed to K. Versprille research work for his PhD [15]. NURBS is the current geometry industrial standard for representing curves, surfaces, and solids in CAD (Computer-Aided Design) systems; see, e.g., [16], [17], [18], [19] and [20]. Although they have been used and improved in modelling for over 40 years, there are many gaps remain unsolved. T-splines is a generalisation of NURBS for surfaces and -to some extent- solids, exhibiting several advantages over NURBS:

- Local refinement with NURBS generates superfluous control points due to the global-knot-vector nature of this representation (Fig.1.3). T-splines are more efficient in this aspect, as they allow a row of control points to terminate, creating a *hanging-node*, as referred to in the FEM parlance [21], [22], [23]. This hanging-node configuration in T-splines is called a *T-junction*. The locally rectangular control mesh, where T-junctions exist, is called a *T-mesh* [8]. A control point $\mathbf{P}_A \in \mathbb{R}^3$, and a control weight $w_A > 0$ are assigned to every vertex of the T-mesh, where the index A denotes a global control-point counter, $A = 0, 1, 2, \dots, n$, $n \in \mathbb{N}$.
- NURBS need multiple patches with non-coinciding knot vectors to represent complex shapes, therefore complicating analysis, as well as optimisation when it comes to parametric modelling (Fig. 1.1). In addition, degenerate shapes are produced, lacking smoothness at the locations where the neighbouring patches meet and where at least G^1 -continuity is needed, as well as with reduced overall fairness. T-splines are capable of representing complex shapes with a single patch, producing models with increased smoothness (at least G^1 -continuous) and fairness.
- Trimmed NURBS are not mathematically watertight as parametric trimming curves are mathematically incapable (non-NURBS curves) to represent the intersectional curve between two trimmed surfaces, resulting to the development of non-watertight (non-solid) objects. This issue is of great importance in design engineering and optimisation, including ship design, as solid objects are a prerequisite in CFD analysis. Trimming can be avoided with T-splines, due to their advantage by using T-junctions [24] (Fig. 1.2).

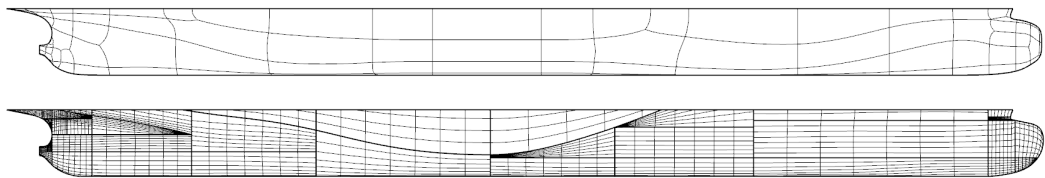


Figure 1.1: T-splines single patch (top) and NURBS multi-patch (bottom) ship-hulls.

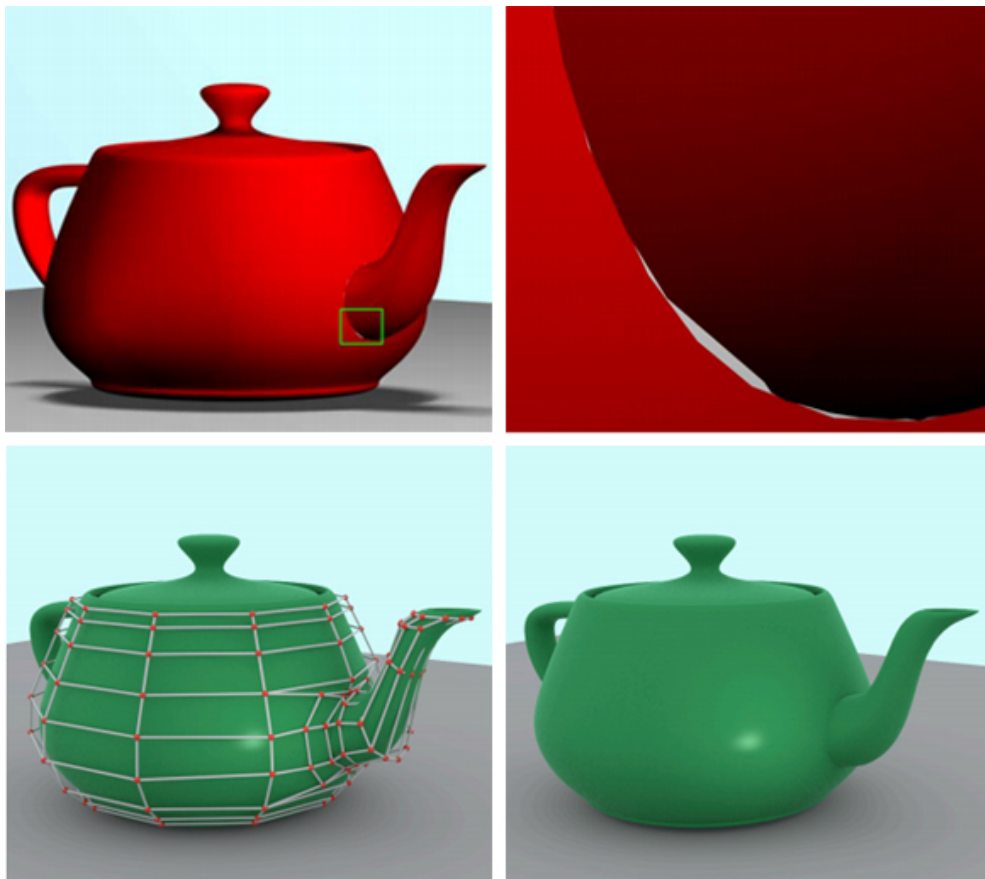


Figure 1.2: Trimmed NURBS produce non-watertight models with gaps (top), while T-splines solve this problem taking advantage of T-junctions (bottom) [24].

In this work we opt for odd-degree splines, which possess the “well known” minimum pseudo-norm property (in engineering terms: linearised-elastic-energy minimisers), each member of a T-splines basis is associated in a one-to-one manner with each of the control points. Assuming that $R_{A,p}(s, t)$ represents the A^{th} polynomial-spline basis function of bi-degree $p = 2\rho + 1$, $\rho = 1, 2, \dots$, for surfaces, the resulting T-splines surface can be represented as:

$$\mathbf{c}(s, t) = \sum_{A=0}^n \mathbf{P}_A N_{A,p}(s, t), \quad (s, t) \in \Omega \subset \mathbb{R}^2, \quad (1.1)$$

where

$$N_{A,p}(s, t) = \frac{w_A R_{A,p}(s, t)}{\sum_{J=0}^n w_J R_{J,p}(s, t)} \quad (1.2)$$

T-splines bases inherit the basic properties of NURBS bases [25] as summarised below:

- (i) **partition of unity**: $\sum_{i=1}^n N_{A,p}(s, t) = 1$ with (s, t) varying in a subdomain ω of the global domain of definition $\Omega \subset \mathbb{R}^2$;
- (ii) **non-negativity**: $N_{A,p}(s, t) \geq 0$ with $(s, t) \in \mathbb{R}^2$;
- (iii) **compact support**:

$$N_{A,p}(s, t) = 0 \text{ if } (s, t) \notin [s_{A-(\rho+1)}, s_{A+(\rho+1)}] \times [t_{A-(\rho+1)}, t_{A+(\rho+1)}]$$
 $(\rho = 1 \text{ for the typical bicubic case});$
- (iv) **continuity**: in the neighbourhood of a knot line of multiplicity k , $N_{A,p}(s, t)$ is C^{p-k} continuous;
- (v) **convex-hull property**: a T-splines $\sum_{A=0}^n \mathbf{P}_A N_A(s, t)$, $(s, t) \in \tilde{\omega} \subset \mathbb{R}^2$, lies within the convex hull of the subset of control points \mathbf{P}_A , $i = 1, \dots, n$, for which the compact support of the corresponding basis functions intersects $\tilde{\omega}$;
- (vi) **affine invariance**: a T-splines surface is invariant with respect to affine transformations;
- (vii) **boundary-curve interpolation**, via allocating appropriate multiplicity at the boundary knots, and
- (viii) **linear independence** for a topologically restricted subset of T-splines, referred to as *analysis-suitable T-splines*, optimised to meet the needs for both design and analysis; see, e.g., [22], [26], [27], [28], [29], [30], [31], [32]).

While in NURBS all basis functions are defined via the tensor product of two 1D global knot vectors, in T-splines a local knot vector is assigned to every basis function, with each local knot vector having an association with the topology of the whole patchwork of the object [33]. This difference enables T-splines to permit T-junctions, while for the same refinement (knot insertion) shorter knot lines are generated. Fig. 1.3 illustrates this fundamental difference between a NURBS mesh and a T-mesh. In this work, due to the fact that it suffices for the needs of our application to work with G^1 -continuous surfaces, we shall limit our discussion to bicubic T-splines.

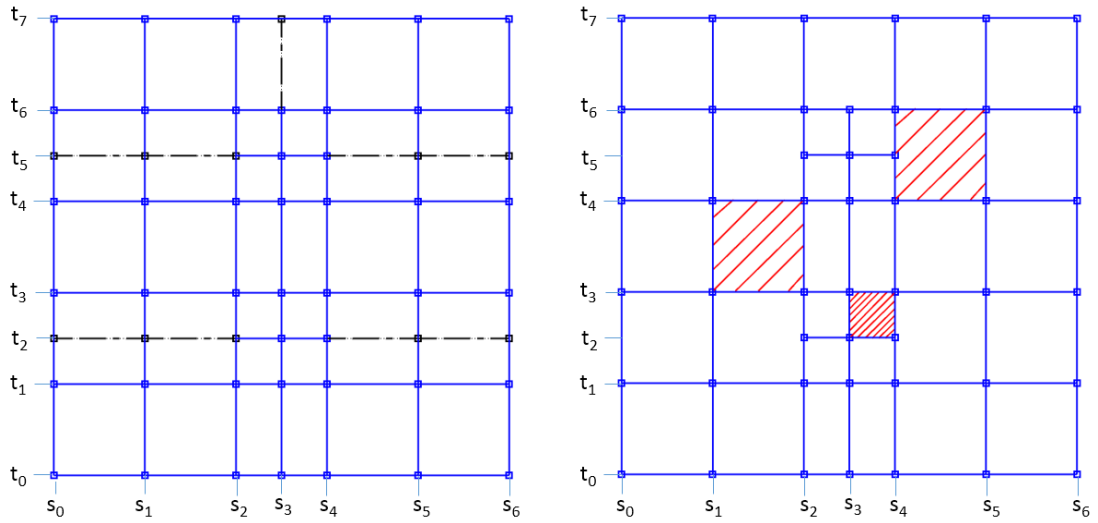


Figure 1.3: NURBS (left) knot lines lie on a global rectangular grid, while T-splines (right) can form T-junctions due to locally defined knot vectors. Extraneous knot lines and control points in NURBS are depicted in black border-style lines. Examples of parametric faces, depicted in red diagonal stripe pattern: one face (rightmost) with 5 vertices (four corner vertices and one T-junction vertex) and two faces with four corner vertices. Note that all faces, in both NURBS and T-splines meshes, are rectangular.

It is well known from CAD practice that NURBS need multiple patches to represent complex shapes, complicating the analysis and optimisation of the design. On the contrary, it is not unlikely that T-splines are capable to represent the same shapes with a single patch. In addition, refinement with NURBS generates superfluous control points, as NURBS must lie topologically on a rectangular grid, which is not the case in T-splines due to the fact that they permit T-junctions. Although, T-splines are not totally free of superfluous control points. Furthermore, in various cases, *trimming* of NURBS surfaces cannot be avoided. A curve stemmed by trimming is generally not a NURBS curve, therefore approximation is needed to represent it. On the other hand, T-splines are able to represent such surfaces without trimming with the aid of T-junctions [24]. In addition, T-splines can produce a valid merging of multiple NURBS patches into a watertight surface without gaps, see, e.g., Fig. 3 in [8].

Given a T-mesh in the parametric space and a valid knot-interval configuration, we describe the process of creating the T-splines bicubic basis corresponding, e.g., to the vertex \mathbf{V}_0 depicted in Fig. 1.4:

- (i) Starting from the vertex \mathbf{V}_0 , the horizontal knot vector is obtained by marching towards each of the two horizontal directions until two sequential edges or vertices are encountered. The corresponding univariate basis is the cubic B-spline $N_{0,3}(s)$ defined on the local knot vector $\{s_1, s_2, s_3, s_4, s_5\}$:

$$N_{0,3}(s) \equiv N[s_1, s_2, s_3, s_4, s_5](s). \quad (1.3)$$

- (ii) Create in directly analogous manner the univariate basis along the vertical direction.

$$N_{0,3}(t) \equiv N[t_3, t_4, t_5, t_6, t_7](t). \quad (1.4)$$

- (iii) The product $N_{0,3}(s)N_{0,3}(t)$ of the two univariate basis functions gives the sought-for T-splines bicubic basis function corresponding to \mathbf{V}_0 .

Note that a vertex in the parametric space corresponds to a specific control point of the T-mesh in the physical space.

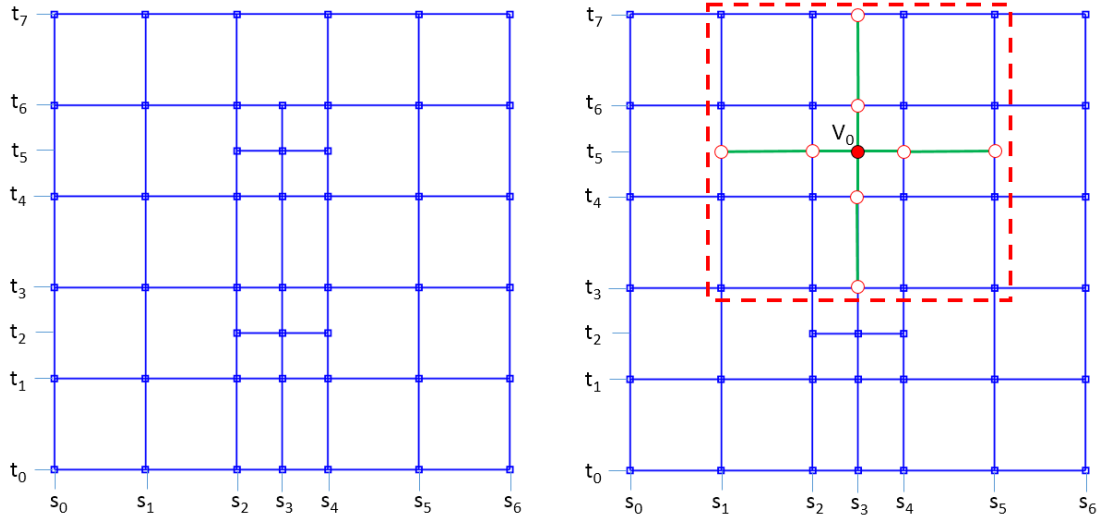


Figure 1.4: Constructing the T-splines basis function corresponding to the vertex \mathbf{V}_0 of a valid T-mesh. The extents of the constituting univariate knot vectors are colored in **green**, while the boundary of the support of the resulting basis function is depicted as dashed polygonal line, colored in **red**.

Finally, it is worth mentioning that T-splines have been successfully used in conjunction

with Isogeometric Analysis (IGA) in several areas of computational mechanics; see e.g., [6], [34], [35], [33], [36], [37], [38], [39]. IGA is a new methodology that eliminates the need of geometry discretisation (meshing), as it is the case for standard FEM (Finite Element Methods) and BEM (Boundary Element Method), enabling a seamless and strong coupling between the geometry representation and the solver ¹.

1.2 Parametric modelling

Parametric Modellers (PM) play a crucial role in shape optimisation of engineering systems, for they have to represent robustly and efficiently every solid object. In general, a PM is an algorithmic scheme depending on a set of geometric parameters, referred to as *external parameters*, accessible to the user, the number of which determines the dimension of the design space associated with the given PM. Attributing values to all members of this set, PM is expected to deliver an acceptable instance of the shape to be optimised. Acceptability is tested against the fundamental requirement that the bounding surface of the generated object should be free from self-intersections.

We are explaining the concept of parametric modelling quoting the example in [43], as Fig. 1.5 illustrates. We assume a planar solid L-shaped object, with the lengths of 4 out of 6 edges controlled by 4 independent parameters. The shape and the characteristics of the final object depend on the assigned values of those 4 parameters. Comparing the two objects at the left-hand side, one can observe that $d_1 > d'_1$ and $d_0 > d'_0$, hence the difference in their shape. Note that the topology remains the same, which is not the case when we focus at the third object at the right-hand side of the figure. Here, due to the fact that $d''_3 > d''_1$ the object is invalid, as to retain the L-shape it should stand that $d''_3 < d''_1$. A constrain therefore should be imposed to ascertain the validity of the final object.

¹The reader interested in the theoretical foundations and research on the analytical properties of T-splines, e.g., dimension of T-splines spaces, linear independence of T-splines basis functions, can appeal to an abundance of pertinent papers in the area of *Computer-Aided Geometric Design*; see, e.g., [22], [26], [27], [28], [29], [30], [31], [32], [40], [41], [42].

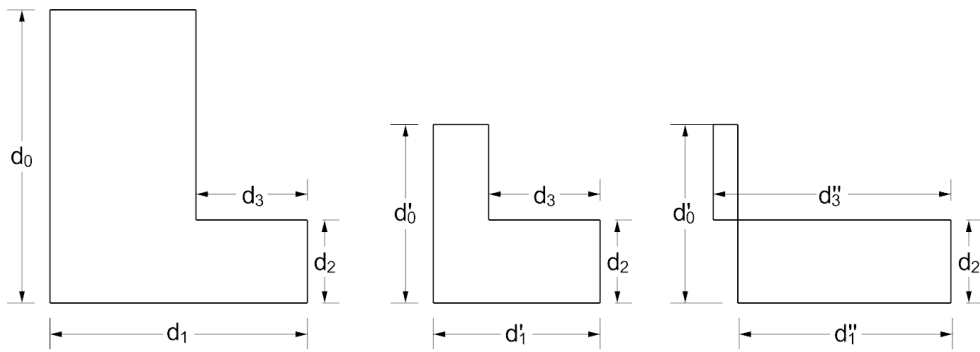


Figure 1.5: Parametric modelling: a simple example. Two different, topologically equivalent valid objects (left and middle) with $d_1 > d'_1$ and $d_0 > d'_0$. Invalid shape due to wrong parameter value assignment and lack of constraint(s) imposition.

Furthermore, various application-stemming requirements have to be met, governed by shape descriptors selected for evaluating the quality of the generated instance. Typical examples of such descriptors are the *smoothness*, which is a result of *parametric* or *geometric continuity* [44], and the *fairness*, i.e., non-oscillatory distribution of intrinsic geometric features, such as curvature, torsion, Gaussian curvature, and integral characteristics, such as moments of various orders, e.g., volume, centroids, moments of inertia.

Adopting the timeline introduced by Prof. H. Nowacki [45], who pioneered in *Computer-Aided Ship Design (CASD)* via coupling form parameters with the then novel technology of B-splines [5], the research and user communities celebrated in 2020 the key developments of CASD since its inception about six decades ago. An early attempt for building a ship parametric model is due to Lackenby [46], in which hull variants are obtained by modifying the prismatic coefficient, the centre of buoyancy, and the extent and position of the cylindrical mid-body of a parent hull. This approach has been subsequently generalised towards expanding the geometric coverage and portability of the parametric model through the use of NURBS. As a result, a number of PMs are currently available to ship-design practitioners, such as CAESES^{®2}, CATIA^{®3}, the Rhino3D-based

²<https://www.caeses.com/>

³<https://www.3ds.com/>

GRASSHOPPER^{®4}, MARIN/GMS^{®5}, NAPA^{®6} and PARAMARINE^{®7}.

NURBS-based technology is not however free of shortcomings when applied to parametric modelling of complex ship forms involving many and functionally critical regions of shape transition nature; see, e.g., [47]. One can attribute these limitations to the underlying tensor-product character of NURBS, which implies the need for building smoothly connected (at least G^1 -continuous) NURBS patches over areas with different shape characteristics (convex, flat, cylindrical, saddle) and varying scale. As a result, it is not unlikely that designers and engineers have to deal with highly complex NURBS models, involving large populations of control points, which may also need costly healing intervention before feeding them to the solver and optimiser for shape optimisation. In order to handle the challenge of seamless integration between parametric modelling and shape optimisation, more generally, integrating CAD with CAE (Computer-Aided Engineering), several alternative representations have been developed in pertinent literature, including hierarchical splines [48], PHT-splines [49], LR-splines [50] and T-splines [8], [21], [51]. T-splines constitute a generalisation of NURBS technology that removes several of NURBS deficiencies, e.g., enabling refinement without the need of adding redundant control points (see §1.1).

Regarding T-splines technology in parametric modelling, it is capable to develop modellers for complex ship forms, which provide acceptable and smooth (at least G^1 -continuous) geometries at the expense of lower complexity in comparison with PMs employing the standard NURBS-based technology. Herein, for a given set of design parameters and their ranges, we measure complexity via the number of DoF (Degrees of Freedom) required for representing the geometry of each instance, i.e., the number of control points involved in the two competing representations. Since T-splines achieve to span the same design space with considerable less DoF than NURBS, it is legitimate to assert that their efficiency, expressed as the ratio of shape-richness over DoF, is higher than that of NURBS. Furthermore, it is worth noticing that DoF downsizing is beneficial in

⁴<https://www.grasshopper3d.com/>

⁵<http://www.marin.nl/>

⁶<https://www.napa.fi/>

⁷<https://paramarine.qinetiq.com/>

the subsequent steps of shape optimisation, e.g., by decreasing analogously the size of stiffness matrices involved in the adopted FEM/BEM solver.

Chapter 2

A T-splines-based Parametric Modelling Tool for ship-hull design

2.1 Introduction

In this chapter we present the method of constructing a T-splines-based parametric modelling tool (*TshipPM*) for ship-hull design. *TshipPM*'s development has been based on two different ship-hulls: a) the MOERI KVLCC tanker [10], and b) the MOERI KCS container [9] ship-hulls. Herein we use as reference the latter to present *TshipPM*'s functionality and features. After a short introduction in the current section on the philosophy adopted for building the modeller, §2.2 presents the physical (dimensional) parameters along with a concise description of each one of them, their classification in global and non-global categories and, finally, their interrelation via the non-dimensional parameters. Lastly, §2.3 is devoted to the process of defining the geometry and connectivity (topology) of the control cage underlying the T-splines surface. Initially, we refer to the partition of the ship-hull in segments and elaborate on the concept of control curves [6], defining all such curves *TshipPM* makes use of. Additionally, we provide in detail the mapping of the user-specified parameters to the control points of the control

cage, using as an example the forward transition part of the hull and especially from the parallel middle part up to the flat of side. Finally, we illustrate the full output of TshipPM, i.e., control cage and surface, and discuss the gains with respect to the required number of control points when using T-splines versus NURBS representation.

TshipPM has been developed in two environments:

- C#, using RhinoCommon^{®1}, which is the cross-platform .NET plug-in Software Development Kit (SDK) for Rhino ^{®2};
- RhinoScript ^{®3}, Rhino's scripting functionality, based on Visual Basic[®].

It employs Autodesk[®] T-splines plug-in[®] v.4.0 for Rhino5 to create the T-splines surface using as input the *control cage* created by TshipPM. Rhino is a broadly used CAD software, based on NURBS, for modelling curves, surfaces and solids. Autodesk plug-in adds T-splines functionality to Rhino which with the support of RhinoScript generates T-splines models. In analogy to the control polygon of curves, the geometry and topology of the control cage influences decisively the shape of the underlying T-splines surface to be generated. It is a network of linear edges connecting the given control points and, in conjunction with the associated T-splines basis, it delivers the shape of the surface.

Our approach of constructing TshipPM is threefold: firstly to develop a PM that a) it is considerably flexible in representing ship-hulls, b) it is robust, meaning it delivers valid geometries for the whole of the external parameter values, and c) provides instances lying in the proximity of a given parent ship-hull in terms of geometric variation and ship-design criteria, such as up to 2nd order moments and SAC. For this purpose, TshipPM is equipped with the functionality to produce additional transition curves, enabling a finer control of the geometry construction from midship towards forward (FP) and afterward (AP) perpendiculars, as well as a flexible and -with regards to the parent ship- accurate representation of bow and stern. The above enhancements are

¹<https://developer.rhino3d.com/guides/rhinocommon/>

²<https://www.rhino3d.com/>

³<https://developer.rhino3d.com/guides/rhinoscript/>

achieved at the expense of only three parameters in comparison to [6] and a set of constraints described in 3.3. The number of control points increased from 134 in [6] to 276, nevertheless the order of magnitude remains the same, $O(10^2)$, i.e., well below than that required when using NURBS, $O(10^3)$.

2.2 Parameters: introduction and handling

TshipPM's main purpose is to generate the control cage of a ship-hull with the aid of parameters of both external and internal character. It is currently relying on 27 *physical parameters* P_i , $i = 0, \dots, 26$, which characterise the geometry of the control cage under construction. These are dimensional quantities and are classified in global and non-global, the latter being further categorised in 5 groups, according to which part of the ship they belong; see Table 1. Besides the first three physical parameters, $P_0 = \text{Lwl}$ (length of the waterline), $P_1 = \text{B}$ (beam) and $P_2 = \text{T}$ (draft), the remaining physical parameters are defined according to the following scheme:

$$P_i = \hat{P}_i \cdot f_i(P_0, \dots, P_{i-1}), \quad p_i \in (0, 1), \quad i = 3, \dots, 26, \quad (2.1)$$

where \hat{P}_i are the *non-dimensional parameters* ranging in $[0.01, 0.99]$, and f_i are linear functions of the physical parameters. In matrix form the above relations can be represented as:

$$\mathbf{P}' = \text{diag}(\hat{\mathbf{P}})(\mathbb{A}\mathbf{P}), \quad \mathbf{P}' = [P_3, \dots, P_{26}]^T, \quad \hat{\mathbf{P}} = [\hat{P}_3, \dots, \hat{P}_{26}]^T, \quad (2.2)$$

where $\text{diag}(\hat{\mathbf{P}})$ denotes the diagonal matrix defined by the non-dimensional parameters \hat{P}_i , and \mathbb{A} is a rectangular matrix of block-triangular structure. Both physical, P_i , and non-dimensional, \hat{P}_i , parameters are of external (user-visible) nature. On the contrary, *internal parameters*, which range in $(0, 1)$, are accessible only to the developer and are deployed for shape stabilisation via retaining key-shape characteristics of the type of the hull the designer is interested in. Their number varies, depending on the complexity and the number of control curves which identify challenging areas, such as bow and

stern, but also areas of shape transition between the midship and the bow/stern areas of the ship. Internal parameters are not inter-dependent to each other or to any external parameters. Once parameters values are provided to the first three physical and the 24 non-dimensional parameters, Autodesk's T-splines plug-in is called to create the smooth surface, using as input the control cage created by TshipPM to produce the final T-spline model. Fig. 2.1 provides a representative diagram of the execution process.

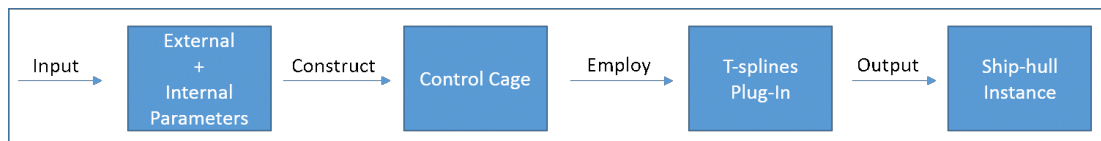


Figure 2.1: Steps to create a TshipPM instance: we first attribute values to $P_0 = \text{Lwl}$, $P_1 = \text{B}$, $P_2 = \text{T}$ and the non-dimensional parameters $\hat{P}_i \in (0, 1)$, $i = 3, \dots, 26$, which, alongside the internal parameters, determine the control cage of the ship-hull; TshipPM plug-in is then called to produce the smooth surface of the final instance.

Table 2.1: TshipPM parameters and groups; P_i 's: physical parameters.

| P_i 's | Description | Linear functions f_i of |
|------------------------------|--|---------------------------|
| Global Parameters | | |
| $P_0 = \text{Lwl}$ | Waterline Length | - |
| $P_1 = \text{B}$ | Beam | - |
| $P_2 = \text{T}$ | Draft | - |
| non-Global Parameters | | |
| Mid_Part Parameters | | |
| $P_4 = \text{Mid_Pos}$ | Longitudinal middle position of Middle Body | Lwl |
| $P_3 = \text{Mid_L}$ | Length of Middle Body | Lwl, Mid_Pos |
| $P_5 = \text{BilgeR}$ | Bilge Radius | B or T |
| Fwd_Part Parameters | | |
| $P_6 = \text{BatFP}$ | Breadth at FP | B |

Continued on next page

Table 2.1 – *Continued from previous page*

| P_i 's | Description | Linear functions f_i of |
|----------------------------|--|------------------------------|
| P_7 =FoS_Fwd_L | Flat of Side | Mid_Pos, Mid_L |
| P_8 =FoB_Fwd_L | Flat of Bottom | Mid_Pos, Mid_L |
| P_9 =FoS_Trans | Height between keel and FoS at the transition of Midship to Fwd_Part | T, BilgeR |
| P_{10} =FoB_Trans | Breadth between keel and FoB at the longitudinal position where FoB ends | B |
| P_{11} =FP_L_fromMS | Length from Mid Body to FP | Mid_Pos, Mid_L, FoB_Fwd_L |
| Bow Parameters | | |
| P_{12} =FwdRise | Rise of Fwd_Part | Bulb_H |
| P_{13} =BulbL | Bulbous Length (from FP) | Lwl |
| P_{14} =BulbH | Bulbous Height | T |
| P_{15} =BulbB | Bulbous Width | B |
| P_{16} =BulbTipH | Height of tip of Bulbous | Bulb_H |
| Aft_Part Parameters | | |
| P_{17} =BatAP | Breadth at AP | B |
| P_{18} =BatAPLow | Breadth at the bottom at AP | B |
| P_{19} =FoS_Aft_L | Flat of Side | Lwl, Mid_Pos, Mid_L |
| P_{20} =FoB_Aft_L | Flat of Bottom | Lwl, Mid_Pos, Mid_L |
| P_9 =FoS_Trans | Height between keel and FOS at the transition of Midship to Aft_Part | T, BilgeR |
| P_{10} =FoB_Trans | Breadth between keel and FoB at the longitudinal | B |

Continued on next page

Table 2.1 – *Continued from previous page*

| P_i 's | Description | Linear functions f_i of |
|-------------------------|--|--|
| | position where FoB ends | |
| P_{21} =AP_L.FromMS | Length from Middle Body to AP | Lwl, Mid_Pos, Mid_L, FoB_Aft_L |
| Stern Parameters | | |
| P_{22} =TransomH | Transom Height | T |
| P_{23} =TransomB | Transom Breadth | BatAP |
| P_{24} =TubeL | Tube Length | Lwl, Mid_Pos, Mid_L, FoB_Aft_L, AP_L.FromMS |
| P_{25} =TubeR | Tube Radius | T |
| P_{26} =SternFullness | Bulkiness of Stern at the waterline | Lwl, B |

TshipPM favours the use of non-dimensional parameters, where possible, in order to handle effectively the interdependency that occurs among them, as it is the case in Eq. 2.1. For example, if we define ship's draft T as a physical parameter, the permissible value would be upper-bounded by the ship's depth D . On the other hand, by employing the ratio parameter: $r = \frac{T}{D}$, its valid range bounds are 0 and 1 regardless of ship's depth value, which permits a simpler checking algorithm for the validation of input parameter values. The benefits of this approach are more evident when we have a series of interdependent parameters, as in the case of ship's midship part definition in the example below.

Let us consider a ship of length l , and two points defining the starting and terminating points of the midship part respectively. Furthermore, let l_S and l_E denote the corresponding distances of each of these points from the common origin O; see Fig. 2.2. If we were to implement l_S and l_E as physical parameters, we would have to include validation checks so that, for example, the inequality $0 < l_S < l_E < l$ stands true,

whereas, by introducing ratio parameters $r_E = \frac{l_E}{l}$ and $r_S = \frac{l_S}{l_E}$, we ensure the right order of the corresponding points by simply bounding both ratios in $(0,1)$; see Fig. 2.2. Obviously, in this case, no further validation checking is required.

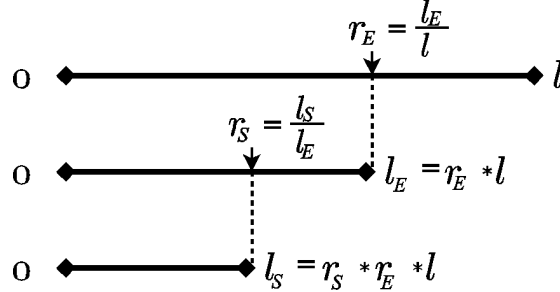


Figure 2.2: Relation between physical (dimensional) and non-dimensional parameters.

Thus, the use of non-dimensional parameters, ranging in subintervals of $(0, 1)$, supports the robustness of TshipPM, by avoiding the allocation of values to parameters that would result in delivering non-valid geometries suffering, e.g., from self-intersections. These subintervals are currently pre-specified by the developer via experimentation with the available parent hulls. Figs. 2.3 and 2.4 illustrate all physical parameters used in TshipPM. In the rest of the text the term *parameters* will be used for referring only to the three physical and the twenty-four non-dimensional parameters, denoted by the parameter name and a wide hat symbol, e.g., \widehat{BulbL} denotes the non-dimensional parameter corresponding to the physical parameter of bulbous-bow length, $BulbL$. Note that parameters $\widehat{FoS_Trans}$ and $\widehat{Fob_Trans}$ are used both in forward and afterward parts, but their corresponding lengths differ, as an internal parameter has been applied to both of them for the afterward part.

2.3 From parameters to the control cage and the output surface

The shape of a typical ship-hull consists of three main parts, namely, a) the *midship*, which is the part bounded by FP and AP, b) the *bow*, which extends from FP to the forward end of the hull, and c) the *stern*, extending from AP and afterwards. Midship consists of the cylindrical (parallel) middle part (Mid_Part), the forward (Fwd_Part)

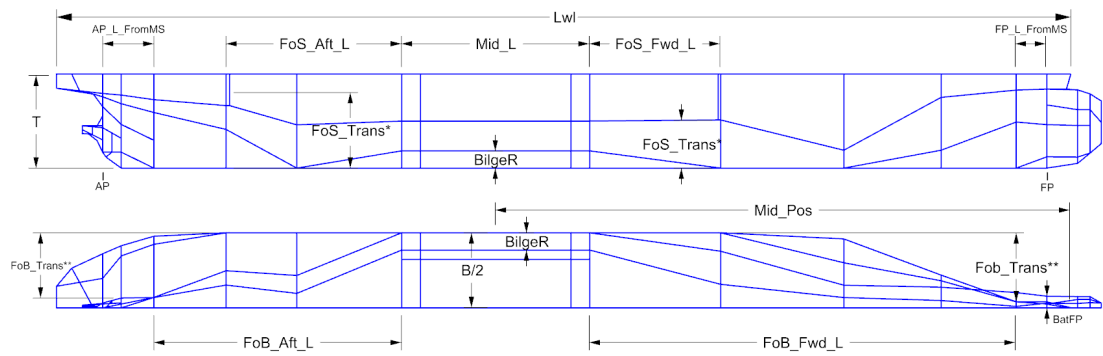


Figure 2.3: TshipPM input parameters for Fwd.Part and Aft.Part.

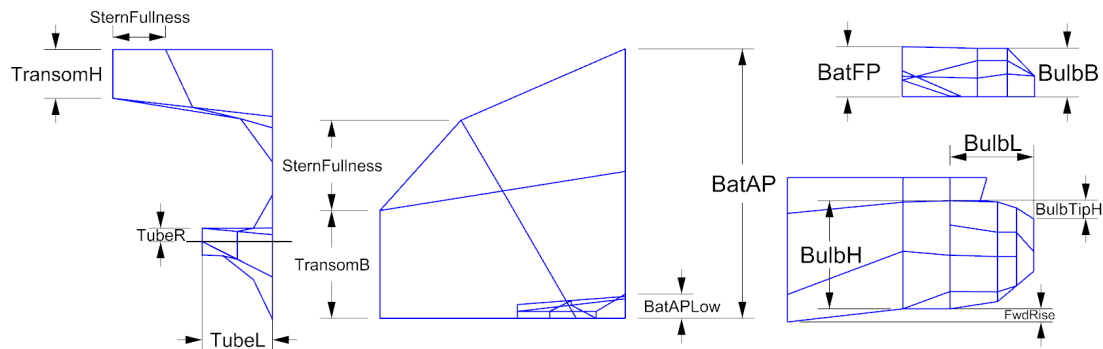


Figure 2.4: Bow and Stern Parameters.

and the afterward (Aft.Part) parts. To describe adequately the main parts of a ship-hull, global physical parameters are needed, such as Lwl, B and T. This multi-level segmentation of the ship-hull should be reflected on the control cage; see Figs. 2.5 and 2.6.

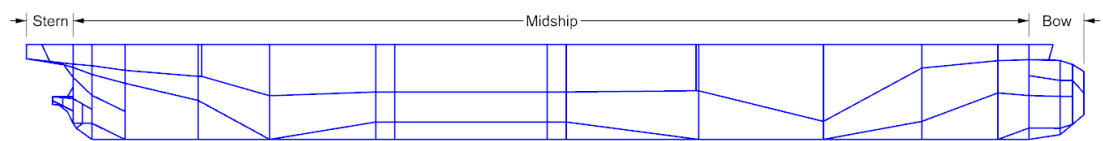


Figure 2.5: Typical segmentation of a ship-hull. Main parts are midship, bow, and stern.

In order to give shape to the bounding surfaces of a ship-hull and guarantee a fair transition from a part to its neighbouring ones, *control curves* are required. Control curves are of two types, namely, *bounding* and *transition curves*. Bounding curves, such as bow (BowPrfCrv) and stern (SternPrfCrv) profile curves, are used to specify

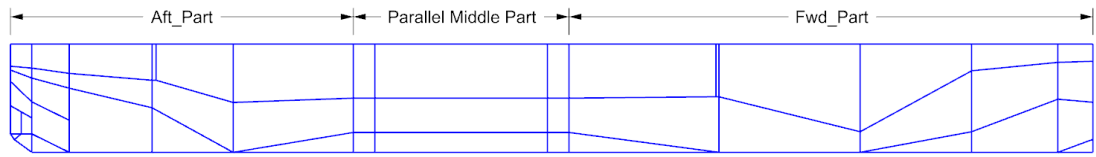


Figure 2.6: Segmentation of midship, consisting of the cylindrical (or parallel) middle part, forward and afterward Parts.

the boundaries of the ship-hull, while transition curves are employed to indicate the transition between different parts and/or further segmentation of the same part. Pre-images of these curves on the control cage level are introduced by the so-called *control cage paths*, as illustrated in Fig. 2.7.

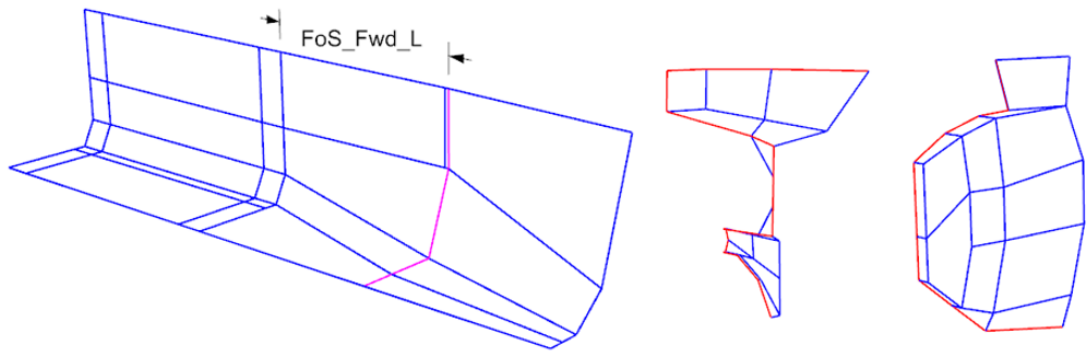


Figure 2.7: Examples of control cage paths of transition type (left: from flat of side to forward part, depicted in magenta) or boundary type (middle: stern boundary, right: bow boundary, depicted in red).

An example of determining control cage paths using both internal and external parameters is given in Fig. 2.8, where it is assumed that the extent of Mid.Part is already fixed. The external parameter controlling the longitudinal transition from the terminating point of Mid.Part to the forward transition curve at FoS_Fwd_L, is parameter $\widehat{FoS_Fwd_L}$, while the forward transition curve itself makes use of parameters FoS_Trans, FoB_Trans and the internal parameters a_1 and a_2 .

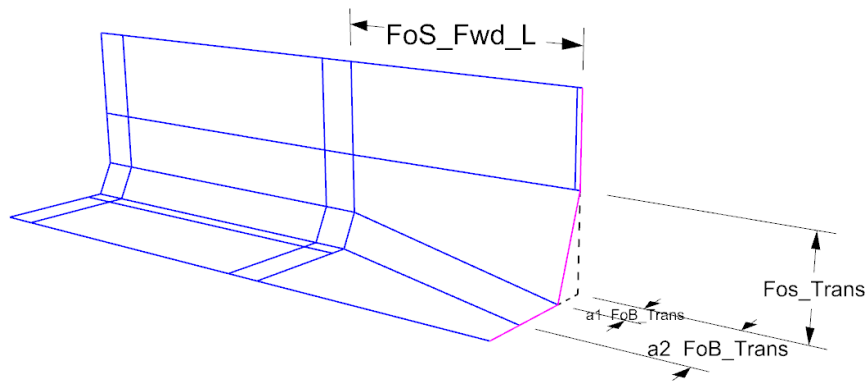


Figure 2.8: Control cage path of forward transition curve (magenta) and its defining parameters.

We now elaborate on and define the control curves used by TshipPM with the aid of Figs 2.9 to 2.18. Distances, which are linear functions of both external and internal parameters, are denoted as functions of the form: $f(\text{“physical parameter”})$, while dimensions which are dependent only on one external parameter, without any internal parameters applied, are denoted as “physical parameter”. For example, for FTCD in Fig. 2.12, the horizontal distance between P_A and P_B is defined by both Fob_Trans (physical dimension) and an internal parameter, therefore it is noted as $f(Fob_Trans)$. On the contrary, the horizontal distance between P_B and the dash-dot vertical line on the right hand side (as Fig. 2.11 indicates the two vertical dash-dot lines define the breadth of the ship-hull), determines the transition dimension for the end of the flat of bottom, and it is not a function of any internal parameter but only of the external parameter FoB_Trans , hence the scheme $f(\text{“physical parameter”})$ is not needed. Note that, in the same figure, the repetition $f(Fob_Trans)$ denotes that each of these functions use a different internal parameter. This is the case in all Figs from 2.9 to 2.18.

Each transition curve in the context of the control cage is a set of contiguous straight lines (polylines). For the description of the control curves we are segmenting the ship-hull into 5 segments, namely parallel middle part (Mid_Part), forward part (Fwd_Part), bow, afterward part (Aft_Part), and stern as Fig. 2.9 illustrates.

Parallel middle part consists of four identical U-shaped polylines, two of them framing the parallel middle part and another two located slightly closer towards the Mid_Pos of

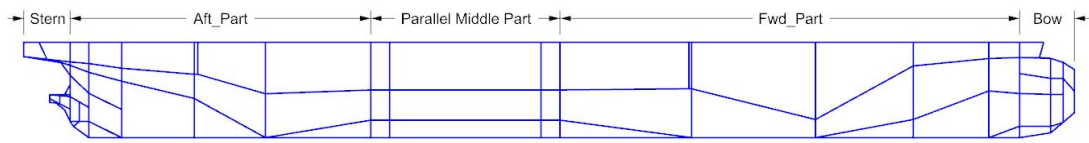


Figure 2.9: Segmentation of the ship-hull into five segments.

the ship-hull. Since TshipPM uses bicubic T-splines, four collinear control points are needed (or three collinear line segments) to create a straight line along one parametric direction. This set-up, in addition to the fact that the four polylines are identical, ascertains the middle part is cylindrical. Each transition curve is constructed by the use of three external parameters, as Fig. 2.10 illustrates, and it is comprised of six coplanar control points (five contiguous coplanar line segments).

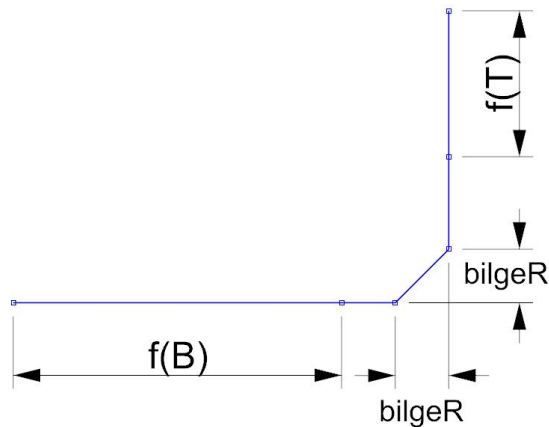


Figure 2.10: U-shaped polylines and control points of the parallel middle part and their interdependency with physical parameters.

Forward and afterward parts consist of four transition curves each. Forward part transition curves (FTC) FTCA, FTCB, FTCC, and FTCD are formed by five coplanar control points. FTCA uses parameters T, B, FoB_Trans and FoS_Trans, while the rest of the FTCs make use of the first three of them. FTCA and FTCD are located at the longitudinal position of the end of the flat of side and flat of bottom respectively, while the rest of the transition curves are in between. Note that the vertical blue line segment exactly before the magenta-depicted polyline in Fig. 2.8 is necessary to ascertain the flatness of the flat of the side. The same applies at the afterward part. Figs 2.11 and

2.12 illustrate all four forward transition curves.

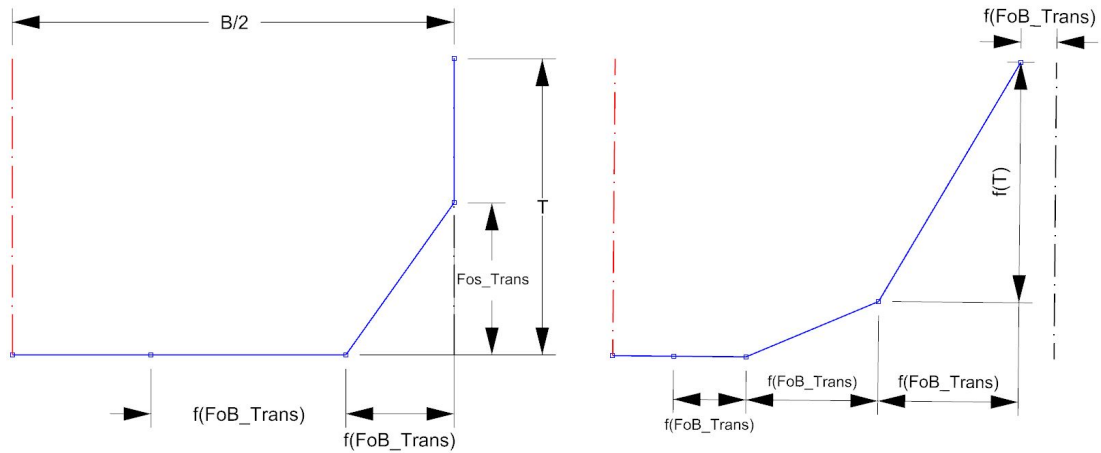


Figure 2.11: First (FTCA) and second (FTCB) forward transition curves. FTCA is located at the longitudinal position where the flat of side of the forward part ends.

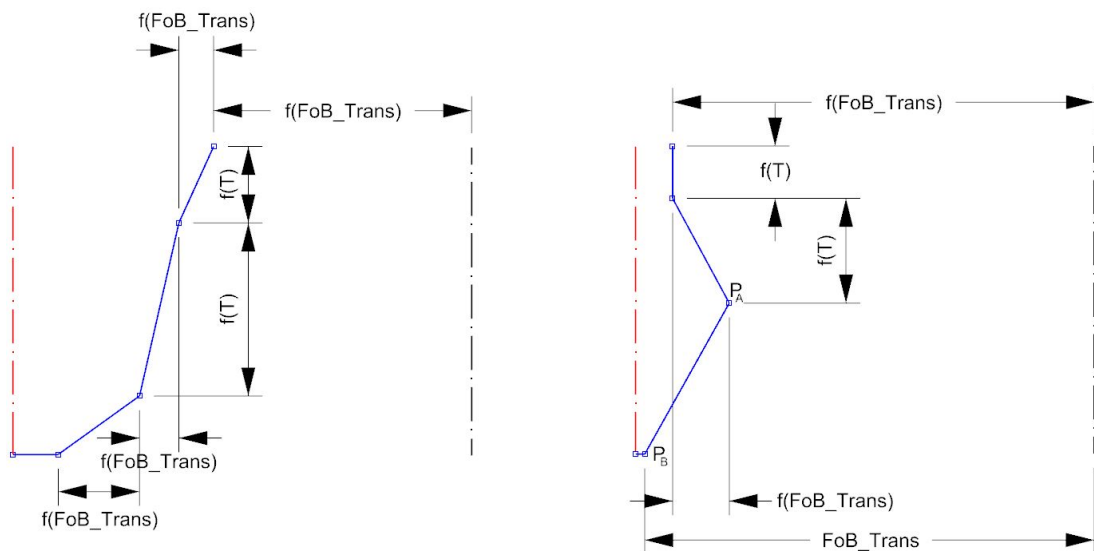


Figure 2.12: Third (FTCC) and fourth (FTCD) forward transition curves. FTCD is located at the longitudinal position where the flat of bottom of the forward part ends.

Afterward part transition curves (ATC) ATCA, ATCB, and ATCC consist of five coplanar control points, while ATCD consists of six. More control points are needed for the fourth transition curve as the complexity of hull's surface increases as we approach the stern. First three curves use parameters T , B , FoB_Trans , FoS_Trans , and TransomH , while FTCD makes use of all of them except from FoS_Trans . ATCB and ATCD are

located at the longitudinal position of the end of the flat of side and flat of bottom respectively, while ATCA is between the middle part and ATCB, and ATCC between ATCB and ATCD. Figs 2.13 and 2.14 illustrate all four afterward transition curves.

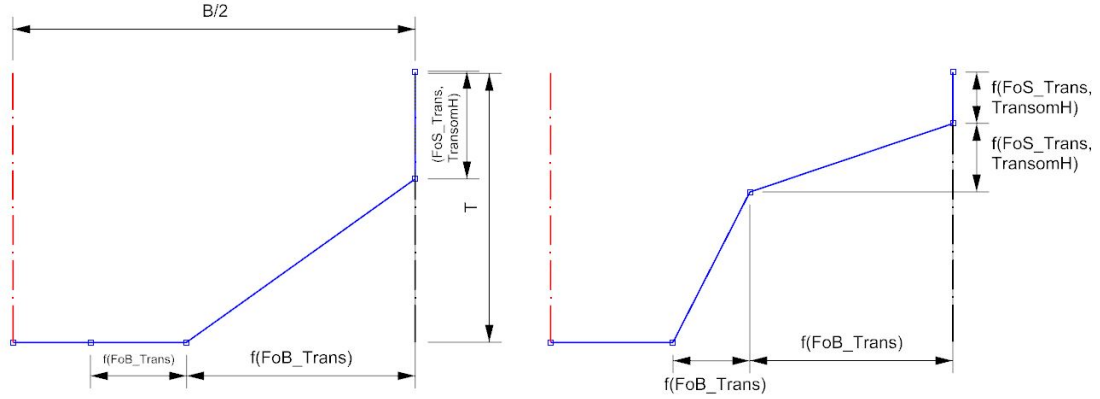


Figure 2.13: First (ATCA) and second (ATCB) afterward transition curves. ATCB is located at the longitudinal position where the flat of side of the afterward part ends.

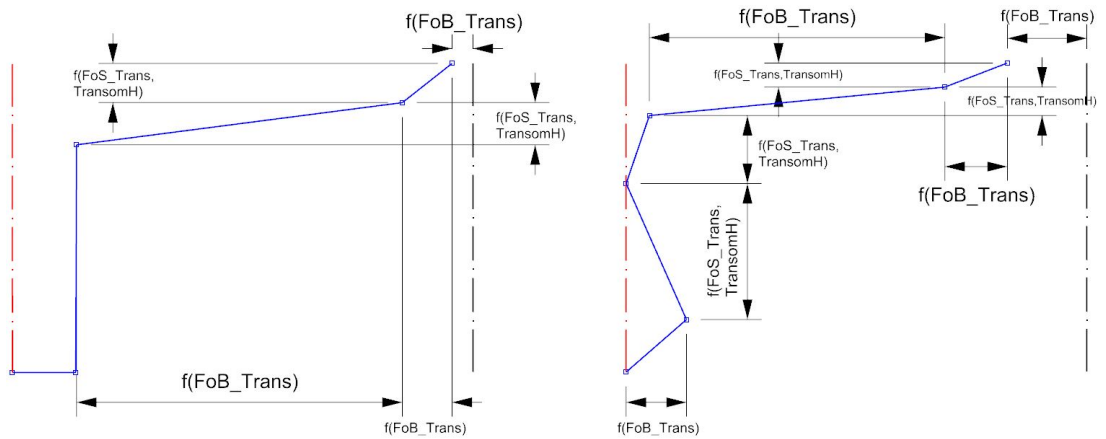


Figure 2.14: Third (ATCC) and fourth (ATCD) afterward transition curves. ATCD is located at the longitudinal position where the flat of bottom of the afterward part ends.

Bow consists of three transition curves, namely the forward perpendicular transition curve (FP_Crv) and two bulb transition curves (BulbCrvA and BulbCrvB), and one boundary curve, the BowProfCrv. FP_Crv is consisted by six coplanar control points and uses parameters T , B , $BatFP$, $BulbH$, $BulbB$, and $FwdRise$. More control points are needed for the FP_Crv transition curve as the surface of the hull at this location and

towards the bow becomes more complex compared with the regions at Fwd.Part. Bulb curves are comprised of six coplanar control points which are defined by parameters B, BulbB, BulbH, and FwdRise. Finally, BowProfCrv consists of nine coplanar control points along the longitudinal direction and uses parameters BulbL, BulbH, BulbTipH, and FwdRise. The top line segment of the BowProfCrv has also been offset by a function of BatFP (details are shown in Fig. 4.10, subsection 4.3.4.1). See Figs 2.15 and 2.16 for the control curves of the bow part.

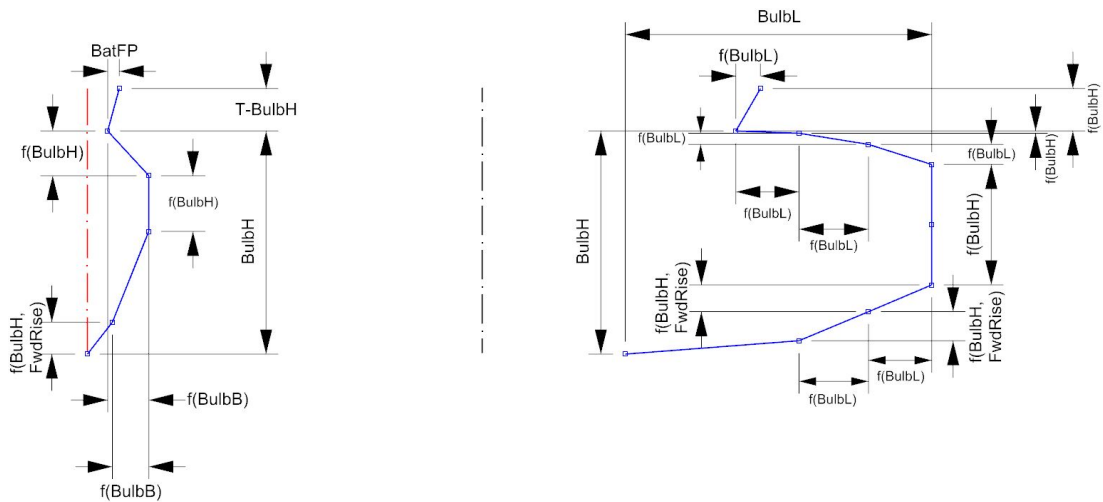


Figure 2.15: Forward perpendicular (FP_Crv) and bow profile (BowProfCrv) curves.

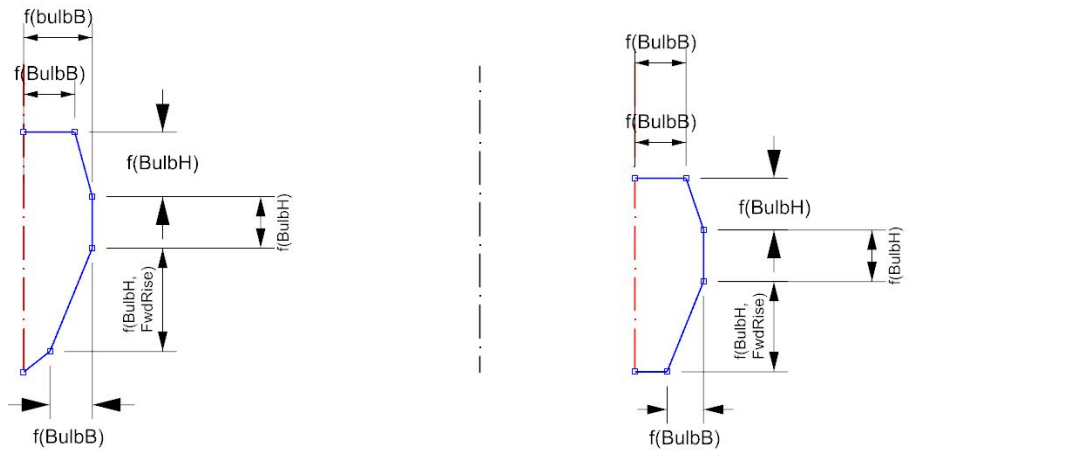


Figure 2.16: Bulb Curves.

Stern consists of the afterward perpendicular transition curve (AP_Crv), and the SternPrfCrv. AP_Crv is formed by eight coplanar control points, making use of parameters T, B, BatAP, BatApLow, TransomH for its representation, while SternPrfCrv is consisted of twenty-two control points which are not coplanar and it is represented using parameters Lwl, B, T, TransomB, TransomH, and SternFullness. The latter is responsible for the location of one control point which runs along the red straight line (P_0, P_1) (see Fig. 2.18) according to the following equation:

$$\mathbf{r}(t) = (1 - c) \cdot \mathbf{P}_0 + c \cdot \mathbf{P}_1, \quad (2.3)$$

where $r(t)$ is the location of the control point on the xy -plane and $c = \widehat{SternFullness}$. SternPrfCrv includes eight control points which construct the tube profile curve, using parameters T, TubeL, and TubeR. Figs 2.17 to 2.18 depict AP_Crv and three different views of SternPrfCrv.

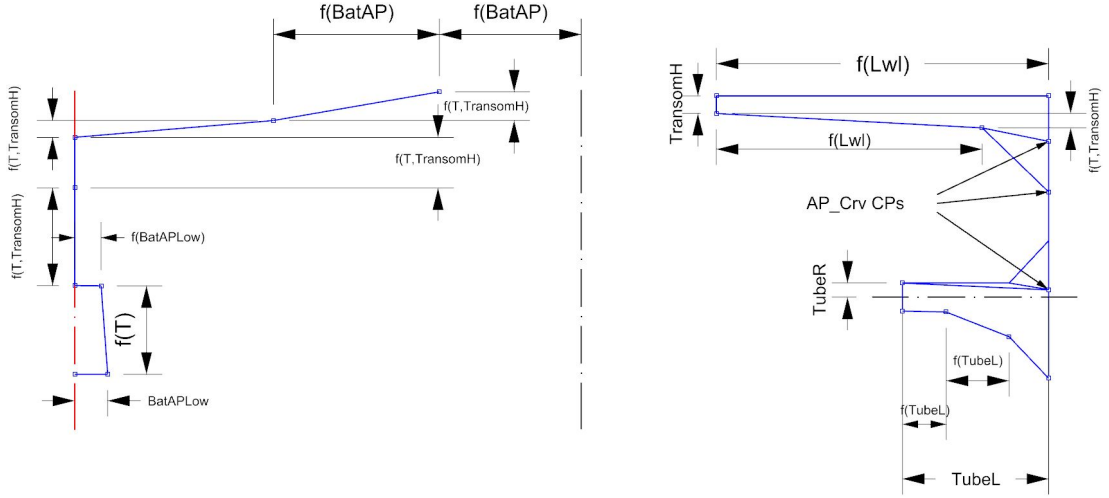


Figure 2.17: Afterward perpendicular (AP_Crv) and stern profile (SternPrfCrv) curves.

Let us now describe in more detail the process adopted by TshipPM for mapping user-defined parameters to the control points of the control cage for the configuration depicted in Fig. 2.19, associated with the challenging transition area from the middle body towards FP. To proceed, we assume that the control points $\mathbf{M}_{0,i} = (M_{0,i}^x, M_{0,i}^y, M_{0,i}^z)^T$,

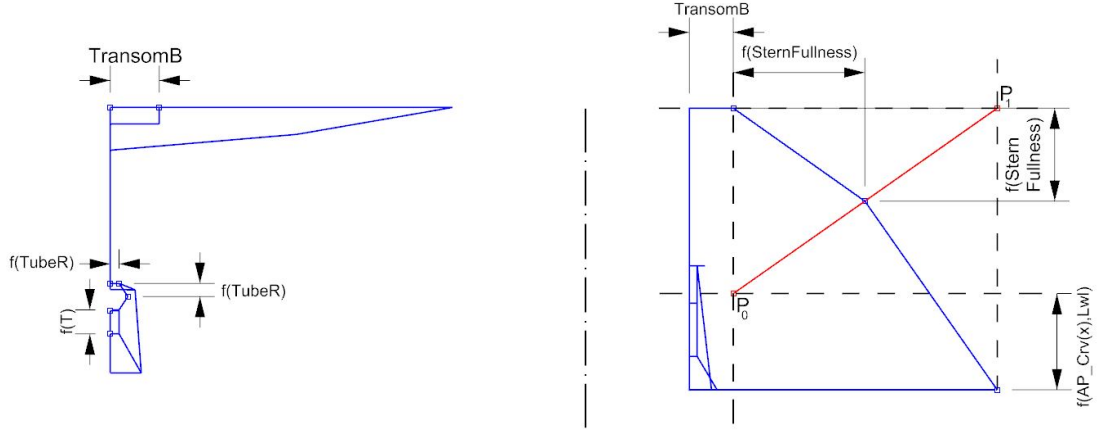


Figure 2.18: Two more views of the stern.

$i = 0, \dots, 5$, of the U-shaped control polygon \mathcal{M}_0 (depicted in black) have been calculated, adopting the choices made in Fig. 6 in §2 of [6], and we aim to compute the control points of the first two FTCs, namely FTCA (depicted in red) with control points $\mathbf{P}_{0,i}, i = 0, 1, \dots, 5$ and FTCB (depicted in green) with $\mathbf{P}_{1,i}, i = 0, 1, \dots, 5$. TshipPM calculates the location of the afore mentioned control points using the below formulae (2.4)-(2.8) for FTCA and (2.9)-(2.13) for FTCB.

$$\mathbf{P}_{0,0} = \mathbf{M}_{0,0} - (FoS_Fwd_L, 0, 0)^T, \quad (2.4)$$

$$\mathbf{P}_{0,1} = \mathbf{P}_{0,0} - (0, 0, FoS_Trans)^T, \quad (2.5)$$

$$\mathbf{P}_{0,2} = (M_{0,0}^x - FoS_Fwd_L, O^y - a_0 \cdot FoB_Trans, O^z - T)^T, \quad (2.6)$$

$$\mathbf{P}_{0,3} = (P_{0,2}^x, O^y - a_1 \cdot FoB_Trans, P_{0,2}^z)^T, \quad (2.7)$$

$$\mathbf{P}_{0,4} = (P_{0,3}^x, O^y - \frac{B}{2}, P_{0,3}^z)^T. \quad (2.8)$$

$$\mathbf{P}_{1,0} = \mathbf{M}_{0,0} - (a_2 \cdot FoB_Fwd_L, 0, 0)^T, \quad (2.9)$$

$$\mathbf{P}_{1,1} = (P_{1,0}^x, P_{1,0}^y - a_3 \frac{B}{2}, O^z - a_4 \cdot T)^T, \quad (2.10)$$

$$\mathbf{P}_{1,2} = (M_{0,0}^x - a_2 \cdot FoB_Fwd_L, O^y - a_5 \cdot FoB_Trans, O^z - T)^T, \quad (2.11)$$

$$\mathbf{P}_{1,3} = (P_{1,2}^x, O^y - a_6 \cdot FoB_Trans, P_{1,2}^z)^T, \quad (2.12)$$

$$\mathbf{P}_{1,4} = (P_{1,3}^x, O^y - \frac{B}{2}, P_{1,3}^z)^T. \quad (2.13)$$

where \mathbf{O} denotes the origin $(0, 0, 0)^T$ of the coordinate system, and a_i , $i = 0, 1, \dots, 6$, are internal parameters with $a_0 < a_1$, and $a_5 < a_6$. Regarding the functionality of these inequalities, they control the relative spatial positioning of control points, e.g., $a_0 < a_1$, secures that $\mathbf{P}_{0,3}$ is placed further than $\mathbf{P}_{0,2}$ and towards $\mathbf{P}_{0,4}$. Furthermore, formulae (2.4)-(2.13) readily imply that mapping user-defined parameters to control points is done through affine mappings.

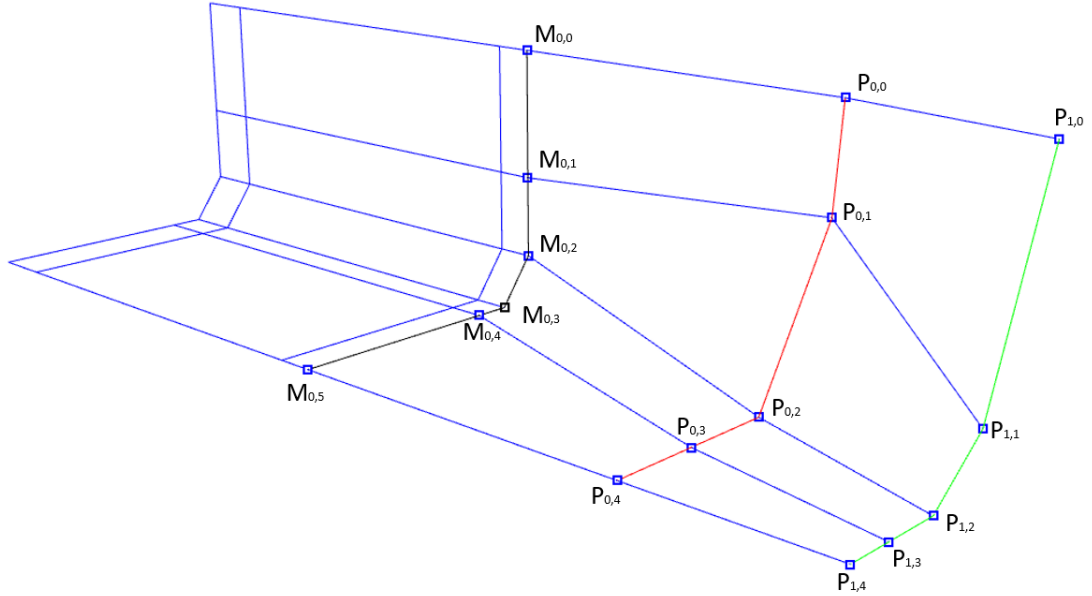


Figure 2.19: Control points of the U-shaped polygonal line \mathcal{M}_0 (depicted in black) and the first two Forward Transition Curves, FTC_0 (depicted in red) and FTC_1 (depicted in green).

Once the control polygons of each of the transition curves FTC_i , $i=0,1$ are defined, the connectivity between them and the vertices of the U-shaped control polygon \mathcal{M}_0 is set

as shown in Fig. 2.19. Despite the fact that all three control polygons do not share the same number of points, T-splines functionality enables us to circumvent this issue by introducing T-junctions (valence-3 control points), in this case at $\mathbf{M}_{0,3}$. Between transition curves with the same number of control points, as is the case for FTC_0 and FTC_1 of the same figure, we connect each control point of each transition curve with only one control point of the other, ending with connecting line segments of the form $\overline{\mathbf{P}_{0,i}\mathbf{P}_{1,i}}$, where \mathbf{P}_0 and \mathbf{P}_1 are the two transition curves, and $i = 0, 1, \dots, n$ is n the number of the control points of each one of them. Working in this way we are able to retain, in most cases, quadrilateral topology, i.e., create quadrilateral elements, to reduce the complexity and increase the quality of the developed surfaces.

In case a control point is necessary to be connected with fewer or more than 4 edges, *extraordinary points* have to be introduced. For control points on the boundary of the T-mesh, but not on the corners, extraordinary points are created when their valence is higher than 3, while for corner control points, extraordinary are called the points which are connected with more than 2 lines. An example of a control point of valence-5 is shown in Fig. 2.20. \mathbf{b}_0 is an extraordinary point on the FP_Crv (depicted in magenta), connected with \mathbf{b}_1 of the first BulbCrv (depicted in red).

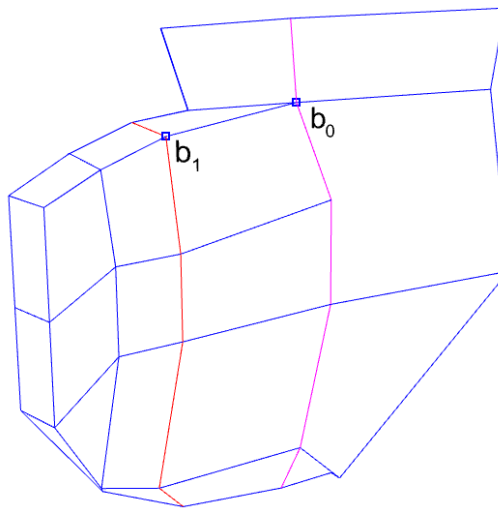


Figure 2.20: An example of a valence-5 extraordinary point. \mathbf{b}_0 is located on the Forward Perpendicular Transition Curve (depicted in green) and is connected with \mathbf{b}_1 of the first Bulb Curve (depicted in red).

The computed control cage is then fed to the T-splines plug-in, which in its turn delivers the corresponding ship-hull instance, represented by one single T-spline polysurface. Figs. 2.21-2.23 depict a typical output, with hollow rectangles indicating the control points \mathbf{P}_A , $A = 0, \dots, n$, of the obtained T-splines hull surface along with their connectivity (dashed lines, showing the control cage), while solid lines constitute the mapping of T-mesh edges on the surface.

Considering the control points \mathbf{P}_A , $A = 0, \dots, n$, as the DoF (degrees of freedom) of the surface representation (1.1), their population of which ($= n + 1$) is a measure of the computational complexity of the resulting surface. For the surface in Figs. 2.21-2.23, $n+1 = 157$ (for the whole hull $n+1=276$). On the contrary, DoF of parent ship-hull and is approximately 5,000. Since every T-splines surface can be accurately represented as a NURBS multi-patch surface, one can experience the DoF overloading effect of NURBS by transforming T-splines into NURBS and count the control points. For the surface in question this operation leads to a multi-patch NURBS surface with approximately 18,000 control points.

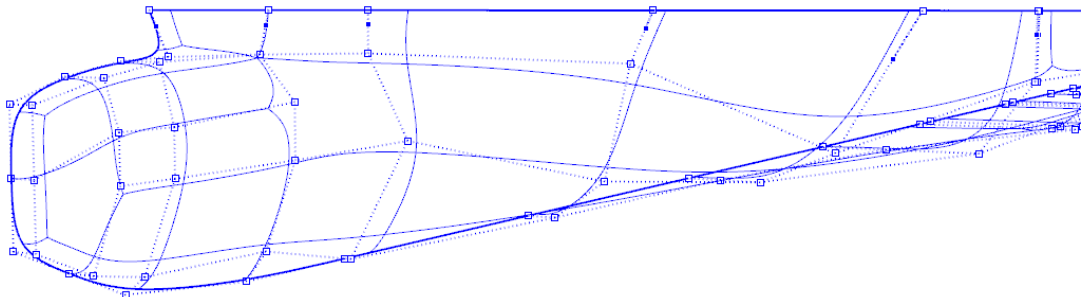


Figure 2.21: TshipPM output: perspective view from the bow with control points and on-surface mappings of T-mesh edges.

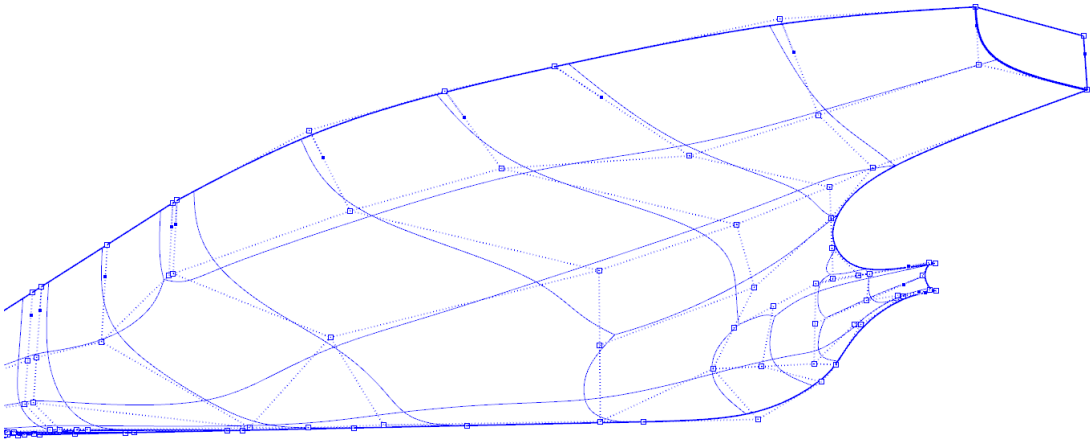


Figure 2.22: TshipPM output: perspective view from the stern.

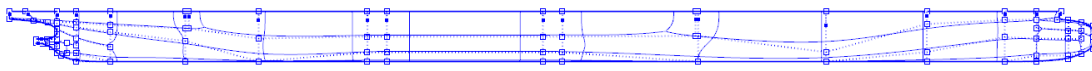


Figure 2.23: TshipPM output: side view.

Chapter 3

Geometric Validity of TshipPM and Data Analysis

3.1 Introduction

The question about PMs' robustness and their capability to produce valid geometries for the whole design space they cover, as well as which parameter values may lead to non-valid models, are "long standing" problems in parametric CAD design, which are easy to state but difficult to solve; see, e.g., [43]. Herein, by *design space* we refer to the subset of \mathbb{R}^{27} , determined by the three physical parameters ($P_0=Lwl>0$, $P_1=T>0$, $P_2=B>0$) and the remaining physical parameters P_i , $i=3,\dots,26$ (see Table 2.1) defined by the linear functions of Eq. 2.1. In this chapter we elaborate on the constraints imposed to TshipPM to achieve high levels of robustness regarding to the geometric validity of the outputs. Firstly, in §3.2 we introduce the geometric validity evaluation criteria of the outputs and the auxiliary terms. Afterwards, in §3.3 we elaborate on the design constraints imposed to TshipPM, while in §3.4 we illustrate the behaviour of TshipPM against strong parameter values variation with respect to a parent ship-hull. Lastly, in §3.5 we assess the robustness of TshipPM against a large set of 100,000 models for each of the KCS and KVLCC ship-hulls, and we analyse the data collected.

3.2 Definitions and geometric validity evaluation criteria

Constraints in CAD are limitations set either by the nature of the model (e.g. symmetry, perpendicularity, etc.), known as *geometric constraints*, or by the designer to achieve the desired end product, the so-called *design constraints*. We define as *geometric validity* of complex shapes their attribute to be represented by a single surface or a set of surfaces capable to form watertight (or solid or closed or not open) objects, therefore presenting no unnecessary trimmings (holes), to be properly welded (no naked edges), and for which no self-intersections (intersection of neighbouring surfaces) occur.

We explore the geometric validity of the produced instances with the aid of Rhino5. As TshipPM creates one single polysurface and not a number of surfaces, it's out of our interest to explore areas where neighbouring surfaces are not properly welded. Hence, we focus our evaluation on two criteria, namely a) self-intersections and b) trimmings. The evaluation is being conducted using two techniques:

A) We determine whether a model is geometrically valid or not by exploring its watertightness. Open objects can occur due to two main reasons: a) the presence of trimmings (and/or unwelded surface edges which are out of our scope) as a result of topologically incorrect control cages (Fig. 3.1), and b) as an end product of self-intersections. These objects, in most cases, can be identified automatically by Rhino using the *isObjectSolid* property. Although, there are some open instances that cannot be identified in Rhino as such, therefore, on top of the *isObjectSolid* property, we calculate instance's volume; if volume is negative, the instance is open, thus invalid. Such an example is shown in Fig. 3.2, where one symmetric half of the hull is depicted in red colour. The invalidity at the area of AP is obvious: the encircled surface, part of the red-depicted half-hull intersects the plane of symmetry and further penetrates the other half-hull.

B) There are some cases of self-intersections, though, where the final object is identified by Rhino as a closed object, the volume has a positive value, but the geometry presents self-intersections; in these cases we appeal to user-based visual evaluation. Fig 3.3 illustrates two cases of evident, undetected self-intersections, while Fig. 3.4 shows a self-intersection which is more cumbersome to identify. The latter group of self-

intersections is most probable to be encountered in the complex areas of the hull, such as the stern, as it is the case in Fig. 3.4. Via visual evaluation, one can also identify shape-abnormalities with respect to a regular-practical container or taker ship-hull.

The development of a PM which produces valid geometries in the whole range of the input parameter values, or even in most areas of the range, is a matter of extensive experimentation and very close examination of the sensitive areas of the models produced. It should be also stressed that, usually, the problematic cases are occurring in extreme parameter values, lying very close to the boundaries of the values range.

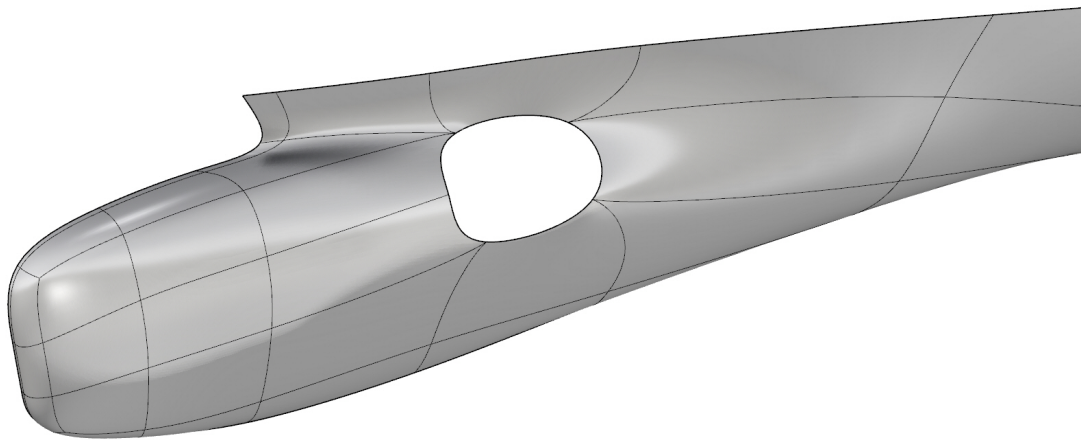


Figure 3.1: An invalid geometry consisting of a trimmed surface.

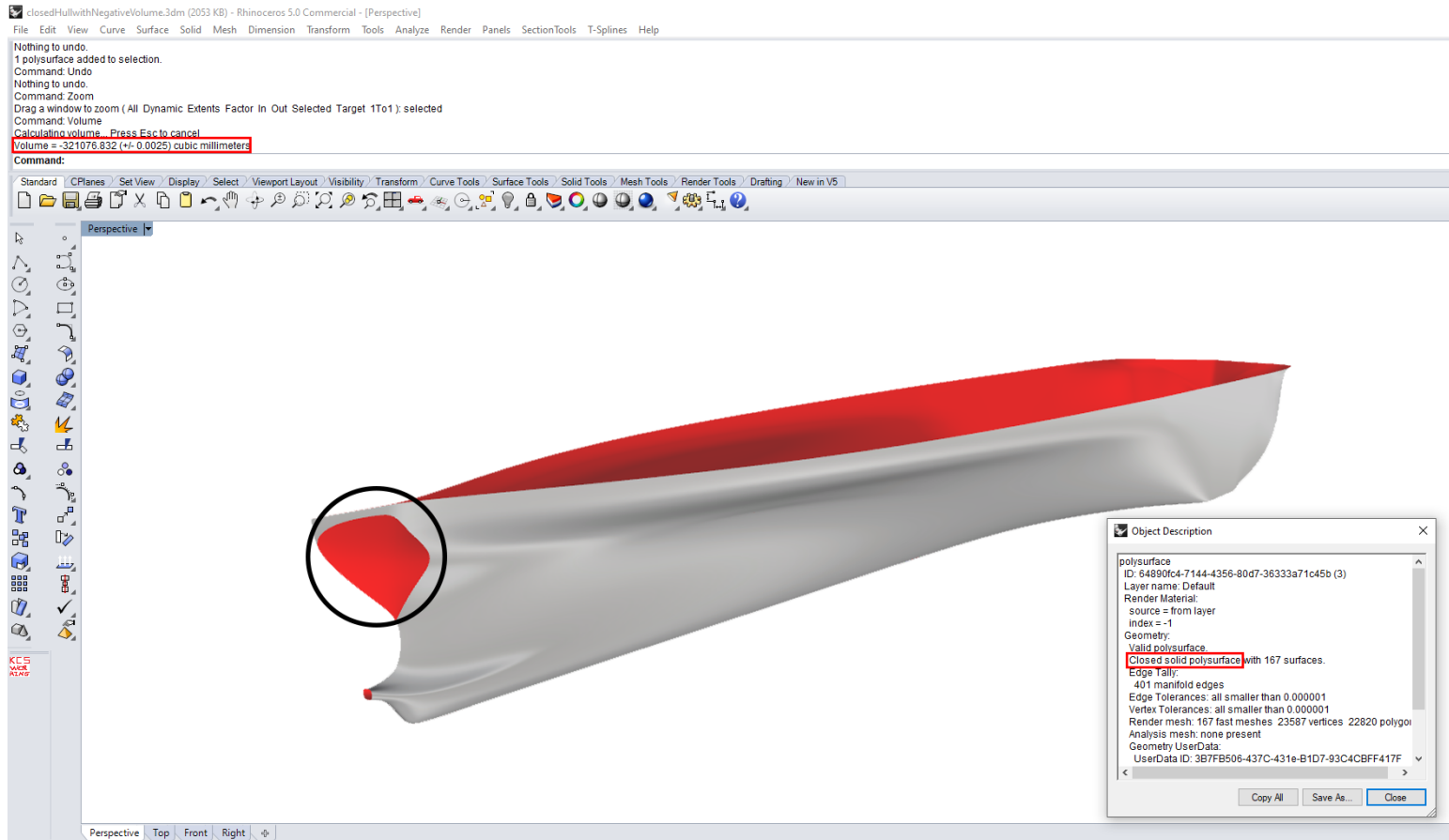


Figure 3.2: Example of an invalid, closed object with negative volume. The self-intersection is encircled.

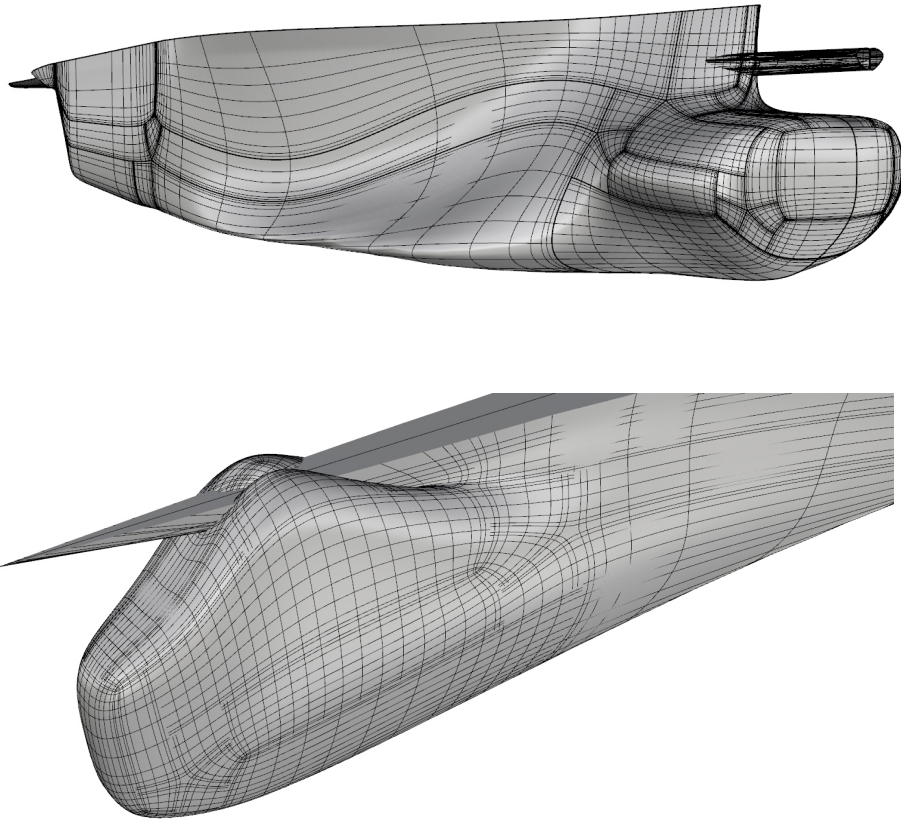


Figure 3.3: Undetected by Rhino invalid geometries consisting of evident self-intersections.

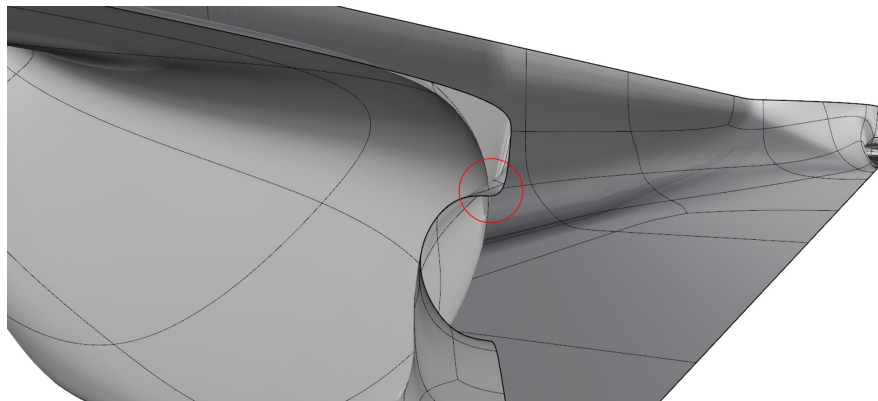


Figure 3.4: Undetected by Rhino invalid geometries consisting of non-evident self-intersections.

At this point it would be important to note that, although part of the detection of the invalid surfaces can be conducted automatically by integrating certain criteria into the algorithm, the automated imposition of constraints depends on a set of criteria such as the ship-hull type, the detailed characteristics of the hull, the richness of the design space, etc., and it is an intricate process which requires further research.

3.3 Constraints

3.3.1 Transition curve constraints

As described before, TshipPM uses a set of transition curves to control the geometry construction of the produced ship-hulls. The increased number of transition curves compared to [6] alongside with the large design space the modeller covers might cause failures in the construction process, characterised as *overlaps* and *misplacements* of the transition curves. An overlap is when two curves coincide (they share the same longitudinal position), and a misplacement is when a transition curve, which was supposed to be located after/before another transition curve, is located before/after that. An example of misplacement is given in Fig. 3.5, where $FoS_Fwd_L > FoB_Fwd_L$, therefore FTCB is drawn before FTCA, and as a result a shape-abnormality occurs.

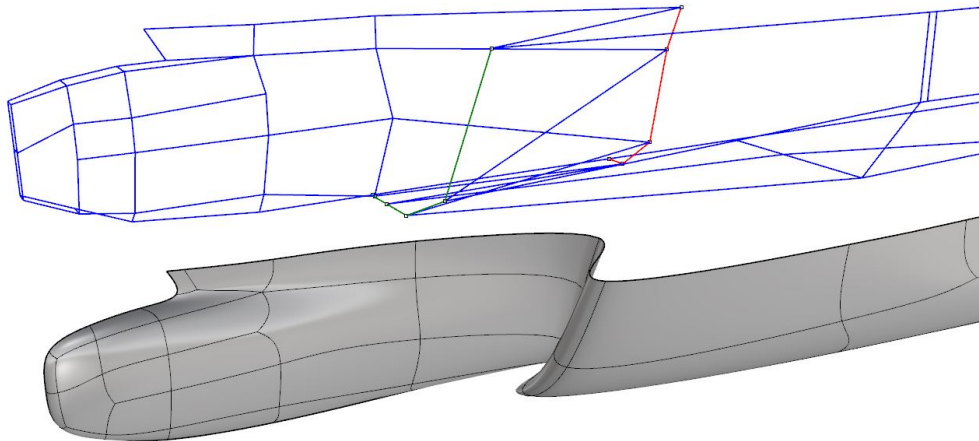


Figure 3.5: Misplacement of the second Forward Transition Curve (FTCB), depicted in red, as it is located before the first Forward Transition Curve (FTCA), depicted in green.

To ascertain no overlaps or misplacements among the transition curves which belong to a certain part of the ship-hull, we have adopted two measures:

Measure 1: Assignment of internal parameters to explicitly determine the topological location of each transition curve.

We are going to discuss the first measure with the aid of Fig. 3.6. For the forward part of the ship-hull we make use of 4 transition curves. FTCA and FTCD are located at the end of the flat of side and flat of bottom respectively. No internal parameters are needed for these two curves, as their topological position is determined by only the use of physical parameters FoS_Fwd_L (for FTCA) and FoB_Fwd_L (for FTCD). FTCB should be located between FTCA and FTCD, and FTCC between FTCB and FTCD. Assuming the closest U-shaped polyline of the middle part is at the longitudinal positions $x = 0$, FTCA, FTCB, FTCC, FTCD are located at the longitudinal position $FTCA(x) = FoS_Fwd_L$, $FTCB(x) = a_1 \cdot FoB_Fwd_L$, $FTCC(x) = a_2 \cdot FoB_Fwd_L$, and $FTCD(x) = FoB_Fwd_L$, where it should finally stand that $FTCA(x) \leq FTCB(x) \leq FTCC(x) \leq FTCD(x)$, achieved by introducing internal parameters a_1 and a_2 , where $\frac{FoS_Fwd_L}{FoB_Fwd_L} < a_1 < a_2 < 1$. The afore mentioned process is in tandem with the constraint $FoS_Fwd_L \leq FoB_Fwd_L$ which imposes the length of the flat of side to be less than or equal to the length of the flat of bottom, and which applies for both forward and afterward parts. This guarantees that there will be no misplacement among FTCs when $FoS_Fwd_L \leq FoB_Fwd_L$, but it does not suffice to prevent FTCA and FTCD from overlapping (see **Measure 2**). FP_Crv and BulbCrvs are not subject to any constraints. The afterward part has been constructed in a similar way, with no interdependency between internal parameters a_3 and a_4 , and both of them lie in $(0,1)$ (Fig. 3.7).

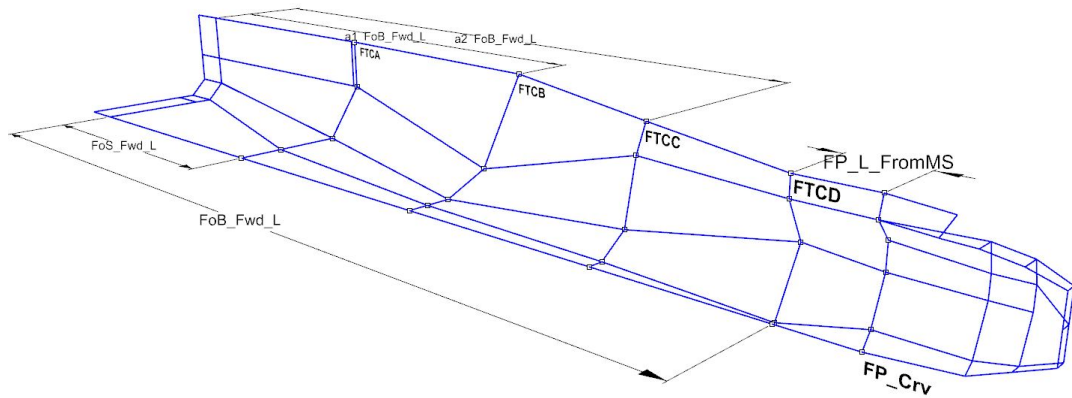


Figure 3.6: Longitudinal position of the forward transition curves. Internal parameters ascertain there are no misplacements.

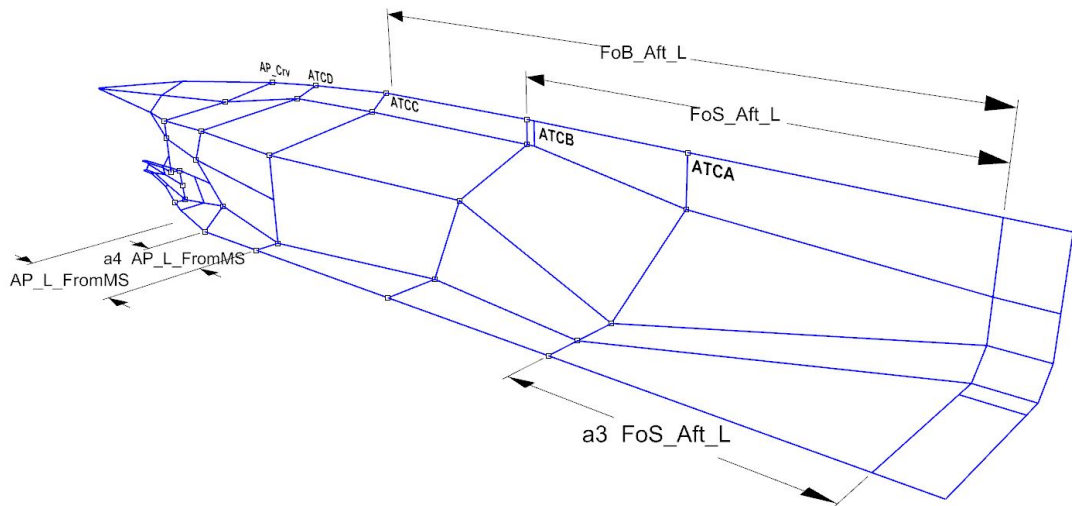


Figure 3.7: Longitudinal position of the afterward transition curves. Internal parameters ascertain there are no misplacements.

Measure 2: Internal decisions for drawing/neglecting transition curves.

Measure 1 does not ascertain the prevention of overlaps. There are cases for which a decision should be taken in the algorithm about whether or not a transition curve is to be constructed or neglected. The constraints taken for these cases are outlined below:

- (a) If $FTCA(x) = FTCB(x)$, $FTCB$ is neglected.
- (b) If $FTCA(x) = FTCC(x)$, both $FTCB$ and $FTCC$ are neglected.

- (c) If $FTCA(x) = FTCD(x)$, all forward transition curves but FTCA are neglected.
- (d) If $ATCB(x) = ATCC(x)$, ATCC is neglected.

3.3.2 Other constraints and conditions

Keel plane constraint: Along the afterward part, the second control point ($P_{2,ATCi}$, $i = A, B, C, D$) of each transition curve counting from the waterline length is assigned as a function of parameters $\widehat{FoS_Trans}$ and $\widehat{TransomH}$, i.e., $P_2 = f(\widehat{FoS_Trans}, \widehat{TransomH})$. In case extremely high parameter values (e.g., 0.99) are assigned for both of them, control point P_2 of the first two transition curves might be located lower than the keel of the ship-hull, and as a result a shape-abnormality is created, as shown in Fig. 3.8. A geometric constraint needs to be imposed to ascertain that $P_{2,ATCi}(z) \geq Origin(z) - T$, for $i = A, B$. The same applies for $P'_2(z)$. The constraint is not applied to ATCC, and ATCD, as $\widehat{FoS_Trans}$ doesn't have such a significant contribution as it has for ATCA, and ATCB.

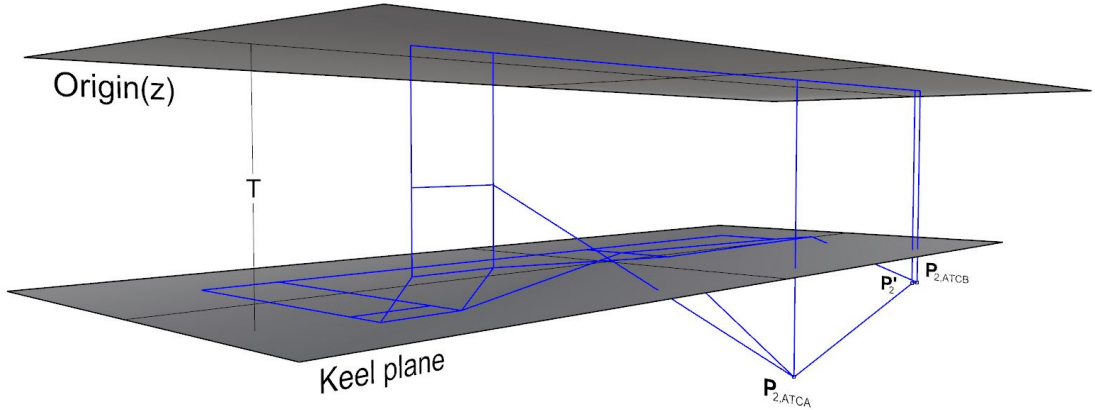


Figure 3.8: $P_{2,ATCA}$, $P_{2,ATCB}$, and P'_2 have to be constrained so they are always at the same level with or higher than the keel of the ship-hull.

Stern constraint: Assuming large values for parameters $\widehat{TransomH}$ and \widehat{TubeR} and extremely low values for $\widehat{FoS_Aft_L}$, $\widehat{FoB_Aft_L}$, $\widehat{AP_L_FromMS}$, and \widehat{TubeL} , a self-intersection is caused at the location where the bottom part of the stern meets the main body of the hull (Fig. 3.9). To understand the reason behind this invalidity we are

focusing on the output surface of the invalid model and we compare it with the surface of the constrained, valid one (Fig. 3.10). We observe that the stern in the proximity of ship-hull's main body is bulkier compared to the valid one's and it intersects with the surface created between the tube and the main body. The problem occurs because the profile of the stern at this particular location is created by 3 control points (Fig. 3.9) as compared to the 4 control points of the valid surface, hence the surface of the stern near the main body is more curved. We prevent this phenomenon by introducing a series of control points (4 in total, 3 line segments) along the span of the stern in the proximity of the main body of the hull (Fig. 3.11). These control points are imposed only when very long stern and smaller main bodies are produced and not for regular (more practical) instances, since the constraint is not necessary then.

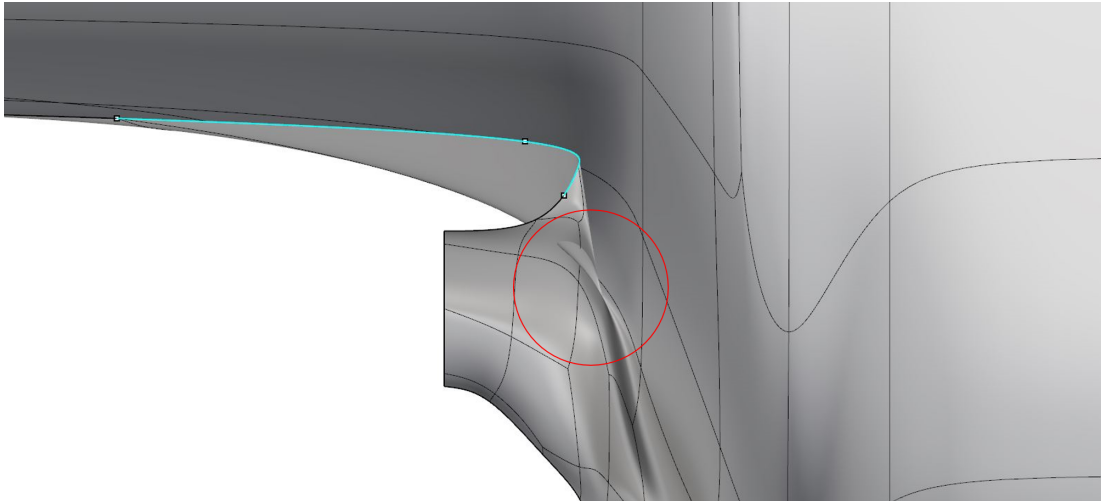


Figure 3.9: Invalid surface: intersection of the neighbouring surfaces of stern and tube-main body.

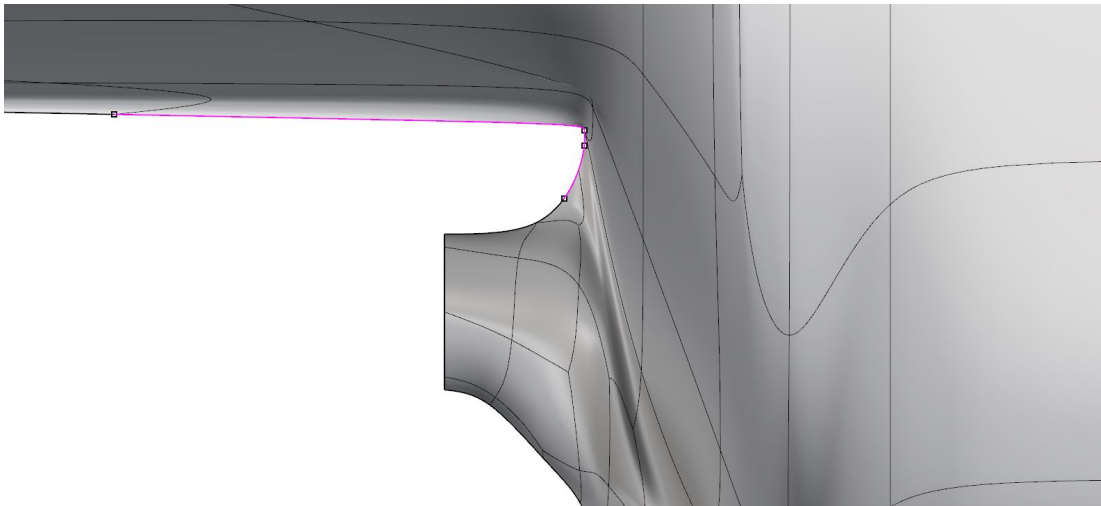


Figure 3.10: Valid surface: the problem has been addressed by introducing a series of control points along the span of the stern (see Fig. 3.11).

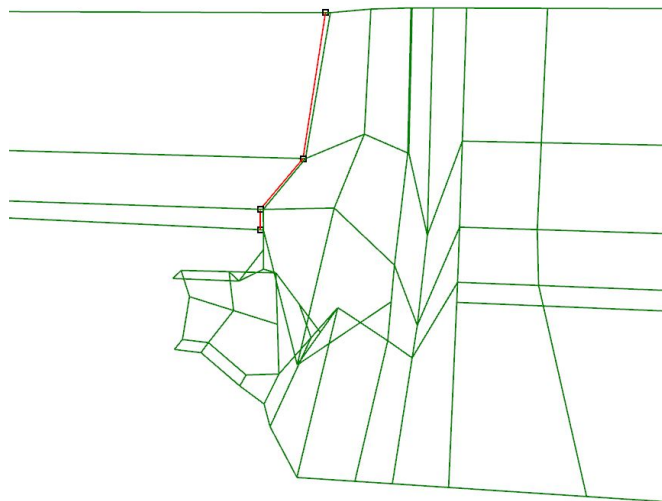


Figure 3.11: Control cage of the valid object: 4 control points along the span of the stern are needed to ascertain geometric validity at the transition from the main body-tube to stern.

Middle part constraint Finally, Mid_Pos and Mid_L have been constrained between $[Lwl/4, 3Lwl/4]$ and $[Lwl/5, 4Lwl/5]$ respectively, as it is impractical to obtain objects with their parallel middle part -which is the part with the largest sectional area- in the proximity of the bow or the stern - where smaller sectional areas are met.

Apart from the imposed constraints, we have incorporated some conditions which allow the construction of models with less control points. This is accomplished by neglecting the construction of some of the transition curves, when one curve is very close ($< 0.1m$) to another one. In brief:

- (i) If FTCA is very close to FTCB, the latter is neglected.
- (ii) If FTCA is very close to FTCC, both FTCB and FTCC are neglected.
- (iii) If FTCA is very close to FTCD, all forward transition curves except FTCA are neglected.
- (iv) If ATCB is very close to ATCC, ATCC is neglected.

3.4 Behaviour of TshipPM against strong parameter values variation

To provide an experimental indication of the robustness of the constructed PM, Figs. 3.12 to 3.19 collect TshipPM's response to strong variations of 3 design parameters, namely the bilge radius (BilgeR), the bulb length (BulbL) and the bulb height (BulbH). Variation is measured in percentage of the parameter values corresponding to the parent hull (KCS).

In particular, Fig. 3.12 and 3.13 depict TshipPM's response to BilgeR decrease and increase by 30% respectively. Figs. 3.14-3.17 illustrate deviations of BulbH and BulbL up to $\pm 20\%$. Finally, Fig. 3.18 shows a TshipPM instance where BulbH has been decreased by 60%, while Fig. 3.19 is an instance where both BulbH and BulbL have been decreased by 50%.

TshipPM provides at least G^1 -continuous and fair surfaces for the afore mentioned significant variations of parameter values. At this point we have to mention that, in some cases at the stern, and especially at the area where the tube meets the main body of the hull, assuming extreme parameter values, thus very long sterns with large tube radius and ATCD very close to the afterward perpendicular, there occur few cases

where smoothness and fairness are very cumbersome to identify and evaluate, let alone to demonstrate, mainly in the proximity and around the symmetric axis, and they seem to be of lower quality than in the rest of the hull, but still no less than G^1 .

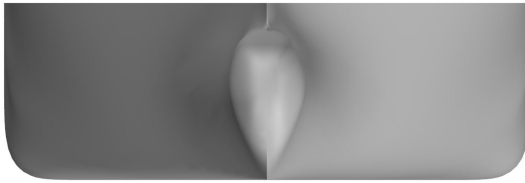


Figure 3.12: Decreasing \widehat{BilgeR} by 30%.

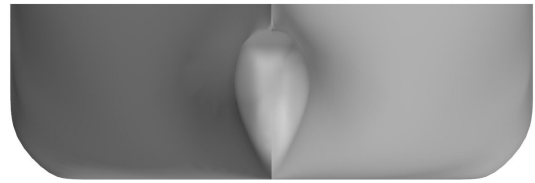


Figure 3.13: Increasing \widehat{BilgeR} by 30%.

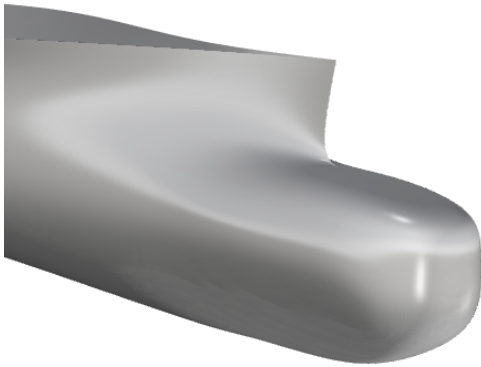


Figure 3.14: Decreasing \widehat{BulbH} by 20%.

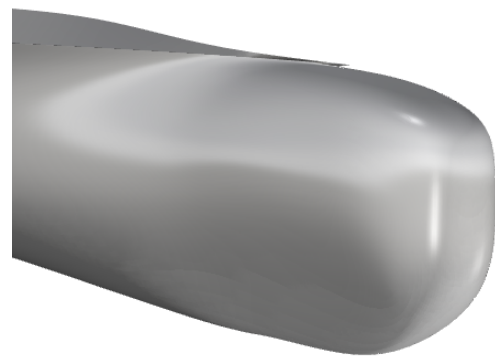


Figure 3.15: Increasing \widehat{BulbH} by 20%.

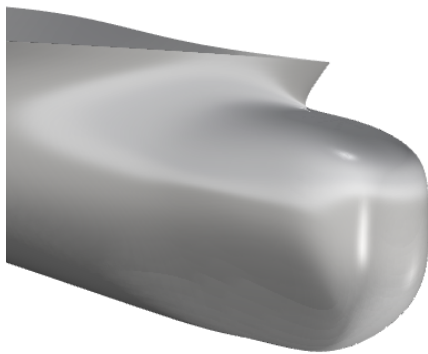


Figure 3.16: Decreasing \widehat{BulbL} by 20%.

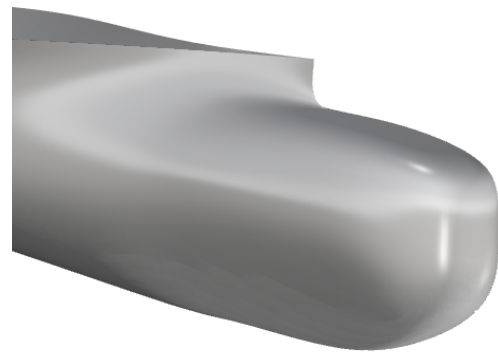


Figure 3.17: Increasing \widehat{BulbL} by 20%.

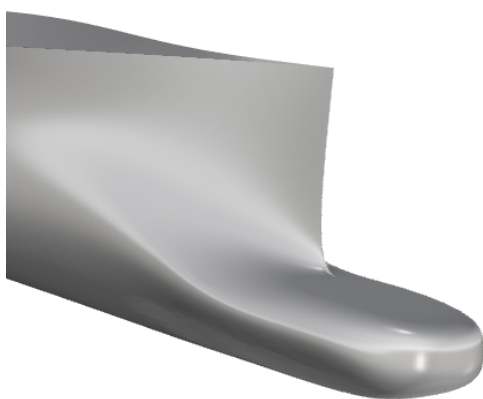


Figure 3.18: Decreasing \widehat{BulbH} by 60%.

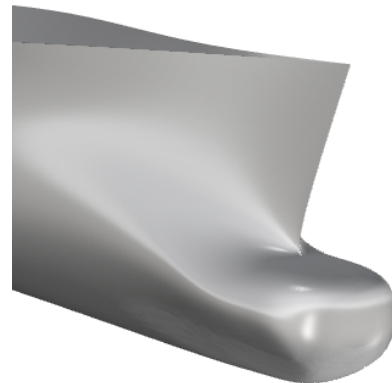


Figure 3.19: Decreasing both \widehat{BulbL} and \widehat{BulbH} by 50%.

3.5 Data analysis and design space

In this section we present the output of a Monte Carlo sampling of ship-hull instances produced by TshipPM, both for containers and tankers. Firstly, we explain how the sampling process was conducted and we elaborate on the extracted information of the sampling. Furthermore, we analyse the data output to determine the *actual design space* of TshipPM, and we measure its flexibility with regards to geometric characteristics, namely volume and moments of inertia.

Sampling: An iteration process was conducted, using RhinoScript, to create a sample of 100,000 designs to use as a basis for the evaluation of the robustness of each TshipPM version, one for containers (*TshipPM_C*) and one for tankers (*TshipPM_T*), after the imposition of the constraints. We evaluate TshipPM's capability to produce geometrically valid instances for any combination of parameter values provided in between the range [0.01,0.99] using Rhino's GUI and its features, as explained in §3.2. The parameter values for each instance were assigned randomly using the *Rnd* function, that returns a floating-point value between 0.0 and 1.0. The returned values have been adjusted affinely to lie in between the desired parameter value range.

Out of the total sampling of the 200,000 produced models of both versions of TshipPM and after evaluating them with the aid of Rhino5, the results showed that TshipPM produces geometrically valid objects for the whole range of the external parameters, i.e. it produces watertight models for its complete design space.

A database which includes the following information for each model has been created:

- (i) input file: parameter values;
- (ii) output file: dimensions of the physical parameters in absolute lengths, *C_b* (block coefficient), volume, volume centroid, and moments of inertia with respect to volume centroid;
- (iii) a 3d object in Rhino's *.3dm* format;

Part of the database for the 3d objects of containers ¹ and tankers ² are accessible in the online design community of GrabCAD.

Data Analysis and Design Space: One of the most crucial characteristics of a PM is its capability to produce models of diverse shapes and geometric characteristics, such as volume moments, therefore creating models of an extensive *design space*. The *theoretical design space* (TDS) of a PM is defined based on the values the physical parameters can theoretically take according to the underlying linear functions defining them, assuming no constraints are imposed. On the other hand, the *actual design space* (ADS) is created by the values the physical parameters can actually take after the constraints are imposed. We're explaining this notion with the aid of the following example: we assume TshipPM uses just one parameter, the physical parameter responsible for the middle position of the ship-hull (Mid_Pos), and its corresponding non-dimensional parameter ($\widehat{Mid_Pos}$). Mid_Pos is defined as:

$$Mid_Pos = \widehat{Mid_Pos} \cdot 0.7 \cdot Lwl \quad (3.1)$$

with $\widehat{Mid_Pos} \in [0.01, 0.99]$. Assuming that $Lwl = 232.5\text{m}$, Mid_Pos can take values, before the constraints, in the range $[1.74\text{m}, 172.63\text{m}]$. Since Mid_Pos's lower bound is constrained and cannot be lower than $Lwl/4$, as explained in §3.3, the actual range of Mid_Pos's position is $[58.125\text{m}, 172.63]$, therefore ADS of the 1-dimensional TshipPM is 67% of TDS.

To compare TDS and ADS we first calculate the volume of the hypercube they cover in \mathbb{R}^{24} (Lwl , T , B are assumed constant), which is acquired by the following formula:

$$V = \prod_{n=0}^{23} \Delta P_i \quad (3.2)$$

where with \prod we denote the product, and ΔP_i is the length of the range of the physical parameter P_i can take. Therefore, for TshipPM_C the volume of TDS and ADS

¹<https://grabcad.com/library/tshippm-container-models-1>

²<https://grabcad.com/library/tshippm-tanker-models-1>

are $V_{container}^{TDS} = 1.2 \cdot 10^{32}m^{24}$ and $V_{container}^{ADS} = 7 \cdot 10^{30}m^{24}$, while for TshipPM_T are $V_{tanker}^{TDS} = 1.1 \cdot 10^{38}m^{24}$ and $V_{tanker}^{ADS} = 5.9 \cdot 10^{36}m^{24}$. For TshipPM_C TDS is 5.6% of ADS, while for TshipPM_T is 5.4%. Tables 3.1 for TshipPM_C and 3.2 for TshipPM_T, illustrate all the above information, as well as the range of physical parameters values for both TDS and ADS, and the percentage difference of each of them between TDS and ADS. Physical Parameters have been sorted in ascending order as per their percentage difference between TDS and ADS. In both tables one can observe that only 9 out of the 24 physical parameters are limited after the imposition of the constraints, while just 6 parameters have significant contribution to the decrease of ADS, namely Mid_L, TubeL, Mid_Pos, AP_L_From_MS, FoB_Aft_L, and FoS_Aft_L.

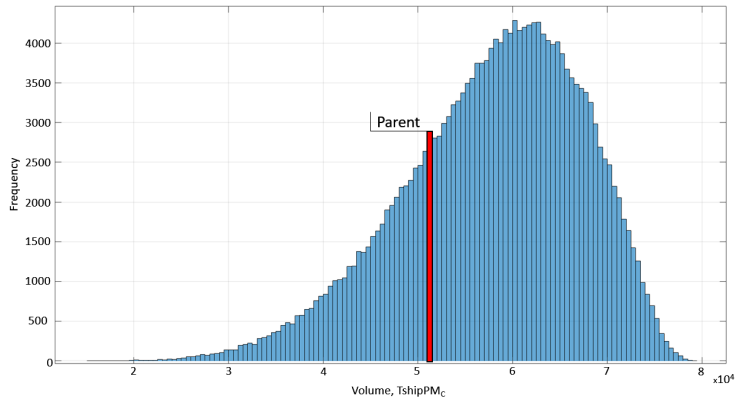
To investigate further the flexibility of TshipPM with regards to geometric characteristics, we explore the frequency distribution of the samples' volume and moments of inertia. Using a dense, representative sampling, strong frequency discontinuities in histograms may reveal inadequacies of the parametric modeller, therefore histograms can be exploited to heal those inadequacies. Histograms of Figs 3.20a to 3.20d and Figs 3.21a to 3.21d illustrate the frequency distribution of the samples of TshipPM_C and TshipPM_T respectively, while Table 3.3 provides information for the samples, namely volume and MoI range, mean value, standard deviation, and skewness. Both versions show a very broad range with respect to models' volume and moments of inertia, with the range of TshipPM_C's volume lying in $[15,302m^3 ; 79,027m^3]$ and TshipPM_T's in $[122,219m^3 ; 532,231m^3]$. Taking as well into consideration the plethora of shapes the PM can produce, TshipPM is proved to be a very flexible and robust tool for modelling ship-hull instances.

Table 3.1: TshipPM_C design space.

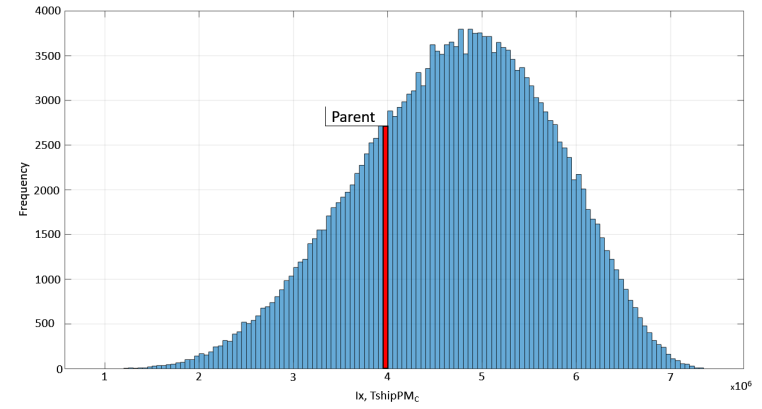
| Parameters | TDS range(m) | ADS range(m) | $\frac{V^{ADS}}{V^{TDS}}$ (%) |
|--------------------------------|----------------------------|--------------------------|-------------------------------|
| Mid_L | 0.42 - 228.45 | 0.42 - 95.51 | 41.7 |
| TubeL | 0.01 - 228.22 | 0.01 - 141.69 | 62.1 |
| Mid_Pos | 1.74 - 172.63 | 58.125 - 172.63 | 67.0 |
| AP_L_from_MS | 0.01 - 228.23 | 0.01 - 156.06 | 68.4 |
| FoB_Aft_L | 0.01 - 228.24 | 0.9 - 171.63 | 74.8 |
| FoS_Aft_L | 0.01 - 228.24 | 0.49 - 171.63 | 75.0 |
| FP_L_from_MS | 0.01 - 172.41 | 0.01 - 151.57 | 88.0 |
| FoS_Fwd_L | 0.01 - 170.7 | 0.47 - 167.01 | 97.6 |
| FoB_Fwd_L | 0.01 - 170.70 | 0.51 - 168.96 | 98.7 |
| BatAP | 0.16 - 15.94 | 0.16 - 15.94 | 100 |
| BatAPLow | 0.04 - 3.98 | 0.04 - 3.98 | 100 |
| BatFP | 0.03 - 3.19 | 0.03 - 3.19 | 100 |
| BilgeR | 0.04 - 3.56 | 0.04 - 3.56 | 100 |
| BulbB | 0.06 - 6.38 | 0.06 - 6.38 | 100 |
| BulbH | 0.11 - 10.70 | 0.11 - 10.70 | 100 |
| BulbL | 0.16 - 15.35 | 0.16 - 15.35 | 100 |
| BulbTipH | 0.01 - 5.3 | 0.01 - 5.3 | 100 |
| FoBS_Trans | 0.16 - 15.94 | 0.16 - 15.94 | 100 |
| FoS_Trans | 0.06 - 10.65 | 0.06 - 10.65 | 100 |
| FwdRise | 0.01 - 2.65 | 0.01 - 2.65 | 100 |
| SternFullness | 0.01 - 0.99 | 0.01 - 0.99 | 100 |
| TransomB | 0.01 - 15.78 | 0.01 - 15.78 | 100 |
| TransomH | 0.05 - 5.35 | 0.05 - 5.35 | 100 |
| TubeR | 0.02 - 1.76 | 0.02 - 1.76 | 100 |
| Volume (m²⁴) | 1.2·10³² | 7·10³⁰ | 5.6 |

Table 3.2: TshipPM_T design space.

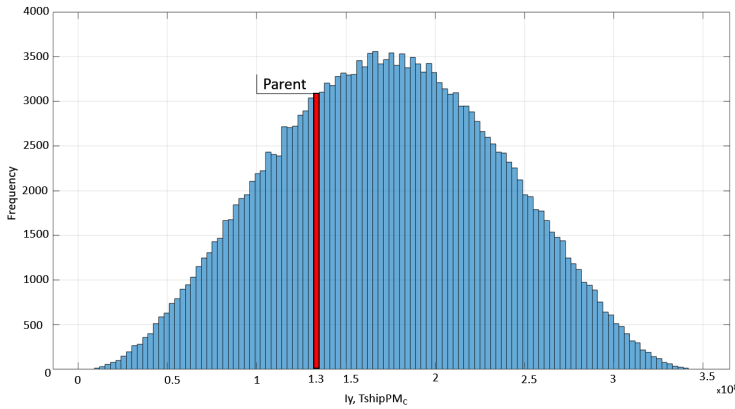
| Parameters | TDS range(m) | ADS range(m) | $\frac{V_{ADS}}{V_{TDS}}$ (%) |
|--------------------------------|----------------------------|----------------------------|-------------------------------|
| Mid_L | 0.59 - 323.27 | 0.64 - 134.88 | 41.6 |
| TubeL | 0.01 - 322.94 | 0.01 - 203.36 | 63.0 |
| Mid_Pos | 2.46 - 244.28 | 82.25 - 244.28 | 67.0 |
| AP_L_from_MS | 0.01 - 322.96 | 0.01 - 219.20 | 67.9 |
| FoB_Aft_L | 0.01 - 322.96 | 1.36 - 241.95 | 74.5 |
| FoS_Aft_L | 0.01 - 322.97 | 0.67 - 241.80 | 74.7 |
| FP_L_from_MS | 0.01 - 243.97 | 0.01 - 209.05 | 85.7 |
| FoS_Fwd_L | 0.01 - 241.55 | 0.66 - 236.43 | 97.6 |
| FoB_Fwd_L | 0.01 - 241.55 | 1.02 - 237.62 | 98.0 |
| BatAP | 0.29 - 28.71 | 0.29 - 28.71 | 100 |
| BatAPLow | 0.07 - 7.17 | 0.07 - 7.17 | 100 |
| BatFP | 0.05 - 5.75 | 0.05 - 5.75 | 100 |
| BilgeR | 0.09 - 9.23 | 0.09 - 9.23 | 100 |
| BulbB | 0.11 - 11.49 | 0.11 - 11.49 | 100 |
| BulbH | 0.29 - 27.74 | 0.29 - 27.74 | 100 |
| BulbL | 0.22 - 21.71 | 0.22 - 21.71 | 100 |
| BulbTipH | 0.01 - 13.74 | 0.01 - 13.74 | 100 |
| FoB_Trans | 0.29 - 28.71 | 0.29 - 28.71 | 100 |
| FoS_Trans | 0.14 - 27.61 | 0.14 - 27.61 | 100 |
| FwdRise | 0.01 - 6.87 | 0.01 - 6.87 | 100 |
| SternFullness | 0.01 - 0.99 | 0.01 - 0.99 | 100 |
| TransomB | 0.01 - 28.42 | 0.01 - 28.42 | 100 |
| TransomH | 0.13 - 13.87 | 0.13 - 13.87 | 100 |
| TubeR | 0.05 - 4.56 | 0.05 - 4.56 | 100 |
| Volume (m²⁴) | 1.1·10³⁸ | 5.9·10³⁶ | 5.4 |



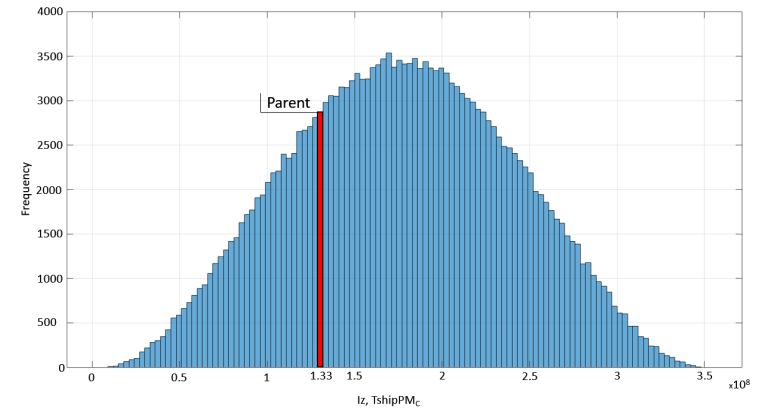
(a) TshipPM_C histogram of moments of Volume.



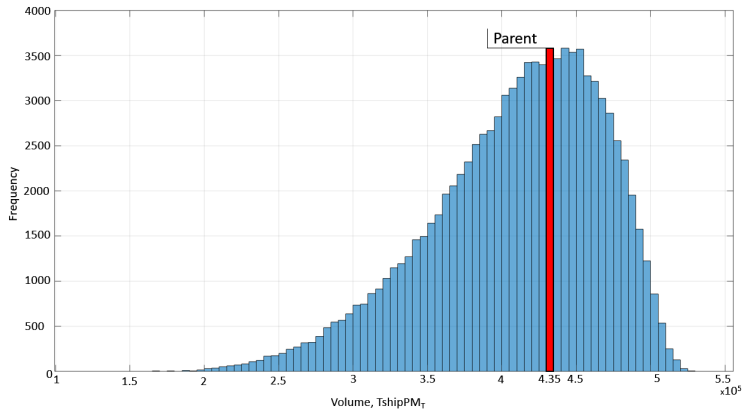
(b) TshipPM_C histogram of moments of inertia (I_x).



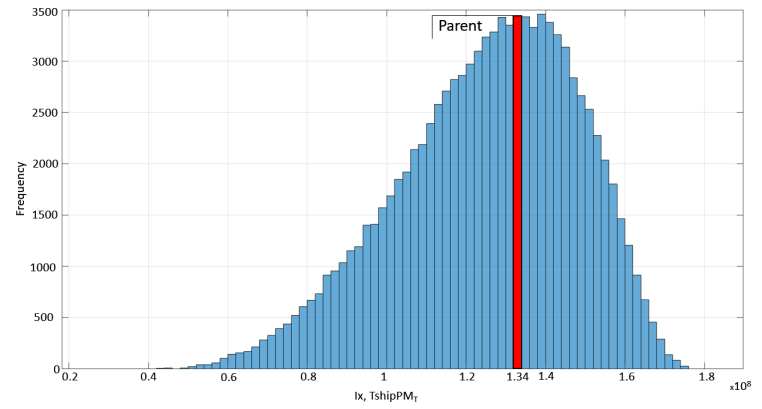
(c) TshipPM_C histogram of moments of inertia (I_y).



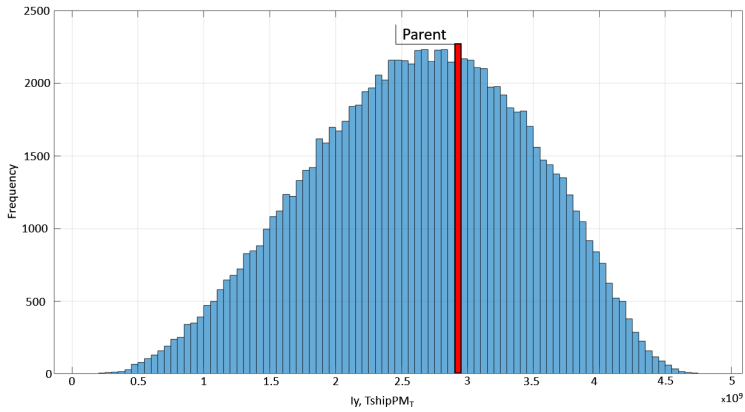
(d) TshipPM_C histogram of moments of inertia (I_z).



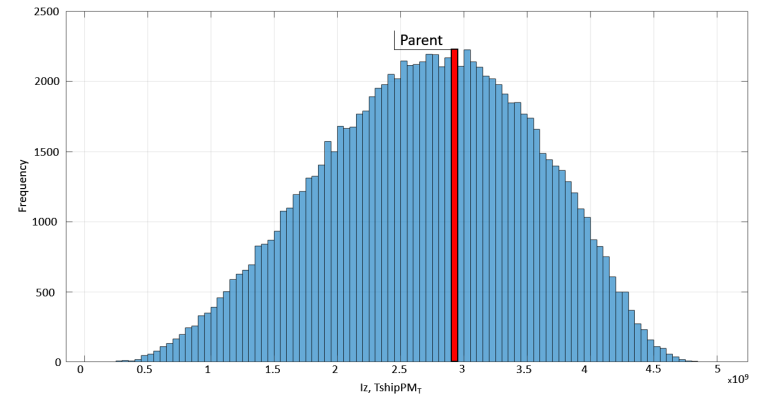
(a) TshipPM_T histogram of volume.



(b) TshipPM_T histogram of moments of inertia (I_x).



(c) TshipPM_T histogram of moments of inertia (I_y).



(d) TshipPM_T histogram of moments of inertia (I_z).

Table 3.3: TshipPM data analysis: Histograms information.

| | TshipPM_C | TshipPM_T |
|--|---|---|
| Volume (m³) | | |
| Volume of parent hull | 52,053 | 435,577 |
| Range | 15,302 - 79,027 | 122,219 - 532,231 |
| Mean | 57,855 | 409,437 |
| Standard deviation | 9,497 | 58,065 |
| Skewness | -0.55 | -0.69 |
| Moments of Inertia (I_x, m⁵) | | |
| Inertia of parent hull | 3.89·10 ⁶ | 1.35·10 ⁸ |
| Range | 9.02·10 ⁵ - 7.42·10 ⁶ | 2.69·10 ⁷ - 1.81·10 ⁸ |
| Mean | 4.71·10 ⁶ | 1.25·10 ⁸ |
| Standard deviation | 1.02·10 ⁶ | 2.27·10 ⁷ |
| Skewness | -0.25 | -0.47 |
| Moments of Inertia (I_y, m⁵) | | |
| Inertia of parent hull | 1.30·10 ⁸ | 2.88·10 ⁹ |
| Range | 5.01·10 ⁶ - 3.47·10 ⁸ | 1.15·10 ⁸ - 4.85·10 ⁹ |
| Mean | 1.73·10 ⁸ | 2.64·10 ⁹ |
| Standard deviation | 6.22·10 ⁷ | 8.18·10 ⁸ |
| Skewness | 0.01 | -0.16 |
| Moments of Inertia (I_z, m⁵) | | |
| Inertia of parent hull | 1.33·10 ⁸ | 2.96·10 ⁹ |
| Range | 6.00·10 ⁶ - 3.52·10 ⁸ | 1.33·10 ⁸ - 4.96·10 ⁹ |
| Mean | 1.77·10 ⁸ | 2.71·10 ⁹ |
| Standard deviation | 6.29·10 ⁷ | 8.32·10 ⁸ |
| Skewness | 0.01 | -0.17 |

Chapter 4

A Method to Remodel a Parent Ship-hull using TshipPM

4.1 Introduction

A basic need when it comes to parametric modelling is the capability to reconstruct a given model. This process, which we will be referring to as *remodelling*, is deployed to build a parent CAD model, with the aid of a PM, and use it as a basis for optimisation. In this chapter we provide in detail the method of remodelling the MOERI Container (KCS) ship-hull instance from waterline level and below, employing TshipPM. This very elaborated process consists of two major steps:

- a. Control cage construction using TshipPM;
- b. Optimisation of the control cage to more accurately remodel the parent ship-hull.

The optimisation process of the control cage is being conducted manually at a control-point level and it could be used in tandem with the automated optimisation process (based on ship-design criteria) which is presented in Chapter 6.

4.2 Geometric and ship-design criteria for remodelling evaluation

The evaluation of the remodelled instance has been based on a set of criteria of geometric (a-c) and ship-design (c-d) nature:

- a. Smoothness: at least G^1 -continuity. The evaluation of smoothness has been conducted in Rhino5 3D using the isophotes feature (*Zebra*);
- b. Fairness: smooth distribution and proper sign of curvatures, e.g., nearly zero Gaussian curvature in the area of the cylindrical mid-part. Gaussian curvature is an intrinsic geometrical property of surfaces used by designers to evaluate the fairness of the boundary of a 3D object and locate deficiencies versus the anticipated shape of the surface under consideration. The Gaussian curvature $K(\mathbf{P})$ at a point \mathbf{P} of a smooth surface is expressed as:

$$K(\mathbf{P}) = \kappa_1(\mathbf{P}) \cdot \kappa_2(\mathbf{P}) \quad (4.1)$$

where κ_i , $i = 1, 2$ are the *principal curvatures* at \mathbf{P} ;

- c. Deviation from prescribed k^{th} -order moments, e.g., volume ($k = 0$), volume centroid ($k = 1$), moments of inertia ($k = 2$);
- d. Deviation from a prescribed *sectional area curve* (SAC). Plotting and comparing SAC among different ship-hulls is a useful tool for evaluating the longitudinal distribution of sectional area values (SAV) as well as extracting the longitudinal positions where the major variations of SAVs exist.

4.3 Control cage construction

The remodelling process aims to the reconstruction of a parent ship-hull CAD model. A prerequisite is the construction of the control cage of the model (using TshipPM) which

will approximate the parent ship-hull and will be later fed into Rhino5 to create the resulting surfaces with the aid of the T-splines plug-in. For the control cage construction the following steps are to be followed:

1. **Physical parameters extraction:** extract physical parameters from the CAD model of the parent ship-hull (§4.3.1);
2. **Parameter values assignment:** assign non-dimensional parameter values in accordance to the physical parameters and their linear functions describing them (§4.3.2). Dimensional parameters corresponding to the values of the global physical parameters Lwl, T, and B, is necessary to be assigned as well;
3. **Midship control cage construction:** build and interconnect the U-shaped polylines of the cylindrical middle part, as well as the transition curves at Fwd_Part and Aft_Part (§4.3.3);
4. **Bow control cage construction:** build and interconnect BowPrfCrv, BulbCrvs and BowOffCrv, and connect them with the forward part of the midship (§4.3.4);
5. **Stern control cage construction:** build and interconnect SternPrfCrv and TubeCrv, and connect them with the afterward part of the midship (§4.3.5).

4.3.1 Extraction of physical parameters

We measure on the parent hull's CAD model all physical parameters used in TshipPM (Table 2.1), by focusing each time on a specific part of the ship-hull, namely Midship, Bow, and Stern. We measure longitudinal, vertical, and lateral physical parameters. If not provided by the technical characteristics of the ship, we measure all three global physical parameters (Lwl, T, B), and we then proceed to measure all physical parameters with local nature.

We identify the physical parameters used to define Midship, Bow, and Stern:

1. Midship
 - a. Cylindrical Middle Part:

- Longitudinal: Mid_Pos, Mid_L;
 - Vertical: BilgeR;
 - Lateral: BilgeR;
- b. Forward Part:
- Longitudinal: FoS_Fwd_L, FoB_Fwd_L;
 - Vertical: FoS_Trans;
 - Lateral: FoB_Trans;
- c. Afterward Part:
- Longitudinal: FoS_Aft_L, FoB_Aft_L;
 - Vertical: FoS_Trans;
 - Lateral: FoB_Trans;
2. Bow:
- Longitudinal: FP_L_FromMS, BulbL;
 - Vertical: BulbH, BulbTipH, FwdRise;
 - Lateral: BatFP, BulbB;
3. Stern:
- Longitudinal: AP_L_FromMS, TubeL, SternFullness;
 - Vertical: TransomH, TubeR;
 - Lateral: BatAP, BatAPLow, TransomB, TubeR, SternFullness;

Now we elaborate on the extraction of physical parameters. We first determine the physical parameters of a more “global” nature, Mid_Pos and Mid_L, by assuming two different cases:

Case 1: Ship-hull incorporates a parallel middle part; Mid_Pos is then located at the longitudinal middle of the parallel middle part. Mid_L represents the whole length of the middle part.

Case 2: Ship-hull doesn't incorporate a parallel middle part; Mid_Pos is then located at the same longitudinal position as the location of the largest section of the hull. To determine Mid_L, we march towards both forward and afterward parts, and we identify two sections -one at each direction- with section areas approximately 2% less than the area of the largest section. Then, we locate Mid_L/2 at the longitudinal dimension from Mid_Pos to that section. Mid_L is then the dimension bounded by that section and the symmetric one with respect to Mid_Pos.

Secondly, we proceed to the extraction of local physical parameters, a set of which is straightforward to measure, such as TransomH, TransomB, BulbL, etc., as Fig 4.1 shows, using as an example the KVLCC tanker CAD model which has a transom stern. Another set of local parameters related to flat surfaces, such as flat of side/bottom of forward and afterward parts, is measured with the aid of Rhino5 3D properties. We use single false-color analysis provided by Rhino's *curvature analysis* feature, which evaluates visually the *Gaussian curvature* (K) of the surfaces. Surfaces for which $K = 0$ are either cylindrical -in case one principle curvature is zero- or flat -when both principle curvatures are zero.

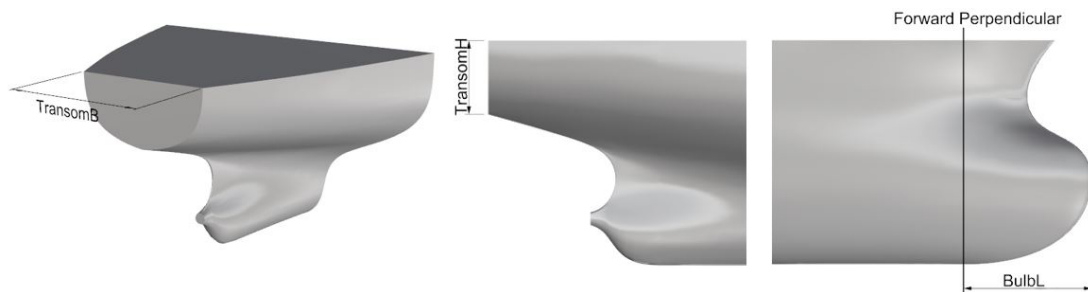


Figure 4.1: Measuring TransomH, TransomB, and BulbL on the KVLCC tanker CAD model.

An example of measuring physical parameters (i.e., dimensions) on a CAD model is shown in Fig. 4.2. Flat parts ($K=0$) are depicted in green and the lengths can be measured accurately. Having assigned Mid_Pos and Mid_L physical parameters, we measure the longitudinal length of the flat of bottom (FoB_Fwd_L) towards the forward part, with point of reference the Mid_L/2 at the Fwd.Part. The breadth at the end of

the flat of bottom which starts from the symmetric axis and travels laterally (vertically in the figure) to the location where $K \neq 0$, provides the length of the flat of bottom transition dimension (FoB_Trans) by subtracting it from the breadth of the half-hull ($B/2$).

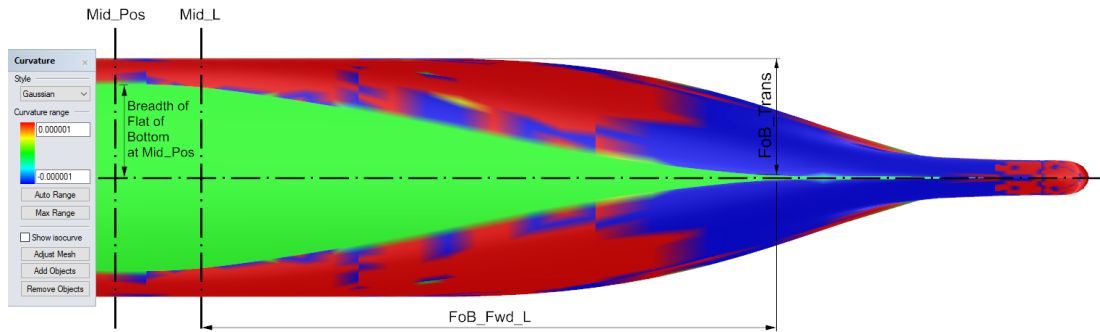


Figure 4.2: Measuring the length of flat of bottom (FoB_Fwd_L) at the forward part and the breadth of flat of bottom transition (FoB_Trans), by using *Gaussian curvature* from the *curvature analysis* feature in Rhino5.

4.3.2 Assigning non-dimensional parameter values with respect to physical parameters

As soon as the process of extracting the physical parameters is completed, we assign the corresponding non-dimensional parameter values, which are used as input in TshipPM. The linear functions describing the relationship between a dimensional and its corresponding non-dimensional parameter are described by Eq. 2.1. In example, the linear function corresponding to Mid_Pos is given by:

$$Mid_Pos = \widehat{Mid_Pos} \cdot 0.75 \cdot Lwl$$

Assuming $Mid_Pos = 135m$ and $Lwl = 232.5m$, we should assign $\widehat{Mid_Pos}$ with the value of:

$$\widehat{Mid_Pos} = \frac{Mid_Pos}{0.75 \cdot Lwl} = 0.77$$

The linear function corresponding to Mid_L is given by:

$$Mid_L = \widehat{Mid_L} \cdot 0.7 \cdot (Lwl - Mid_Pos)$$

To approximately have a middle part measuring 35.5m and Lwl, Mid_Pos as before, we should assign $\widehat{Mid_L}$ with the value of:

$$\widehat{Mid_L} = \frac{Mid_L}{0.7 \cdot (Lwl - Mid_Pos)} = 0.52$$

The linear function indicating the location of TransomH with respect to vertical axis, is given by:

$$TransomH = \widehat{TransomH} \cdot 0.5 \cdot T$$

and given that $TransomH = 2m$, $\widehat{TransomH} = 0.37$.

4.3.3 Midship control cage construction

In this section the construction of the midship part of the ship-hull is demonstrated. Midship construction requires the allocation of all necessary U-shaped lines and transition curves which fully describe the parallel, forward and afterward parts. We should recall at this point that TshipPM uses a cylindrical middle part which is not the case for the KCS container parent hull.

4.3.3.1 Cylindrical middle part: U-shaped polylines

We first determine the U-shaped polylines of the Mid_Part by extracting the largest section of the parent hull and which will be located at the Mid_Pos of the TshipPM model. Furthermore, we extract the control points of this section and connect them with lines, thus forming a *polyline*, and we determine all four polylines comprising the cylindrical middle part, by translating the initial polyline. Finally we interconnect the polylines with parallel transverse polylines. With the aid of Figs. 4.3 and 4.4 we describe the process, which consists of the following steps:

- a. Orient parent ship to match TshipPM instance's orientation, therefore the tip of the waterplane is located at (0,0,0).

- b. Extract largest section (located at Mid_Pos);
- c. Derive sectional curve, which we call *extracted* curve from the symmetric half of the largest section. The control polygon of any of the extracted curves shown herein consists of more than 30 control points, therefore the next step is always necessary. For this step we use Rhino's *Section* command.
- d. Rebuild sectional curve with a degree-3 curve and as many control points as TshipPM uses for the specific curve, forming the *rebuilt* curve. For this step we use Rhino's *Rebuild* command.
- e. Extract control points of rebuilt curve and adjust them for an accurate match, forming the final, *adjusted* curve.
- f. Assign proper values to external parameters of linear functions (Eq. 2.1) to match the coordinates of the control points obtained in the previous step (Fig. 2.10);
- g. Determine Mid_L as described in §4.3.1, and build all four U-shaped polylines and interconnect them.

As described earlier (Eq. 2.1) the physical parameters are expressed as linear functions of internal (where necessary) and external parameters. In Fig. 4.3 the variables of each linear function determining the control points of the U-shaped polylines are illustrated. The extracted curve is depicted in **red**, the rebuild curve and its control points in **green**, and the adjusted curve in **blue**. Note that one control point of the rebuilt curve lies below the flat-of-bottom level of the hull, which, if permitted, would create an invalid geometry with respect to design criteria, as the bottom of the hull would not be *flat* any more. All control points at the bottom part of the parallel middle part have to be collinear. c_1 and c_2 are internal parameters.

Once control points are extracted and the adjusted curve is defined, we construct the control cage of the cylindrical middle part of the ship-hull, as Fig. 4.4 illustrates. Starting from Mid_Pos and moving longitudinally towards the forward part of the parent hull we calculate the sectional areas at the neighbour of the largest section,

up to the location where sectional area's value is not more than 2% smaller than the largest one. At this point we locate the position of $\text{Mid}_L/2$, determining at the same time the symmetric one with respect to Mid_Pos ($-\text{Mid}_L/2$). Finally, we build all four U-shaped polylines and interconnect them with parallel transverse lines.

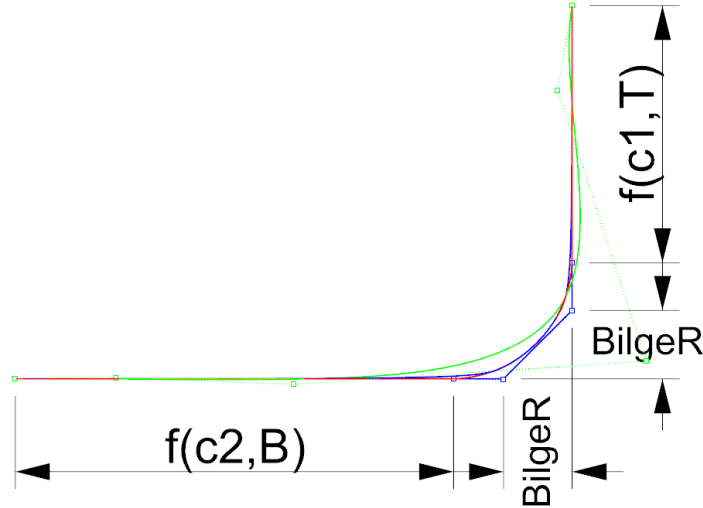


Figure 4.3: Control points extraction of the largest section at Mid_Pos and their inter-relation to external and internal parameters. In red is depicted the extracted curve, while in green and blue are the rebuilt and adjusted curves, respectively, and their control points respectively. c_1 , c_2 are internal parameters.

As a reminder, the y - and z -coordinates of the control points of each U-shaped polyline are the same as translations of the corresponding coordinates of the control points of the adjusted sectional curve at the largest section, and the U-shaped polylines are planar. Their x -coordinates are dependent on $\text{Mid}_L/2$ for the two outermost and on Mid_L and an internal parameter c for the two inner ones, with $c < 0.5$ ¹. The number of U-shaped polylines is not random, as for bicubic T-splines there should exist four identical polylines to construct a cylindrical middle part.

¹As presented in §2.2, all internal parameters range in $(0,1)$. The only exception is parameter c which ranges in $(0,0.5)$.

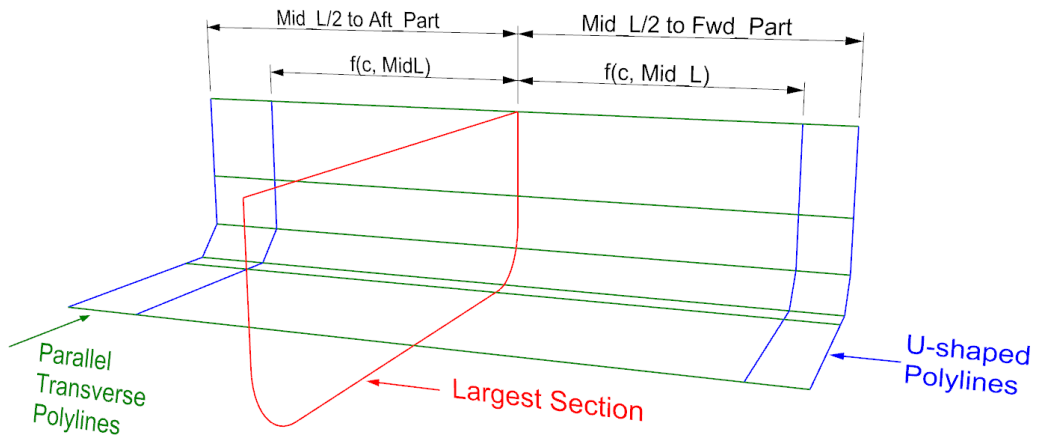


Figure 4.4: Construction of the cylindrical middle part: define a) $\text{Mid_L}/2$ from largest section (red) of the parent hull, b) U-shaped polylines (blue), and c) parallel transverse lines (green). With c we denote the internal parameter introduced to determine the x-position of the two inner U-shaped polylines.

4.3.3.2 Fwd_Part/Aft_Part: transition curves

Once the control cage of the parallel middle part is set, we proceed to the construction of the transition curves of the forward and afterward parts in two major steps: Firstly, we set the longitudinal position of each transition curve depending on physical parameter values, i.e., the longitudinal position of the flat of side and bottom. Secondly, at the longitudinal location of each transition curve, we extract the section of the symmetric half of the hull and define sectional curve's control points in the same way as in 4.3.3.1, expressing their coordinates with linear functions (see 2.1) to construct the corresponding part of the control cage, as described in 2.3 with the aid of Fig. 2.19.

Hereof we demonstrate the process of determining the transition curves of the forward part of the hull and their control points, as well as the construction of the transition curve at FoS_Fwd_L . The process is the same for the afterward part. We describe it with the aid of Figs. 4.5 and 4.6, and consists of the following steps:

A. Setting the location of Fwd_Trans_Crvs.

1. Extract waterplane's boundary curve (black curve in Fig. 4.5) from $\text{Mid_L}/2$ up to the tip of the waterplane;

2. Rebuild curve (degree = 3) with as many control points as the number of the forward transition curves plus two: one for the control point at the location of \mathcal{M}_0 (Mid_L/2) and one for the control point at the waterplane tip. Therefore, the waterplane boundary curve is described by 6 control points;
3. Repeat steps 2 to 4 to determine the control points of the boundary curve at the flat of bottom (blue curve in Fig. 4.5), starting from \mathcal{M}_0 up to FoB_Fwd_L. At the end of this process both boundary curves should have the same number of control points for the parts from \mathcal{M}_0 up to FoB_Fwd_L. Therefore, the flat-of-bottom boundary curve is described by 5 control points;
4. Adjust control points of both rebuilt curves so as every pair of control points, one at the waterplane and one at the bottom plane boundary curves share the same x-coordinate (Fig. 4.6). At the end of this process, all transition curves should be planar.

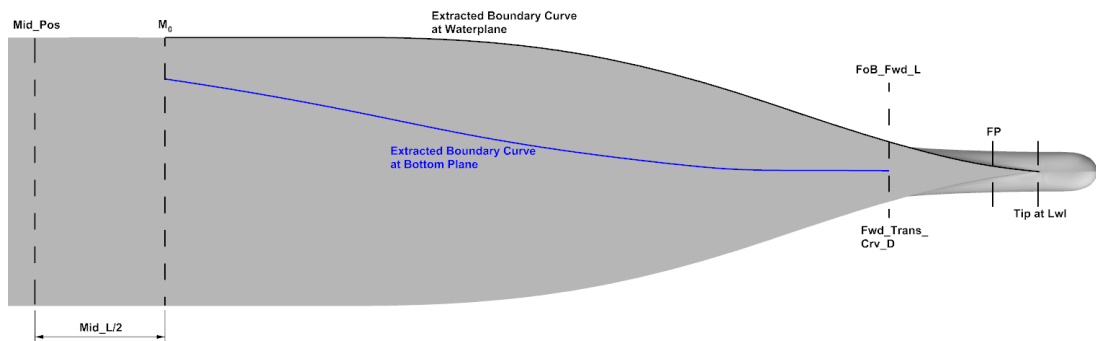


Figure 4.5: Extracted boundary curves at waterplane and flat of bottom. Both are extracted starting from \mathcal{M}_0 , with the former extended up to the tip of the waterplane, and the latter up to the FoB_Fwd_L, consisting of 6 and 5 control points respectively.

In Fig. 4.6, the waterplane and bottom curves and their control polygons are shown before and after adjustment. The extracted curves are depicted in red, the rebuilt curves in green, while in blue are the final, adjusted curves. The control points of FTCA- and FTCD-rebuilt curves have been adjusted regarding the x-coordinate to be placed at the same position as the FoS and FoB. The rebuilt curves are then relocated only towards the y-direction (vertical as for the figure) to build the final adjusted

curves, as the x-coordinates have to remain the same to impose planarity. Note that the flat-of-bottom boundary curve has been taken up to the length of the FoB_Fwd_L because, from that position and forward, it becomes a straight line up to the bottom part of the forward perpendicular.

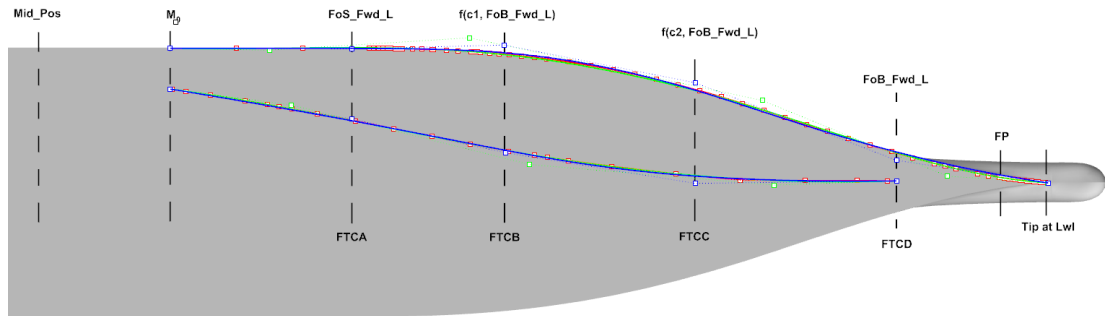


Figure 4.6: Setting the location of FTCs. Extracting control points of the waterplane and bottom plane curves and adjustment. In red are the originally extracted curves and their control polygons, in green are the rebuilt ones, while in blue are the final, adjusted curves. The control points of both boundary curves have been relocated in a manner so they share the same x-coordinate. $c1$ and $c2$ are internal parameters with $c1 < c2$.

- B. Extraction of the sections corresponding to forward transition curves (extracted curves).
- C. Rebuild extracted curves (degree-3) with as many control points as TshipPM uses for each of them (rebuilt curves).
- D. Extract control points of rebuilt curves and adjust them for an accurate match, forming the final, *adjusted* curves.
- E. Assign proper values to external parameters of linear functions (Eq. 2.1) to match the coordinates of the control points obtained in the previous step (see Figs 2.11 to 2.17);
- F. Interconnection of forward and afterward part with the parallel middle part.

4.3.3.3 Forward and afterward perpendicular curves

Before proceeding to Bow and Stern control cage construction, we need to derive the sectional curves at FP and AP, and create the final adjusted curves. Similarly as in the previous section, we extract the control points of the symmetric halves of the sectional curves at forward (FP_Crv) and afterward (AP_Crv) perpendiculars, and we express their coordinates using linear functions. Fig 4.7 illustrates in red the FP and AP sections, and the final, adjusted FP and AP curves in blue with their control polygons.

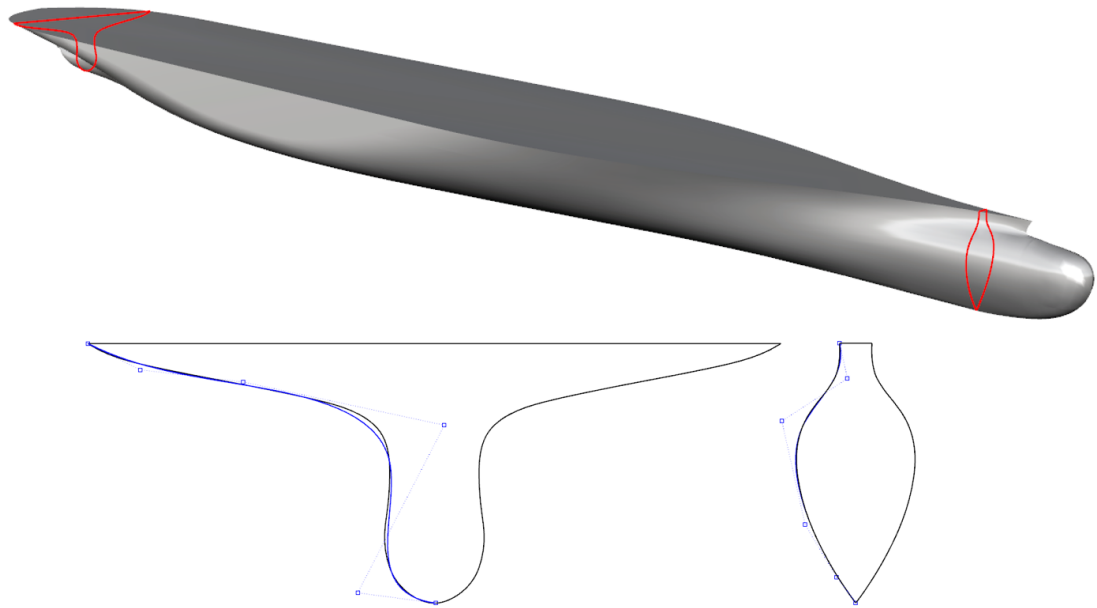


Figure 4.7: FP and AP sections in red (top). Adjusted FP (right) and AP (left) curves with their control polygons depicted in blue (bottom).

4.3.4 Bow control cage construction

Next in the remodelling process of the parent ship-hull comes the construction of the control cage of the bow and its interconnection with the FP_Crv. The process consists of the following steps:

- a. Bow profile curve extraction and rebuilding;
- b. Bulb curve extraction and rebuilding, bow offset and overhang offset polylines determination;

- c. Construction of the bow control cage assigning proper values to external parameters of linear functions (Eq. 2.1) to match the coordinates of the control points obtained in the previous steps;
- d. Interconnection of the bow control cage with the midship.

4.3.4.1 Bow profile curve

The extraction of bow profile curve's (BowPrfCrv) control points, which is illustrated in Fig. 4.8, consists of the following steps:

1. Extract BowPrfCrv from parent ship;
2. Rebuild curve (degree = 3) using as many control points as in TshipPM;
3. Adjust if necessary the rebuilt curve for an accurate match with the extracted one. Note that the control points $P_1-P'_1$ and $P_2-P'_2$ are adjusted to be collinear as shown in Fig. 4.8, as they will define the location of the (planar) bulb curves on a later stage;
4. Assign proper values to external parameters of linear functions (Eq. 2.1) to match the coordinates of the control points obtained in the previous step (see Fig. 2.15).

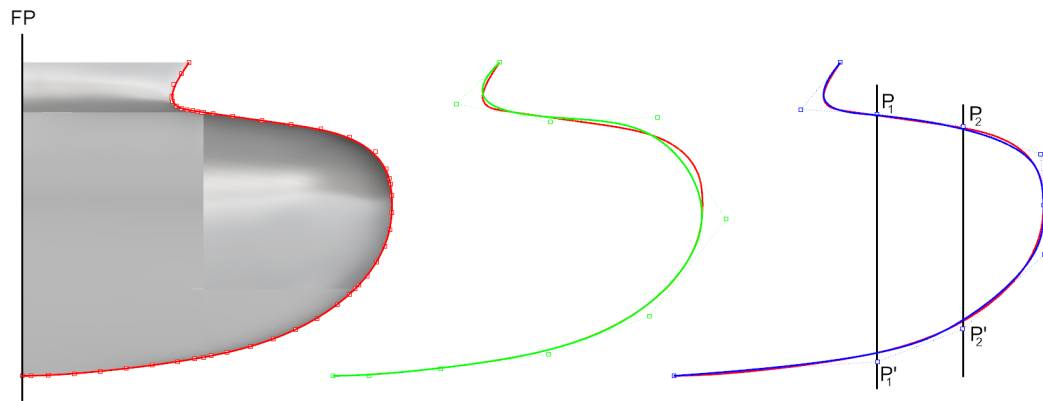


Figure 4.8: Extraction of bow profile curve (BowPrfCrv) and its control points, and adjustment. Depicted in red color is the extracted BowPrfCrv consisting of 45 control points. In black is the degree-3 rebuilt curve with 10 control points. In blue is depicted the final adjusted curve with a more accurate match in comparison with the rebuilt curve. Control points $P_1-P'_1$ and $P_2-P'_2$ are adjusted to be collinear and they define the location of the bulb curves.

4.3.4.2 Bulb curves, bow offset and overhang offset polylines

Next step is to provide the bulb its final shape. For this, the construction of Bulb (BulbCrvs) curves and offset polylines is necessary (Figs. 4.9 and 4.10). TshipPM is using 2 bulb curves, which are extracted from the parent hull by at the sections located at P_1 and P_2 of the rebuilt bow profile curve, as shown in Fig. 4.8. We demonstrate the extraction and rebuild of one bulb curve, while we show both bow offset and bow overhang polylines. The steps are as follows:

1. Extract BulbCrv from parent ship;
2. Rebuild curve (degree = 3) using as few as many control points as TshipPM uses;
3. Adjust if necessary the rebuilt curve with an accurate match with the extracted curve;
4. Assign proper values to external parameters of linear functions (Eq. 2.1) to match the coordinates of the control points obtained in the previous step (see Figs. 2.15 and 2.16).

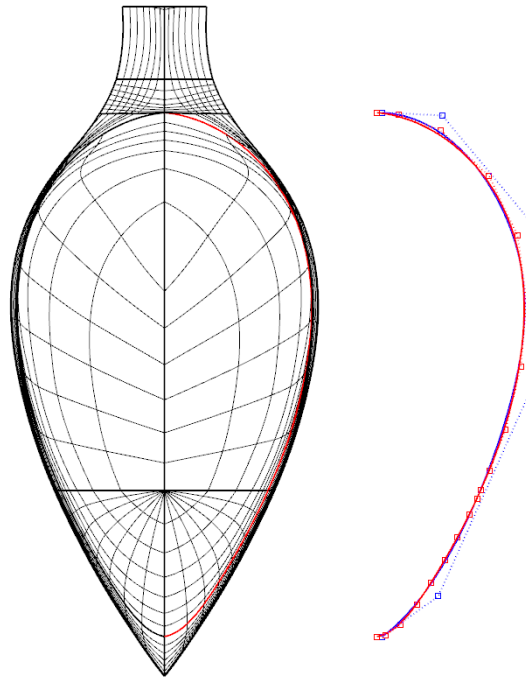


Figure 4.9: Extraction of bulb curve and its control points from parent ship. Depicted in **red** color is the extracted curve, in **blue** is the adjusted curve.

The offset polyline are constructed by directly offsetting a set of control points of *BowPrfCrv*, as shown in Fig. 4.10. The bow offset polyline (**blue**) is essential for providing shape to bulbous bow, while the overhang offset polyline (**red**) is necessary to create the thickness at the straight segment connecting the bow overhang with the bulb.

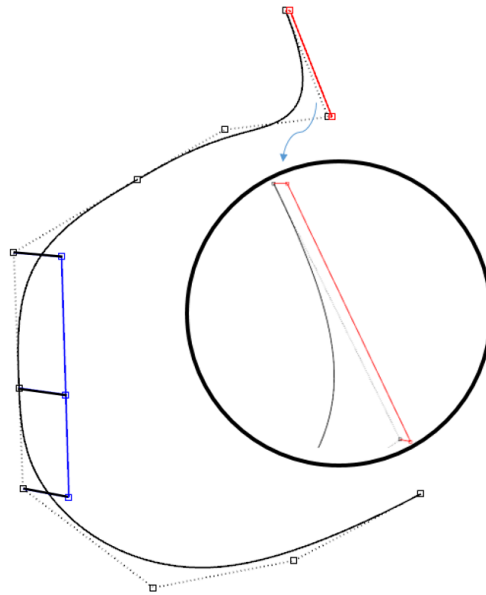


Figure 4.10: Construction of offset curves: depicted in blue is the bow offset polyline and in red is the overhang offset polyline. In black are the lines used for their interconnection with the bow profile curve.

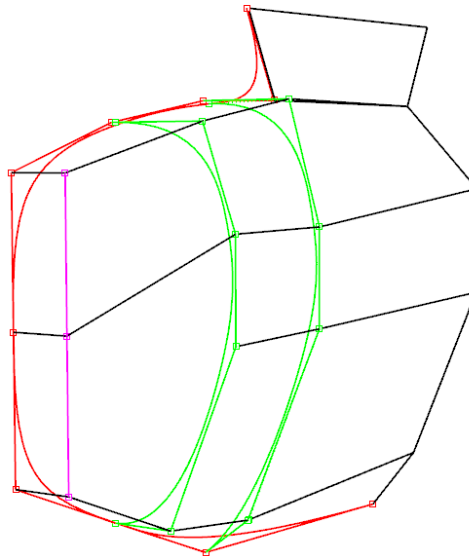


Figure 4.11: The control cage of the bulb. BowPrfCrv depicted in red, bulb curves in green, offset polylines in magenta, and interconnecting lines in black color.

Once bulb curves and offset polylines are set, they are connected with the BowPrfCrv to eventually form bow's control cage (Fig. 4.11), which is then interconnected with

FP_Crv's control polygon.

4.3.5 Stern control cage construction

To finalise the remodelling of the hull we develop the control cage of the stern and interconnect it with AP_Crv. The process consists of the following steps:

- A. Construction of AP_Crv which shares two control points with SternPrfCrv;
- B. SternPrfCrv and Tube Curve (TubeCrv) extraction;
- C. Interconnection of the stern with the midship.

4.3.5.1 Afterward perpendicular transition curve and tube curve

Fig. 4.12 illustrates the extraction of Afterward Perpendicular (AP_Crv) and Tube (TubeCrv) curves' control points. The steps are as follows:

1. Extract AP_Crv and TubeCrv from parent ship-hull;
2. Rebuild curves (degree = 3) using as many control points as in TshipPM;
3. Adjust if necessary the rebuilt curves with an accurate match with the extracted one;
4. Assign proper values to external parameters of linear functions (Eq. 2.1) to match the coordinates of the control points obtained in the previous step (see Figs 2.17 and 2.18).

The result of the afore-mentioned process is illustrated in Fig. 4.12. It is important to note that the rebuilt (green) curve has a control point (P_0) which is located further to the left of the vertical symmetric plane. This is not an acceptable location of any control point as, in case it is moved even further to the left, an invalid geometry would occur, as the two symmetric halves of the hull would intersect. We resolve this issue by imposing the adjusted curve (blue) to approximate the extracted (red) curve while

having three collinear control points (P'_0, P'_1, P'_2) which are aligned with the vertical symmetric plane and constrained to be placed further to the left.

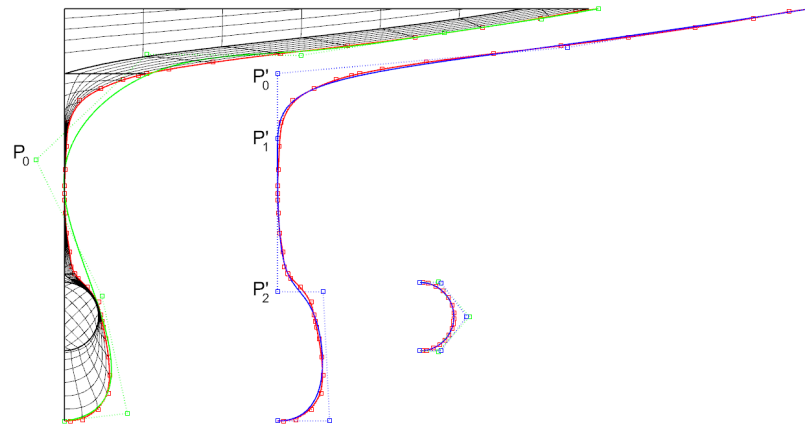


Figure 4.12: Extraction of the control points for Afterward Perpendicular (AP_Crv, middle) and Tube (TubeCrv, right) curves, depicted in blue. In red and green are depicted the extracted curves (left).

4.3.5.2 Stern profile curve

The extraction of SternPrfCrv is as follows:

1. Extract SternPrfCrv from parent ship, excluding the planar curve which represents the circumference of the TubeCrv. We result with 3 curves (Figs 4.13 and 4.14);
2. Rebuild curves (degree = 3) using as many control points as TshipPM uses;
3. Adjust the rebuilt curves with a more accurate match with the extracted curves;
4. Assign proper values to external parameters of linear functions (Eq. 2.1) to match the coordinates of the control points obtained in the previous step (see Fig. 2.17).

In the case of the stern, we end up with 3 curves instead of 1. To begin with, the part of the SternPrfCrv at the waterline level is extracted. Fig. 4.13 illustrates the extracted (red), rebuilt (green), and adjusted (blue) curves. Note that both rebuilt and adjusted curves consist of 4 control points. The latter seemingly has 3 control points because the KCS parent ship-hull has a transom stern ($\widehat{TransomB} = 0.01$ and

$TransomB \approx 0m$). Next we extract the part of the SternPrfCrv which instantiates at the end of the waterline length on the waterline level and ends at the bottom part of the FP_Crv (Fig. 4.14). Again, the extracted curve is depicted in red, the rebuilt in green, and the adjusted in blue color. Adjusted curve's control points P_A and P_B belong to the FP_Crv, so we do not create any new points, but we adjust P'_A and P'_B of the rebuilt curve to coincide with P_A and P_B . P_C is located at the middle of the straight line segment (P_AP_B) . Fig. 4.15 illustrates the control cage of the stern with all the interconnected lines.

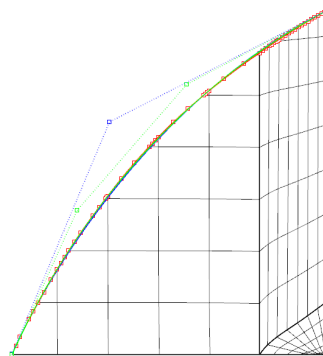


Figure 4.13: Extraction of the part of the Stern Profile Curve (SternPrfCrv) at the waterline level and adjustment. Depicted in red color is the originally extracted Stern-PrfCrv, while in green is the rebuilt and in blue is the adjusted curve.

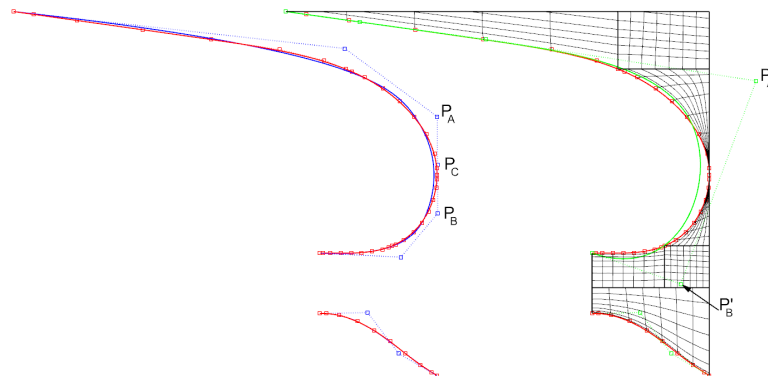


Figure 4.14: Extraction of the stern profile curves and their control points, and adjustment. Depicted in red color are the originally extracted stern profile curves, in green the rebuilt, and in blue the adjusted one.

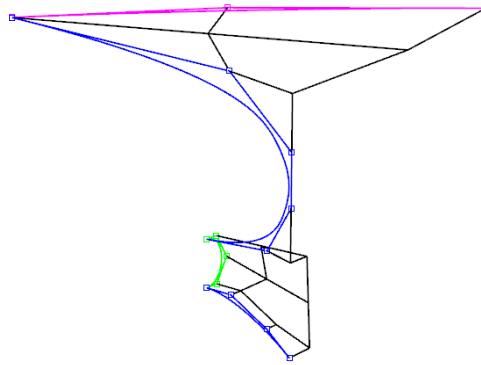


Figure 4.15: Control cage of the stern. SternPrfCrv depicted in magenta, Stern_Crv in blue, TubeCrv in green, interconnections depicted in black color.

4.4 Evaluation and optimisation

With the completion of the control cage construction of the TshipPM model, evaluation under a predefined set of criteria (§4.2) is conducted, which provides important feedback to users and developers, indicating whether or not and at which part and/or exact location of the parametric model there is a necessity for adjustments, during which both external and internal (only for developers) parameter values should be reconsidered and re-evaluated.

The optimisation of the parametric model is conducted in two steps:

1. Redefine internal (for developers, if necessary) and external (for both developers and users) parameter values;
2. Evaluate the parametric model using the set of geometric and design criteria described in §4.2 for comparing the parametric and parent models.

At this point we should stress that, from a developer's point view, evaluation process is of immense importance as it can aid towards the enhancement of the PM under development. The evaluation of the model is split in two phases:

1. Phase 1: Pre-optimisation evaluation of the parametric model, which is the subsequent phase of the control cage construction as described so far, and indicates the

locations on TshipPM's produced model which need adjustments to approximate closer the parent hull;

2. Phase 2: Post-optimisation evaluation of the optimised parametric model.

In between the two phases, the optimisation of TshipPM is conducted, as described later on (§4.4.2). Scheme of Fig. 4.16 provides diagrammatically the steps of evaluation and optimisation.

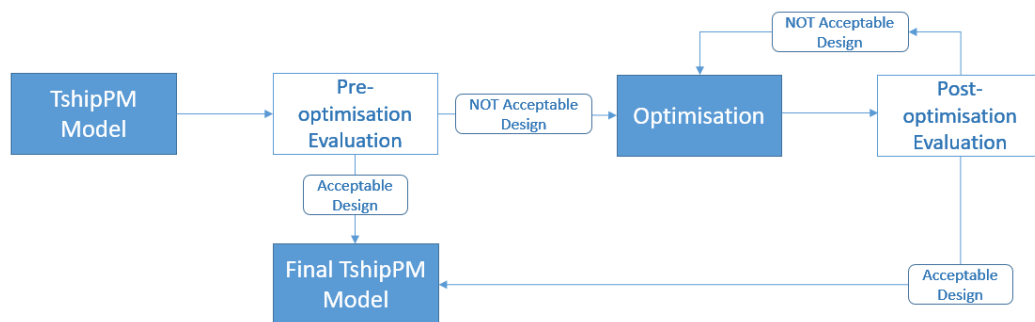


Figure 4.16: Diagram of the evaluation and optimisation process of the TshipPM model.

4.4.1 Pre-optimisation process and results

After the construction of the control cage, Rhino's T-splines plug-in is executed to obtain the smooth model. The model is undergoing evaluation with respect to the parent ship-hull, using a pre-determined set of criteria. For ease of reference we call the TshipPM model prior to optimisation as PM_BO and the parent ship-hull as Parent. It is important to note at this point that the volume-moment values for the segmented parts have been calculated for slightly different segmentation locations for the PM_BO model than the optimised model presented in Chapter 5, as the work for the pre-optimised model had been conducted before the final decision for the exact coordinates of segmentation was taken.

Smoothness: We are focusing on the bow and stern areas which are the most complex parts of the ship-hull shape-wise. As Figs. 4.17 and 4.18 show, TshipPM, taking advantage of T-splines which can represent an object with a single or very few patches,

produces at least G^1 -continuous surfaces throughout the ship-hull's surface. On the other hand, Parent has locations where neighbouring surfaces have been joined with C^0 -continuity, mainly due to the fact that the CAD parent hull model has been build using many patches around the bow and stern areas, which makes the joining process of the neighbouring surfaces, with -at least- tangent continuity, cumbersome.

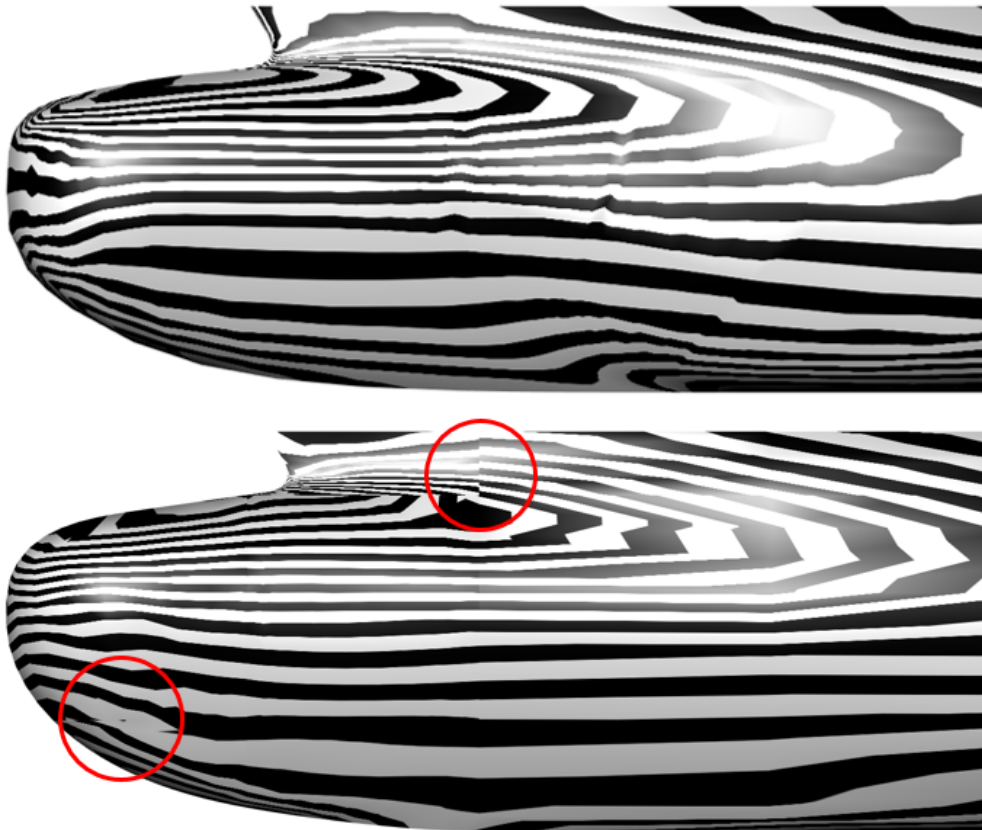


Figure 4.17: Smoothness at bow of PM_BO (top) and Parent (bottom). PM_BO's surface is at least G^1 -continuous everywhere, while Parent consists of areas of C^0 -continuity (indicated by red circles).

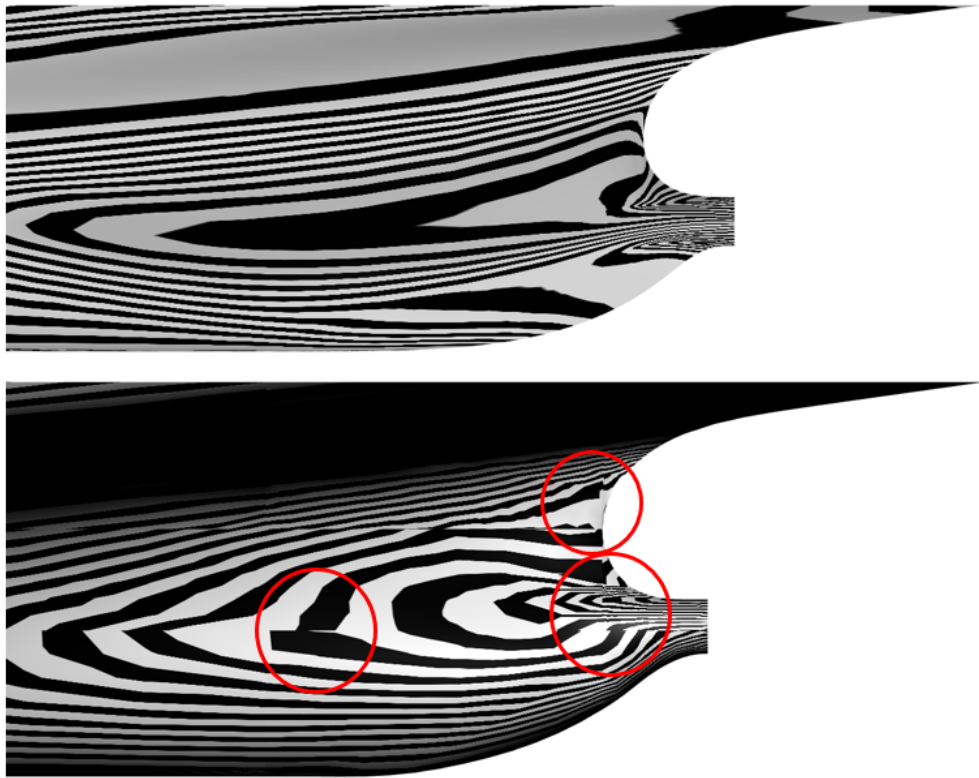


Figure 4.18: Smoothness at stern of PM_BO (top) and Parent (bottom). PM_BO's surface is at least G^1 -continuous everywhere, while Parent consists of areas of C^0 -continuity, encircled in red.

Fairness: Figs 4.19 and 4.20 illustrate the fairness of PM_BO and Parent, with views taken to focus on the bow and stern areas respectively. Fairness has been measured in terms of the *Gaussian curvature* provided in Rhino5 by the *Curvature Analysis* property, using a curvature range of $(-10^{-6}, 10^{-6})$. We have opted for a strict range in the very proximity of zero to identify only the absolutely flat areas and ignore those who are seemingly flat, but actually are not.

It is clear enough that TshipPM produces instances with much more fair surfaces comparing to the Parent CAD model all along the length of the hull, and especially in the more complex areas of bow and stern, and in the transition towards them. This is due to the fact that TshipPM creates one surface comparing to the multi-patch object of the parent CAD model, and extraneous sign-variations of the Gaussian curvature are limited.

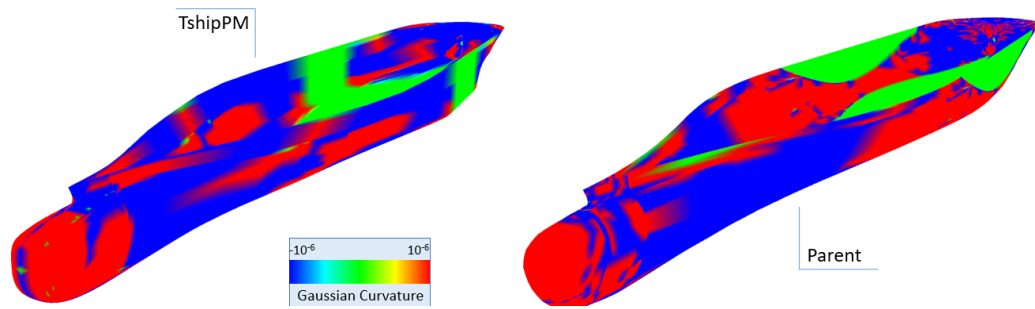


Figure 4.19: Color map indicating the Gaussian curvature of TshipPM and parent ship-hulls; Bow view.

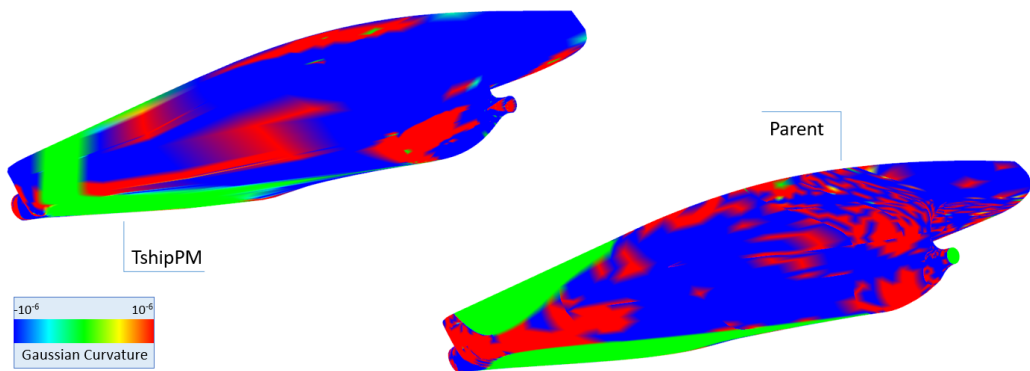


Figure 4.20: Color map indicating the Gaussian curvature of TshipPM and parent ship-hulls; Stern view.

Moments: First step is to examine whether or not moment values converge with respect to Parent. Following the control cage construction process as described in this chapter, one is to expect deviations for volume and volume centroid of around 1%. As far as inertia is concerned, deviations of more than 2% is normal to occur, as 2nd order moments are more sensitive compared to lower order moments.

To investigate with increased accuracy the deviation of moments, we segment PM_BO and Parent into three parts, namely Fwd_Part, Parallel Middle Part, and Aft_Part. To do so, we derive the longitudinal positions of starting and end points of the cylindrical middle body of PM_BO, which are $x_1 = 104.7\text{m}$, $x_2 = 151.7\text{m}$. In Rhino5's environment, we create the three segments for both models (Fig. 4.21).

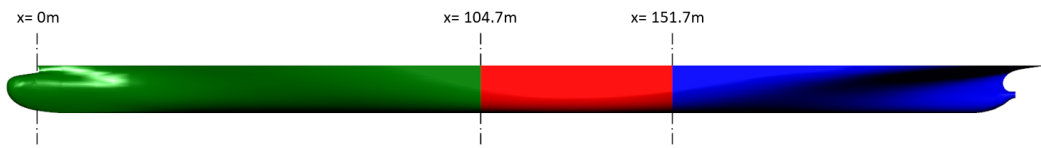


Figure 4.21: Segmentation of PM_BO and Parent to calculate volume moments for Fwd_Part, Parallel Middle Part, and Aft_Part.

We then calculate the volume moments up to 2nd-order for each segment of both models separately, as well as their percentage differences. Although results look pretty acceptable as far as moments up to 1st are concerned, they witness deviations larger than 1% for 2nd-order moments with respect to x-axis for Fwd_Part, while for Aft_Part, there are deviation of more than 1% for volume and inertia. This is an indication that we have to look closer and optimise Fwd_Part and Aft_Part. Tables 4.1 and 4.2 include the values of moments up to 2nd-order and their percentage deviations for the full ship-hull and all three segments of both models respectively. All deviations larger than 1% are highlighted in red color.

Table 4.1: Comparison of moments between KCS parent ship-hull and PM_BO model. (Full model)

| Models | Volume (m ³) | Volume Centroid (m) | Inertia (m ⁵) |
|----------------|-----------------------------|---------------------------|------------------------------|
| Parent Ship | 52,053 | x: 118.407 | Ix: 3,892,800 |
| | | y: 0 | Iy: 130,372,480 |
| | | z: -4.894 | Iz: 133,304,065 |
| PM_BO | 51,663 | x: 117.849 | Ix: 3,825,544 |
| | | y: 0 | Iy: 128,455,542 |
| | | z: -4.918 | Iz: 131,316,381 |
| Difference (%) | | x: 0.473 | Ix: 1.758 |
| PM1 vs | 0.754 | y: 0 | Iy: 1.492 |
| Parent Ship | | z: -0.476 | Iz: 1.514 |

Notes: Inertia has been calculated with respect to Centroid Coordinate Axis. Negative percentage indicates that parent ship-hull's absolute value is lower than that of PM_BO.

Table 4.2: Comparison of moments between KCS parent ship-hull and PM_BO model. (Segmentation at $x^1 = 104.7\text{m}$, and $x^2 = 151.7\text{m}$)

| Models | Volume (m^3) | Volume Centroid (m) | Inertia (m^5) |
|-----------------------|-----------------------------------|----------------------------------|------------------------------------|
| Forward Part | | | |
| Parent Ship | 20,139 | x: 66.570 | Ix: 1,281,763 |
| | | y: 0 | Iy: 12,289,833 |
| | | z: -4.902 | Iz: 13,213,492 |
| PM_BO | 19,990 | x: 66.151 | Ix: 1,235,346 |
| | | y: 0 | Iy: 12,365,082 |
| | | z: -4.910 | Iz: 13,241,339 |
| Difference (%) | | x: 0.633 | Ix: 3.757 |
| PM_BO vs | 0.750 | y: 0 | Iy: -0.609 |
| Parent Ship | | z: -0.156 | Iz: -0.210 |
| Middle Part | | | |
| Parent Ship | 14,993 | x: 123.972 | Ix: 1,393,339 |
| | | y: 0 | Iy: 2,550,009 |
| | | z: -5.308 | Iz: 3,658,949 |
| PM_BO | 15,030 | x: 123.972 | Ix: 1,401,532 |
| | | y: 0 | Iy: 2,563,768 |
| | | z: -5.321 | Iz: 3,679,001 |
| Difference (%) | | x: 0.076 | Ix: -0.585 |
| PM_BO vs | -0.250 | y: 0 | Iy: -0.537 |
| Parent Ship | | z: -0.247 | Iz: -0.545 |
| Afterward Part | | | |

Continued on next page

Table 4.2 – *Continued from previous page*

| Models | Volume (m ³) | Volume | Inertia (m ⁵) |
|----------------|-----------------------------|-----------------|------------------------------|
| | | Centroid (m) | |
| Parent Ship | 16,919 | x: 175.178 | Ix: 1,212,449 |
| | | y: 0 | Iy: 6,415,948 |
| | | z: -4.519 | Iz: 7,319,577 |
| PM_BO | 16,642 | x: 174.501 | Ix: 1,184,132 |
| | | y: 0 | Iy: 6,136,896 |
| | | z: -4.563 | Iz: 7,010,778 |
| Difference (%) | | x: -0.388 | Ix: 2.391 |
| PM_BO vs | 1.667 | y: 0 | Iy: 4.547 |
| Parent Ship | | z: -0.974 | Iz: 4.405 |

Notes: Inertia has been calculated with respect to Centroid Coordinate Axis. Negative percentage indicates that parent ship-hull's absolute value is lower than that of PM_BO.

Sectional Area Curve: We already have an indication that Fwd_Part and Aft_Part do not behave as desired. Using sectional area curves (SAC) we can locate the exact intervals or longitudinal positions where sectional area values (SAV) are different between PM_BO and Parent. As illustrated in Fig. 4.22 sectional area values differ significantly at the longitudinal intervals [10m, 34m], [67m, 100m], and [181m, 224m]. This is an indication the optimisation process should mainly focus in the transition curves bordered by the limits of those intervals.

4.4.2 Optimisation for a more accurate remodelling of the parent ship-hull

The evaluation process showed that the instance needs adjustments both in forward and afterward parts. Herein we illustrate only the adjustments to better approximate the Fwd_Part. SAC comparison showed critical deviations in the longitudinal

intervals [10m, 34m] and [67m, 100m]. As shown in Fig. 4.23, the former interval is located between the FP_Crv and the FTCD, while the latter starts a few meters before the FTCC and extends up to the cylindrical middle part. We need to adjust all the transition curves which lie between those intervals. We are extracting from PM_BO the sectional curves at the same location as the transition curves are located and we compare them one-by-one with the corresponding sectional curves extracted from the Parent at the same longitudinal position.

Fig. 4.24 shows all four sectional curves of the Fwd_Part both for PM_BO and Parent in blue and red color respectively. Although the sectional curves of the models under investigation are close enough, the mismatch between the corresponding sectional curves is apparent. The coordinates of the control points used in TshipPM should be adjusted by manipulating the values of the external parameters (and internal for developers) which control the location of each curve's control points. We manipulate them until the sectional curve of the TshipPM instance approximates -as closely as desired- Parent's corresponding sectional curve. The process consists of the following steps, which are illustrated in the diagram of Fig. 4.25:

1. Extract symmetric-half's critical sections;
2. Adjust the control points of the TshipPM model so that the section of the TshipPM model is approximating parent's corresponding one;
3. Build the T-spline surface from the new control cage;
4. Evaluate the sectional curves;
5. Re-adjust control points and evaluate if required.

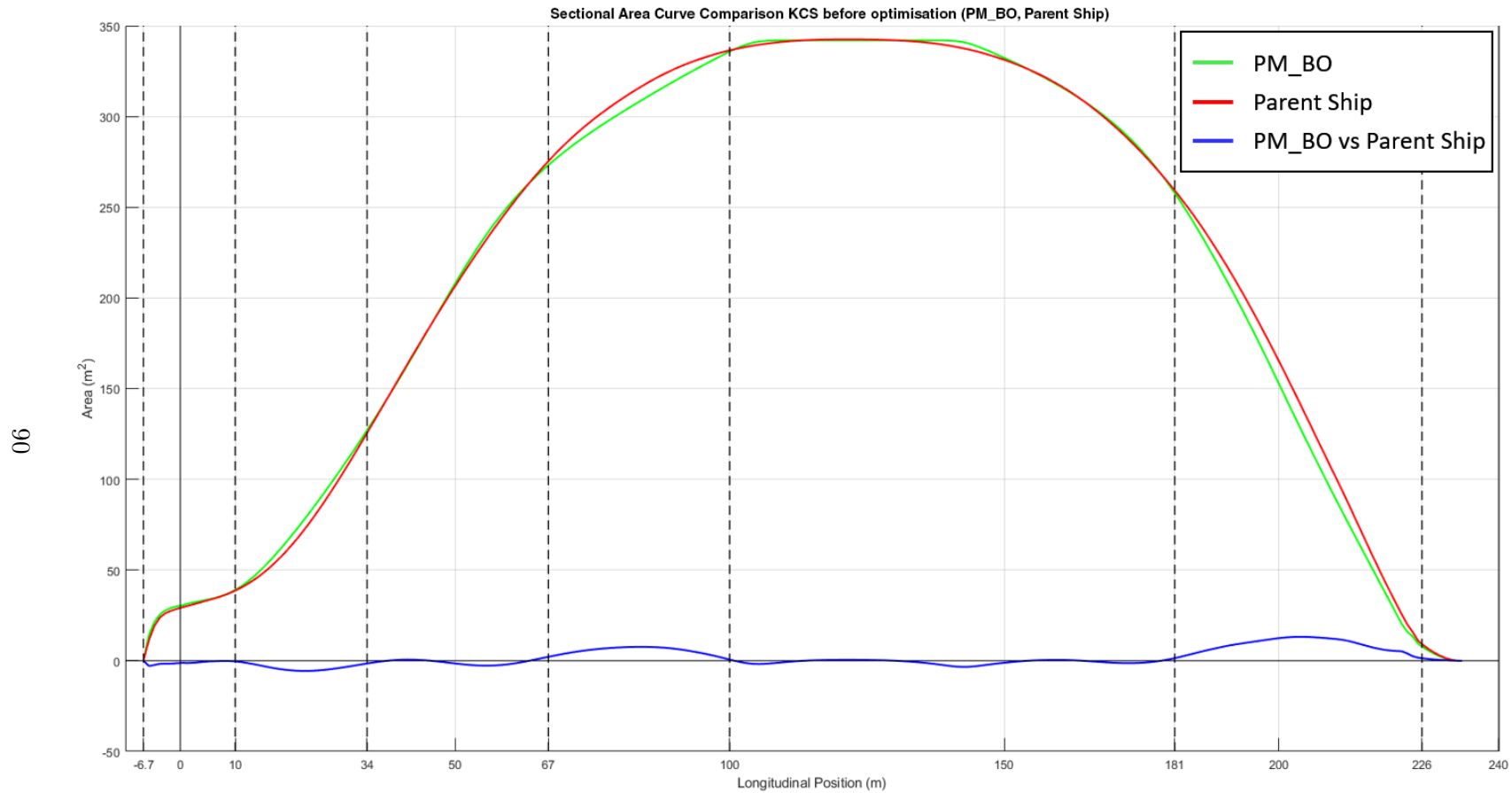


Figure 4.22: Comparison of Sectional Area Curves (SAC) between PM_BO and Parent. In green color PM_BO SAC is depicted, Parent ship SAC in red, while their deviation is depicted in blue color.

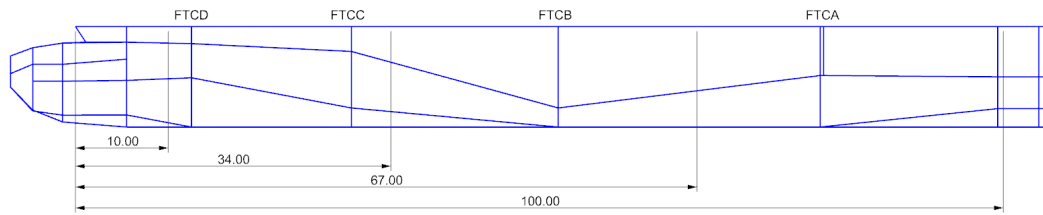


Figure 4.23: Longitudinal intervals where analysis using SAC showed critical deviations between the sectional areas of TshipPM and parent models.

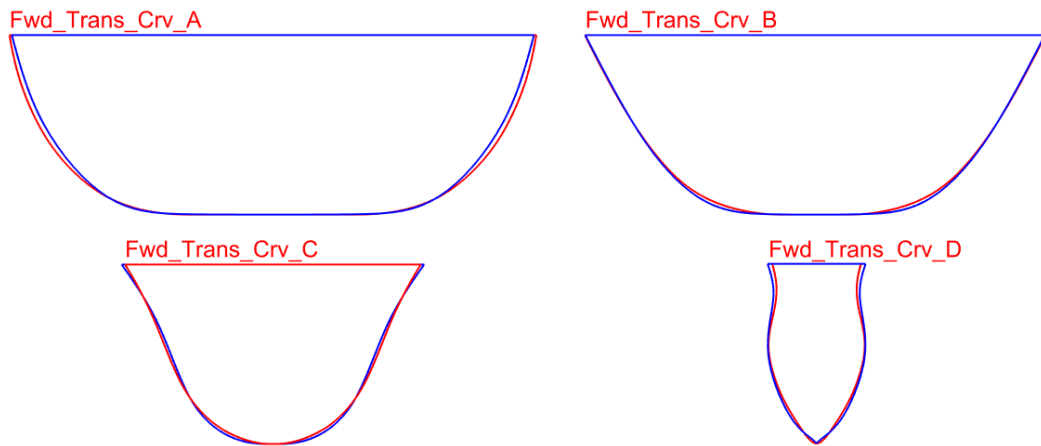


Figure 4.24: Sectional areas of PM_BO (blue) and Parent (red) models of the forward part.

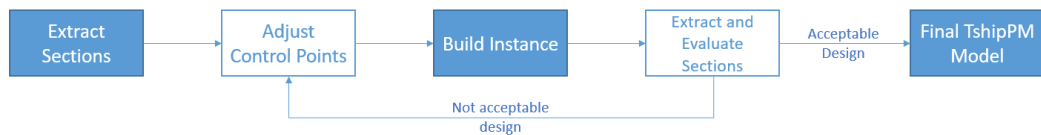


Figure 4.25: Evaluation-optimisation workflow to attain an acceptable ship-hull instance which lies geometrically in the proximity of the parent hull.

In some cases during the evaluation process developers might consider necessary to implement additional control points on the control curves, or even new control curves, mainly in areas where the representation of the objects is not accurate in comparison with the parent object, therefore the optimisation process is unable to create a parametric model in the proximity of the Parent. The extra control points will increase the flexibility of the PM to produce a broader variety of ship-hulls, therefore providing the capability to approximate the parent hull with increased accuracy.

4.4.3 Post-optimisation evaluation: results and comparison with parent hull

In this chapter we present the results and conduct the comparison between TshipPM’s KVLCC ship-hull instance and Parent, which has been remodelled from the deck line (not waterline) and below, with $L_{wl} = 329\text{m}$, $T = 28\text{m}$, and $B = 58\text{m}$. On top of the employed evaluation criteria, we evaluate TshipPM using *Hausdorff distance*, as a metric for geometric variation. The results for KCS, which was the case study for the remodelling process, are presented in Chapter 5.

Before we proceed to the results, it is necessary to mention that, purposefully, the version of TshipPM used for the remodelling of KVLCC (TshipPM_T) uses fewer transition curves than that of KCS (TshipPM_C). Specifically, as Fig. 4.26 shows, there are 3 forward and 3 afterward transition curves (including FP_Crv and AP_Crv), and 1 bulb curve, instead of 5, 5, and 2 respectively in TshipPM_C. The purpose of this is twofold: firstly, KVLCC ship-hull’s geometry is less complex than KCS’s, therefore fewer control points are needed to remodel the parent ship-hull. In fact, TshipPM uses 190 control points to represent the KVLCC hull, instead of 276 for KCS remodelling. Secondly, we would like to emphasise on the fact that more accurate remodelling can be succeeded with the introduction of just a set of 2-3 control curves.

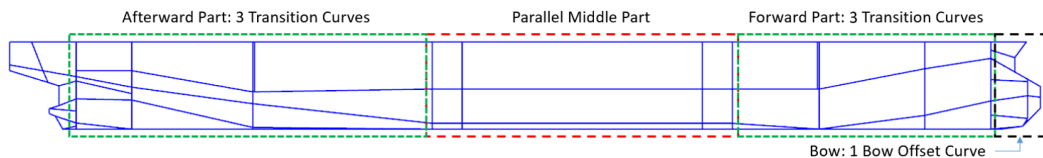


Figure 4.26: KVLCC’s control cage with fewer control curves compared to the version of TshipPM used for the remodelling of the KCS ship-hull.

Smoothness: As already mentioned, TshipPM is taking advantage of T-splines which produce at least G^1 -continuous surfaces throughout the ship-hull’s surface. The smoothness of both hulls is evaluated with the aid of Figs. 4.27 and 4.28.

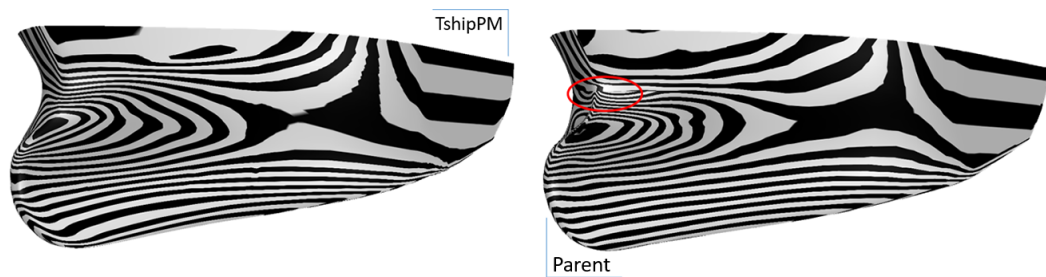


Figure 4.27: Bow view: smoothness of TshipPM's (left) and Parent's (right) models. TshipPM's surface is at least G^1 -continuous everywhere, while Parent consists of areas of C^0 -continuity. An example is indicated by the red circle.

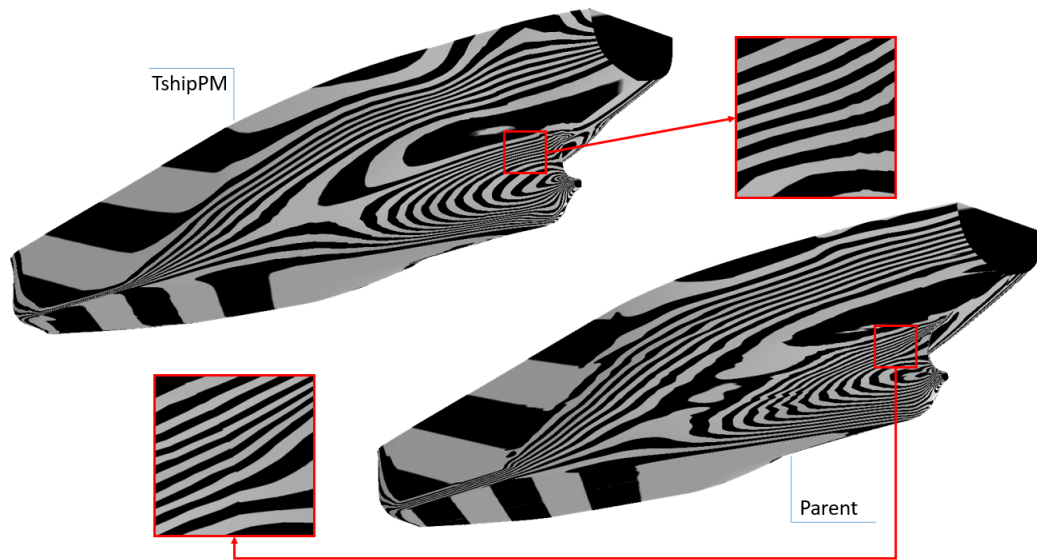


Figure 4.28: Stern view: smoothness of TshipPM's (left) and Parent's (right) models. TshipPM's surface is at least G^1 -continuous everywhere, while Parent consists of areas of C^0 -continuity. An example is indicated by the red square-frame.

Fairness: In order to use a color map capable to easily reveal sign variations of the Gaussian curvature, a very short interval around $K = 0$ has been selected, i.e., $[-10^{-6}, 10^{-6}]$. Blue color illustrates areas of negative K , i.e., principal curvatures are of opposite sign, implying that the surface has locally a saddle-like (non-convex) shape. On the contrary, red color implies that both principal curvatures share the same sign, equivalently the surface is locally convex, resembling the shape of a sphere. Finally,

green color is used when $K = 0$, in which case the surface is cylindrical or flat. Figs 4.29 and 4.30 illustrate the fairness of the TshipPM model and the Parent, focusing at the bow and the stern area respectively. TshipPM produces instances with much more fair surfaces comparing to the Parent CAD model, while the cylindrical middle part is represented adequately.

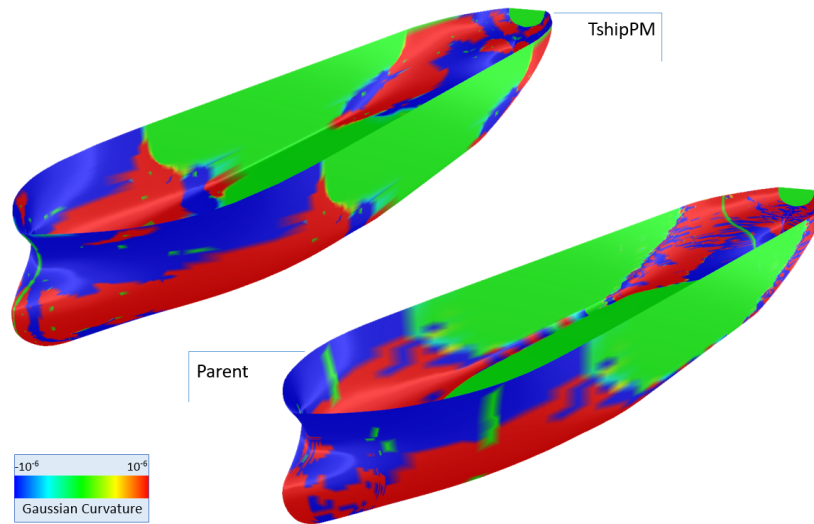


Figure 4.29: Color map indicating the Gaussian curvature of the TshipPM model and the Parent; Bow view.

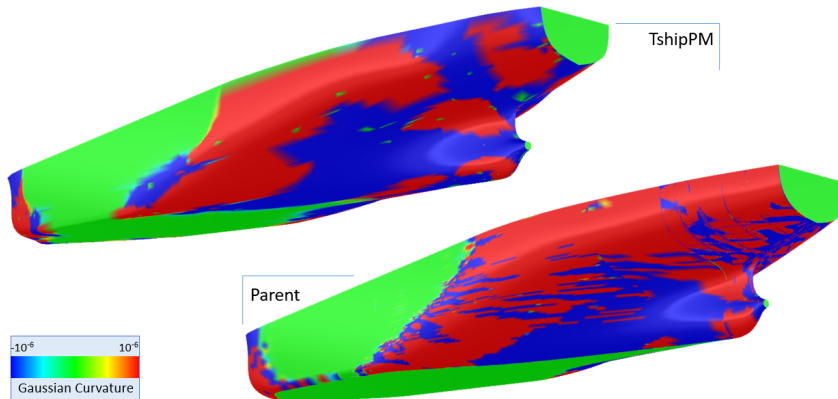


Figure 4.30: Color map indicating the Gaussian curvature of the TshipPM model and the Parent; Stern view.

Moments: Tables 4.3 and 4.4 present the moments for the whole models and the segmented ones respectively. For the whole model, volume has a difference of less than 0.5%, and volume centroid values are almost everywhere identical, while difference in moments of inertia, which are more sensitive, are lower than 1%. As far as the segmented models are concerned, all differences between the corresponding parts are lower than 1%, with parallel middle part's values very close for the two models, while only moments of inertia with respect to the x-coordinate are higher than 1% but still lower than 2%.

Table 4.3: Comparison of moments between KVLCC parent ship-hull and TshipPM model.

| Models | Volume (m ³) | Volume Centroid (m) | Inertia (m ⁵) |
|--------------------|------------------------------------|---|-------------------------------------|
| Whole Model | | | |
| Parent Ship | 435,577 | x: 155.714 | Ix: 1.347·10 ⁸ |
| | | y: 0 | Iy: 2.884·10 ⁹ |
| | | z: -13.281 | Iz: 2.964·10 ⁹ |
| TshipPM | 433,711 | x: 155.636 | Ix: 1.335·10 ⁸ |
| | | y: 0 | Iy: 2.861·10 ⁹ |
| | | z: -13.281 | Iz: 2.940·10 ⁹ |
| Difference (%) | | x: 0.05 | Ix: 0.9 |
| TshipPM vs | 0.431 | y: 0 | Iy: 0.8 |
| Parent Ship | | z: 0 | Iz: 0.8 |

Notes: Inertia has been calculated with respect to Centroid Coordinate Axis. Negative percentage indicates that parent ship-hull's absolute value is lower than that of TshipPM's instance.

Table 4.4: Comparison of moments between KVLCC parent ship-hull and TshipPM model.

 (Segmentation at $x^1 = 99.69\text{m}$, and $x^2 = 192.30\text{m}$)

| Models | Volume (m^3) | Volume Centroid (m) | Inertia (m^5) |
|-----------------------|-----------------------------------|----------------------------------|------------------------------------|
| Forward Part | | | |
| Parent Ship | 121,166 | x: 55.714 | Ix: $3.492 \cdot 10^7$ |
| | | y: 0 | Iy: $7.905 \cdot 10^7$ |
| | | z: -13.538 | Iz: $9.876 \cdot 10^7$ |
| TshipPM | 120,379 | x: 55.788 | Ix: $3.441 \cdot 10^7$ |
| | | y: 0 | Iy: $7.883 \cdot 10^7$ |
| | | z: -13.542 | Iz: $9.810 \cdot 10^7$ |
| Difference (%) | | x: -0.1 | Ix: 1.4 |
| TshipPM vs | 0.132 | y: 0 | Iy: 0.2 |
| Parent Ship | | z: -0.03 | Iz: 0.8 |
| Middle Part | | | |
| Parent Ship | 157,522 | x: 143.732 | Ix: $5.430 \cdot 10^7$ |
| | | y: 0 | Iy: $1.341 \cdot 10^8$ |
| | | z: -13.982 | Iz: $1.679 \cdot 10^8$ |
| TshipPM | 157,649 | x: 143.731 | Ix: $5.443 \cdot 10^8$ |
| | | y: 0 | Iy: $1.342 \cdot 10^9$ |
| | | z: -13.992 | Iz: $1.681 \cdot 10^8$ |
| Difference (%) | | x: 0.001 | Ix: -0.2 |
| TshipPM vs | -0.081 | y: 0 | Iy: -0.07 |
| Parent Ship | | z: -0.071 | Iz: -0.1 |
| Afterward Part | | | |
| Parent Ship | 156,889 | x: 244.976 | Ix: $4.526 \cdot 10^7$ |
| | | y: 0 | Iy: $1.863 \cdot 10^8$ |

Continued on next page

Table 4.4 – *Continued from previous page*

| Models | Volume (m ³) | Volume | Inertia |
|----------------|-----------------------------|-----------------|---------------------------|
| | | Centroid (m) | (m ⁵) |
| | | z: -12.379 | Iz: 2.125·10 ⁸ |
| | | x: 244.897 | Ix: 4.444·10 ⁸ |
| TshipPM | 155,682 | y: 0 | Iy: 1.849·10 ⁹ |
| | | z: -12.360 | Iz: 2.106·10 ⁸ |
| Difference (%) | | x: 0.03 | Ix: 1.8 |
| TshipPM vs | 0.78 | y: 0 | Iy: 0.7 |
| Parent Ship | | z: 0.01 | Iz: 0.9 |

Notes: Inertia has been calculated with respect to Centroid Coordinate Axis. Negative percentage indicates that parent ship-hull's absolute value is lower than that of TshipPM's instance.

Sectional Area Curve: Fig. 4.31 illustrates the SACs of the instances under investigation. TshipPM's instance is shown in green, Parent's in red, and their deviation in black. Green and red curves overlap almost all along the length of the hull, with more significant, though small deviations, in the complex bow area, as well as before the transition area between the afterward part and the stern, due to the lower number of transition curves TshipPM_T uses in the afterward part. In a nutshell, TshipPM is capable to produce models with SACs in the proximity of the Parent's SAC.

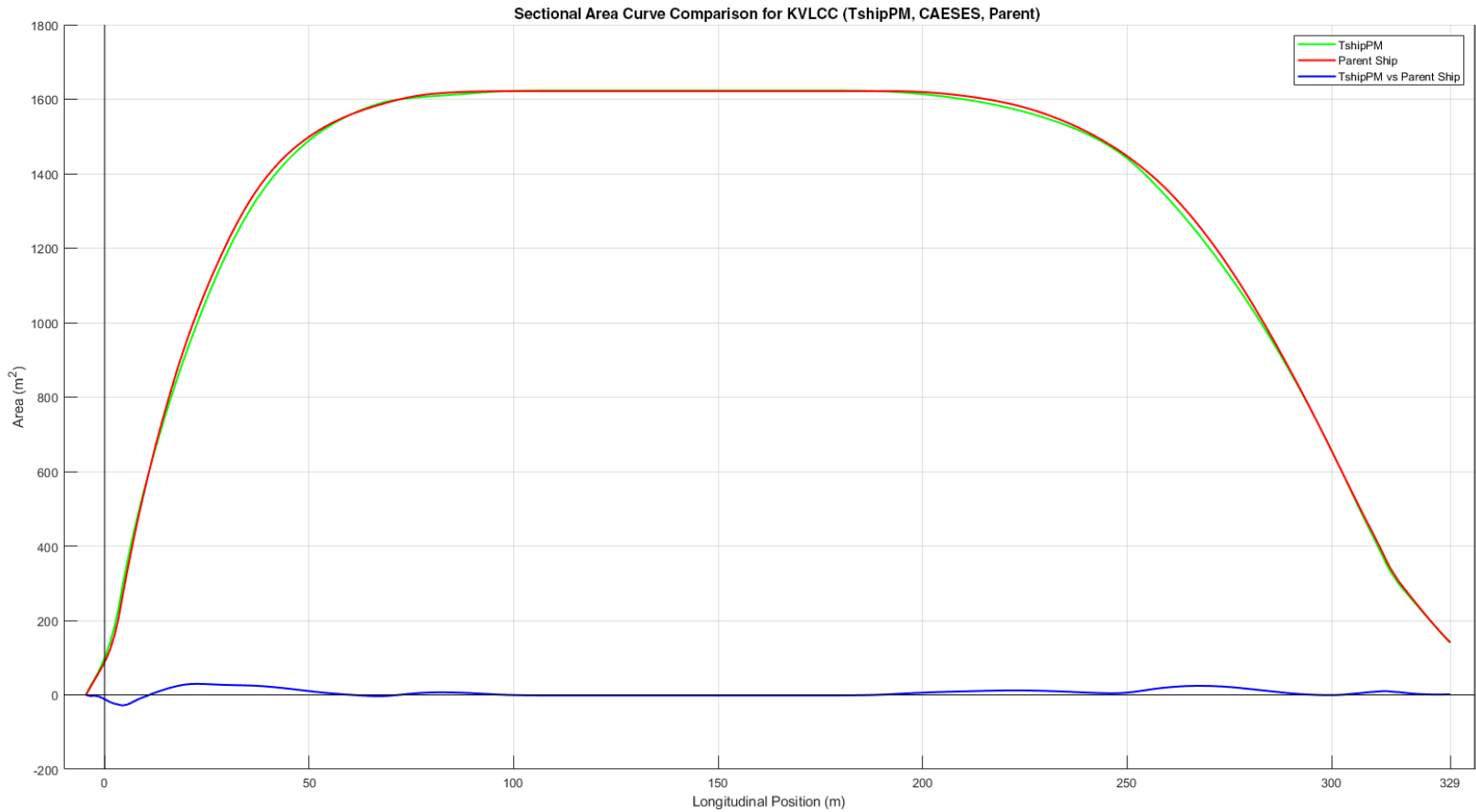


Figure 4.31: Comparison of sectional area curves (SAC) between TshipPM and Parent. TshipPM's SAC is depicted in green color, Parent's in red, while their deviation is depicted in blue color.

Hausdorff distance: In this section we adopt the concept of Hausdorff distance for evaluating TshipPM’s model in terms of its geometric deviation from the Parent. Using the MeshLab[®]² tool and a dense sampling on the surfaces involved (3.6×10^5 points), Fig. 4.32 depicts the distribution of deviation $\delta(\mathbf{x})$ of TshipPM’s model from the parent hull, defined as:

$$\delta_i(\mathbf{x}) = \min_{\mathbf{y} \in \text{PKCS}} d(\mathbf{x}, \mathbf{y}), \quad \mathbf{x} \in \text{PM}_i, \quad i = 1, 2 \quad (4.2)$$

where $d(\mathbf{x}, \mathbf{y})$ denotes the Euclidean distance between two points \mathbf{x} and \mathbf{y} . The color map ranges over $[0, 0.9\text{m}]$, which covers the maximum deviation measured for TshipPM’s model. **Red** areas indicate higher deviations while **blue** lower ones. Mean deviation is 0.11m. The areas of large deviations are limited at the bow and the transition area from the parallel middle part towards the afterward part, closer to the waterplane. Although the Hausdorff distance metric indicates acceptable results in terms of geometric variation, better results could be attained by introducing an extra bulb curve and an additional afterward transition curve in between the parallel middle part and the first afterward transition curve (see results for KCS in chapter 5, §5.8, where additional control curves are implemented).

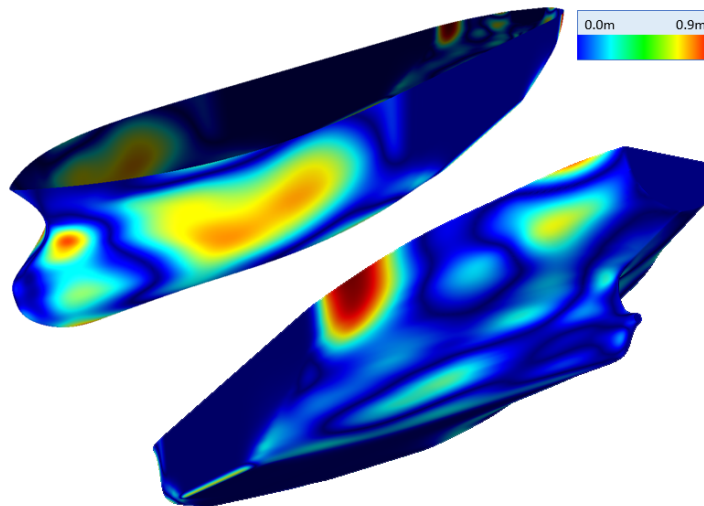


Figure 4.32: Hausdorff distance plot illustrating the geometric deviation between TshipPM’s produced model and Parent.

²<http://www.meshlab.net/>

Chapter 5

TshipPM vs CAESES: A comparative study

5.1 Introduction

This chapter focuses on comparing the performance of TshipPM with a well established commercial PM for ship design, namely CAESES [52]. This comparison will use the container-ship model KCS [9] as parent hull and the following criteria:

1. Common external parameters;
2. k^{th} -order moments ($k = 0, 1, 2$), i.e., volume, volume centroid, moment of inertia;
3. Sectional Area Curve (SAC);
4. Gaussian curvature as a measure of surface shape and fairness;
5. Sectional curvature for assessing the shape and fairness of sections at specific longitudinal positions;
6. Hausdorff distance to evaluate geometric variation.

The work presented in this chapter has been published in November 2019 in the *Ocean Engineering*¹ international journal, titled "A *T-splines-based parametric mod-*

¹<https://www.journals.elsevier.com/ocean-engineering>

eller for computer-aided ship design [7]”. Note that the results are slightly different than those in [7], as the work for the current document has been revisited and improved, but also the ship-hulls have been segmented in different locations compared to [7].

5.2 CAESES parametric modeller

CAESES adopts the classic naval architecture approach: a set of longitudinal lines, so-called *basic curves*, is laid out from which all information can be retrieved to subsequently establish the geometry of the ship-hull. This is done in a 3-stage process:

1. Parametric design of a suitable set of basic curves such as deck profile, design waterline, flat of side and flat of bottom, centerplane profile etc. The basic curves are built in agreement with a few prominent transversal curves, e.g., the main frame section, the transom, and, optionally, additional sections in the forward or afterward body.
2. Parametric modelling of design sections derived from the basic curves.
3. Generation of a set of B-spline surfaces that interpolate or closely approximate the design sections.

Curve design in CAESES is based on the concept of the *F*-spline curves [53]. These curves are formulated considering a base of 4 parameters: a) the starting and terminating positions of the curves, b) tangents at starting and terminating positions, c) the area, and d) the centroid of the area. They secure an inherently smooth and convex behaviour and reduce the need for further heuristic computations. Furthermore, *Property Distribution Curves* are introduced, representing the variation of important geometric properties of the hull form along, e.g., the longitudinal direction. Typical examples include SAC, the distribution curve of rise at the bottom, the design-waterline, deck profile and others.

In the present study, CAESES has been employed using 31 input (external in the terminology of TshipPM) parameters, depicted graphically in Fig. 5.1 and categorised

in Table 5.1 in three groups, namely a) *Main Dimensions*, b) *XmainFrame*, and c) *Forebody* and *Aftbody* parameters.

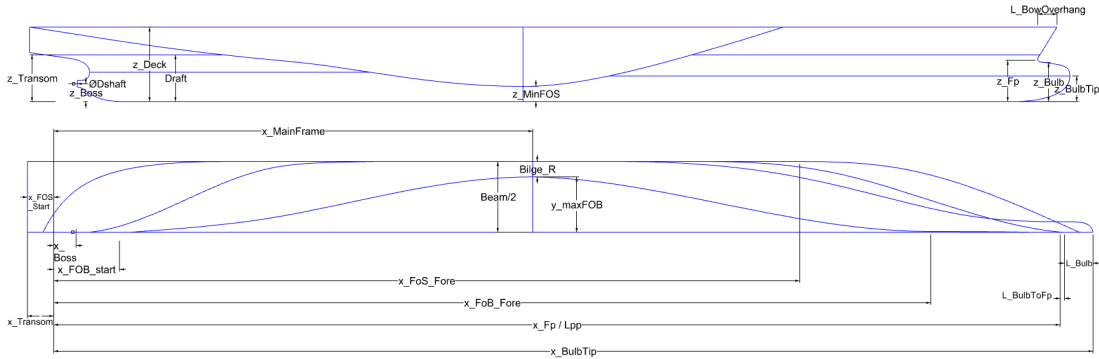


Figure 5.1: CAESES input parameters.

Table 5.1: CAESES Parameters and Groups.

| Parameter | Description |
|------------------------|--|
| Main Dimensions | |
| Lpp | Distance between perpendiculars |
| Beam | Beam (breadth) of ship |
| Draft | Design waterline of the ship |
| z_Deck | Deck height of the ship |
| y_MaxFOB | $\text{HalfBeam} \cdot y_MaxFOB_Fore_Factor$ |
| y_MaxFOB_Fore_Factor | Factor used in y_maxFOB |
| z_MinFOS | $\text{Height} \cdot zMinFOS_Fore_Factor$ |
| z_MinFOS_Fore_Factor | Factor used in z_MinFOS |
| XmainFrame | |
| x_MainFrame | x position of the largest section area $x_FP \cdot x_MainFrame_Factor$ |
| x_MainFrame_Factor | Factor used in x_MainFrame |
| Forebody | |
| x_Fp | x position of Forward Perpendicular |

Continued on next page

Table 5.1 – *Continued from previous page*

| Name of Parameter | Association to Dimensional Parameters |
|--------------------------|---|
| x.Peak | x position of end point of Bow Overhang |
| x.BulbTip | x position of Bulb Tip |
| x.FOB.Fore | x position of end point $x_{FP} \cdot x_{FOB.Fore.Factor}$ |
| x.FOB.Fore.Factor | Factor used in x.FOB.Fore |
| x.FOS.Fore | x position of end point $x_{FP} \cdot x_{FOS.Fore.Factor}$ |
| x.FOS.Fore.Factor | Factor used in x.FOS.Fore |
| z.Fp | z position of highest point of the Forward Perpendicular |
| z.Bulb | Bulb Height |
| z.BulbTip | Bulb Tip Height |
| L.Bulb | Length of Bulbous Bow |
| L.BulbToFp | Distance from Bulbous Bow Tip to FP |
| L.BowOverhang | Length of Bow Overhang |
| Aftbody | |
| x.FOB.Start | x position of starting point of Flat of Bottom $x_{FP} \cdot x_{FOB.Start.Factor}$ |
| x.FOB.Start.Factor | Factor used in x.FOB.Start |
| x.FOS.Start | x position of starting point of Flat of Side $x_{FP} \cdot x_{FOS.Start.Factor}$ |
| x.FOS.Start.Factor | Factor used in x.FOS.Start |
| x.Boss | x position of Shaft |
| x.Transom | x position of Transom |
| z.Transom | Transom Height |

Continued on next page

Table 5.1 – *Continued from previous page*

| Name of Parameter | Association to Dimensional Parameters |
|-------------------|--|
| Dshaft | Diameter of Shaft |

In the ensuing sections §§5.3-5.7 we present and discuss the results obtained from using TshipPM and CAESES for approximating the KCS parent hull. For ease of reference we call as PM1 and PM2 the models produced by TshipPM and CAESES respectively. All calculations have been conducted with models spatially located as shown in Fig. 5.2, with the tip of the bow overhang (most front part of the Lwl) being located at the start of the coordinate system.

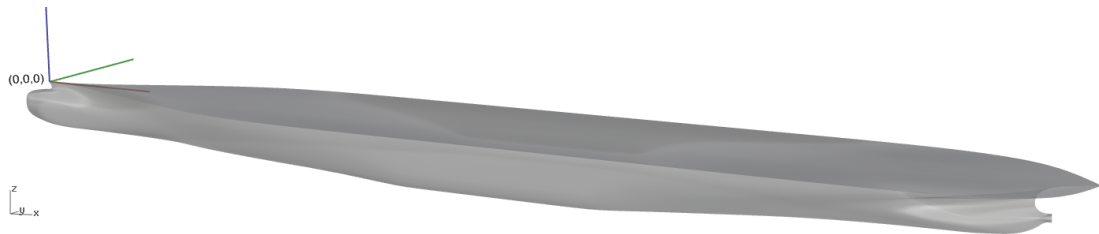


Figure 5.2: Coordinate system and models' orientation.

5.3 Common parameters

Though the adopted geometric representation and the parametric-modelling methodology employed by TshipPM and CAESES are different, the two PMs do share many common input parameters. In particular, the modellers have 14 common parameters (52% of TshipPM's and 45% of CAESES's parameters) collected in Table 5.2.

Table 5.2: TshipPM-CAESES common parameters list.

| CAESES | TshipPM |
|----------------------------------|------------------|
| Parameter | Parameter |
| Global Parameters | |
| Beam | B |
| Draft | T |
| Middle Part Parameters | |
| z_MinFOS | BilgeR |
| x_MainFrame | Mid_Pos |
| Forward Part Parameters | |
| x_Fp | FP_L_FromMS |
| x_FOB_Fore | FoB_Fwd_L |
| x_FOS_Fore | FoS_Fwd_L |
| z_Bulb | BulbH |
| L_BulbToFp | BulbL |
| Afterward Part Parameters | |
| x_FOB_start | FoB_Aft_L |
| x_FOS_start | FoS_Aft_L |
| x_Boss | TubeL |
| z_Transom | Transom_H |
| Dshaft | TubeR |

5.4 Moments comparison

In this section we present the results from comparing volumetric moments up to 2nd order, provided by PM1 and PM2 against the KCS parent hull. Comparison has been conducted both for the whole ship-hulls and for their segmented parts, namely forward ($x \in [-6.7m, 104.7m)$), middle ($x \in [104.7m, 151.7m)$) and afterward parts

($x \in [151.7m, total\ length]$). PM1 and parent ship-hull moments have been calculated via Rhino3D, while PM2 moments have been calculated using ANSYS Fluent[®]. Negative differences indicate that the parent ship-hull value is smaller than the corresponding value of PM1 or PM2 and vice versa. The obtained values and resulting deviations are collected in Tables 5.3 and 5.4.

Focusing on the whole ship-hull, volume, volume centroid, and moment of inertia for both PM1 and PM2 lie very close to those of the parent hull, with differences less than 1%, with the exception of the moments of inertia of PM2 in the x-coordinate, where a difference of 1.288% is observed. As for the volume, PM1 overestimates about 0.146%, while PM2's estimate differs by 0.073%. As far as centroid location is concerned, PM1's estimate deviates by 0.099% and -0.143% along the x - and z -axis respectively, while PM2's centroid estimate exhibits a variation of 0.038% and -0.001% , correspondingly. For both parametric models and the parent hull, the y -coordinate of centroid is zero in single precision accuracy. In terms of inertia, PM1's estimate differs from that of the parent hull by $\delta I_x = 0.026\%$, $\delta I_y = -0.458\%$ and $\delta I_z = -0.448\%$. For PM2, the corresponding quantities vary by 1.288%, 0.756%, 0.767%.

For the segmented hulls, both PM1 and PM2 show satisfactory behaviour for all calculated moments and for all three segments as differences of lower than 1% occur for all calculated moments. Specifically, PM1's volume is larger by 0.116%, x- coordinate of volume centroid underestimates by 0.069% while z-coordinate overestimates by 0.354%, $\delta I_x = -0.460\%$, $\delta I_y = 0.104\%$, $\delta I_z = 0.013\%$. For PM2, volume is larger by 0.150%, x- and z- coordinates of volume centroid are equal to -0.053% and 0.253% , $\delta I_x = -0.719\%$, $\delta I_y = -0.134\%$, and $\delta I_z = -0.340\%$.

Table 5.3: Comparison of moments among KCS parent ship-hull, PM1 and PM2 models (full model).

| Models | Volume (m³) | Volume Centroid (m) | Inertia (m⁵) |
|---|-----------------------------------|------------------------------------|------------------------------------|
| Parent Ship | 52,053 | x: 118.407 | Ix: 3.892·10 ⁶ |
| | | y: 0 | Iy: 1.304·10 ⁸ |
| | | z: -4.892 | Iz: 1.333·10 ⁸ |
| PM1 | 52,129 | x: 118.332 | Ix: 3.901·10 ⁶ |
| | | y: 0 | Iy: 1.310·10 ⁸ |
| | | z: -4.899 | Iz: 1.339·10 ⁸ |
| PM2 | 52,015 | x: 118.290 | Ix: 3.843·10 ⁶ |
| | | y: 0 | Iy: 1.294·10 ⁸ |
| | | z: -4.896 | Iz: 1.323·10 ⁸ |
| Difference (%) PM1 vs Parent Ship | -0.146 | x: 0.099 | Ix: 0.026 |
| | | y: 0 | Iy: -0.458 |
| | | z: -0.143 | Iz: -0.448 |
| Difference (%) PM2 vs Parent Ship | 0.073 | x: 0.038 | Ix: 1.288 |
| | | y: 0 | Iy: 0.756 |
| | | z: -0.001 | Iz: 0.767 |

Notes: Inertia has been calculated with respect to Centroid Coordinate Axis. Negative percentage indicates that parent ship-hull's value is lower than that of PM1/2.

Table 5.4: Comparison of moments among KCS parent ship-hull, PM1 and PM2 models.

(Segmentation at $x^1 = 104.7\text{m}$, and $x^2 = 151.7\text{m}$)

| Models | Volume (m³) | Volume Centroid (m) | Inertia (m⁵) |
|---------------------|-----------------------------------|------------------------------------|------------------------------------|
| Forward Part | | | |
| Parent Ship | 18,793 | x: 64.175 | Ix: 1.159·10 ⁶ |
| | | y: 0 | Iy: 1.066·10 ⁷ |
| | | z: -4.876 | Iz: 1.149·10 ⁷ |
| | | x: 64.081 | Ix: 1.155·10 ⁶ |

Continued on next page

Table 5.4 – Continued from previous page

| Models | Volume (m ³) | Volume Centroid (m) | | Inertia (m ⁵) |
|--------------------|------------------------------------|----------------------------------|---------|-------------------------------------|
| PM1 | 18,798 | y: | 0 | Iy: 1.073·10 ⁷ |
| | | z: | -4.874 | Iz: 1.155·10 ⁷ |
| | | x: | 64.151 | Ix: 1.160·10 ⁶ |
| PM2 | 18,809 | y: | 0 | Iy: 1.069·10 ⁷ |
| | | z: | -4.877 | Iz: 1.152·10 ⁷ |
| Difference (%) | | x: | 0.9 | Ix: 0.685 |
| PM1 vs | -0.105 | y: | 0 | Iy: -0.133 |
| Parent Ship | | z: | 0.122 | Iz: -0.053 |
| Difference (%) | | x: | 0.025 | Ix: -0.652 |
| PM2 vs | -0.198 | y: | 0 | Iy: -0.192 |
| Parent Ship | | z: | -0.131 | Iz: -0.225 |
| Middle Part | | | | |
| Parent Ship | 16,004 | x: | 121.505 | Ix: 1.486·10 ⁶ |
| | | y: | 0 | Iy: 3.082·10 ⁷ |
| | | z: | -5.305 | Iz: 4.264·10 ⁷ |
| PM1 | 16,046 | x: | 121.499 | Ix: 1.495·10 ⁶ |
| | | y: | 0 | Iy: 3.100·10 ⁶ |
| | | z: | -5.319 | Iz: 4.29·10 ⁶ |
| PM2 | 16,002 | x: | 121.501 | Ix: 1.485·10 ⁶ |
| | | y: | 0 | Iy: 3.081·10 ⁶ |
| | | z: | -5.304 | Iz: 4.263·10 ⁶ |
| Difference (%) | | x: | 0.005 | Ix: -0.602 |
| PM1 vs | -0.261 | y: | 0 | Iy: -0.582 |
| Parent Ship | | z: | -0.263 | Iz: -0.606 |
| Difference (%) | | x: | 0.003 | Ix: 0.0673 |

Continued on next page

Table 5.4 – Continued from previous page

| Models | Volume (m ³) | Volume Centroid | | Inertia (m ⁵) |
|-----------------------|-----------------------------|--------------------|-----------|------------------------------|
| | | | (m) | |
| PM2 vs Parent Ship | 0.012 | y: 0 | z: 0.019 | Iy: 0.032 Iz: 0.023 |
| Afterward Part | | | | |
| Parent Ship | 17,254 | x: 174.602 | | Ix: 1.243·10 ⁶ |
| | | y: 0 | | Iy: 6.709·10 ⁶ |
| | | z: -4.533 | | Iz: 7.636·10 ⁶ |
| PM1 | 17,281 | x: 174.722 | | Ix: 1.245·10 ⁶ |
| | | y: 0 | | Iy: 6.702·10 ⁶ |
| | | z: -4.517 | | Iz: 7.635·10 ⁶ |
| PM2 | 17,280 | x: 174.695 | | Ix: 1.252·10 ⁶ |
| | | y: 0 | | Iy: 6.718·10 ⁶ |
| | | z: -4.544 | | Iz: 7.662·10 ⁶ |
| Difference (%) | | x: -0.069 | | Ix: -0.46 |
| PM1 vs Parent Ship | -0.116 | y: 0 | z: 0.354 | Iy: 0.104 Iz: 0.013 |
| Difference (%) | | x: -0.053 | | Ix: -0.719 |
| PM2 vs Parent Ship | -0.150 | y: 0 | z: -0.253 | Iy: -0.134 Iz: -0.340 |

Notes: Inertia has been calculated with respect to Centroid Coordinate Axis. Negative percentage indicates that parent ship-hull's absolute value is lower than that of PM1/2.

5.5 SAC comparison

Fig. 5.9 illustrates the SACs of PM1 (red curve), PM2 (blue curve) and parent (green curve) ship-hulls. It also shows the difference of SAVs between PM1 and parent hull

(magenta curve), as well as between PM2 and parent (black curve) with the maximum deviation $|SAV_{PM1} - SAV_{parent}|_{max} = 9.5m^2$ occurring at $x = 168m$, and $|SAV_{PM2} - SAV_{parent}|_{max} = 3m^2$ occurring at $x = 59m$, which correspond to 2.8% and 9‰ of the maximum SAV respectively.

It is worth noticing that the red curve is constant along the interval $104.7 \leq x \leq 151.7$ because TshipPM imposes by construction a strictly cylindrical part along the middle part of the hull and, as a result, possesses constant SAV in this interval. This is the reason of the significant deviation occurring at PM1's SAC with respect to the parent ship. The problem can be mitigated by introducing an additional afterward transition curve between the middle part and the first afterward transition curve. Fig. 5.10 shows the improved performance of TshipPM after introducing the extra transition curve, with $|SAV_{PM1} - SAV_{parent}|_{max} = 3.8m^2$ at $x = 163m$, 250% smaller than in the case of 4 transition curves.

5.6 Gaussian-curvature comparison

Figs. 5.3-5.6 depict side, bow, stern and bottom views of the Gaussian curvature plot of the surfaces of PM1 and PM2. A color map capable to easily reveal sign variations of the Gaussian curvature has been selected ($[-10^{-6}, 10^{-6}]$). In general, one can easily spot extraneous Gaussian-curvature sign variations in several areas of both plots. A more focused view, though, reveals that the Gaussian-curvature distribution of PM2 exhibits more extraneous small-scale oscillations between red (convex) and blue (non-convex) areas; see; Figs. 5.8-5.7. This is attributed to the fact that CAESES's model has been constructed using multi-patch NURBS surfaces of high complexity, consisting of many tiny patches, resulting to unevenly spaced isoparametric curves, and thus it is more challenging to achieve fairness in transition regions. For example, in Fig. 5.8 one can observe curvature oscillations at the mid- and bottom-stern which consist of a total of 50 patches. Nevertheless, Fig. 5.6 indicates that, in large scale, the shape of the flat of side/bottom provided by PM2 is better than that of PM1. In this connection it should be noted that TshipPM imposes by construction a strictly cylindrical parallel middle

part, indicated by the green rectangle in Fig. 5.6.

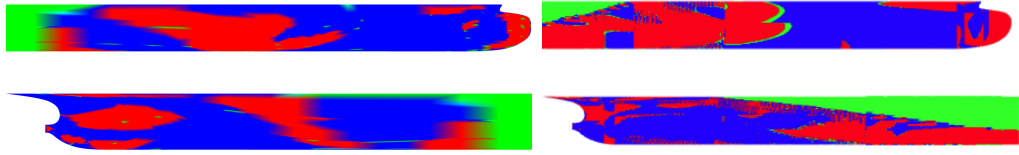


Figure 5.3: Side view of Gaussian-curvature plot of Fwd.Part (top) and Aft.Part (bottom) of PM1 (left) and PM2 (right).

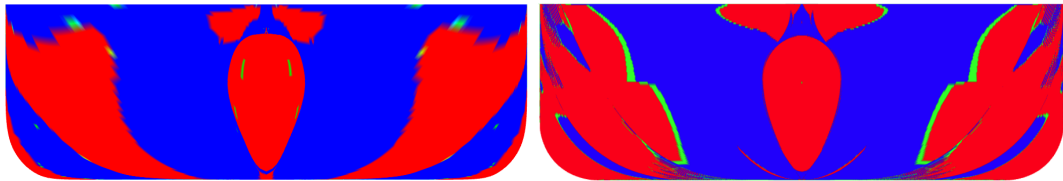


Figure 5.4: Bow view of Gaussian-curvature plot of PM1 (left) and PM2 (right).

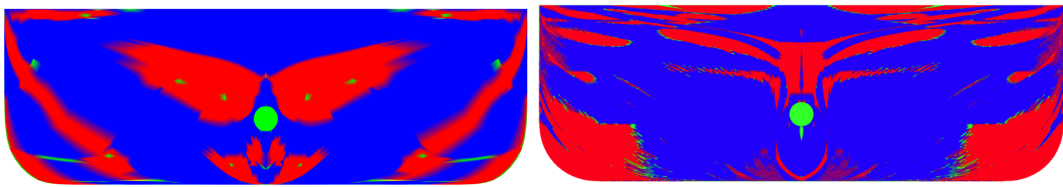


Figure 5.5: Stern view of Gaussian-curvature plot of PM1 (left) and PM2 (right).

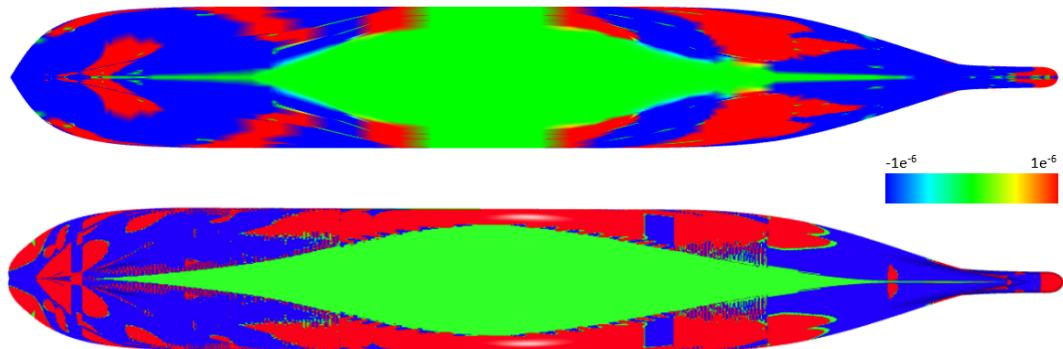


Figure 5.6: Bottom view of Gaussian-curvature plot of PM1 (top) and PM2 (bottom).

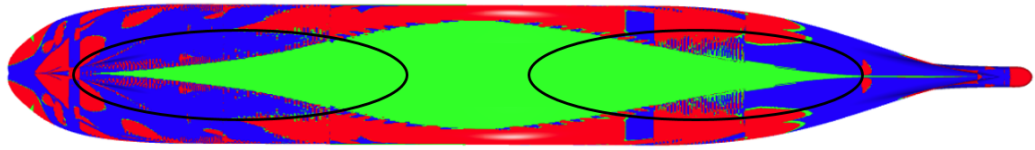


Figure 5.7: Small-scale extraneous oscillations of the Gaussian-curvature distribution of PM2: bottom view.

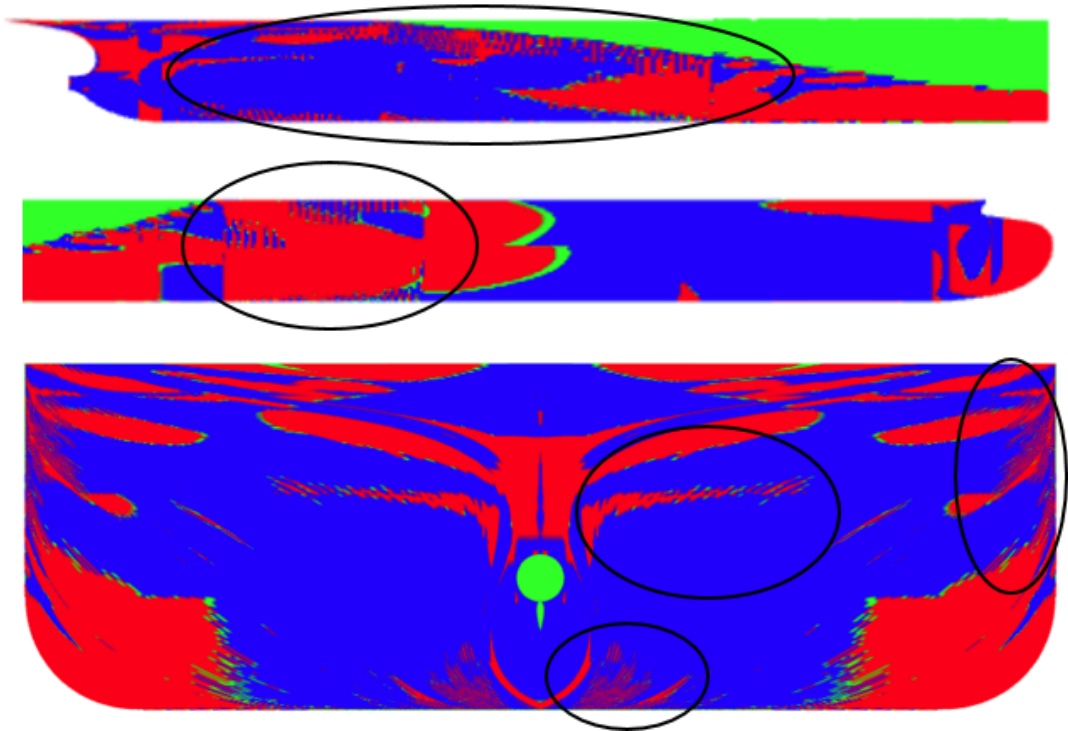


Figure 5.8: Small-scale extraneous oscillations of the Gaussian-curvature distribution of PM2: side (top) and stern view (bottom).

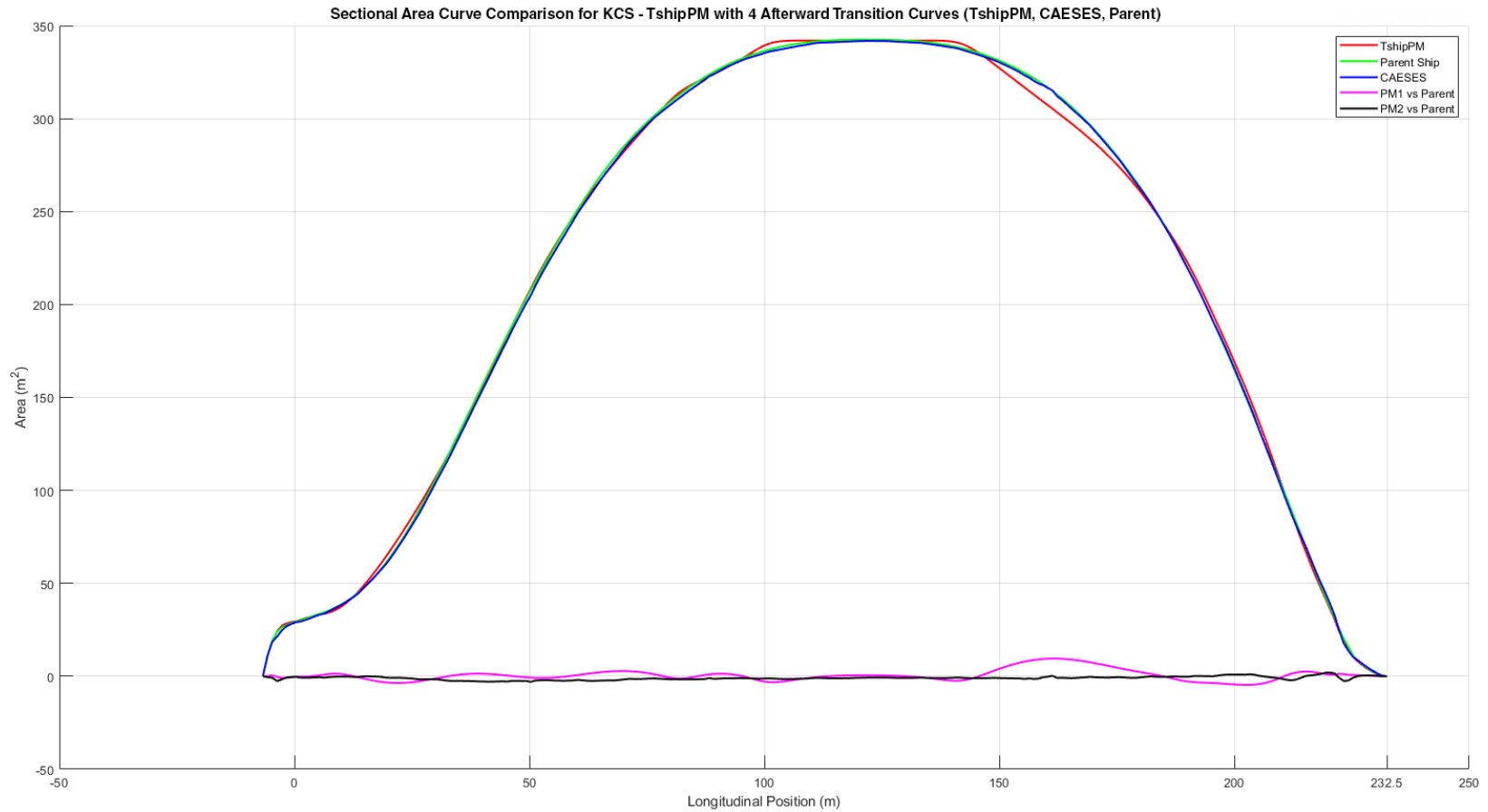


Figure 5.9: SAC comparison among PM1 (red curve), PM2 (blue curve) and the parent hull (green curve): the magenta curve depicts the deviation $SAV_{PM1} - SAV_{PM2}$ with the absolute maximum deviation for PM1 (9.5m) and PM2 (3m) occurring at $x = 168m$ and $x = 59m$ respectively.

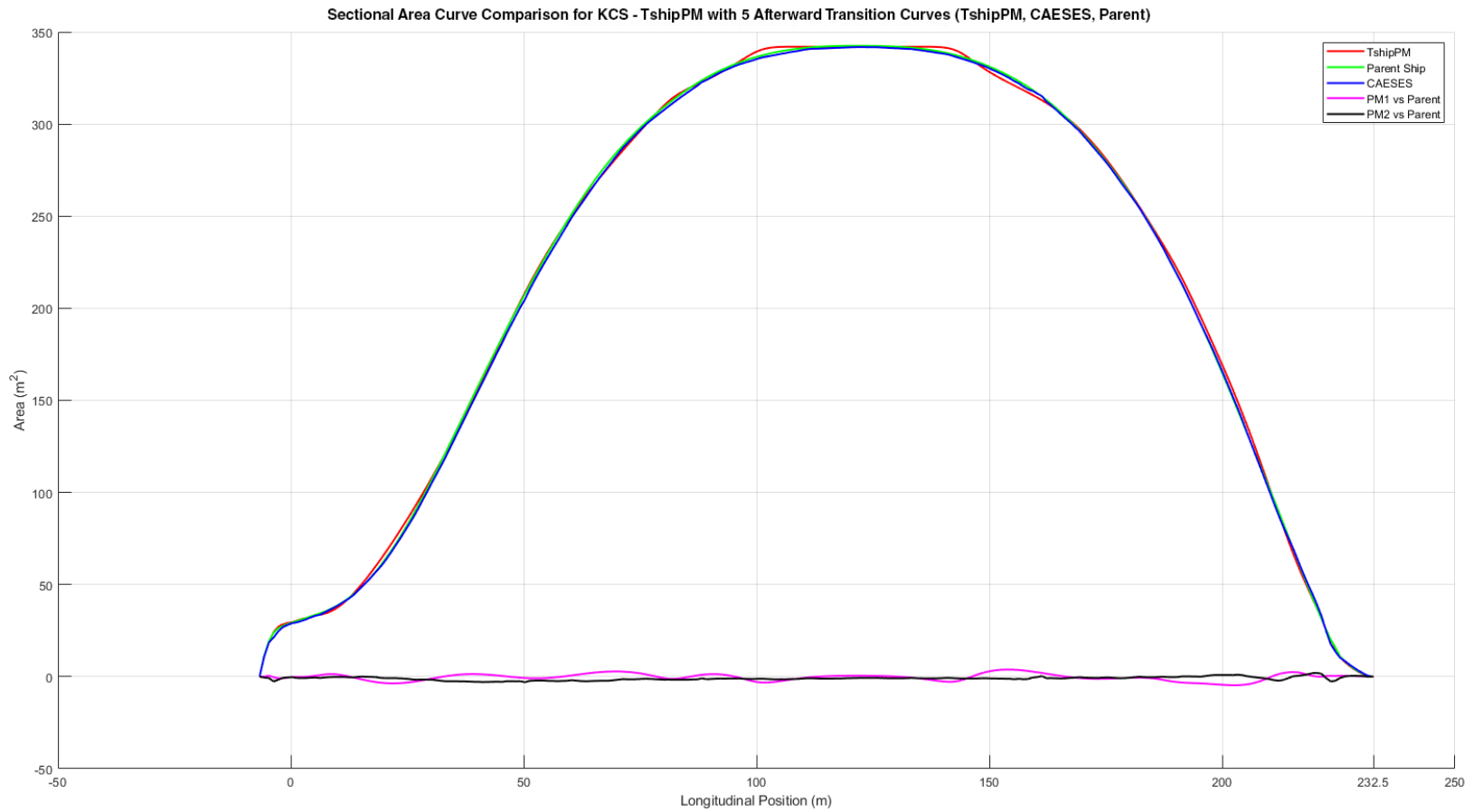


Figure 5.10: Improved SAC for PM1 (red curve) after introducing an additional afterward transition curve between the parallel middle part and the first afterward transition curve. The absolute maximum deviation for PM1 (3.8m) occurs at $x = 163m$.

5.7 Sectional-curvature comparison

By sectional curvature we refer to the curvature distribution of the planar curves obtained by intersecting the ship-hull with planes vertical to its longitudinal axis. When selected at prescribed distances, called *stations*, the resulting family of curves is referred to as the *body plan* of the hull. In design practice, the fairness of these sections is evaluated via their curvature distribution depicted as a function of a parameter running along the curve or in the porcupine format, adopted herein. Figs. 5.11-5.14 indicate that PM1 and PM2 provide, in general, fair sections which are also free from extraneous inflection points, with exception to PM2's sections at $x = 146.3m$ and $x = 205.3m$, as shown in Fig. 5.15.

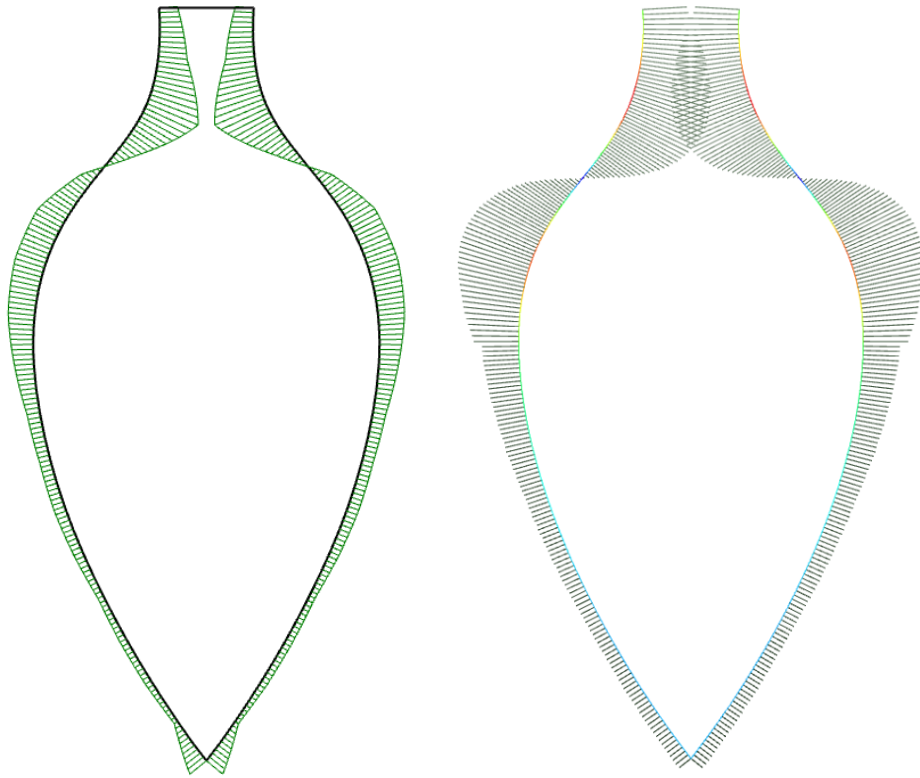


Figure 5.11: PM1 (left), and PM2 (right) porcupine plots of section at $x = -1m$.

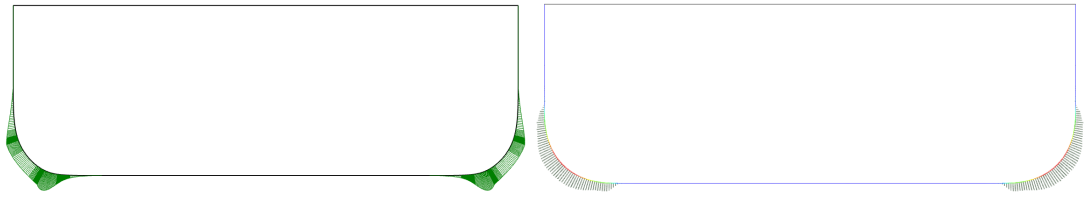


Figure 5.12: PM1 (left), and PM2 (right) porcupine plots of section at $x = 104.7m$.

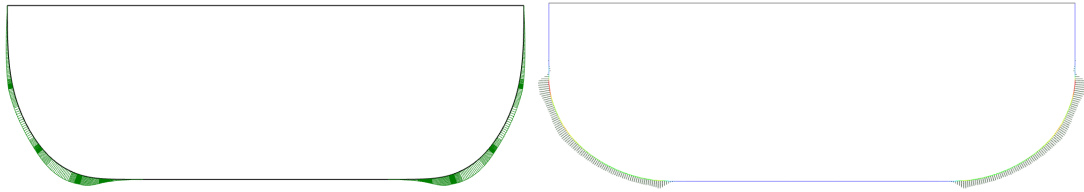


Figure 5.13: PM1 (left), and PM2 (right) porcupine plots of section at $x = 151.7m$.

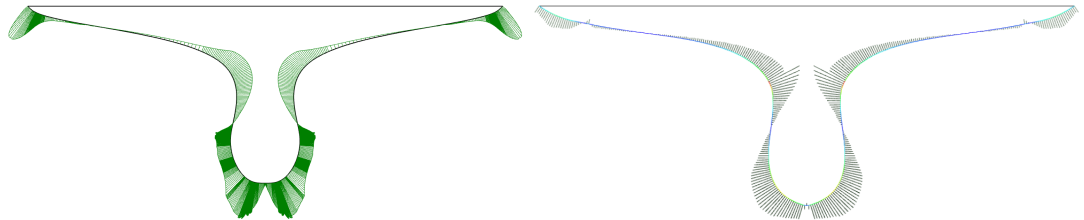


Figure 5.14: PM1 (left), and PM2 (right) porcupine plots of section at $x = 205.3m$.

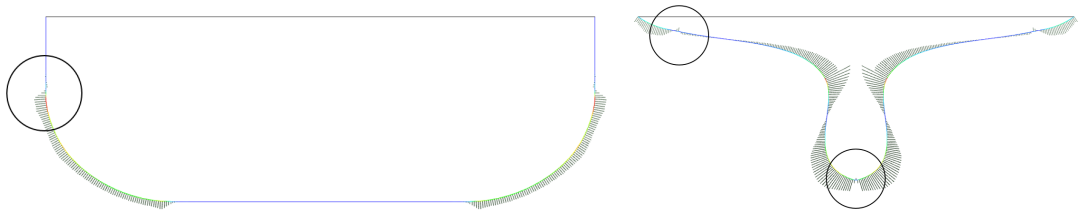


Figure 5.15: Extraneous inflection points of PM2 for sections at $x = 151.7m$ (left) and $x = 205.3m$ (right).

5.8 Hausdorff distance comparison

For this comparison we use the MeshLab[®] ² tool and a dense sampling as in §4.4.3. Figs. 5.16 and 5.17 depict the distribution of deviation $\delta(\mathbf{x})$ of PM1 and PM2 from the

²<http://www.meshlab.net/>

parent hull respectively. The color map for both models ranges over $[0, 0.69\text{m}]$, which covers the maximum deviation measured for PM1 with respect to the parent ship-hull. PM2's maximum deviation is 0.3m, while the mean deviation for PM1 is 0.07m and 0.012m for PM2.

The obtained results indicate a considerably better behaviour for PM2, attributed to the large DoF (276 for PM1 vs $O(10^5)$ for PM2). PM1's significant deviations occur in the transition area from midship towards the afterward part of the hull which is attributed to the fact that PM1 surface changes abruptly from the cylindrical middle body towards, especially, the afterward part; see also Fig. 5.6 (top).

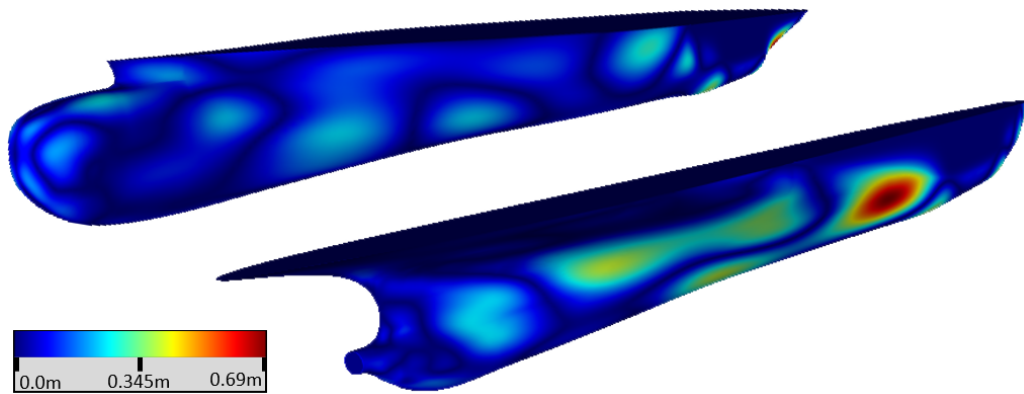


Figure 5.16: Hausdorff distance plot of PM1 with respect to the parent ship-hull. Color map range is $[0, 0.69\text{m}]$

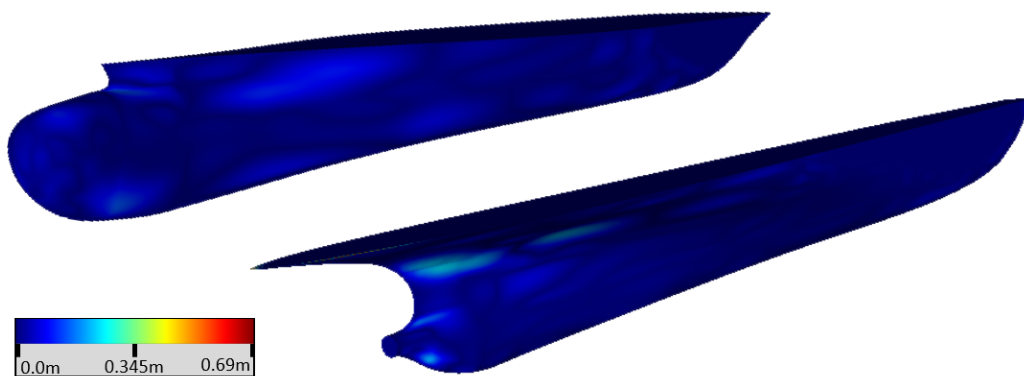


Figure 5.17: Hausdorff distance plot of PM2 with respect to the parent ship-hull. Color map range is $[0, 0.69\text{m}]$

A better performance in the geometric variation of PM1 occurs after introducing an additional afterward transition curve as we did for the SAC comparison. As Fig. 5.18 shows the maximum deviation after the addition of the 5th transition curve is 0.42m, located at the same transition area as before. Fig 5.19 also illustrates the geometric deviation of PM1 with respect to the parent hull, but the range of the color map is as before introducing the extra transition curve, i.e., $[0, 0.69\text{m}]$, and makes obvious the better performance of TShipPM after introducing the additional transition curve at the afterward part.

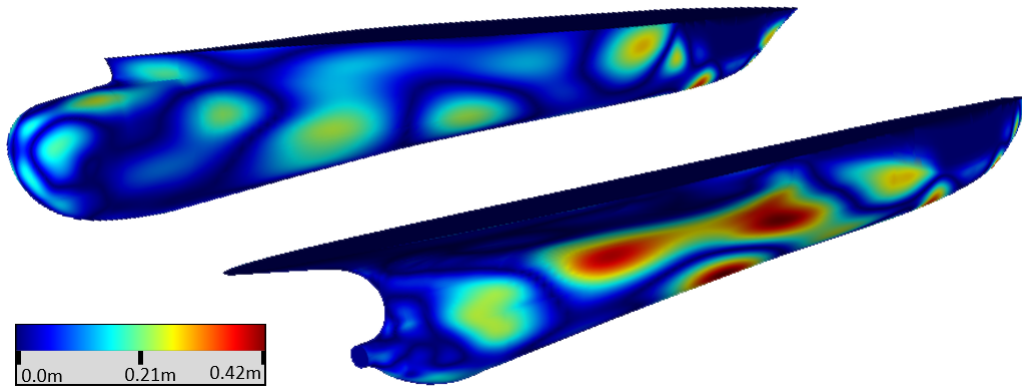


Figure 5.18: Hausdorff distance plot of PM1 with respect to the parent ship-hull. Color map range is $[0, 0.42\text{m}]$.

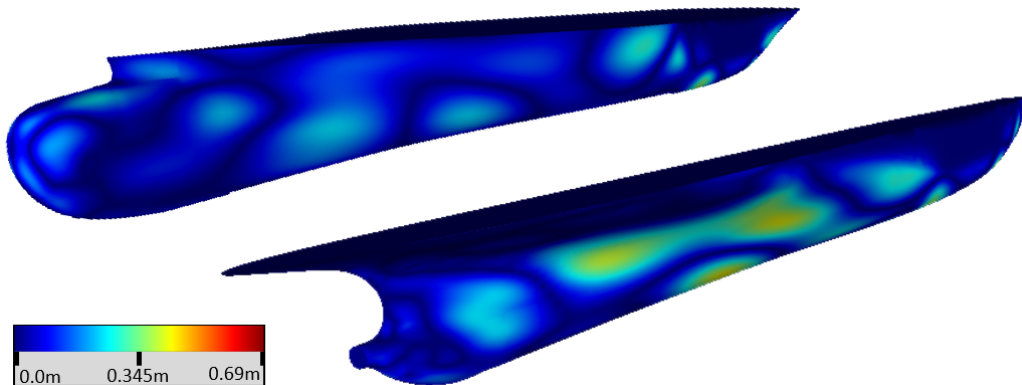


Figure 5.19: Hausdorff distance plot of PM1 with respect to the parent ship-hull. Color map range is $[0, 0.69\text{m}]$

Chapter 6

A shape-optimisation tool integrating a MO-TLBO algorithm and TshipPM

6.1 Introduction

In this section we present a shape-optimisation tool based on ship-design criteria. Specifically, it uses up to 2nd order moments, i.e., volume, volume centroid, and moments of inertia with respect to volume centroid (VC), as *objective functions* (or *cost functions* or *costs*) to minimise, aiming to create ship-hulls with geometries in the proximity of a parent hull. It combines both TshipPM and a new, in-house version of the recently introduced *Teaching-Learning-based Optimisation* (TLBO) meta-heuristic optimisation method. The tool has been built in C# and its method has been adjusted accordingly, to serve the needs of the current work. The goal of the work presented in the ensuing sections is to introduce the meta-heuristic algorithms in parametric modelling for ship design, integrate them with TshipPM, and create a CAD parametric modelling tool which will constitute the initial step towards developing automated tools for remodelling container and tanker ship-hulls, but not limited to them.

6.2 Introduction to TLBO

Throughout the history of engineering advancements, following the introduction of computing devices and digital design, intelligent optimisation algorithms have been used in excess to create high-end products and services, increasing dramatically the cost-effectiveness in the field of manufacturing technologies. Meta-heuristic algorithms, either *evolutionary*, such as *genetic algorithms* (GA) [54]–[58], *genetic programming* (GP) [59]–[62], *evolutionary programming* (EP) [63]–[65], *differential evolution* (DE) [66]–[68], etc., or *swarm intelligence*, such as *stochastic diffusion search* (SDS) [69], [70], *ant colony optimisation* (ACO) [71], [72], *particle swarm optimisation* (PSO) [73] and *artificial swarm intelligence* (ASI) [74], have significant contribution towards the rise of a new digital industrial technology, known as *Industry 4.0*.

Researchers and practitioners who are involved in a variety of complex engineering optimisation problems in real-world settings are using all kind of optimisation techniques to reach the best possible outcomes. Although all of the above population-based optimisation methods are potentially good candidates for shape optimisation, there is a necessity of either enhancing the existing algorithms or introducing innovative ones, since the existing ones are subject to limitations [75]. A major limitation is the requirement of assigning values to several algorithm-specific parameters. The proper parameter tuning is not trivial, it depends on the characteristics and the nature of each problem and invalid assignment can lead to algorithmic ineffectiveness, ending up to incapability of the algorithm to identify the optimum solution. TLBO, introduced in 2011 by R.V. Rao et al. [76], overcomes this issue and solves both single and multi-objective continuous and discrete optimisation problems. It is based on a teaching-learning principle, where the best solution of the population, considered as the *teacher*, is teaching the *learners*.

In short, TLBO uses only common control parameters such as population size (PS), number of generations or stopping criteria based on the objective function(s) and has two main phases, namely *Teaching* (TP) and *Learning* (LP). During the teaching phase, the goal is to increase the mean result of the class (learners) by a teacher. As a teacher is assigned the learner with the best result so far, i.e., the solution of which the objective

function value is closer to the stopping criteria. The best solution (teacher) and the mean result of the class are then used to provide the final best solution. As far as the learning phase is concerned, the learners increase their knowledge among themselves by interacting with each other, where a learner learns from another if and only if it is more knowledgeable. Time-efficiency in TLBO method depends mainly on the *dimensionality* of the problem and the *number of the objective functions*.

TLBO comes in two basic versions: a) the Elitist TLBO algorithm (ELTBO) which utilises the concept of elitism where the worst solutions are replaced by the best (elite) solutions, and b) the non-Dominated Sorting TLBO algorithm (NSTLBO) which is used for multi-objective problems, where no single solution exists that optimises the cost function, but a set of optimal Pareto solutions instead. The latter is an *a posteriori* approach, which generates all the Pareto optimum solutions and the designer selects one of them. On the other hand, an *a priori* approach requires a set of preferences assigned by the designer in the form of weights assigned to the cost functions. Finally, TLBO incorporates the concept of *crowding distance* as a technique for obtaining the best solutions [58].

TLBO has been already used in many engineering fields, for both constrained and unconstrained single or multi-objective (MO-TLBO) problems. A detailed and easy to follow presentation of the TLBO algorithm can be found in R. Venkata Rao's book, published by Springer (London, UK), under the title " *Teaching Learning Based Optimization Algorithm And Its Engineering Applications*" [77].

6.3 TshipPM MO-TLBO: objective functions

TshipPM MO-TLBO has been built in C# with a multi-objective approach, using in total 18 cost functions. The ship-hull is segmented in 3 main parts, i.e. a) parallel middle part, b) forward part including bow, and c) afterward part including stern, as in Fig. 4.21, but with the bow stretching towards the right-hand side, with 6 costs of same nature correlated to each of them. The 6 costs correspond to a set of ship-design criteria, namely the 0th, 1st, and 2nd order volume moments. Specifically, the costs are

the volume (V), the x- and z-coordinates (y-coordinate is zero) of the volume centroid (VC), and the x-, y-, and z- coordinates of the moments of inertia (MoI) with respect to the volume centroid of the KCS ship-hull, and are defined by the following equations:

$$V = \int_V dx dy dz \quad (6.1)$$

$$VC_x = x_0 = \frac{M_x}{V}, \quad VC_y = y_0 = \frac{M_y}{V}, \quad VC_z = z_0 = \frac{M_z}{V} \quad (6.2)$$

$$I_x = M_{yy} + M_{zz} - 2(y_0 \cdot M_y + z_0 \cdot M_z) + (y_0^2 + z_0^2) \cdot V \quad (6.3)$$

$$I_y = M_{xx} + M_{zz} - 2(x_0 \cdot M_x + z_0 \cdot M_z) + (x_0^2 + z_0^2) \cdot V \quad (6.4)$$

$$I_z = M_{xx} + M_{yy} - 2(x_0 \cdot M_x + y_0 \cdot M_y) + (x_0^2 + y_0^2) \cdot V \quad (6.5)$$

where M_x , M_y , M_z are the first volume moments, given by:

$$M_x = \int_V x dV, \quad M_y = \int_V y dV, \quad M_z = \int_V z dV \quad (6.6)$$

V is the volume, and M_{xx} , M_{yy} , and M_{zz} are the second volume moments, defined by the equations:

$$M_{xx} = \int_V x^2 dV, \quad M_{yy} = \int_V y^2 dV, \quad M_{zz} = \int_V z^2 dV \quad (6.7)$$

In brief, the tool calculates the 18 corresponding values for the parent-hull, 6 for each segment, i.e., V^P , VC_x^P , VC_z^P , I_x^P , I_y^P , I_z^P , where the superscript P denotes the parent ship-hull. TshipPM MO-TLBO generates a set of instances (solutions) by providing random values to the 24 non-dimensional parameters in the range [0.01,0.99] (Lwl, T, and B are constant), using the .NET *Random* class. Subsequently, it calculates the corresponding to each model V, VC, and MoI values (V^M , VC_x^M , VC_z^M , I_x^M , I_y^M , I_z^M ,

6 costs for each part, M denotes the model). Having already set the stopping criteria, at the end of each iterative process, a set of optimum solution $\{\mathbf{P} = [p_0, p_1, \dots, p_{23}]^T\}$ is returned, where $p_1 \dots p_{23}$ are the external, non-dimensional parameters of TshipPM. The objective functions for each part to minimise are:

$$\left| \frac{V^P}{V^M} - 1 \right|, \quad (6.8)$$

$$\left| \frac{VC_x^P}{VC_x^M} - 1 \right|, \left| \frac{VC_z^P}{VC_z^M} - 1 \right|, \quad (6.9)$$

$$\left| \frac{I_x^P}{I_x^M} - 1 \right|, \left| \frac{I_y^P}{I_y^M} - 1 \right|, \left| \frac{I_z^P}{I_z^M} - 1 \right| \quad (6.10)$$

and they should be upper bounded by the stopping criteria value SC. The goal at the end of the process is to create a set of ship-hull models possessing V, VC, and MoI values in the proximity of a parent ship-hull.

6.4 TshipPM MO-TLBO: method and features

In this section we elaborate on the characteristics and functionality of the TshipPM MO-TLBO tool. The main characteristic of the tool is that it optimises the three main parts of the ship-hull separately. Although the total number of the objective functions to minimise is 18, the tool is developed to handle the three distinct main ship-hull parts separately and in a specific order. First, the parallel middle part is optimised, then the forward and last the afterward part, while the latter two are dependent on the middle part. This provides a significant advantage to TshipPM MO-TLBO, as each time only 6 costs are required to be minimised, but at the end of the algorithmic process all 18 have been minimised. Figure 6.1 illustrates the approach adopted: We have broken down a global optimisation problem of 24 variables and 18 costs to:

1. Middle part optimisation: number of Variables = 3, number of costs = 6;
2. Forward part optimisation: number of Variables = 11, number of costs = 6;

3. Afterward part optimisation: number of Variables = 10, number of costs = 6;

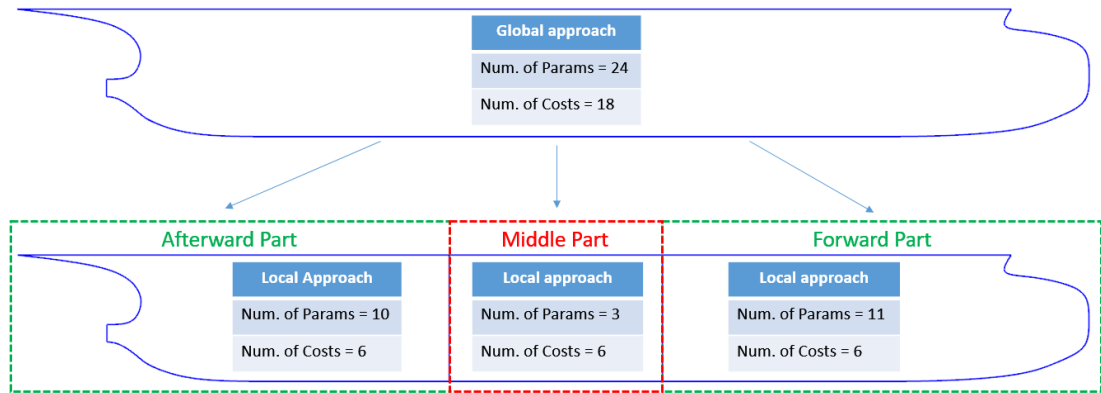


Figure 6.1: Costs distinction per segment. Breaking down the problem for a more time-efficient approach. Parameters $\widehat{FoS_Trans}$ and $\widehat{Fob_Trans}$ are calculated only for the forward part, and their value is constant for the afterward part.

According to our numerical experience, a global approach where the 18 costs were to be minimised simultaneously, in the best case scenario would result in a dramatical increase of the calculation time, while in the worst case some of the stopping criteria would never be satisfied. The above-mentioned breakdown is attributed to the method utilised to develop TshipPM:

- The parallel middle part uses 3 parameters, Mid_Pos , Mid_L , and $BilgeR$. The first two have a more global nature, as their values are used to define the parameters of the forward and afterward parts. As soon as the costs of the parallel middle part have been minimised, parameters $\widehat{Mid_Pos}$, $\widehat{Mid_L}$, and \widehat{BilgeR} have been obtained, and the algorithm proceeds to minimise the costs of the other two parts.
- The forward and afterward parts are using 11 and 10 parameters respectively, and are not interdependent, hence the parameters of the forward part do not interfere in the afterward part and vice versa. Parameters and their corresponding ship hull parts are described in Table 2.1. Parameters $\widehat{FoS_Trans}$ and $\widehat{Fob_Trans}$ are calculated only for the forward part, and their value is constant for the afterward part.

It is of crucial importance to start and finish the optimisation of the middle part as it depends on $\widehat{Mid_Pos}$ and $\widehat{Mid_L}$, two parameters that affect both the forward and the afterward parts. As soon as the middle part is optimised according to the stopping criteria, the algorithm progresses to the optimisation of the remaining segmented parts.

A second characteristic of the tool is that it calls TshipPM to create only the part of the hull under optimisation, including always the middle part. Normally, a PM tool draws the lines for the control cage of the whole hull, for the final smooth model to be produced. In this case, TshipPM is called to draw only the lines of the control cage for the part under optimisation alongside with the middle part, neglecting the third part. When the middle part is being optimised it is drawn alone. Assuming the middle part is already optimised, TshipPM is called to draw in the CAD GUI only the lines of the middle and the forward part, neglecting the afterward part. When the afterward part is being optimised, the control cage lines of the forward part are neglected. Note that, time-consumption for drawing the lines as well as for the construction of the final smooth surface increases with a raise in the number of lines required to be drawn in the CAD tool. Therefore, this characteristic increases the time-efficiency of the tool by more than 200%.

The tool uses 6 inputs:

- a. Initial population size: $IPS = 2,000$. The size of the initial population sampled within the design space of TshipPM. A high value is used to sample instances covering an area of the design space as broad as possible. IPS size is an important parameter of the MO-TLBO algorithm, as the lower the value the less area of the design space of the PM is scanned, making it almost impossible to find an optimum solution, while much higher values increase time consumption. The optimum choice of IPS value depends on the problem. Note that, initial population is only sampled once per ship-hull part, therefore only 3 times in the whole process;
- b. Population size: $PS = 200$. The size of the population used in teaching and learning phases. Low values is most probable to limit the diversity. Both populations

are created using *Monte Carlo* sampling;

- c. Number of variables: $NoV = 24$. All non-dimensional parameters of TshipPM. Global parameters Lwl , T , and B remain constant.
- d. Number of costs: $NoC = 6$. Volume, volume centroid and moments of inertia, i.e., V , VC_x , VC_z , I_x , I_y , I_z ;
- e. Stopping criteria: $SC = 0.02$. A deviation of less than 2% is desirable;
- f. Rhino's absolute tolerance: $AT = 10^{-6}$;

Although in optimisation algorithms the rule of thumb is to use a PS ten times larger than the dimensionality of the problem to improve diversity, i.e., $PS = 10 \cdot NoV$, the final choice depends upon the individuals and the nature of the problem. After experimentation we opted for a high IPS to sample instances covering as much of the design space of TshipPM as possible, and a lower PS to use in teaching and learning phases to increase the time efficiency of the tool. Lower values of IPS are making impossible to scan a broad area of TshipPM's design space, and, subsequently, the algorithm might stick in local optima which do not satisfy the stopping criteria.

The main body of the algorithm consists of the following 5 basic steps, each of which is a separate method in the algorithm, while its flowchart is illustrated in Fig. 6.2.

1. Create initial population: Creates the *initial population* (IP), comprised of 2,000 solutions, by randomly assigning values in the range $[0.01,0.99]$ for all 24 non-dimensional parameters, except for the following:
 - a. $\widehat{Mid_Pos}$: $[0.533,0.99]$;
 - b. $\widehat{Mid_L}$:
 - if $(Mid_Pos + Mid_L / 2 > 4 \cdot Lwl / 5)$ then Mid_L takes values in the range $[0.01,0.68]$
 - if $(Mid_Pos - Mid_L / 2 < Lwl / 5)$ then Mid_L takes values in the range $[0.01,0.57]$.

This choice of the parameter values range of `Mid_Pos` and `Mid_L` has been made to to secure designs with the middle part close to the longitudinal middle of the ship. A middle part which is located in the proximity of the bow or the stern would be impractical, and it would result to increase IPS and PS as the design space to explore would be larger. Each solution consists of 24 different parameter values, same as the number TshipPM uses to provide the hull geometry, assuming that the global parameters `Lwl`, `T`, and `B` are constant.

2. TshipPM execution: We execute TshipPM to build the initial population of 2,000 ship-hull instances, calculating, as well, the 6 costs for each model, using Rhino's *VolumeMoments* command. The execution is faster as we disable redrawing of the instances, and we convert the T-splines surface into a *mesh* object using Rhino's command *Mesh*, opting for the default options. The geometric variation occurring by the transformation from a T-splines to a mesh object is negligible (approximately 1‰).
3. Sorting: Making use of the non-dominated sorting functionality, we determine the Pareto fronts (each front contains a set of solutions sorted according to the non-dominance rank) for the initial population. Solutions of which the value of at least one cost is not dominated by any other solution's cost-value (for the same cost type) are forming the first Pareto front (rank = 0). The rest of the fronts have a rank higher than zero, hence they are appended in ascending order of rank. Our goal is to end up with the best 200 solutions (PS=200) and feed them into the teaching and learning phases, which are responsible to teach the learners and find the optimum solution(s) respectively. The desired-best solutions are contained in the first Pareto front, which during the teaching-learning iterative process, but especially after calculating the costs of the initial population (IPS>PS), it might contain more than 200 solutions. Therefore, we sort in ascending order the costs of the first front regarding to their *mean value*, i.e., the mean value of all 6 costs of each solution. This process ascertains that an updated population (sorted population) of a maximum size of 200 is acquired.

4. Teaching phase (TP): Sorted population is fed into the TP. TP uses a random solution from the first front (where best solutions are stored) of the sorted population to modify it and create a new one, based on the best solution (or teacher), the teaching phase population (TP population). The reason behind using a random solution instead of the best mean solution (or any other type of a “best” solution) is to prevent the optimisation of being stuck in a local optimum which is not satisfying the stopping criteria. Here we have opted for a deeper scanning of the design space than for a faster but not robust process. On top of this, the 40 bottom-of-the-list solutions, i.e. the 20% of TP population with the highest mean values, are replaced by 40 random solutions to increase the diversity and sample more solutions out of the design space. TP population is then fed into TshipPM to build the new instances and calculate the corresponding costs. Both sorted and TP populations are then fed back to the sorting process, to find the new first front and return the updated -after sorting- population (TP’ population), which contains the 200 best solutions of the TP.

5. Learning phase (LP): TP’ population is fed into the LP. LP creates a new population (LP population) based on the best mean cost of TP’ population. Both populations are fed into the sorting process again, as in teaching phase, resulting to the final (for this iteration) LP’ population, which includes -among others- the best costs so far at the top of the list. The concept of crowding distance is not utilised in this version of the optimisation algorithm, as we opted for the *mean-value* approach.

The iteration process terminates when at least one solution of the LP’ population satisfies the stopping criteria. As soon as the middle part is optimised, an initial population of IPS is sampled for the optimisation of the forward part. The afterward part is optimised last and only when the optimisation process of the forward part has been completed. At the end of the whole process, the algorithm provides one or more optimised models, as the optimum solutions are usually more than one for each part.

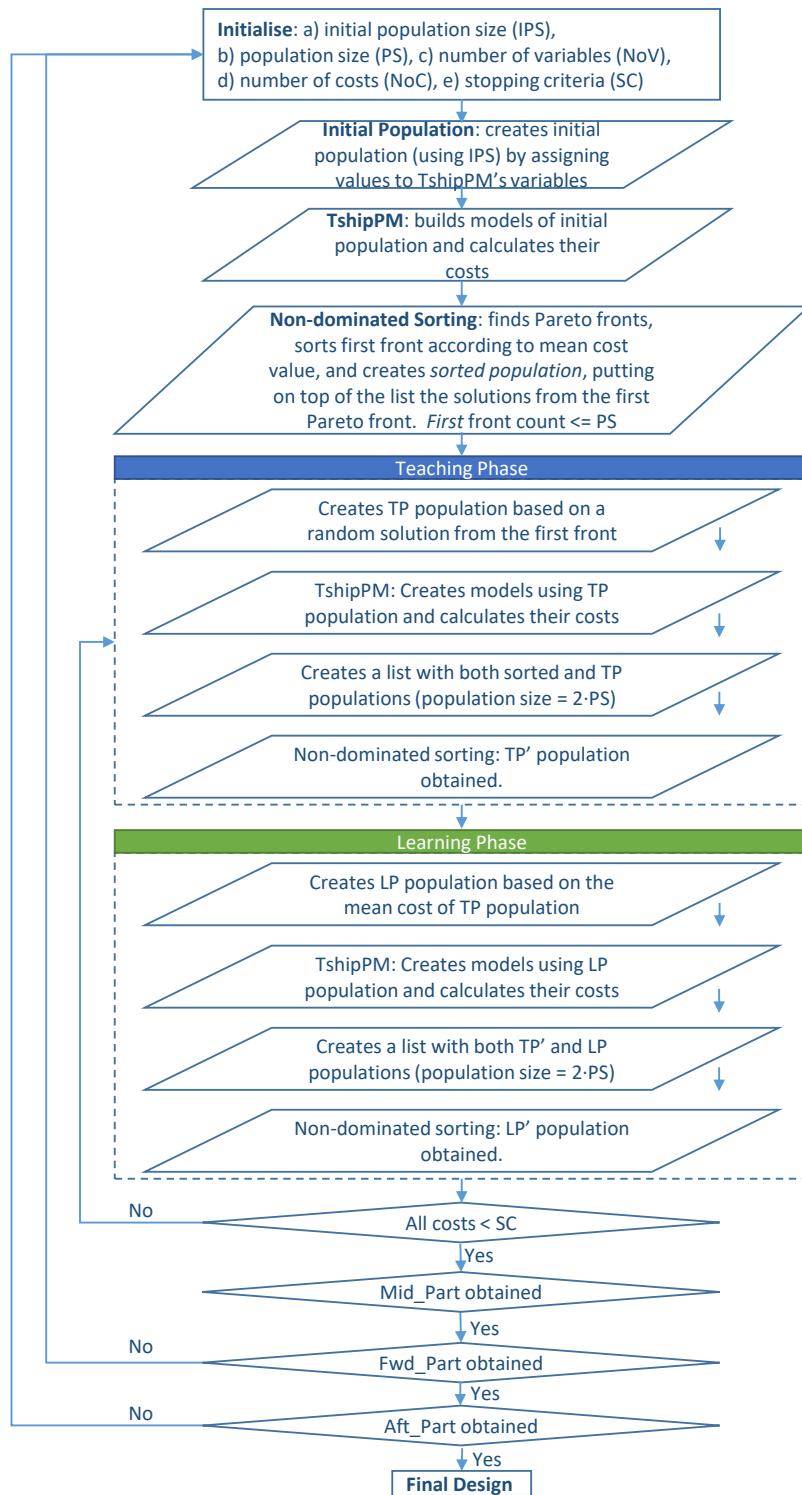


Figure 6.2: TshipPM MO-TLBO flowchart for shape optimisation using as cost functions ship-design criteria, i.e., volume moments up to 2nd order, namely volume, volume centroid, and moments of inertia with respect to volume centroid.

6.5 Results

In this section we present the results of TshipPM MO-TLBO with regards to the convergence of the produced models with respect to the parent ship-hull. The results are based in 10 randomly selected models. To evaluate the results the following criteria are used:

- Objective Function Values (OFV) for each segmented part separately;
- Comparison of V, VC and MoI for the whole hull. Stopping criteria = 0.02, therefore the results of this comparison should be lower than 0.02;
- Visual evaluation of the whole hull with the aid of Rhino's interface;
- Hausdorff distance metric for identifying the geometric variation of the parallel middle part using MeshLab[®];

Each of the models presented has been derived from a different execution of the tool, therefore none of them belongs to the same optimal Pareto with another. At this point we should remind that, usually, in the final optimal Pareto front there is contained a set of optimal solutions and not a single one. In addition, we provide the run time of each of the executions, as well as the mean run time of all executions. The tool was executed on a regular desktop PC with Windows[®]10 on an *Intel(R) Core(TM) i7-6700 CPU 3.40GHz*, with 16GB of RAM, and a display adapter with a *Nvidia Quadro P400* chipset.

Table 6.1 illustrates the OFV for the whole hull but also for each part separately for the 10 models under evaluation, which are illustrated in Fig. 6.3. The time needed for optimising and producing each model is shown in the same table as well. Run time isn't consistent, as meta-heuristic algorithms are sampling random solutions of the design space, and the number of iterations can vary significantly. The mean run time for optimising the KCS container ship-hull using the TshipPM MO-TLBO tool is 155.8 minutes.

One can observe that, although the costs for all models have been minimised and the volume moments for the whole hull show sufficient behaviour and they do satisfy the stopping criteria, the majority of the models' forward and afterward parts are far from lying in the proximity of the parent hull's geometry, with exceptions model 5 and models' 10 forward part, which show a better behaviour, not lying in the proximity of the parent though. Especially, one can observe that the afterward is the most cumbersome part to approximate, particularly at the stern due to its high complexity, attributed to the topology of the control cage. The forward part shows a better behaviour in general. Focusing on the bow, we can observe that the results are better than those at the stern because of its reduced topological complexity in comparison with the stern. Nevertheless, the complexity of the bow and the whole forward part is still high, with an increased local dimensionality (11 parameters). We could conclude that using volume moments up to 2nd order is not sufficient to optimise the shape of complex shapes such as bow and stern of ship-hulls with respect to a parent hull.

On the other hand, as Fig. 6.4 illustrates, the parallel middle part shows a much better behaviour. The maximum geometric variation is observed in Model_5 with max and mean values of 1.26m and 0.26m respectively. Model_8 demonstrates the minimum-max geometric variation (0.36m) with a mean value 0.08. Model_2 has the lowest mean variation (0.07m), and a max value of 0.44m. Shape-wise, middle part is almost identical in all models, attributed to its decreased complexity, i.e., low dimensionality with only 3 parameters and a simpler compared to the other parts control cage. The parallel middle part needs approximately 20-40 minutes to be optimised, the upper bound indicating the time-consumption for stricter stopping criteria. It should be reminded at this point that TshipPM uses a cylindrical middle part, while the parent hull does not, therefore it is expected to have small variations with respect to geometry at this part. Note that, as far as the orientation of the ship-hull is concerned, the forward part of the ship-hull is stretched towards the right-hand side of the middle part.

The simplicity of the parallel middle part allow us to impose stricter stopping criteria, but only for this segmented part. The complexity of the rest demands a substantial

increase to the PS and IPS of the algorithm, therefore a dramatical increase in the run time, otherwise the optimisation is cumbersome to achieve. Assuming $SC = 0.002$, TshipPM MO-TLBO produces always models with a middle part lying in the proximity of Model_11 of Fig. 6.5, with max and mean deviation with respect to the parent ship-hull of approximately 0.30m and 0.05m respectively. The expense in time-efficiency is no more than several minutes.

Table 6.1: TshipPM MO-TLBO results.

| Model | OFV^{VC} (%) | OFV^{MoI} (%) | Volume (%) | Run time (min) |
|--------------|--------------------------------|---------------------------------|----------------------|--------------------------|
| Model_1 | Whole ship-hull | | | |
| | VCx: 0.33 | Ix: 1.31 | 0.30 | 117 |
| | VCz: 0.08 | Iy: 0.46 | | |
| | | Iz: 0.49 | | |
| | Forward part | | | |
| | VCx: 0.03 | Ix: 1.34 | 0.84 | |
| | VCz: 0.63 | Iy: 1.34 | | |
| | | Iz: 1.51 | | |
| | Middle part | | | |
| | VCx: 0.24 | Ix: 0.98 | 0.84 | |
| | VCz: 0.08 | Iy: 1.28 | | |
| | | Iz: 1.22 | | |
| | Afterward part | | | |
| | VCx: 0.16 | Ix: 1.09 | 1.10 | |
| | VCz: 0.89 | Iy: 1.80 | | |
| | | Iz: 1.54 | | |
| Model_2 | Whole ship-hull | | | |
| | VCx: 0.84 | Ix: 0.46 | 0.08 | 57 |
| | VCz: 0.33 | Iy: 0.22 | | |

Continued on next page

Table 6.1 – *Continued from previous page*

| Model | OFV^{VC} (%) | OFV^{MoI} (%) | Volume (%) | Run time (min) |
|--------------|--------------------------------|---------------------------------|----------------------|--------------------------|
| | | Iz: 0.24 | | |
| | | Forward part | | |
| | VCx: 0.78 | Ix: 0.55 | 1.53 | |
| | VCz: 1.32 | Iy: 0.21 | | |
| | | Iz: 0.40 | | |
| | | Middle part | | |
| | VCx: 0.02 | Ix: 0.95 | 0.81 | |
| | VCz: 0.03 | Iy: 1.59 | | |
| | | Iz: 1.42 | | |
| | | Afterward part | | |
| | VCx: 0.48 | Ix: 0.74 | 1.17 | |
| | VCz: 0.22 | Iy: 1.87 | | |
| | | Iz: 1.82 | | |
| Model_3 | | Whole ship-hull | | |
| | VCx: 0.48 | Ix: 1.99 | 0.06 | 78 |
| | VCz: 1.10 | Iy: 0.32 | | |
| | | Iz: 0.33 | | |
| | | Forward part | | |
| | VCx: 0.71 | Ix: 0.34 | 0.02 | |
| | VCz: 0.68 | Iy: 0.14 | | |
| | | Iz: 0.02 | | |
| | | Middle part | | |
| | VCx: 0.49 | Ix: 0.10 | 0.32 | |
| | VCz: 0.14 | Iy: 0.51 | | |
| | | Iz: 0.40 | | |
| | | Afterward part | | |

Continued on next page

Table 6.1 – *Continued from previous page*

| Model | OFV^{VC} (%) | OFV^{MoI} (%) | Volume (%) | Run time (min) |
|----------------|--------------------------------|---------------------------------|----------------------|--------------------------|
| | VCx: 0.49 | Ix: 0.87 | 0.46 | |
| | VCz: 0.36 | Iy: 1.32 | | |
| | | Iz: 1.64 | | |
| Model_4 | Whole ship-hull | | | |
| | VCx: 0.23 | Ix: 0.19 | 0.49 | 170 |
| | VCz: 1.14 | Iy: 0.59 | | |
| | | Iz: 0.58 | | |
| | Forward part | | | |
| | VCx: 0.28 | Ix: 1.38 | 1.09 | |
| | VCz: 1.91 | Iy: 0.80 | | |
| | | Iz: 0.43 | | |
| | Middle part | | | |
| | VCx: 0.01 | Ix: 1.81 | 0.07 | |
| | VCz: 0.96 | Iy: 1.70 | | |
| | | Iz: 0.69 | | |
| | Afterward part | | | |
| | VCx: 0.10 | Ix: 0.16 | 0.37 | |
| | VCz: 0.56 | Iy: 1.07 | | |
| | | Iz: 0.82 | | |
| Model_5 | Whole ship-hull | | | |
| | VCx: 1.23 | Ix: 0.46 | 0.24 | 92 |
| | VCz: 0.53 | Iy: 1.4 | | |
| | | Iz: 1.36 | | |
| | Forward part | | | |
| | VCx: 1.62 | Ix: 0.54 | 0.04 | |
| | VCz: 1.74 | Iy: 0.15 | | |

Continued on next page

Table 6.1 – *Continued from previous page*

| Model | OFV^{VC} (%) | OFV^{MoI} (%) | Volume (%) | Run time (min) |
|--------------|--------------------------------|---------------------------------|----------------------|--------------------------|
| | | Iz: 0.23 | | |
| | | Middle part | | |
| | VCx: 0.97 | Ix: 1.48 | 0.32 | |
| | VCz: 0.62 | Iy: 0.14 | | |
| | | Iz: 0.54 | | |
| | | Afterward part | | |
| | VCx: 0.95 | Ix: 0.68 | 0.98 | |
| | VCz: 1.14 | Iy: 1.68 | | |
| | | Iz: 1.29 | | |
| Model_6 | | Whole ship-hull | | |
| | VCx: 1.14 | Ix: 0.33 | 0.03 | 214 |
| | VCz: 0.74 | Iy: 0.42 | | |
| | | Iz: 0.42 | | |
| | | Forward part | | |
| | VCx: 1.92 | Ix: 1.88 | 0.02 | |
| | VCz: 1.90 | Iy: 0.28 | | |
| | | Iz: 0.50 | | |
| | | Middle part | | |
| | VCx: 0.98 | Ix: 0.08 | 0.11 | |
| | VCz: 0.06 | Iy: 0.22 | | |
| | | Iz: 0.13 | | |
| | | Afterward part | | |
| | VCx: 0.92 | Ix: 1.22 | 0.01 | |
| | VCz: 0.13 | Iy: 1.69 | | |
| | | Iz: 1.79 | | |
| Model_7 | | Whole ship-hull | | |

Continued on next page

Table 6.1 – *Continued from previous page*

| Model | OFV^{VC} (%) | OFV^{MoI} (%) | Volume (%) | Run time (min) |
|--------------|--------------------------------|---------------------------------|----------------------|--------------------------|
| | VCx: 0.87 | Ix: 1.10 | 0.11 | 87 |
| | VCz: 0.29 | Iy: 1.30 | | |
| | | Iz: 1.24 | | |
| | Forward part | | | |
| | VCx: 1.72 | Ix: 0.42 | 1.10 | |
| | VCz: 0.16 | Iy: 0.95 | | |
| | | Iz: 0.72 | | |
| | Middle part | | | |
| | VCx: 0.32 | Ix: 1.62 | 0.49 | |
| | VCz: 0.51 | Iy: 0.49 | | |
| | | Iz: 0.15 | | |
| | Afterward part | | | |
| | VCx: 0.36 | Ix: 0.50 | 0.40 | |
| | VCz: 0.06 | Iy: 1.94 | | |
| | | Iz: 1.79 | | |
| Model_8 | Whole ship-hull | | | |
| | VCx: 0.77 | Ix: 1.09 | 0.54 | 42 |
| | VCz: 0.33 | Iy: 1.36 | | |
| | | Iz: 1.34 | | |
| | Forward part | | | |
| | VCx: 0.60 | Ix: 0.09 | 0.32 | |
| | VCz: 1.11 | Iy: 0.01 | | |
| | | Iz: 0.09 | | |
| | Middle part | | | |
| | VCx: 0.25 | Ix: 0.74 | 0.39 | |
| | VCz: 0.09 | Iy: 0.42 | | |

Continued on next page

Table 6.1 – *Continued from previous page*

| Model | OFV^{VC} (%) | OFV^{MoI} (%) | Volume (%) | Run time (min) |
|--------------|--------------------------------|---------------------------------|----------------------|--------------------------|
| | | Iz: 0.54 | | |
| | | Afterward part | | |
| | VCx: 0.54 | Ix: 0.80 | 1.61 | |
| | VCz: 0.15 | Iy: 0.04 | | |
| | | Iz: 0.04 | | |
| Model_9 | | Whole ship-hull | | |
| | VCx: 0.55 | Ix: 0.98 | 0.13 | 153 |
| | VCz: 0.74 | Iy: 0.62 | | |
| | | Iz: 0.63 | | |
| | | Forward part | | |
| | VCx: 0.09 | Ix: 0.65 | 1.43 | |
| | VCz: 0.15 | Iy: 0.21 | | |
| | | Iz: 0.34 | | |
| | | Middle part | | |
| | VCx: 0.32 | Ix: 0.40 | 0.14 | |
| | VCz: 0.09 | Iy: 0.49 | | |
| | | Iz: 0.22 | | |
| | | Afterward part | | |
| | VCx: 0.23 | Ix: 0.92 | 1.81 | |
| | VCz: 0.92 | Iy: 0.31 | | |
| | | Iz: 0.18 | | |
| Model_10 | | Whole ship-hull | | |
| | VCx: 0.84 | Ix: 0.03 | 0.30 | 457 |
| | VCz: 0.74 | Iy: 0.02 | | |
| | | Iz: 0.02 | | |
| | | Forward part | | |

Continued on next page

Table 6.1 – *Continued from previous page*

| Model | OFV^{VC} (%) | OFV^{MoI} (%) | Volume (%) | Run time (min) |
|--------------|--------------------------------|---------------------------------|----------------------|--------------------------|
| | VCx: 0.92 | Ix: 1.63 | 1.94 | |
| | VCz: 0.40 | Iy: 0.29 | | |
| | | Iz: 0.53 | | |
| | Middle part | | | |
| | VCx: 0.34 | Ix: 1.83 | 0.49 | |
| | VCz: 0.69 | Iy: 0.57 | | |
| | | Iz: 0.94 | | |
| | Afterward part | | | |
| | VCx: 0.38 | Ix: 0.56 | 1.67 | |
| | VCz: 1.45 | Iy: 1.21 | | |
| | | Iz: 0.87 | | |

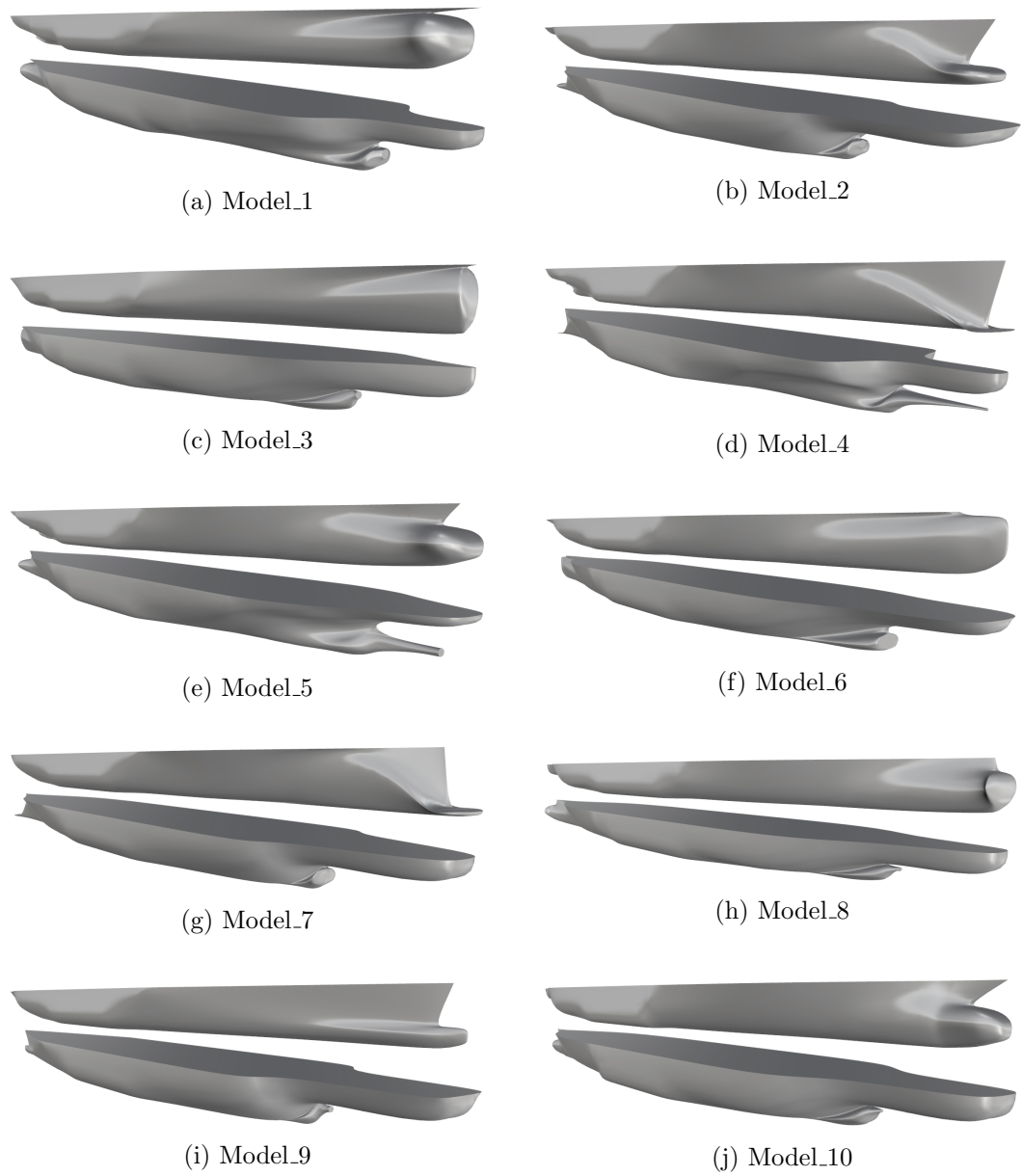


Figure 6.3: Optimised models resulted from TshipPM MO-TLBO. Bow and stern view captures are illustrated for each model.

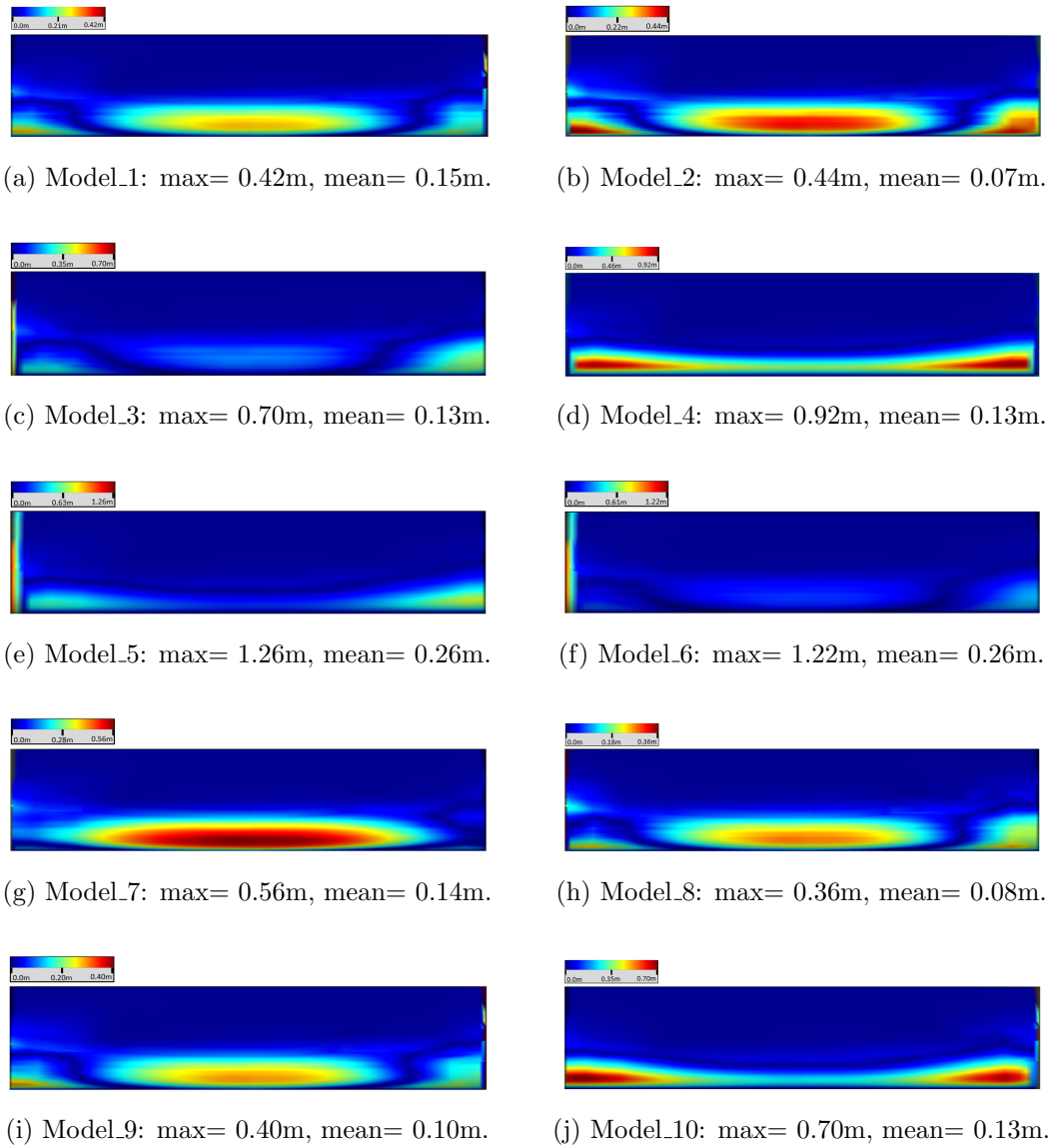


Figure 6.4: Geometric variation between the parallel middle parts of TshipPM MO-TLBO models and the parent hull, using Hausdorff distance.

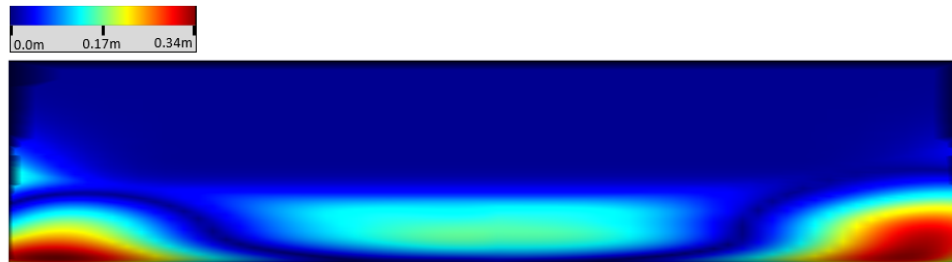


Figure 6.5: Geometric variation between the parallel middle part of TshipPM MO-TLBO models and the parent hull, using stricter stopping criteria. Model_11: max= 0.34m, mean= 0.05m.

Discussion and Summary

The work presented herein is about an innovative, robust, and cost-effective CAD parametric modelling tool for ship-hull design and especially for container and tanker ship-hulls, developed both in RhinoScript and C#. TshipPM is responsible for constructing the control cage of the desired ship-hull. It uses 27 user-specified parameters, 3 dimensional/physical of global nature, namely length at waterline level (Lwl), draft (T), and breadth (B), and 24 non-dimensional of local nature lying in the range [0.01, 0.99], and which are mapped to control points through linear mappings. The control points are interconnected with straight line segments to construct the desired control cage. TshipPM uses the T-splines representation in the context of Autodesk's T-splines plug-in to create the final smooth surface(s). T-splines favour the development of surfaces with a much lower number of DoF (degrees of freedom) compared to the standard NURBS technology, as it takes advantage of T-junctions. For the same reason, this fairly new technology, provides advantages for local refinement of complex areas. In addition, TshipPM is able to produce a single model in less than 10 seconds, performance which is mainly contributed to the very low number of control points (276) it deploys for the production of a ship-hull.

This work transforms the parametric modelling tool used in [6] along several directions, which constitute the contributions to the field of parametric modelling in Computer-Aided Ship Design (CASD): TshipPM is capable of producing ship-hull models for both containers and tankers, taking into consideration the detailed characteristics of both ship types. TshipPM enables a flexible representation of the ship-hull in the geometrically challenging areas of bow and stern, and the transition areas from midship

towards the forward and afterward perpendiculars, with a set of newly introduced transition curves, and a more effective representation of the bow and stern boundary curves. Therefore, it is eventually capable of delivering instances lying in the proximity of any ship-hull type, as long as its complexity does not surpass the complexity of tankers and containers.

Furthermore, a list of all parameters that characterise the geometry of the produced models is provided, along with a concise description of each one of them and their functional interrelation. We have described graphically all control curves used in TshipPM, and we provide the functional relations of their control points with the physical and non-dimensional parameters. In addition, the current work illustrates the mapping of the external parameters to the control points of the T-splines representation. On top of this, we present a step-by-step *remodelling* method for any given container or tanker ship-hull, using as a case study the MOERI Container Ship's (KCS) hull instance, which is extensively used by the research community for CAD and Computational Fluid Dynamics (CFD) benchmarking purposes. The process consists of a detailed description of the control cage construction using TshipPM, and the manual optimisation of the control cage to more accurately remodel the given ship-hull.

Significant contribution has been attained as far as the cumbersome and complicated issue of *geometric validity* is concerned, following the imposition of a set of *geometric* and *design* constraints, which resulted in the production of geometrically valid models for the whole range of the non-dimensional parameter values. A special attention has been given and thorough investigation has been conducted as far as the intricate issue of self-intersection of neighbouring surfaces is concerned. The issue was addressed using automated techniques provided by Rhino5's functionality to investigate the watertightness of the output models, as well as a user-based visual evaluation for the cases of invalid models that could not be identified automatically in Rhino.

Furthermore, we have compared TshipPM with CAESES against a set of comparison criteria, namely the common external parameters, volume, volume centroid, moments of inertia, sectional area curve (SAC), Gaussian and sectional curvatures for assessing

surface and curve fairness respectively, and the Hausdorff distance for measuring the geometric distance between the hulls. The outcome was extremely positive for TshipPM, as it performed at the same level with CAESES, and even better in some aspects, with major advantages: a) the production of models with very low computational complexity of the resulting surface, with just 276 points compared with the vs $O(10^5)$ points of the high fidelity CAESES model, b) the increased fairness with much fewer extraneous Gaussian-curvature sign variations compared to CAESES and c) increased smoothness (least G^1 -continuity) of the surfaces. On the other hand, TshipPM showed not as good performance as CAESES in representing the flat sides of the ship-hull, and especially the flat of side of both forward and afterward parts. Although out of scope for the current work, it is worthwhile to mention that techniques and methods to tackle this issue have been developed, and they mainly require the introduction of a set of control points around the areas of interest, without significant expense in the DoF number of the produced hulls.

Finally, a method and a tool for shape-optimisation against a set of ship-design criteria (volume moments up to 2nd order) has been introduced, incorporating in TshipPM an adjusted for the purpose of the current work *Multi-objective Teaching-Learning-based optimisation* method. The method takes advantage of TshipPM's developing methodology and it breaks down a very demanding and complex optimisation problem of 24 parameters and 18 costs to 3 separate optimisation problems of 6 costs and 3, 10, and 11 parameters for the middle, afterward, and forward part of the hull respectively. As a result, TshipPM MO-TLBO finds a set of optimal solutions in about 155 minutes, time which can decrease in half given that the optimisation problem of the forward and afterward parts can be conducted in parallel. The main shortcoming of the method, proven by the results, is that volume moments up to 2nd order are insufficient shape optimisers when it comes to complex objects, such as the bow and the stern areas of a tanker or container ship-hull. A better solution could be attained by using higher order volume moments.

Bibliography

- [1] P. de Casteljaou, *Courbes et surfaces a poles*, Technical report. Citroën, 1963.
- [2] S. N. Bernstein, “Démonstration du théorème de weierstrass fondée sur le calcul des probabilités,” *Comm. Soc. Math. Kharkov*, vol. 2.Series, XIII, pp. 1–2, 1912.
- [3] —, “O nailuchshem priblizhenii nepreryvnykh funktsii posredstvom mnogochlenov dannoi stepeni,” *Comm. Soc. Math. Kharkov*, vol. 2.Series, XIII No. 2-5, pp. 49–194, 1913.
- [4] P. Bézier, *The Mathematical Basis of the UNISURF CAD System*. Butterworth & Co (Publishers) Ltd, 1986, ISBN: 0-408-22175-5. DOI: 10.1016/B978-0-408-22175-7.50003-0.
- [5] H. Nowacki, G. Creutz, and F. Munchmeyer, “Ship lines creation by computer - objectives, methods and results,” in *Proceedings of the Ship-Symposium on Computer-Aided Hull Surface Definition*, Annapolis, MD, USA, 1977.
- [6] K. Kostas, A. Ginnis, C. Politis, and P. 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, Jan. 2015. DOI: 10.1016/j.cma.2014.10.030.
- [7] T. Katsoulis, X. Wang, and P. Kaklis, “A T-splines-based parametric modeller for computer-aided ship design,” *Ocean Engineering*, vol. 191, 2019.
- [8] T. Sederberg, J. Zheng, A. Bakenov, and A. Nasri, “T-splines and T-NURCCs,” *ACM Transactions on Graphics (TOG)*, vol. 22, pp. 477–484, 2003. DOI: 10.1145/882262.882295.

- [9] MOERI. (2008). “Container ship (KCS),” [Online]. Available: <http://www.simman2008.dk/KCS/container.html> (visited on 02/01/2018).
- [10] —, (2008). “Tanker ship (KVLCC),” [Online]. Available: <http://www.simman2008.dk/KVLCC/KVLCC1/tanker1.html> (visited on 02/01/2018).
- [11] I. J. Schoenberg, “On the problem of smoothing or graduation. A first class of analytic approximation formulae,” *Contributions to the problem of approximation of equidistant data by analytic functions*, vol. 4, pp. 45–99, 1 1946.
- [12] H. B. Curry and I. J. Schoenberg, “On spline distributions and their limits: The pólya distribution functions,” *Abstract 380t, Bulletin of the American Mathematical Society*, vol. 53, p. 1114, 1947.
- [13] C. de Boor, *A Practical Guide to Splines*, 1st. Springer-Verlag New York, 1978, ISBN: 978-0-387-95366-3.
- [14] J. Favard, “Sur l’interpolation,” *Journal de Mathématiques Pures et Appliquées*, vol. 19, pp. 281–306, 105 1940.
- [15] K. Versprille, “Computer aided design applications of the rational B-spline approximation form,” Ph.D. dissertation, New York, 1975.
- [16] S. Harries and H. Nowacki, “Form-parameter approach to the design of fair hull shapes,” Cambridge, MA, USA, 1999.
- [17] H.-C. Kim, “Parametric design of ship hull forms with a complex multiple domain surface topology,” Ph.D. dissertation, Berlin, Germany, 2004.
- [18] C. Abt and S. Harries, “A new approach to integration of CAD and CFD for naval architects,” in *6th International Conference on Computer Applications and Information Technology in Maritime Industries (COMPIT)*, Cortona, Italy, 2007.
- [19] S. Harries, “Serious play in ship design - tradition and future of ship design,” in *Berlin Colloquium*, Berlin, Germany, 2008.
- [20] P. Zhang and D.-X. Zhu, “Parametric approach to design of hull forms,” *Journal of Hydrodynamics*, vol. 20, pp. 804–810, 2008. DOI: 10.1016/S1001-6058(09)60019-6.

- [21] T. Sederberg, D. Cardon, G. Finnigan, N. North, J. Zheng, and T. Lyche, “T-spline simplification and local refinement,” *ACM Transactions on Graphics (TOG)*, vol. 23, pp. 276–283, 2004. DOI: 10.1145/1015706.1015715.
- [22] M.A.Scott, X.Li, T.W.Sederberg, and T.J.R.Hughes, “Local refinement of analysis-suitable T-splines,” *Computer Methods in Applied Mechanics and Engineering*, vol. 206-222, 2012. DOI: 10.1016/j.cma.2011.11.022.
- [23] J. Zhanga and X. Li, “Local refinement for analysis-suitable++ T-splines,” *Computer Methods in Applied Mechanics and Engineering - COMPUT METHOD APPL MECH ENG*, vol. 342, pp. 32–45, Jul. 2018. DOI: 10.1016/j.cma.2018.07.024.
- [24] T. W. Sederberg, X. Li, H. Lin, and H. Ipson, “Watertight trimmed NURBS,” *ACM Transactions on Graphics (TOG)*, vol. 27, pp. 01–08, 3 Aug. 2008. DOI: 10.1145/1360612.1360678.
- [25] C.-K. Shene, *NURBS: Important properties*, 2015. [Online]. Available: <https://pages.mtu.edu/~shene/COURSES/cs3621/NOTES/spline/NURBS/NURBS-property.html>.
- [26] A. Buffa, D. Cho, and G. Sangalli, “Linear independence of the T-spline blending functions associated with some particular T-meshes,” *Computer Methods in Applied Mechanics and Engineering - COMPUT METHOD APPL MECH ENG*, vol. 199, pp. 1437–1445, Apr. 2010. DOI: 10.1016/j.cma.2009.12.004.
- [27] X. Li, J. Zheng, T. Sederberg, T. Hughes, and M. Scott, “On linear independence of T-splines,” *Computer Aided Geometric Design*, vol. 29, pp. 63–76, Jan. 2012. DOI: 10.1016/j.cagd.2011.08.005.
- [28] L. Veiga, A. Buffa, D. Cho, and G. Sangalli, “Analysis-suitable T-splines are dual-compatible,” *Computer Methods in Applied Mechanics and Engineering*, vol. 249–252, 42–51, Dec. 2012. DOI: 10.1016/j.cma.2012.02.025.
- [29] L. Veiga, A. Buffa, G. Sangalli, and R. Vázquez, “Analysis-suitable T-splines of arbitrary degree: Definition, linear independence and approximation properties,”

- Mathematical Models and Methods in Applied Sciences*, vol. 23, Jul. 2013. DOI: 10.1142/S0218202513500231.
- [30] X. Li, “Some properties for analysis suitable T-splines,” *Journal of Computational Mathematics*, vol. 33, 428–442, 4 2015. DOI: 10.4208/jcm.1508-m4493.
- [31] X. Li and Z. Jingjing, “AS++ T-splines: Linear independence and approximation,” *Computer Methods in Applied Mechanics and Engineering*, vol. 333, Feb. 2018. DOI: 10.1016/j.cma.2018.01.041.
- [32] X. Li and M. A. Scott, “Analysis-suitable T-splines: Characterization, refineability, and approximation,” *Mathematical Models and Methods in Applied Sciences*, vol. 24, pp. 1141–1164, 2014. DOI: 10.1142/S0218202513500796.
- [33] Y. Bazilevs, V. Calo, J. Cottrell, J. Evans, T. Hughes, S. Lipton, M. Scott, and T. Sederberg, “Isogeometric analysis using T-splines,” *Computer Methods in Applied Mechanics and Engineering*, vol. 199, pp. 229–263, Jan. 2010. DOI: 10.1007/978-3-7091-1843-6_5.
- [34] J. Cottrell and T. H. Y. Bazilevs, *Isogeometric analysis: Toward Integration of CAD and FEA*. Wiley, Chichester, 2009.
- [35] T. Hughes, J. Cottrell, and Y. Bazilevs, “Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement,” *Computer Methods in Applied Mechanics and Engineering*, vol. 194, pp. 4135–4195, 2005. DOI: 10.1016/j.cma.2004.10.008.
- [36] M. Scott, R. Simpson, J. Evans, S. Lipton, S. Bordas, T. Hughes, and T. Sederberg, “Isogeometric boundary element analysis using unstructured T-splines,” *Computer Methods in Applied Mechanics and Engineering*, vol. 254, pp. 177–221, 2013. DOI: 10.1016/j.compstruc.2012.12.021.
- [37] H. Lian, R. Simpson, and S. Bordas, “Sensitivity analysis and shape optimisation through a T-spline isogeometric boundary element method,” in *International Conference on Computational Mechanics (CM13)*, Durham, UK, 2013.

- [38] A.-A. Ginnis, K. Kostas, C. Politis, P. Kaklis, K. Belibassakis, T. Gerostathis, M. Scott, and T. R. Hughes, “Isogeometric boundary-element analysis for the wave-resistance problem using T-splines,” *Computer Methods in Applied Mechanics and Engineering*, vol. 279, pp. 425–439, 2014. DOI: 10.1016/j.cma.2014.07.001.
- [39] P. Kaklis, C. Politis, K. Belibassakis, A.-A. Ginnis, K. Kostas, and T. Gerostathis, “Boundary element methods and wave loading on ships,” in *Encyclopedia of Computational Mechanics, 2nd Edition*, ed. by Erwin Stein and René de Borst and Thomas J.R. Hughes, Chichester UK, John Wiley & Sons, Ltd, 2018, pp. 1–35. DOI: 10.1002/9781119176817.ecm2115.
- [40] X. Li, J. Zheng, T. Sederberg, T. Hughes, and M. Scott, “On the linear independence of T-spline blending functions,” *Comput. Aided Geom. Design*, vol. 29, pp. 63–76, 2012. DOI: 10.1016/j.cagd.2011.08.005.
- [41] A. Bressan, A. Buffa, and G. Sangalli, “Characterisation of analysis-suitable T-splines,” *Comput. Aided Geom. Design*, vol. 39, pp. 17–49, 2015.
- [42] J. Zhang and X. Li, “On degree-elevation of T-splines,” *Comput. Aided Geom. Design*, vol. 46, pp. 16–29, 2016. DOI: 10.1016/j.cagd.2016.05.010.
- [43] C. Hoffmann and K.-J. Kim, “Towards valid parametric CAD models,” *Computer-Aided Design*, vol. 33, pp. 81–90, 2001. DOI: 10.1016/S0010-4485(00)00073-7.
- [44] G. Farin, *Curves and surfaces for computer aided geometric design. A practical guide*, 5th. Morgan Kaufmann, 2002, ISBN: 978-1-55860-737-8.
- [45] H. Nowacki, “Five decades of computer-aided ship design,” *Computer-Aided Design*, vol. 42, pp. 956–969, 2010. DOI: 10.1016/j.cad.2009.07.006.
- [46] H. Lackenby, “On the systematic geometrical variation of ship form,” in *Transactions of the Royal Institute of Naval Architects (RINA)*, vol. 92, 1950. DOI: 10.5574/KSOE.2013.27.2.001.
- [47] H. Koelman, I. Horváth, and A. Aalbers, “Hybrid representation of the shape of ship hulls,” *International Shipbuilding Progress*, vol. 48, pp. 247–269, 2001.

- [48] D. Forsey and R. Bartels, “Hierarchical B-spline refinement,” *SIGGRAPH Computer Graphics*, vol. 22, pp. 205–212, 1988. DOI: 10.1145/54852.378512.
- [49] J. Deng, F. Chen, X. Li, C. Hu, W. Tong, Z. Yang, and Y. Feng, “Polynomial splines over hierarchical T-meshes,” *Graphical Models*, vol. 70, pp. 76–86, 2008. DOI: 10.1016/j.gmod.2008.03.001.
- [50] T. Dokken, T. Lyche, and K. Pettersen, “Polynomial splines over locally refined box partitions,” *Computer Aided Geometric Design*, vol. 30, pp. 331–356, 2013. DOI: 10.1016/j.cagd.2012.12.005.
- [51] M. Dörfel, B. Jüttler, and B. Simeon, “Adaptive isogeometric analysis by local h-refinement with T-splines,” *Computer Methods in Applied Mechanics and Engineering*, vol. 199, pp. 264–275, 2009. DOI: 10.1016/j.cma.2008.07.012.
- [52] A. Harries, C. Abt, and M. Brenner, “Upfront CAD – parametric modeling techniques for shape optimization,” in *Advances in Evolutionary and Deterministic Methods for Design, Optimization and Control in Engineering and Sciences, Minisci E. et al. (Eds.), Springer 978-3-319-89986-2*, Int. Conf. Evolutionary et al., 2018, pp. 74–81.
- [53] S. Harries and K. Abt, “Parametric curve design applying fairness criteria,” in *International Workshop on Creating Fair and Shape-Preserving Curves and Surfaces – Network Fairshape*, (eds) H. Nowacki and P.D. Kaklis, Berlin, Potsdam, 1998.
- [54] J. Holland, *Adaptation in Natural and Artificial Systems*. The University of Michigan Press, 1975, ISBN: 9780262082136.
- [55] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., 1989, ISBN: 978-0-201-15767-3.
- [56] D. Whitley, “A genetic algorithm tutorial,” *Statistics and Computing*, vol. 4, pp. 65–68, 1994. DOI: 10.1007/BF00175354.
- [57] M. Mitchell, *An Introduction to Genetic Algorithms*. The University of Michigan Press, 1998, ISBN: 9780262133166.

- [58] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION*, vol. 6, pp. 182–197, 2 2002. DOI: 10.1109/4235.996017.
- [59] J. R. Koza, *Genetic programming*, 1994.
- [60] M. Willis, H. Hiden, P. Marenbach, B. McKay, and G. Montague, “Genetic programming: An introduction and survey of applications,” Oct. 1997, pp. 314–319, ISBN: 0-85296-693-8. DOI: 10.1049/cp:19971199.
- [61] J. R. Koza and R. Poli, *A genetic programming tutorial*, E. Burke, Ed., 2003.
- [62] J. Koza and R. Poli, in *Genetic Programming*. Jan. 2005, pp. 127–164, ISBN: 9780387283562. DOI: 10.1007/0-387-28356-0_5.
- [63] T. Bäck, *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford University Press, Inc., 1996, ISBN: 0-19-509971-0.
- [64] T. Bäck, D. Fogel, and Z. Michalewicz, *Handbook of Evolutionary Computation*. Boca Raton: CRC Press, 1997. DOI: 10.1201/9780367802486.
- [65] X. Yao, Y. Liu, and G. Lin, “Evolutionary programming made faster,” *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 2, pp. 82–102, 1999.
- [66] R. Storn and K. Price, “Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces,” *Journal of Global Optimization*, vol. 11, pp. 341–359, Jan. 1997. DOI: 10.1023/A:1008202821328.
- [67] K. Price, R. Storn, and J. Lampinen, *Differential Evolution: A practical approach to global optimization*. Springer, 2005, ISBN: 9783540313069.
- [68] U. Chakraborty, *Advances in Differential Evolution*. Springer, 2008, ISBN: 978-3-540-68827-3.
- [69] J. Bishop, “Stochastic searching networks,” Nov. 1989, pp. 329–331.
- [70] M. M. Al-Rifaie and J. Bishop, “Stochastic diffusion search review,” in. Jan. 2013, vol. 4, pp. 155–173. DOI: 10.2478/pjbr-2013-0021.

- [71] M. Dorigo and G. Di Caro, “Ant colony optimization: A new meta-heuristic,” in *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, vol. 2, 1999, 1470–1477 Vol. 2.
- [72] M. Dorigo, M. Birattari, and T. Stutzle, “Ant colony optimization,” *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28–39, 2006.
- [73] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proceedings of ICNN’95 - International Conference on Neural Networks*, vol. 4, 1995, 1942–1948 vol.4.
- [74] E. Bonabeau, M. Dorigo, and G. Theraulaz, *From Natural to Artificial Swarm Intelligence*. Oxford University Press, Inc., 1999, ISBN: 978-0-19-513158-1.
- [75] E. Alba, C. Blum, P. Isasi, C. León, and J. A. Gómez, *Optimization Techniques for Solving Complex Problems*. John Wiley & Sons, Inc., 2008, ISBN: 9780470293324. DOI: 10.1002/9780470411353.
- [76] R. Rao, V. Savsani, and D. Vakharia, “Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems,” *Computer-Aided Design*, vol. 43, pp. 303–315, 3 2011. DOI: 10.1016/j.cad.2010.12.015.
- [77] R. V. Rao, *Teaching Learning Based Optimization Algorithm And Its Engineering Applications*. Springer, Cham, 2016, ISBN: 978-3-319-22732-0.