



NATIONAL TECHNICAL UNIVERSITY OF ATHENS

SCHOOL OF
NAVAL ARCHITECTURE & MARINE ENGINEERING

DOCTORAL DISSERTATION

A Ship Parametric Model for Isogeometric Analysis

Doctoral Candidate:

Dipl.-Ing. Christian FEURER

Advisory Committee:

Dr.-Ing. Alexandros-Alvertos I. GINNIS¹

Dr. Bert JÜTTLER²

Dr.-Ing. Panagiotis D. KAKLIS¹ (Supervisor)

1: National Technical University of Athens

2: Johannes Kepler University, Linz

Athens, July 2013



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΝΑΥΠΗΓΩΝ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

A Ship Parametric Model for Isogeometric Analysis

Υποψήφιος Διδάκτωρ:

Διπλ.-Μηχ. Christian FEURER

Συμβουλευτική Επίτροπη:

Δρ.-Μηχ. Αλέξανδρος-Αλβέρτος Ι. ΓΚΙΝΗΣ¹

Δρ. Bert JÜTTLER²

Δρ.-Μηχ. Παναγιώτης Δ. ΚΑΚΛΗΣ¹ (Επιβλέπων)

1: Εθνικό Μετσόβιο Πολυτεχνείο

2: Johannes Kepler University, Linz

Αθήνα, Ιούλιος 2013

Ευχαριστίες

Πολλά άτομα με υποστήριξαν στην ανάπτυξη αυτής της διατριβής. Είμαι ευγνώμων προς όλους. Παρόλα αυτά θα ήθελα να αναφέρω ονομαστικά τους κάτωθι:

Ευχαριστώ πρωτίστως τους γονείς μου, Alfred και Maria, ολόκληρη την οικογένεια μου και τους φίλους μου. Αυτοί είναι τα θεμέλια μου, το σύστημά μου, η πραγματικότητά μου. Μοιράζομαι με αυτούς την απέραντη χαρά της ολοκλήρωσης αυτής της προσπάθειας.

Αναγνωρίζω επίσης με ευγνωμοσύνη το υπόβαθρο, τους πόρους και τις υπέροχες ευκαιρίες που μου παρείχε το Εθνικό Μετσόβιο Πολυτεχνείο (ΕΜΠ) σε συνδυασμό με το συγχρηματοδοτούμενο ερευνητικό πρόγραμμα του FP7, Exciting¹.

Ευχαριστώ θερμά τη συμβουλευτική επιτροπή μου, Αλέξανδρο-Αλβέρτος Ι. Γκίνη, Bert Jüttler και Παναγιώτη Δ. Κακλή. Η συνεχής υποστήριξή τους σε όλες τις εκφάνσεις της διδακτορικής διατριβής υπήρξε θεμελιώδης. Ήταν μεγάλο προνόμιό μου να τους έχω ως επιβλέποντες και ως μέντορες μου.

Εκφράζω τη βαθιά μου εκτίμηση στον Παναγιώτη Κακλή για τη συνολική βοήθειά του. Οι συμβουλές και η κρίση του καθώς και η υπομονή και η διορατικότητά που επέδειξε κατά τη συνεργασία μας ξεπέρασαν κατά πολύ το επαγγελματικό καθήκον. Σας ευχαριστώ για όλα!

Η δημιουργική ανταλλαγή ιδεών με τον Αλέξανδρο Γκίνη συντέλεσε στην παραγωγή πολλών αρχών αυτού του έργου. Είμαι ευγνώμων για την συμβολή και το ενδιαφέρον του σε αυτή τη διατριβή.

Ο Bert Jüttler κατηύθυνε την ακαδημαϊκή μου σταδιοδρομία ήδη από τις σπουδές μου στο πανεπιστήμιο Johannes Kepler Universität στο Linz. Είμαι ιδιαίτερος ευγνώμων σε αυτόν για τη μακροχρόνια επαγγελματική και προσωπική υποστήριξη του.

Ευχαριστώ όλους του συναδέλφους μου στην ερευνητική ομάδα του Exciting στην Αθήνα, Θεόδωρο Π. Γεροστάθη, Κωνσταντίνο Β. Κώστα, Δημήτρη Ν. Μουρκογιάννη, Κωνσταντίνο Α. Μπελιμπασάκη και Κωνσταντίνο Γ. Πολίτη. Ο ρόλος τους στην ακαδημαϊκή μου εξέλιξη είναι ιδιαίτερα σημαντικός και οι συμβουλές και η κριτική τους είχαν θετική επιρροή σε αυτό το έργο από τις θεμελιώδεις αρχές του ως τις τεχνικές λεπτομέρειες.

Πολλοί συνάδελφοι, συμφοιτητές, φίλοι αλλά και μέλη της οικογένειάς μου με βοήθησαν με το έντονο ενδιαφέρον τους και τον προβληματισμό τους, τις συμβουλές τους και τη βοήθεια στη μετάφραση κατά την εκπόνηση της παρούσας διατριβής. Τους ευχαριστώ όλους για την υποστήριξη και τους ευχαριστώ ειδικά για τη θετική τους ενέργεια.

Είμαι ευγνώμων για την τιμή που δόθηκε να ζήσω, να δουλέψω και να σπουδάσω στην υπέροχη γη και τον πολιτισμό της Ελληνικής Δημοκρατίας. Ευχαριστώ τους ζωντανούς και όμορφους ανθρώπους της Ελλάδας για τη φιλοξενία τους, και την μύησή μου σε αυτήν την μοναδική εμπειρία που ευλόγησε τη ζωή μου.

Σας ευχαριστώ όλους, danke!

Christian Feurer, Αθήνα Ιούλιος 2013.

¹Exciting – Exact geometry simulation for optimized design of vehicles and vessels: Ένα ερευνητικό έργο στο FP7-Framework της Ευρωπαϊκής Ένωσης (FP7- SST- 2007- RTD-1 - 218536).

Σύνοψη

Η παρούσα διατριβή πραγματεύεται την ανάπτυξη ενός παραμετρικού μοντέλου γάστρας πλοίου εμπορευματοκιβωτίων για τις συγκεκριμένες απαιτήσεις της ισογεωμετρικής ανάλυσης.

Κατά τη διαδικασία σχεδιασμού ενός τρισδιάστατου αντικειμένου ο σχεδιασμός της μορφής και η ανάλυση της λειτουργικότητάς του υλοποιείται παραδοσιακά σε δύο διαφορετικά κατά σειρά στάδια. Η διακριτοποίηση της γεωμετρίας με σκοπό τη χρήση αριθμητικών μεθόδων αποτελεί ένα βασικό ενδιάμεσο βήμα το οποίο είναι δαπανηρό και μπορεί να εισάγει ένα εν δυνάμει μεγάλο σφάλμα προσέγγισης. Για την επίλυση αυτού του ζητήματος, το 2005 ο Hughes et.al. εισήγαγαν την μέθοδο της *ισογεωμετρικής ανάλυσης* σύμφωνα με την οποία οι συναρτήσεις βάσης της υποκείμενης γεωμετρικής απεικόνισης υιοθετούνται για την προσέγγιση της αριθμητικής επίλυσης του προαναφερθέντος προβλήματος, δείτε π.χ. [50, 23]. Κατ' αυτόν τον τρόπο, κάθε γεωμετρική προσέγγιση παρακάμπτεται και τα πεδία της σχεδίασης με τη χρήση ηλεκτρονικού υπολογιστή (computer-aided design – CAD) και της ανάλυσης ενοποιούνται.

Αυτή η δυνατότητα αποτελεί το κύριο κίνητρο για την παρούσα διδακτορική διατριβή, στο πλαίσιο της οποίας αναπτύχθηκε διαφανής μεθοδολογία παραγωγής της γενικής γάστρας πλοίου *μεταφοράς εμπορευματοκιβωτίων* ο οποίος ικανοποιεί πλήρως τους βασικούς περιορισμούς μορφής της εν λόγω οικογένειας γαστρών.

Για να βελτιστοποιηθεί αποτελεσματικά η γάστρα ενός πλοίου, όπως και κάθε αντικείμενο, ως προς ένα συγκεκριμένο κριτήριο, θα πρέπει να οριστεί ο μηχανισμός για την αυτόματη δημιουργία της γεωμετρίας. Ως εκ τούτου η έννοια της *παραμετρικής μοντελοποίησης* είναι θεμελιώδης για την παρούσα εργασία. Το προαναφερθέν μοντέλο της γάστρας του πλοίου εξαρτάται από 29 ολικές και τοπικές παραμέτρους σχεδίασης και από 9 συμπληρωματικές εσωτερικές παραμέτρους.

Είναι φυσικό να μοντελοποιηθούν πλοία μέσω της μοντελοποίησης επιφανειών και για αυτό τον λόγο ακολουθήθηκε μία τοπική σχεδίαση βασισμένη σε ένα μοντέλο δικτύου καμπυλών ελέγχου. Η ορθή κατασκευή αυτών αποδίδει κατά κύριο λόγο λόγω τετράπλευρους κλειστούς βρόγχους οι οποίοι «πληρούνται» με την κατασκευή των γνωστών επιφανειακών τμημάτων Coons. Για να διασφαλιστεί η ομαλή μετάβαση μεταξύ αυτών των επιφανειακών τμημάτων, παρουσιάζουμε έναν πρωτότυπο αλγόριθμο εμπλουτισμού των καμπυλών του δικτύου των καμπυλών ελέγχου με κατανομές διανυσμάτων, οι οποίες αποκαλούνται *πλαίσια* (frame), τα οποία χρησιμοποιούνται ως τα κάθετα διανύσματα των κοινών, μεταξύ των επιφανειακών τμημάτων Coons, εφαπτομενικών επιπέδων. Αυτή η πληροφορία χρησιμοποιείται στη συνέχεια για την κατασκευή των συνοριακών εφαπτομενικών λωρίδων του δικτυβικά αναμειγμένου επιφανειακού τμήματος Coons και με αυτόν τον τρόπο επιτυγχάνεται η δημιουργία επιφανειακών τμημάτων που διαθέτουν συνέχεια εφαπτομενικού επιπέδου.

Η επιλεγθείσα απεικόνιση των γεωμετρικών οντοτήτων που συνθέτουν την επιφάνεια της γάστρας συνθετικών είναι αυτή των B-splines, η οποία θεωρείται ως συνήθης στη σχεδίαση με τη χρήση ηλεκτρονικού υπολογιστή και η οποία είναι κατάλληλη για ισογεωμετρική ανάλυση χάρη στον απλό και γεωμετρικά διαισθητικό τρόπο με τον επιτρέπει η αναπαράσταση B-splines την υλοποίηση της εκλέπτυνσης (refinement). Οι αλγόριθμοι που παρουσιάζονται και μία πιθανή διαδικασία σχεδίασης της γεωμετρίας πλαισίου προγραμματίστηκαν κάνοντας χρήση της γλώσσας

C++ και είχαν ως αποτέλεσμα τη δημιουργία του προγράμματος SmoothPM. Στην εργασία αυτή παρουσιάζεται η αξιολόγηση του προγράμματος, εξετάζοντας τα εξαγόμενα της διαδικασίας που προέκυψαν από 18 διαφορετικά σύνολα παραμέτρων εισόδου. Επιπρόσθετα παρέχονται προτάσεις για μελλοντικές εργασίες και επέκταση της παρούσας διδακτορικής διατριβής.

Λέξεις κλειδιά: CAD, CAGD, εξομάλυνση, ισογεωμετρική ανάλυση, ισογεωμετρική ΜΣΣ, παραμετρική μοντελοποίηση, πλαίσιο, συνέχεια G^1 , σχεδίαση γάστρας πλοίου.

Danksagungen

Viele Menschen haben mich in der Entwicklung dieser Dissertation unterstützt. Ich bin ihnen allen dankbar und möchte die Folgenden namentlich erwähnen:

Vor allem danke ich meinen Eltern Alfred und Maria, meiner gesamten Familie und meinen Freunden. Sie sind meine Grundlage, mein System, meine Realität. Ich teile mit ihnen die große Freude über den Abschluss dieses Werkes.

Dankbar erkenne ich den Rahmen, die finanziellen Mittel und die wunderbaren Möglichkeiten an, die mir vonseiten der National Technical University of Athens (NTUA) und durch das FP7-finanzierte Forschungsprojekt Exciting¹ zur Verfügung gestellt wurden.

Sehr herzlich danke ich meinen Betreuern Alexandros-Alvertos I. Ginnis, Bert Jüttler und Panagiotis D. Kaklis. Ihre fortlaufende Unterstützung in allen Facetten dieses Doktoratsstudiums war fundamental. Ich bin privilegiert, sie als meine Betreuer und Mentoren zu haben.

An Panagiotis Kaklis richtet sich meine tiefe Dankbarkeit für seine breit gefächerte Hilfe. Sein Rat und seine Kritik, seine Geduld und Offenheit gingen weit über das Professionelle hinaus. Ich danke Ihnen für alles!

Der kreative Austausch mit Alexandros Ginnis hat viele der Prinzipien dieses Werkes hervorgebracht. Ich bin ihm sehr dankbar für sein Engagement und sein spezielles Interesse an dieser Arbeit.

Bert Jüttler begleitet meine akademische Laufbahn schon seit dem Grundstudium an der Johannes Kepler Universität in Linz. Ich danke ihm besonders für die vielen Jahre voller professioneller und persönlicher Unterstützung.

Ich danke allen meinen Kollegen in unserem Exciting-Forschungsteam in Athen, nämlich Konstantinos A. Belibassakis, Theodoros P. Gerostathis, Konstantinos V. Kostas, Dimitris N. Mourkogiannis und Constantinos G. Politis. Ihre Rolle in meiner akademischen Entwicklung ist besonders bedeutsam und ihr Rat und ihre Kritik haben diese Arbeit von den fundamentalen Prinzipien bis zu den technischen Details sehr positiv beeinflusst.

Viele Kollegen, Mitstudenten, Freunde und Familienmitglieder sind mir darüber hinaus mit Korrekturlesen und Übersetzungen, mit Rat und Reflektion und mit aktivem Interesse zur Seite gestanden. Ich danke ihnen allen dafür und ich danke ihnen besonders für ihre positive Energie.

Voller Dankbarkeit weiß ich die Ehre zu schätzen, die es war in dem wunderbaren Land und in der Kultur der Hellenischen Republik zu leben, zu arbeiten und zu studieren. Ich danke den lebensfrohen und lebenswürdigen Menschen von Griechenland für ihre Filoxenía, dafür mich zu beherbergen in dieser einzigartigen Zeit, die ein Segen in meinem Leben war.

Ich danke Ihnen allen, ευχαριστώ!

Christian Feurer, Athen im Juli 2013.

¹Exciting – Exact geometry simulation for optimized design of vehicles and vessels: Ein Projekt im FP7-Framework der Europäischen Union (FP7- SST- 2007- RTD-1 - 218536).

Zusammenfassung

Diese Dissertation behandelt die Entwicklung eines parametrischen Modells eines Containerschiffsrumpfs für die spezifischen Anforderungen der isogeometrischen Analyse.

Im Konstruktionsprozess eines gegebenen geometrischen Objekts waren sein Design und die darauf durchgeführte Analyse traditionellerweise zwei getrennte Schritte. Eine Diskretisierung der Geometrie für die Anwendung numerischer Methoden war dabei ein unverzichtbarer Zwischenschritt, der teuer ist und der potenziell hohe Approximationsfehler birgt. Als eine Lösung dieses Problems haben Hughes et.al. im Jahre 2005 die *isogeometrische Analyse* vorgestellt, bei der die Basisfunktionen der Darstellung der zugrundeliegenden Geometrie für den Ansatz der numerischen Lösung des gegebenen Problems verwendet werden, siehe e.g. [50, 23]. Durch diese Methode wird jegliche Geometrie-Approximation vermieden und die Welten des rechnerunterstützten Designs (CAD) und der Analyse nahtlos zusammengeführt.

Diese Eigenschaft ist der primäre Antrieb für diese Arbeit, in der wir auf gänzlich transparente Art ein Modell eines *Containerschiffsrumpfes* entwickeln, in dem alle Designvorschriften, wie die Interpolation von Punkten und Glattheitsbedingungen, *exakt* erfüllt werden.

Um einen Schiffsrumpf, wie jedes Objekt, bezüglich eines gegebenen Kriteriums effizient optimieren zu können, muss ein Mechanismus für die automatische Geometrieerzeugung zur Verfügung gestellt werden. Der Aspekt der *parametrischen Modellierung* ist daher fundamental in dieser Arbeit. Wir konstruieren den Schiffsrumpf in Abhängigkeit von 29 globalen und lokalen Designparametern und von 9 zusätzlichen internen Parametern.

Für die geometrische Modellierung von Schiffsrümpfen ist die Verwendung eines Flächenmodells eine natürliche Wahl und dazu folgen wir einem lokalen Konstruktionsprozess, der auf einem Drahtgittermodell, d.h. einem Netzwerk von Kontrollkurven, basiert. Aus der adäquaten Konstruktion dieser Kurven gehen vorwiegend vierseitige Loops hervor, die mithilfe der bekannten Coons-Flächenkonstruktion „gefüllt“ werden. Um glatte Übergänge zwischen diesen Flächen zu garantieren, präsentieren wir einen neuartigen Algorithmus für die Konstruktion von orthogonalen Vektorfeldern. Diese erlaubt es uns, alle Kurven mit ihrem *Frame* zu erweitern, d.h. mit einer Verteilung von Normalvektoren der zugehörigen bewegten Tangentialebene. Diese Information wird zur Konstruktion der randübergreifenden Tangenten verwendet, die für bikubisch geblendete Coons-Flächen nötig sind. Auf diese Art erzeugen wir Flächen, die in ihren Rändern normal zu den Frames sind und somit glatt aneinander grenzen.

Die gewählte Darstellung der geometrischen Komponenten ist die der B-Splines, die als Standard im CAD gelten und die aufgrund ihrer direkten Verfeinerbarkeit gut geeignet sind für die isogeometrische Analyse. Die vorgestellten Algorithmen und ein möglicher Konstruktionsprozess des Drahtgittermodells wurden in C++ implementiert und in dem Programm SmoothPM zusammengefasst. Wir präsentieren die erzeugten Instanzen von 18 verschiedenen Parametersets und stellen Vorschläge für zukünftige Entwicklungsschritte vor.

Schlüsselwörter: CAD, CAGD, Frame, G^1 Stetigkeit, Glattheit, isogeometrische Analyse, isogeometrische BEM, parametrische Modellierung, Schiffsrumpf-Design.

Acknowledgements

Many people have supported me in the development of this dissertation. I am grateful to them all and I wish to name the following personally:

First and foremost, I thank my parents Alfred and Maria, my entire family and my friends. They are my foundation, my system, my reality. I share with them the great joy of finalizing this effort.

I gratefully acknowledge the framework, funds and wonderful opportunities provided to me by the National Technical University of Athens (NTUA) and through the FP7-funded research project Exciting¹.

I very dearly thank my advisory committee Alexandros-Alvertos I. Ginnis, Bert Jüttler and Panagiotis D. Kaklis. Their continuous support in all facets of this doctoral course has been fundamental. I am privileged for having them as my supervisors and mentors.

My deep appreciation is directed towards Panagiotis Kaklis for his universal help. His advice, criticism, patience and openness have expanded far beyond the professional. Thank you for everything!

The creative exchange with Alexandros Ginnis has produced many of the principles in this work. I am very grateful for his commitment and his particular interest in this thesis.

Bert Jüttler has guided my academic career since my studies at the Johannes Kepler University in Linz. I am distinctly thankful to him for the many years of professional and personal support.

I thank all of my colleagues in our Athens Exciting research group, namely Konstantinos A. Belibassakis, Theodoros P. Gerostathis, Konstantinos V. Kostas, Dimitris N. Mourkogiannis and Constantinos G. Politis. Their role in my academic development is particularly significant and their advice and criticism have positively influenced this work from its fundamental principles to its technical details.

Many colleagues, co-students, friends and family members have helped me with proof-reading and translations, with advice and reflection and with active interest. I thank them all for their support and I especially thank them for their positive energy.

Full of gratitude I recognise the honour which it has been to live, to work and to study in the wonderful land and culture of the Hellenic Republic. I thank the lively and lovely people of Greece for their *filoxenía*, for accommodating me in this unique experience which has blessed my life.

I thank you all, ευχαριστώ!

Christian Feurer, Athens in July 2013.

¹Exciting – Exact geometry simulation for optimized design of vehicles and vessels: A projekt in the FP7-Framework of the European Union (FP7- SST- 2007- RTD-1 - 218536).

Abstract

This dissertation presents the development of a parametric model of a container ship hull for the particular requirements of isogeometric analysis.

In the construction process of a certain geometric object the design of its shape and the analysis performed on it have traditionally been two separate steps. Discretisation of the geometry in order to enable numerical analysis methods has been an essential intermediate step which is costly and bears a potentially large approximation error. To address this issue, in 2005 Hughes et.al. introduced the concept of *isogeometric analysis* in which the basis functions of the underlying geometry's representation are adopted for the ansatz of the numerical solution of the given problem, see e.g. [50, 23]. In this way, any geometry approximation is bypassed and the worlds of computer-aided design (CAD) and analysis are integrated seamlessly.

This very quality marks the primary motivation for this thesis, in which we create the model of a *container ship hull* in an entirely transparent manner which *exactly* fulfils all design constraints, such as the interpolation of points and the demands on smoothness.

In order to efficiently optimise a ship hull, like any other object, with respect to a given criterion, a mechanism for the automated geometry generation has to be provided. The aspect of *parametric modelling* is, therefore, fundamental for this work. We create the mentioned ship hull model in dependence of 29 global and local design parameters and of 9 additional internal parameters.

It is natural to model ship hulls by surface modelling and for that we follow a local design approach based on a wireframe model, i.e. a network of control curves. Their proper construction yields principally quadrilateral curve loops which are “filled” by the means of the well-studied Coons patch construction. To guarantee smooth transitions between these patches, we present a novel algorithm to construct orthogonal vector fields, allowing us to enhance all curves with a representation of its tangent plane's normal vector distribution, the *frame*. This information is then employed in the construction of the cross-boundary ribbons of the bicubically blended Coons patch and we achieve to create patches which are normal to the frames in their boundaries.

The chosen representation of the geometric components is that of B-splines which is regarded as standard in CAD and which is suitable for isogeometric analysis due to its straight-forward refinement property. The presented algorithms and a possible design process of the wireframe geometry have been implemented in C++ resulting in the program SmoothPM. We present its evaluations with 18 different input parameter sets and provide suggestions for future development.

Keywords: CAD, CAGD, frame, G^1 continuity, isogeometric analysis, isogeometric BEM, parametric modelling, ship hull design, smoothness.

Contents

Preamble	i
Ευχαριστίες	i
Σύνοψη	ii
Danksagungen	iv
Zusammenfassung	v
Acknowledgements	vi
Abstract	vii
Contents	viii
I Introduction	1
I.1 The Integration of CAD, Analysis and Parametric Modelling	1
I.2 Task and Novelty of this Dissertation	4
I.3 Isogeometric Analysis	6
I.3.1 The Wave Making Resistance Problem	7
I.3.2 The Isogeometric Boundary Element Method	10
I.3.3 Optimisation Loop Concept	11
I.4 Design Parametric Modelling	12
I.5 The Container Ship	14
II Conceptual Ship Hull Design	19
II.1 Geometric Modelling Principles	19
II.1.1 Wireframe-Based Surface Modelling	19
II.1.2 A G^1 Continuous Patch Network	20
II.2 The Parameters of the Ship	21
II.3 Design in Catia TM	29
III Geometric Tools for the Model	33
III.1 Enhanced Interpolation	33
III.1.1 Hermite Interpolation	37
III.1.2 G^1 Piecewise Bézier Curve Construction	42
III.1.3 Interpolation Parameters and Basis	45
III.1.4 Approximation by a C^2 Curve	48
III.1.5 The Interpolation Algorithm	54
III.2 Enhancing Curves with Frames	54
III.2.1 Frame Conditions	55
III.2.2 The Rotation-Minimizing Frame	62
III.2.3 Optimisation of the Frame	63
III.2.4 The Framing Algorithm	69
III.3 From Frames to Ribbons	70
III.3.1 Cross-Boundary Ribbon Conditions	70

III.3.2	Compatibility and Vertex Twist Vectors	73
III.3.3	The Rotation-Minimizing Ribbon	76
III.3.4	Optimisation of the Ribbon	76
III.3.5	Ribbon Construction Algorithm	80
III.4	Coons Patch Construction	81
III.4.1	Bilinearly Blended Coons Patches	81
III.4.2	Bicubically Blended Coons Patches	82
IV	Construction Steps	89
IV.1	Three Stages of the Wireframe	89
IV.1.1	Stage I	91
IV.1.2	Stage II	99
IV.1.3	Stage III	104
IV.2	Ship Hull Patch Network	108
IV.2.1	Patches With Singularities	109
IV.2.2	Midship Patches	110
IV.2.3	Bow Patches	113
IV.2.4	Stern Patches	116
V	Model Instances	125
V.1	The Default Parameter Set	126
V.2	Global Parameter Changes	128
V.3	Midship Parameter Changes	131
V.4	Bow Parameter Changes	133
V.5	Stern Parameter Changes	136
VI	Implementation Notes	141
VI.1	Data Manipulations	141
VI.2	The Environment for the Task	141
VI.3	SmoothPM: An Implementation in C++	143
VI.3.1	The Main Objects	144
VI.3.2	The Construction Classes	146
VI.3.3	Design with SmoothPM	147
VII	Outlook and Future Work	149
A	B-Splines	153
A.1	Curves and Surfaces	153
A.2	Product of Bézier Expressions	155
	List of Figures	159
	References	161
	Index	167

Chapter I

Introduction

I.1 The Integration of CAD, Analysis and Parametric Modelling

Isogeometric Analysis: We consider the model of a certain geometric object. In order to determine its quality with respect to a particular physical, engineering or operational criterion we must perform analysis on it. This task typically results in solving one or more (partial) differential equations whose analytical solution is likely not to be available. The following poses an example to determine the unknown function $u(\mathbf{x})$ for a given linear differential operator L , a given function $f(\mathbf{x})$ and for all points \mathbf{x} in a given domain Ω of the d -dimensional space, compare to Figure I.1:

$$(Lu)(\mathbf{x}) = f(\mathbf{x}), \quad \mathbf{x} \in \Omega \subset \mathbb{R}^d. \quad (\text{I.1})$$

The subsequent paragraphs are based on the efforts by Hughes et.al. [50] (2005).

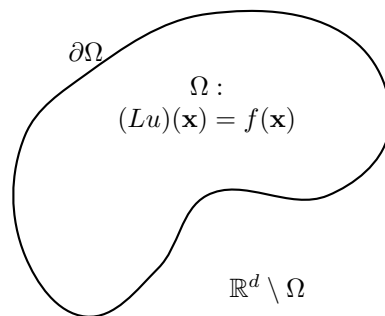


Figure I.1: Problem (I.1) on the domain Ω .

The traditional approach is to solve a possibly simplified problem on a mesh of suitably selected samples points \mathbf{x}_i of Ω , i.e. a *finite element method* (FEM), see Fig. I.2(a). Since its origins in Aerospace engineering in the 1950s and 1960s, this technique has been widely adopted to many scientific fields. However, the lack of integration with geometric design tools has been one particular shortcoming. Typically, the geometric object is created in an environment belonging strictly to the world of computer-aided design (CAD). The generation

of an analysis-suitable mesh for FEM is an intermediate step between geometry generation and analysis and its importance may not be underestimated as it commonly occupies 80% of the overall solving time. An equally unfavourable quality of this *discretisation* is the inevitable approximation error which can have rather large negative influence on the quality of the solution. Problems in fluid mechanics are typical examples with difficulties caused by an approximation error.

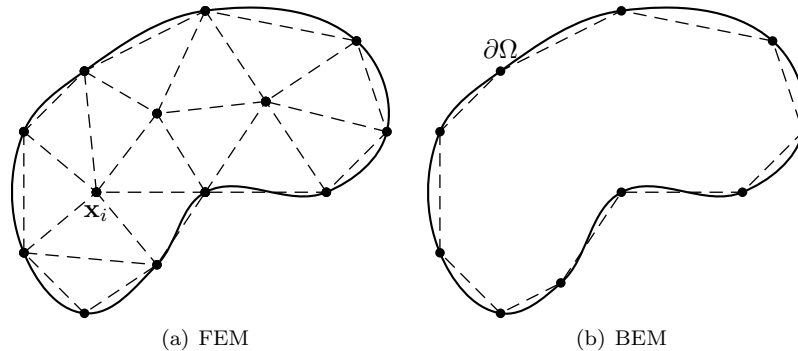


Figure I.2: Numerical approaches for Problem (I.1). (a): A coarse discretisation of the domain Ω for Finite Elements Methods. (b): A coarse discretisation of the boundary $\partial\Omega$ for Boundary Elements Methods.

The concept of *isogeometric analysis* (IGA) has its roots in the integration of the CAD world with analysis and achieves to avoid any error stemming from approximation. To this end, the basis functions which are used to generate the geometry Ω are adopted for the ansatz of the numerical solution for $u(\mathbf{x})$, thus substituting the traditional polynomial-based FEM. One commonly used geometry representation in CAD software packages is that of NURBS (non-uniform rational B-splines) which is particularly well suited for analysis as it allows straight-forward refinement of the function space.

Boundary Element Methods in IGA: Problems like (I.1) can be posed on the geometric domain Ω , as above, or on its complement, e.g. in three dimensional space $\Omega' := \mathbb{R}^3 \setminus \Omega$. They are typically given with conditions on the domain boundary, which we denote by $\partial\Omega$, and we refer to these cases as the inner and outer boundary value problem, respectively. For the details to this paragraph we refer to the literature: e.g. [53, 26, 49]. Both can be similarly transferred to an associated problem on $\partial\Omega$ by the means of the third Green's identity in combination with the corresponding fundamental solution $\gamma(x, y)$. In particular, this transformation yields an identity in the form of an integral equation for the Cauchy data u or $\frac{\partial u}{\partial n}$ on $\partial\Omega$, i.e. a *boundary integral equation* (BIE).

The first advantage of this approach is the reduction of dimension of the problem since then we solve a problem on $\partial\Omega$. The solution on Ω (or Ω') is then obtained by the corresponding *representation formula*. One difficulty associated with the BIE lies in the existence and explicit knowledge of the associated fundamental solution $\gamma(x, y)$. However, if the explicit fundamental

solution of the BIE is available, a numerical solution can be obtained by proper meshing of $\partial\Omega$, a technique referred to as *boundary element method* (BEM), see e.g. [67]. While this discretisation, compare to Fig. I.2(b), is simpler due to the reduced dimension than that of Ω , the main disadvantage lies in the character of the matrix of the associated system of linear equations which is no longer sparse as in the traditional FEM.

This dissertation is strongly linked to BEM since we regard the hull of a ship and the wave making resistance as its measure of quality. The evaluation of this resistance results in a Laplace equation on the infinite exterior of the ship's domain and Neumann conditions on the ship boundary. This problem, to which we have to add conditions on the free water surface, is of elliptic nature and its Green's function, i.e. the fundamental solution, is well studied. The approach of BIE and BEM is, therefore, applicable. We refer to following publications covering the undertaken research in the context of isogeometric BEM:

- Politis et.al. [75] (2009): "An isogeometric BEM for exterior potential-flow problems in the plane", provides a prove-of-concept of the isogeometric boundary element method discussing 2D model problems.
- The isogeometric BEM for the model problem of this very thesis, the wave making resistance problem, is examined in Belibassakis et.al. [9] (2009) for the case of a submerged prolated spheroid, and in Belibassakis et.al. [10] (2011) and [11] (2013), for surface piercing ship hulls like the Wigley hull and the Series 60 ship hull (with block coefficient $C_b=0.60$). In all cases, the isogeometric BEM method is benchmarked against analytical solutions (where available), experimental data and predictions provided by the low-order panel or other similar methods in the pertinent literature, illustrating the superior efficiency of the isogeometric approach.

Parametric Modelling for IGA: A typical problem in the industry is to optimise, under certain conditions, the shape of an object, i.e. its geometric domain Ω in the context of Problem (I.1). While the objective function provides a measure for the quality of a given geometry, efficient optimisation can only be performed if the automated generation of the geometry is possible. A device capable of creating the geometry for a given set of input parameters, i.e. a *parametric model* (PM), is the key to this task. It represents the bridge between low-level design parameters and the core specifications of the geometry. Its creation is, therefore, a difficult task for complex objects, especially if valid geometries are desired for large parameter domains, see Hoffmann [47] (2000).

In Ginnis et.al. [36] (2011), we have presented an isogeometric optimisation loop involving a PM of a container ship hull which we developed in the commercial software package CatiaTM, see Section II.3 for a brief review. While Catia provides vast functionalities for the construction of curves and surfaces, the employed algorithms are, to some extent, based on approximation. Furthermore, it is not possible to fully control the internally employed algorithms for the generation of the geometric entities. While certain tolerances may be specified and altered for most tools, the user of Catia has to generally take into account the "black box" character of its mechanisms. Statements about the continuous dependency of the result on the input parameters are, therefore, impossible to make for parametric models designed in this software package. This makes

Catia an imperfect environment for applications requiring the exact geometry representation, as does an isogeometric solver. In this thesis, we go about the generation of parametric ship hull exclusively employing entirely transparent machinery.

I.2 Task and Novelty of this Dissertation

Parametric Modelling in Ship Hull Design: We briefly overview the pertinent literature of three areas surface modelling, ship hull design and parametric modelling, see e.g. Boehm and Farin [15] (1984), Baer et.al. [6] (1989), Kim [52] (2004) and Nowacki [65] (2010).

The suitability of surface modelling techniques from the field of computer-aided geometric design (CAGD) for the design of ship *hulls* is apparent, see Lee [57] (1999) and (rational) B-splines are a commonly used format for the representation of curves and surfaces, see e.g. Piegl and Tiller [70] (1997) and Farin [32] (2002). They can have any desired piecewise continuity and degree, a characteristic which, in combination with their quality of easy manipulation through control points, accounts for their popularity. As for the construction of surfaces, tensor product patches are predominantly used to create surfaces of quadrilateral topologies. In this connection, the Coons patch construction is a technique to blend opposing boundaries of a quadrilateral curve loop [21]. Gordon-Coons patches are one generalisation to interpolate networks of curves [37] and Gregory patches [7] are another one addressing the issue of compatibility by introducing variable twist vectors.

For complicated topologies, as can have the hull of a ship, one single surface may not be generally suitable. A network of surface patches is typically employed raising the demand for continuity between patches. Significant research in this context was undertaken in the nineteen-eighties introducing the term *geometric* or *visual continuity*, see e.g. Farin [30] (1982), Boehm [16] (1988), and Du and Schmitt [25] (1990). The case of non-quadrilateral topologies was examined with different techniques by Sabin [79] (1983), Hosaka and Kimura [48] (1984), van Wijk [87] (1986) and Loop and DeRose [58] (1989).

Geometric modelling for ship hulls was first approached by Taylor [83] in 1905 and a first step towards *parametric* modelling was taken in 1950 by Lackenby [54] by the systematic variation of sectional area curves defined by explicit polynomials. With the aid of computers, geometric modelling itself has progressed rapidly since 1960, as has, therefore, ship hull design. Reed and Nowacki [76] (1974) created sections and lines interactively and Nowacki et.al. [63] (1977) used Coons patches for surface generation. Methods for the interactive generation of B-spline surfaces include Rogers and Satterfield [78] (1977) and Zhou and Liu [89] (1985).

The FRIENDSHIP-modelerTM by Harries, which has been developed since 1998, employs B-spline surface representations and links the parametric geometry generation with CFD¹ solvers and optimisation tools to propose fully-automated hydrodynamic optimisation: [43, 44, 1, 2, 3]. This modeller is restricted to the generation of a singular patch geometry unlike the work by H.C. Kim [52] (2004). He employs multiple domain surface representations and

¹Computational fluid dynamics

uses skinning and blending techniques for surface design, where spline approximation methods are applied. The model illustrates the validity of modelling by design parameters in the generation of complex ship hulls, a quality which is particularly favourable for optimisation studies: [45]. Recent efforts for the automatic generation of simple ship hulls have been made by Pérez-Arribas et.al. [68] (2006) and Cho et.al. [17] (2006) who achieved to construct a G^1 continuous ship hull using approximations of the wireframe and by Pérez and Clemente [69] (2011). In the work [77] by Rodríguez and Fernández-Jambrina (2012) the concept of programmed design is introduced with an implementation, allowing the user to design by employing specific knowledge and experience.

Note, that the role of parametric modelling in ship design has been explicitly pointed out by Nowacki in his review article [65] of 2010, namely “Five Decades of Computer-Aided Ship Design”, where it is listed as “one of the most essential contemporary reorientations”.

Novelty of this Thesis: One fundamental characteristic of *isogeometric analysis* is that it avoids any geometry approximation, a quality which marks the motivation of this very thesis:

We address the task to create a parametric model of the hull of a container ship in which all design constraints, such as the interpolation of points and the desired geometric continuity, are fulfilled *analytically*, i.e. exactly from a geometric point-of-view. To this end, we develop novel algorithms for the interpolation of point, tangent and zero-curvature data and for the enhancement of curves with a distribution of tangent planes. We achieve to cover the wetted part of the ship hull by surface patches of tensor product format and free of surface-trimming, therefore suitable for standard techniques of (isogeometric) analysis.

The use of entirely transparent machinery for the implementation of this model enables full control over these techniques and provides the possibility for any desired manipulation and further refinement, an essential requirement for the predictability of results.

Furthermore, we determine a set of ship-characteristic design parameters which governs the parametric model.

Formulation of the Task: In this connection, the task of this thesis can be stated as follows:

Development of a parametric model which provides analytically G^1 continuous container ship hulls for a wide range of values of a reasonably-sized set of design parameters.

Methodology: The design of the ship hull takes place on the level of the wireframe for which we develop a custom curve interpolation algorithm allowing to interpolate points, tangents and parameter values with zero curvature. The construction of the wireframe yields a mesh of quadrilateral and triangular C^2 continuous curve loops bounding the wetted part of the ship hull.

In a second design step, each curve is enhanced with a vector field, its *frame*, which is orthogonal to the distribution of curve tangents. Interpreting this vector field as the distribution of normal vectors of a moving plane, we determine

the distribution of tangent planes along the curve. In this way, we provide continuity information across all curves which is used for the construction of the *boundary ribbons* of the bicubically blended Coons patches. Triangular curve loops are treated as degenerate quadrilateral ones and the frames formally define the missing normal vector at the generated singularities.

The use of B-spline for the curve and frame representation allows us to exactly fulfil the orthogonality conditions of the frame to the curve's tangent. Also in this way, we achieve to represent all patches in a standard tensor product format.

Outline of this Thesis: In Chapters II, III and IV we provide the conceptual ship hull design strategy, the necessary geometric machinery and the step-by-step construction for the automated creation of this complex ship hull geometry, respectively. Results of C++ program SmoothPM can be found in Chapter V and we conclude this dissertation with notes on the implementation and possible future development.

The current introductory part of the thesis continues with reviewing the concept of isogeometric analysis in Section I.3 as it motivates our very core requirement. The details of parametric modelling are reviewed in Section I.4 and we close this chapter by providing the necessary context, input and constraints for container ship parametric modelling in Section I.5.

I.3 Isogeometric Analysis for the Wave Making Resistance Problem

The investigation of ship resistance holds significant importance for ship powering predictions and optimisation of ship hulls. Experiments reveal the resistance's high sensibility to principal design parameters which makes this problem particularly well-suited for the development of a parametric ship hull model. In this section we review the problem of wave making resistance as stated in Baar and Price [5] (1988) and present the isogeometric method to obtain the solution in the context of boundary elements following the approach taken in Belibassakis et.al. [11] (2013). For background on the wave resistance and related problems we refer to the literature: see e.g. Pironneau [72] (1989) and Bertram [12] (2000).

Ship resistance depends on viscous and gravitational effects and is, therefore, a rather complicated problem. In the idealized setting of steady ship motion in ideal sea water, which implies an incompressible and inviscid fluid, and with irrotational flows the effects of viscosity can be neglected. Following Froude's hypothesis on the separation of viscous resistance and wave making resistance, it is legitimate to be concerned with the isolated minimization of the ship's *wave making resistance*. This, in turn, has growing influence on the overall resistance as motion speed increases and at high speeds it can contribute more than 50 percent of the overall resistance.

The problem is formulated in the following Subsection I.3.1 and a numerical solution by boundary elements is found in Subsection I.3.2. In that context, the concept of isogeometric analysis is employed, thus motivating an exact geometry representation of the ship body boundary. We close this section with Subsection I.3.3 by an outline of an optimisation loop employing a parametric

ship model and an isogeometric solver.

I.3.1 The Formulation of the Problem

We consider the case of an ideal, i.e. inviscid and incompressible, fluid with irrotational flows. Its surface is pierced by an object, which in our case is a ship hull, in constant motion in direction \mathbf{V} . Regarding [5] as our principal reference, for the following paragraphs, we employ potential flow theory and uniquely define the irrotational flow through the velocity potential function. The conditions on this potential function yield the Laplace equation with additional constraints stemming from the free fluid surface. Representing the unknown potential as an integral of a density function multiplied with an appropriate kernel over the body boundary reduces the problem's dimension and enables the machinery of boundary element methods and furthermore that of isogeometric analysis, see Section I.3.2. A solution for the density function yields the velocity potential from which the wave making resistance is derived.

The Velocity Potential: We introduce the scalar velocity potential $\Phi(\mathbf{x})$ in the convenient decomposition into the uniform parallel flow $\mathbf{V} \cdot \mathbf{x}$ and the disturbance potential $\varphi(\mathbf{x})$:

$$\Phi(\mathbf{x}) := \mathbf{V} \cdot \mathbf{x} + \varphi(\mathbf{x}). \quad (\text{I.2})$$

The originally desired flow velocity field $\mathbf{U}(\mathbf{x})$ is then given by the gradient of $\Phi(\mathbf{x})$:

$$\mathbf{U}(\mathbf{x}) = \nabla\Phi(\mathbf{x}) = \mathbf{V} + \mathbf{u}(\mathbf{x}) \quad (\text{I.3})$$

with the perturbed flow velocity field $\mathbf{u}(\mathbf{x}) := \nabla\varphi(\mathbf{x})$.

Based on the fact that, for our purposes, we can regard the movement to be constant at the fixed sea level, i.e. $z = 0$, we set, without loss of generality, $\mathbf{V} = (-V, 0, 0)^T$.

The Laplace Formulation: The elementary physical principal of *conservation of mass* implies that we have zero flux across the boundary $\partial\Omega$:

$$0 = \int_{\partial\Omega} \nabla\Phi \cdot \mathbf{n} dS = \int_{\Omega} \text{div}\nabla\Phi d\mathbf{x},$$

with the boundary element S , the hull's normal vector \mathbf{n} and the Gauss-Green Theorem justifying the second equality. Since this holds for all smooth domains Ω , we get

$$\text{div}\nabla\Phi = \text{div}(\mathbf{V} + \nabla\varphi) = \Delta\varphi = 0, \quad (\text{I.4})$$

i.e. the Laplace equation. With $\mathbf{n}(\mathbf{x})$ being the normal vector, the normal speed at the body boundary $\partial\Omega$ is determined by $-\mathbf{V} \cdot \mathbf{n}$ which implies the following body boundary condition:

$$\frac{\partial\varphi}{\partial\mathbf{n}} = g(\mathbf{x}) := -\mathbf{V} \cdot \mathbf{n} = Vn^x, \quad \mathbf{x} \in \partial\Omega. \quad (\text{I.5})$$

Equations I.4 and I.5 constitute the classical *Laplace* problem with *Neumann* boundary conditions.

Free-Surface Conditions: We have to incorporate the unknown free-surface $z = \eta(x, y)$ and the three conditions stated below. To linearise these equations according to the theory of infinitesimal waves, we neglect the components of disturbance potential $\varphi(\mathbf{x})$ which are small with respect to the uniform parallel flow $\mathbf{V} \cdot \mathbf{x}$, if they occur in products and squares. In addition to that, the resulting equations are applied on the free surface $z = 0$ instead of the $z = \eta(x, y)$. The conditions are the following:

1. The *kinematic condition* states that the water does not penetrate the water surface. Therefore, on the free surface the flow velocity has to be tangential, i.e.:

$$\Phi_x \eta_x + \Phi_y \eta_y + \Phi_z = 0,$$

and linearisation yields

$$V \eta_x + \Phi_z = 0. \quad (\text{I.6})$$

2. For the *dynamic condition*, which states that pressure on the free surface has to be constant, i.e. we have to fulfil

$$g\eta - V\varphi_x + \frac{1}{2}\|\nabla\varphi\|^2 = c$$

for a constant c . With linearisation we obtain the following equation:

$$g\eta - V\varphi_x = 0. \quad (\text{I.7})$$

3. The fact that the disturbance potential decays at infinity and there are no upstream waves is expressed by *radiation condition* which ensures the existence and uniqueness of the disturbance potential:

$$\varphi(\mathbf{x}) = \begin{cases} O(\frac{1}{|\mathbf{x}|}) & \mathbf{U} \cdot \mathbf{x} > 0, |x| \rightarrow \infty \\ o(1) & \mathbf{U} \cdot \mathbf{x} < 0, |x| \rightarrow \infty. \end{cases} \quad (\text{I.8})$$

Equations (I.6) and (I.7) are combined to the following:

$$\varphi_{xx} + \frac{g}{V^2}\varphi_z = 0, \quad z = 0, \quad (\text{I.9})$$

and we refer to the problem of Equations (I.4), (I.5), (I.8) and (I.9) as the “Neumann-Kelvin Problem”.

The Boundary Integral Equation: In accordance with potential theory, the disturbance potential $\varphi(\mathbf{x})$ as in the following single-layer ansatz with the unknown density function $\mu(\mathbf{x})$, solves the Laplace equation:

$$\varphi(\mathbf{x}) = \int_{\partial\Omega} \mu(\boldsymbol{\xi})G(\mathbf{x}, \boldsymbol{\xi})d\boldsymbol{\xi} + \frac{1}{k} \int_{\ell} \mu(\boldsymbol{\xi})G^*(\mathbf{x}, \boldsymbol{\xi})n^x(\boldsymbol{\xi})\boldsymbol{\tau}^y(\boldsymbol{\xi})d\ell(\boldsymbol{\xi}), \quad (\text{I.10})$$

where ℓ is the waterline of the wetted body boundary $\partial\Omega$ and $\boldsymbol{\tau} = (\tau^x, \tau^y, \tau^z)^T$ with $\tau^z = 0$ denotes the tangent vector along the waterline ℓ . The characteristic wave number $k = g/V^2$ is directly related to the squared inverse of the corresponding Froude number $F = V/\sqrt{gL}$, L being the maximum body length, and

controls the wavelength of the transverse ship waves. The associated Green's function $G(\mathbf{x}, \boldsymbol{\xi})$ is the following:

$$4\pi G(\mathbf{x}, \boldsymbol{\xi}) = \frac{1}{\|\mathbf{x} - \boldsymbol{\xi}\|} - \frac{1}{\|\mathbf{x} - \boldsymbol{\xi}'\|} + G^*(\mathbf{x}, \boldsymbol{\xi}), \quad \boldsymbol{\xi} \in \partial\Omega, \mathbf{x} \in \mathbb{R}^{3-}. \quad (\text{I.11})$$

\mathbb{R}^{3-} denotes the lower half space $\{x, y, z \leq 0\}$, $\boldsymbol{\xi}'$ is the mirror of $\boldsymbol{\xi}$ with respect to the undisturbed free surface $z = 0$ and $G^*(\mathbf{x}, \boldsymbol{\xi})$ represents the regular part of the Green's function which is of exponential decaying and wavelike components.

With the Green's function $G(\mathbf{x}, \boldsymbol{\xi})$, the disturbance potential $\varphi(\mathbf{x})$ as in (I.10) not only directly solves the Laplace equation but also automatically fulfils the free-surface condition (I.9) and the conditions at infinity (I.8). The Neumann-Kelvin problem reduces to solving (I.5) which can be done with the help of the jump relation for, in this case, single-layer potentials, see [26, 49]:

$$\begin{aligned} \frac{\partial}{\partial \mathbf{n}}(\varphi(\mathbf{x})) &= -\frac{1}{2}\mu(\mathbf{x}) \\ &+ \int_{\partial\Omega} \mu(\boldsymbol{\xi}) \frac{\partial G(\mathbf{x}, \boldsymbol{\xi})}{\partial \mathbf{n}(\mathbf{x})} dS(\boldsymbol{\xi}) \\ &+ \frac{1}{k} \int_{\ell} \mu(\boldsymbol{\xi}) \frac{\partial G^*(\mathbf{x}, \boldsymbol{\xi})}{\partial \mathbf{n}(\mathbf{x})} n^x(\boldsymbol{\xi}) \tau^y(\boldsymbol{\xi}) d\ell(\boldsymbol{\xi}) \\ &= g(\mathbf{x}), \quad \mathbf{x} \in \partial\Omega. \end{aligned} \quad (\text{I.12})$$

Derived Quantities: A solution for $\mu(\mathbf{x})$ of Eq. (I.12) yields, in combination with Eq. (I.10), the disturbance potential $\varphi(\mathbf{x})$ and, therefore, the overall perturbed flow velocity field $\mathbf{U}(\mathbf{x})$ is obtained by Eq. (I.3). Along with the overall flow velocity several quantities can be directly derived from $\varphi(\mathbf{x})$, compare e.g. to [60]. Let p_∞ be the ambient reference pressure, ρ the fluid density and g the gravitational acceleration. Then:

1. The *pressure* p is the following:

$$p = p_\infty + \frac{\rho}{2}(V^2 - \|\mathbf{u}\|^2) - \rho g z.$$

2. The *free surface elevation* is obtain by

$$\eta(x, y) = \frac{V}{g} \varphi_z(x, y, 0).$$

3. The *pressure coefficient* C_p is the following

$$C_p = \frac{p - p_\infty}{\frac{1}{2}\rho V^2} = 1 - (\|\mathbf{u}\| - 2gz/V)^2.$$

4. The *wave resistance* R_W is finally obtained by the integration of the pressure on the wetted surface of the body:

$$R_W = \frac{1}{|\partial\Omega|} \int_{\partial\Omega} C_p n_x dS$$

with the area of the wetted hull $|\partial\Omega|$.

5. The *wave resistance coefficient* C_W is derived as follows:

$$C_W = \frac{R_W}{\frac{1}{2}\rho V^2 |\partial\Omega|}. \quad (\text{I.13})$$

I.3.2 The Isogeometric Boundary Element Method

To solve the boundary integral equation in (I.12) for the disturbance potential's density function $\mu(\mathbf{x})$, we observe the reduced physical dimension of this formulation: While in the traditional formulation of the Neumann-Kelvin Problem by the Laplace equation we had to solve for $\mathbf{x} \in (\mathbb{R}^3 \setminus \Omega) \subset \mathbb{R}^3$, the problem in (I.12) is one for all \mathbf{x} on the boundary of the domain: $\partial\Omega$. In our case, the body boundary is the composition of tensor-product surface patches. We, therefore, pose the problem on each patch as in Eq. (I.14) below. Since we only model the wetted part of the hull, the parametrization of each surface patch $\mathbf{S}(u, v)$ can be assumed to be in a way such that it coincides with the waterline at the parameter $v = 0$. For fully submerged patches, the term with the waterline integral is 0.

$$\begin{aligned} & \frac{1}{2}\mu(\mathbf{S}(u, v)) \\ & - \int_0^1 \int_0^1 \mu(\mathbf{S}(\bar{u}, \bar{v})) \frac{\partial G(\mathbf{S}(u, v), \mathbf{S}(\bar{u}, \bar{v}))}{\partial \mathbf{n}(\mathbf{S}(u, v))} m(\bar{u}, \bar{v}) d\bar{u} d\bar{v} \\ & - \frac{1}{k} \int_0^1 \mu(\mathbf{S}(\bar{u}, 0)) \frac{\partial G^*(\mathbf{S}(u, v), \mathbf{S}(\bar{u}, 0))}{\partial \mathbf{n}(\mathbf{S}(u, v))} n^x(\mathbf{S}(\bar{u}, 0)) \tau^y(\mathbf{S}(\bar{u}, 0)) m(\bar{u}, 0) d\bar{u} \\ & = -g(\mathbf{S}(u, v)), \quad u, v \in [0, 1]. \end{aligned} \tag{I.14}$$

with $m(u, v)$ being the surface metric tensor determinant:

$$m(u, v) = \left\| \frac{\partial \mathbf{S}(u, v)}{\partial u} \times \frac{\partial \mathbf{S}(u, v)}{\partial v} \right\|. \tag{I.15}$$

Approximation of the Solution: The isogeometric approach represents the principle to express the unknown density function $\mu(\mathbf{x})$ in Equation (I.12) by the same representation as the domain boundary. It was initially proposed by Hughes et. al. in the context of Finite Element Method: [50, 22, 51, 23]. We review the application of this concept to the solution of a boundary integral equation as can be found in Belibassakis et.al. [11] (2013) in the setting of boundary element method (BEM, see e.g. París and Cañas [67], 1997).

We consider the geometry given in B-spline representation, which is standard for geometry in computer-aided design. Each patch can, therefore, be written as follows:

$$\mathbf{S}(u, v) = \sum_{i=0}^{n_u} \sum_{j=0}^{n_v} N_i^{p_u}(u) N_j^{p_v}(v) \mathbf{S}_{ij}$$

with degrees p_u and p_v , the knot vectors which result in the basis functions $N_i^{p_u}(u)$ and $N_j^{p_v}(v)$ and the control points \mathbf{S}_{ij} . Guided by the isogeometric principle, we approximate the unknown $\mu(u, v) := \mu(\mathbf{S}(u, v))$ by $\tilde{\mu}(u, v)$ which adopts the geometry's bases:

$$\tilde{\mu}(u, v) = \sum_{i=0}^{n_u} \sum_{j=0}^{n_v} N_i^{p_u}(u) N_j^{p_v}(v) \mu_{ij}. \tag{I.16}$$

The $n_u \times n_v$ control values μ_{ij} are, therefore, the unknowns of $\tilde{\mu}(u, v)$ which is formally the projection of the solution $\mu(u, v)$ into the space \mathcal{S}^0 of functions

that can be represented in B-spline form with the bases of the geometry. With this projection, Eq. (I.12) becomes:

$$\begin{aligned}
 & \frac{1}{2} \sum_{i,j=0}^{n_u, n_v} N_i^{p_u}(u) N_j^{p_v}(v) \mu_{ij} \\
 & - \sum_{i,j=0}^{n_u, n_v} \mu_{ij} \int_0^1 \int_0^1 N_i^{p_u}(u) N_j^{p_v}(v) \frac{\partial G(u, v; \bar{u}, \bar{v})}{\partial \mathbf{n}(u, v)} m(\bar{u}, \bar{v}) d\bar{u} d\bar{v} \\
 & - \sum_{i=0}^{n_u} \mu_{i,0} \int_0^1 N_i^{p_u}(u) \frac{\partial G^*(u, v; \bar{u}, 0)}{\partial \mathbf{n}(u, v)} n^x(\mathbf{S}(\bar{u}, 0)) \tau^y(\mathbf{S}(\bar{u}, 0)) m(\bar{u}, \bar{v}) d\bar{u} \\
 & = -g(u, v), \quad u, v \in [0, 1], \tag{I.17}
 \end{aligned}$$

where the $G(u, v; \bar{u}, \bar{v}) := G(\mathbf{S}(u, v); \mathbf{S}(\bar{u}, \bar{v}))$, $\mathbf{n}(u, v) := \mathbf{n}(\mathbf{S}(u, v))$ and $m(u, v)$ is the surface metric tensor determinant as above.

To solve Equation (I.17) for the unknown μ_{ij} we can employ the collocation method and solve the system at the $n_u \times n_v$ determined parameter values (u_i, v_j) . Given a suitable scheme for the evaluation of the integral expressions this yields a system of $n_u \times n_v$ linear equations for as many unknowns. Its solvability depends on the locations of the collocation parameters and is guaranteed if Greville abscissae are employed: [32]. A solution of (I.17) yields, therefore, with Eq. (I.10) the approximated solution of $\varphi(\mathbf{x})$:

$$\tilde{\varphi}(\mathbf{x}) = \int_{\partial\Omega} \tilde{\mu}(\boldsymbol{\xi}) G(\mathbf{x}, \boldsymbol{\xi}) d\boldsymbol{\xi} + \frac{1}{k} \int_{\ell} \tilde{\mu}(\boldsymbol{\xi}) G^*(\mathbf{x}, \boldsymbol{\xi}) n^x(\boldsymbol{\xi}) \tau^y(\boldsymbol{\xi}) d\ell(\boldsymbol{\xi}). \tag{I.18}$$

Refinement of the Approximation Space: A better solution can be achieved by typical space refinements like knot insertion and degree elevation, which are, in the context of finite elements, often referred to as h - and p -refinement, respectively. In this way, we obtain more degrees of freedom to represent the solution. The dimensions of these spline spaces depend on the cardinalities of the knot vectors, i.e. on the number of inserted knots in the case of h -refinement. As above, let \mathcal{S}^0 be the space of functions which can be expressed by in B-spline form with the bases of the original surface patch. For a constant degree, with iterated insertion of knots we obtain a sequence of nested spaces denominated as \mathcal{S}^i :

$$\mathcal{S}^i \subset \mathcal{S}^{i+1}.$$

The fundamental advantage of isogeometric analysis is that in all these refinement steps the geometry remains unchanged. For any numerical strategy used to solve for μ_{ij} we, therefore, guarantee that no geometry approximation error is produced.

For a closer analysis of the problem, its solution by collocation, numerical integration and experiments we refer to [75, 9, 10, 11].

I.3.3 Optimisation Loop Concept

As we have stated in the beginning of this section, it is common and important to minimize the wave resistance coefficient C_W , Eq. (I.13), of a ship hull. This, however, is only possible effectively if we can integrate a solver for the

wave resistance into an optimisation loop with a parametric model of the geometry and the optimiser itself. Ginnis et.al. [36] (2011) present such a loop for the optimisation of the bulb of a container ship. It employs an isogeometric solver as presented in the above Subsections I.3.1 and I.3.2 and the parametric model to produce the geometry of the ship hull was developed in CatiaTM as we review in Section II.3. As for the optimiser, the software package ESTECO modeFrontierTM was employed which uses the following algorithms:

- (μ, λ) -ES and CMA-ES: semi-deterministic evolution strategies suited for multi-modal problems: [59, 42],
- Multi-Directional Search algorithm: deterministic pattern-search method well suited to convex noisy problems: [24].

Figure I.3 depicts the concept of an optimisation loop as described: The optimiser starts the cycle by sending an initial parameter set to the modeller which, in turn, produces the geometric representation of the object. This is sent to the solver and upon successful evaluation, the objective function value, which is here the wave resistance coefficient C_W , is returned to the optimiser which determines a new parameter set and a new cycle starts.

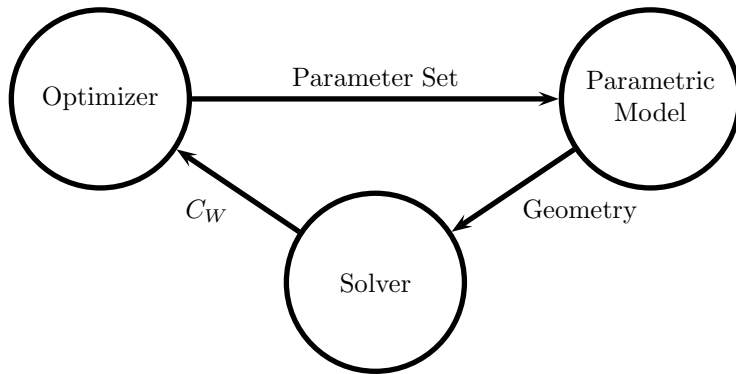


Figure I.3: Schematic diagram of an optimization loop.

Bow Optimisation of a Container Ship: The above mentioned effort tested the optimisation of the bow area of a container ship against the criterion of minimal wave resistance. With given principal ship characteristics, the four bow parameters “bulb length”, “bulb radius”, “bulb rise” and “bulb top position” were chosen to be the design parameters in the optimisation loop. The bounds for each of these parameters are implemented using a penalisation approach while the additional displacement constraint is fulfilled in the construction of the PM. The value of the wave resistance could be reduced by 11.5% and the conclusion was drawn that the length and radius of the bulb hold superior significance with respect to the others.

I.4 Design Parametric Modelling

Simple geometric objects can be defined by few parameters, e.g. a rectangle by its side lengths a and b . In such a task we likely start to create a straight base

line segment of length a . The infinite extension of this line marks the separation of two half planes. In one of the two we create two other straight lines of length b , one from each vertex of the first line and both forming a right angle to it. One final straight line connecting the ends of the two previous lines concludes geometry, compare to Figure I.4.

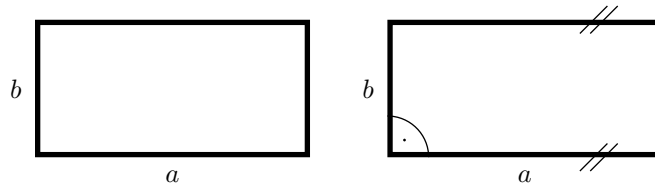


Figure I.4: Left: Model of a rectangle, parametrized by its side lengths a and b . Right: Fundamental assumption of right angles and the resulting parallel opposite sides.

Note that in this construction we have used more information than just two lengths. The intrinsic meaning of the term “rectangle” provides us with the information that all sides are perpendicular. Even more essential is the existence of *four* lines forming a closed loop. These implicit pieces of information are fundamental for the implementation and become more interconnected and involved as the geometry grows in complexity. Well-posed and properly linked design parameters are often subtle yet always a fundamental first step towards constructing a stable parametric model. In this way, a parametric model enables a design by high-level parameters and liberates the designer from handling a high number of low-level parameters with little influence. The character and selection of the design parameters are the content of the following paragraphs.

Dimensionless Parameters: We consider a partition of the above rectangle in Figure I.4. The subdivision shall take place along the base line, providing one part of length a_1 ; see Figure I.5. If we want to model the totality of the two resulting smaller rectangles, we are well advised to impose boundaries for a_1 . Unless the physical or philosophical circumstances permit otherwise, it is only reasonable to demand $a_1 \in (0, a)$ by which we guarantee to obtain non-degenerate sub-rectangles.

In many cases, it is convenient to employ dimensionless parameters representing ratios of lengths. In this example, we can introduce r_1 as the ratio of a_1 to a : $r_1 := a_1/a$. The validity of r_1 , and therefore that of a_1 , is easily checked as $r_1 \in (0, 1)$ yields correct results.

As parameters become more nested, this advantage is emphasized as the optimiser’s task remains to bluntly guarantee values between 0 and 1. Consider, therefore, the further subdivision of the second part of the rectangle at the parameter a_2 . Without ratios, we should firstly check that $a_1 \in (0, a)$ and, a posteriori, that $a_2 \in (a_1, a)$. Dependence of one parameter on another is not at hand if we employ $r_1, r_2 \in (0, 1)$ in combination with a and obtain $a_1 = r_1 a$ and $a_2 = a_1 + r_2(a - a_1)$. All parameters, a , r_1 and r_2 , can then be defined independently and their natural bounds, $r_1, r_2 \in (0, 1)$, yield valid values for a_1 and a_2 .

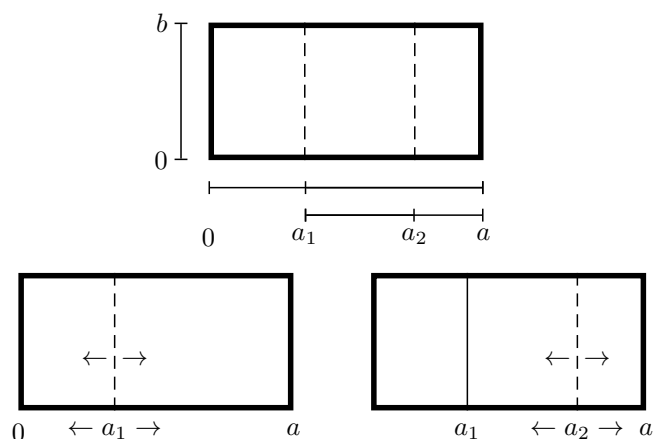


Figure I.5: Parametrization of the partitioning of a rectangle. We express a_1 by its ratio r_1 with respect to a : $a_1 = r_1 a$. Similarly, a_2 is defined by r_2 : $a_2 = a_1 + r_2(a - a_1)$.

Selection of Parameters: There are more possibilities to model our original rectangle from Figure I.4. Instead of giving the principal side lengths as parameters, we can alternatively give the length a of one side and determine the length of the other side by a ratio r_b : $b = r_b \cdot a$. Such an approach to modelling the geometry is advantageous if similar rectangles are to be created as then r_b has to be simply kept constant. If the designer plans to specify lengths at hand by which to construct the rectangle, the original modelling approach is better suited. It is the final purpose that determines the best way to parametrize the geometry, as it is the case for any model. Obviously, in all cases we have two degrees of freedom.

In the much larger scale of our model of the container ship hull a virtually countless number of specifications, such as control points and knot vectors in a possible B-spline geometry representation, determine the final shape. The difficulty lies in linking these low-level specifications to descriptive parameters of the general shape of the hull and, thus, deliberating the designer from tasks like the proper placement of control points for curves and surfaces. The routines in Chapter III provide this bridge in the form of an interpolation algorithm and three steps from a set of curves to a G^1 continuous surface patch network.

I.5 The Container Ship

A parametric model is developed to cover *one* category of objects. In our case, the model should yield valid hulls of *container ships*, see Figure I.6. It would go beyond the scope of this thesis to present the numerous characteristics of such a ship. For this reason, we refer to Watson [86] (1998), Eyres [28] (2001) and Letcher [55] (2009) for details to the following paragraphs where we present an outline of the ship terms and characteristics which we consider fundamental for the construction of our container ship parametric model. Compare to Figure I.8.

It is convenient to partition the ship lengthwise in three areas, namely the following:



Figure I.6: The Eleonora Maersk Container Ship, IMO 9321500.
Source: commons.wikimedia.org.

1. We refer to the front part of the ship as **bow**.
2. The rear part of a ship is called **stern**.
3. The transitional part between the bow and the stern is referred to as **midship**.

By regarding the ship bounding box, we get a rather good idea of her principal dimensions and form. In this respect, the ship length, breadth and depth are the main dimensions:

4. The ship **overall length** measures the distance in *longitudinal* direction between the extreme points at the forward and aft part of the ship.
5. The **breadth** measures the ship's greatest width. Its direction shall be called the *transverse* direction.
6. The **depth** measures the ship height at the side of the middle of the ship length, i.e. the *vertical* distance from her baseline (the "keel") to the freeboard deck.

The following global terms are associated with many types of ships:

7. The ship **waterline** is the horizontal curve that marks the intersection of the ship hull with the undisturbed free water surface. This line is often referred to as summer load line. We do not go into the dependency of the waterline on the ship load and regard only *one* waterline.
8. The vertical distance from the baseline of the ship to the summer waterline is called **draft**.
9. The **aft-perpendicular (AP)** is the vertical line at centreline of the rudder stock.
10. The **forward-perpendicular (FP)** is the imaginary line extending vertically from the point at which the profile of the ship front part is intersected by the waterline.

11. The **length between perpendiculars (LPP or LBP)** is the distance between the AP and the FP. It is often used, generally by merchant ships, instead of the overall length as a means to measure the ship length.

On a more detailed level, we introduce the following local terms (from bow to stern):

12. The front part of the ship which, on the level of the deck, extends forward beyond the FP is called **forward overhang**. In our model, the fore-most part of the deck has no width, i.e. it shrinks to a point.
13. As opposed to a conventionally shaped bow, in a **bulbous bow** a round extension is placed in longitudinal direction onto the transition between the midship and the bow, compare to Figure I.7. Its level is usually below the water surface and its purpose is to decrease the ship's wave resistance.



Figure I.7: A bulbous bow with a complex shape. Source: commons.wikimedia.org.

14. The midship area of container ship hulls is usually a surface created from an extruded section curve. This portion of the hull shall be called **parallel midship**.
15. The flat part of the midship is not limited to the parallel midship. The totality of the flat expansion on the side is called **flat-of-side (FOS)** or simply **flat side**.
16. Analogously, we refer to the flat part on ship's bottom as **flat-of-bottom (FOB)** or **flat bottom**.
17. We restrict the model to cover container ships with only one propeller which finds its placement on the **shaft** of the stern.

18. The type of stern used in this ship model is that of a **transom** stern. As opposed to a round stern, it is characterised by a flat surface behind the AP and above the propeller, bounding the ship in the back. Its width is a fraction of the overall ship breadth.

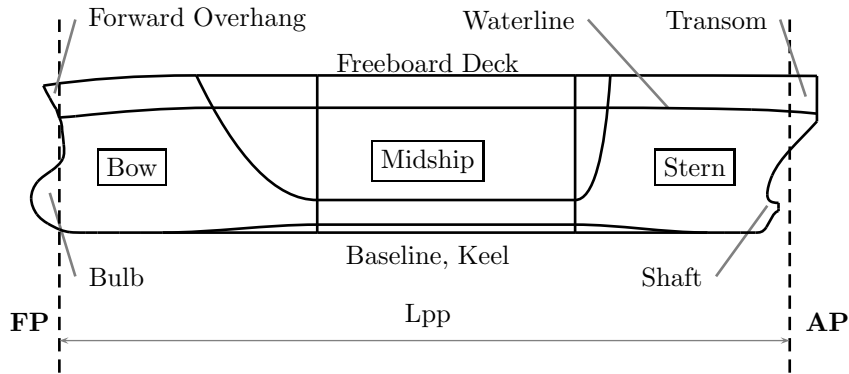


Figure I.8: Principal characteristics of a container ship.

Parent Ships: Specifying the afore-mentioned geometric characteristics still leaves a broad band of possible interpretations for the hull geometry of a container ship. We shall, therefore, use a pair of parent hulls, available in pertinent literature, for building our parametric model.

The first parent hull is referred to in the literature as the KRISO container ship (KCS) [19]. Its principal dimensions are LPP $7.2786m$, draft $0.3418m$, wetted surface $9.4379m^2$ and model speed $2.196m/s^2$, i.e. this ship model is not of real-world dimensions.

The second parent hull, referred to as the MS Exciting container ship, has its roots in commercially interesting ships and was modified and provided by the ship classification society Stiftelsen Det Norske Veritas (DNV) for the Work Package 2 of Exciting [29]. Its principal dimensions are LPP $277m$, breadth $32.2m$ and draft $13m$.

Chapter II

Conceptual Ship Hull Design

This chapter establishes the design strategy for the model, both from a geometric as well as from a parametric point-of-view. We decide on the geometric modelling approach in Section II.1, and, subsequently, the set of parameters for the model is presented in Section II.2. A review of the development of such a model in CatiaTM concludes this chapter: Section II.3.

II.1 Geometric Modelling Principles

The geometry of the ship hull shall be constructed by a surface modelling approach. The advantages of such an approach and its suitability for our purposes are covered in Subsection II.1.1. The demand for smoothness motivates Subsection II.1.2, where we preview the construction of surface patch networks with tangent plane continuity.

The Wetted Part of the Hull: In the context of an optimisation loop to minimize the wave making resistance C_W , see Section I.3, only the wetted part of the hull is relevant and shall, therefore, be constructed. In this way we avoid creating trimmed patches which are likely to be produced in the alternative case, i.e. when, for the purposes of analysis, a model covering the entire hull has to be restricted to the parts below the waterline.

II.1.1 Wireframe-Based Surface Modelling

The Surface Modelling Approach: It is common in Computer-Aided Design to model the geometry of any kind of three-dimensional objects by *surface modelling*: see e.g. Lee [57] (1999). Since the context of isogeometric analysis, in which we apply a boundary element method to evaluate the coefficient C_W of wake making resistance (see Section I.3), poses the primary purpose of this model, our primary interest lies in determining a representation of the ship *hull*. Alternative modelling approaches like spatial decomposition models or constructive solid modelling are, therefore, not suitable.

Local Construction: For our rather complex ship hull geometry, it is suitable to construct a network of connected surface *patches* and join them with the desired continuity, i.e. we follow the construction by a local approach.

B-Spline Representation: The construction of each of these patches depends heavily on the desired representation of the geometry. Rational and non-rational B-spline have become the predominant form for the representation of curves and surfaces, see Nowacki [65] (2010) and, therefore, also in isogeometric analysis.

Modelling with a Wireframe: Our model is governed by a network of control curves, the wireframe, which gives a good idea of the shape of the eventual ship hull, compare to the sketch in Figure II.1. Appropriate design of these curves yields a quadrilateral loops which can be “filled” by well-known methods like the Coons patch construction which we employ, see [21]. As this method blends opposite boundary information, the resulting patches comply with the shape suggested by the wireframe.

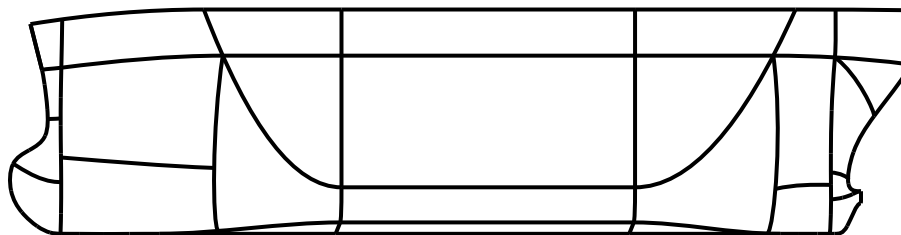


Figure II.1: A sketch of the control curve network, the wireframe.

II.1.2 A G^1 Continuous Patch Network

In accordance with Farin [32] (2002), we introduce an intuitive characterisation for geometric continuity:

Definition II.1.1 (G^1 Continuity). Two surfaces with a common boundary curve are G^1 continuous if the tangent plane changes continuously across that boundary curve. $\#$

Unless otherwise indicated, we shall also employ the term “smooth” to refer to G^1 continuity.

Smooth Surface Construction: To guarantee G^1 continuity, we specify distributions of tangent planes along all curves of the wireframe. The behaviour of the patches in a neighbourhood of their boundaries is, therefore, pre-determined prior to the construction of any patch. As we see in Section II.3, the CatiaTM model takes a similar approach and directly employs the use of “ribbons” and we rely on proper patch creators to comply with this imposition. We can be certain that approximations take place in such an environment and as we discuss below, it is this the fundamental motivation for this thesis.

Without the machinery of an environment like Catia, we have to do the proper “leg work” to ensure fundamental properties like the required G^1 continuity. These requirements can then, however, be fulfilled on an *exact* level and, generally speaking, full control over geometric properties can be provided.

In Section III.2 we elaborate a way to specify a distribution of tangent planes for a curve by a vector field corresponding to the distribution of the tangent plane normals, the *frame*, compare to Figure II.2. The conditions on this frame, which are mainly constituted by the required orthogonality to the tangent, result in a linear equation system which is solved by standard techniques of linear algebra.

One particular well suited technique to construct quadrilateral surface patches for our model is the bicubically blended Coons patch construction as it is governed by a set of vector fields which represent the cross-boundary tangents of the patch. These cross-boundary *ribbons* are constructed orthogonally to the corresponding boundary curve frame and, therefore, lie in the tangent plane of the curve.

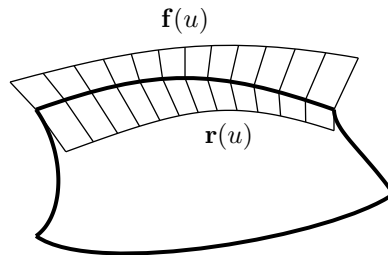


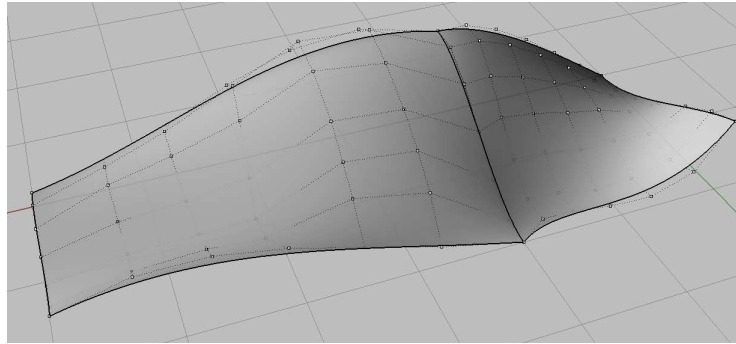
Figure II.2: The frame $\mathbf{f}(u)$ and ribbon $\mathbf{r}(u)$ of a curve in a quadrilateral loop.

The Coons patch construction itself then blends the curves and ribbons of opposite boundaries, thus retaining the orthogonality of the patch to the frames. In this manner we guarantee a smooth tangent plane transition over the entire ship hull. Compare to Figure II.3 for an example of two patches, which join G^1 continuously.

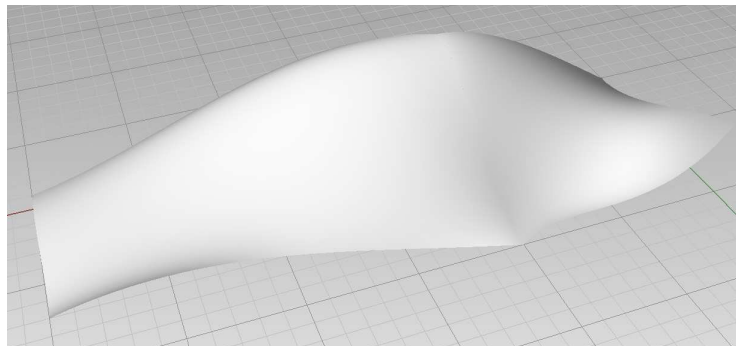
Hierarchically Equal Surface Patches: From a strategical point-of-view this approach holds one additional, fundamental benefit: the construction of one patch is not dependent on that of its neighbours. We, therefore, create a network of hierarchically equal patches. They all, however, strictly depend on the control curve network and the associated frames.

II.2 The Parameters of the Ship

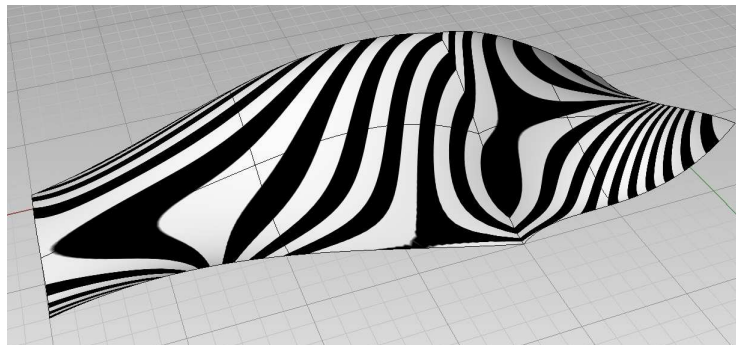
Orientation: We place the ship into a right-handed coordinate system of the coordinates x , y and z . The ship’s **Bottom Plane** shall be in the x - y plane $z = 0$ having the x -axis as its centre in longitudinal direction. Two planes which are parallel to the Bottom Plane are the **Draft Plane** at the height of the design waterline and the **Deck Plane** at the height of the deck. The x - z plane shall be called **Centre Plane** at $y = 0$ and its parallel plane at the



(a)



(b)



(c)

Figure II.3: Two G^1 continuous patches: (a) exhibits the control points of both patches; (b) exhibits the rendered patches; (c) exhibits the isophote curves (see Chapter V and [74, 39, 40, 4]), whose continuity confirms the geometry's G^1 continuity as black and white stripes. All plots are produced in Rhinoceros 3DTM (see page 142).

hull's farthest point from the Centre Plane (at $y = Y_2/2$) shall be called **Side Plane**.

The aft-perpendicular (AP) shall coincide with the z -axis. The ship's longitudinal direction shall be in that of the positive x -axis. Therefore, the forward-perpendicular (FP) intersects the x -axis and has the orientation of the z -axis. See a sketch in Figure II.4.

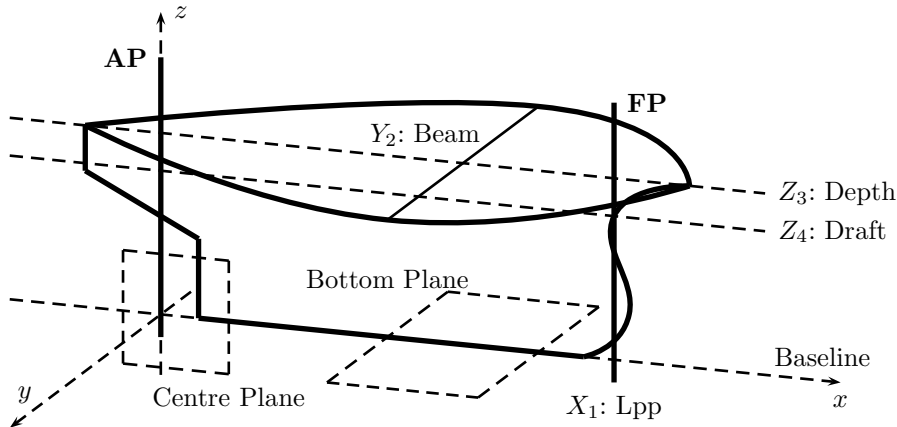


Figure II.4: The global parameters in a simplified ship hull in 3D-space.

Set of Parameters: Below we present a list of the 29 exposed parameters for the modification of the model. Additionally, we give 9 internal parameters which have significant influence on the ship shape. However, these are chosen to be fixed and, therefore, denominated “internal”. All parameters are categorised into four groups. Parameters affecting the entire ship are referred to as *global* parameters and constitute the first group. The influence of the other parameters is mostly *local* and we distinguish the *midship*, *bow* and *stern* parameters.

As parameter names tend to get lengthy, we assign them abbreviations consisting of a base name and an index. The base name depends on the corresponding parameter's direction or nature and the index coincides with parameter number. In this sense, X_1 refers to the first parameter, the LPP, which corresponds to a length in longitudinal direction, thus motivating the use of “ X ” for its basis. On the other hand, R_4 expresses the *ratio* of the parameter Z_4 to Z_3 , as proposed in Section I.4. For angles, we use the base name Φ .

The selection of parameters is based on the technical report [80] prepared by DNV in the context of the EU-funded project Exciting [29] and in accordance with this report we shall provide a default value for each parameter. Table II.1 on page 30 summarizes all 38 parameters and default values and in Chapter V we present ship hull instances which result from structured parameter alterations.

We have four global parameters affecting the entire ship; compare to Figure II.4:

1. The **Length Between Perpendiculars (LPP)** is the principal design parameter for the length of the ship. It measures the distances between

the aft-perpendicular and the fore-perpendicular and is referred to by X_1 . Its default value is $230m$.

2. The **Beam** Y_2 measures the breadth of the ship at its widest part. The default value for this parameter is $32m$.
3. The **Depth** Z_3 with the default value of $23m$ is the vertical distance between the Bottom Plane of the ship and the Deck Plane.
4. The **Draft** Z_4 is the vertical distance from the Bottom Plane of the ship to the design waterline. We put this length in relation to the Depth Z_3 and introduce the ratio $R_4 := Z_4/Z_3$, whose default value is 78%, which implies that Z_4 is roughly $18m$.

The LPP X_1 , the Beam Y_2 and the Depth Z_3 can be seen as the ship's principal dimensions while the Draft Z_4 is central for our model as it marks to upper boundary of the wetted hull.

The parameters of the midship are the following, compare to Figure II.5:

5. The **Parallel Midship Start** X_5 gives the location in longitudinal direction of the start of the midship. We introduce the ratio which puts X_5 in relation to the LPP: $R_5 := X_5/X_1$. In the default setting, this parameter is $R_5 = 30\%$ which yields $X_5 = 69m$.
6. The **Parallel Midship End** X_6 is the location of the end of the parallel midship. As above, we use the dimensionless parameter R_6 as the ratio of X_6 to the distance from the midship start to the FP. We have $R_6 := (X_6 - X_5)/(X_1 - X_5)$. Our default value is $R_6 = 50\%$ implying $X_6 = 149.50m$.
7. The **Bilge Radius** Y_7 is the distance from the transverse expansion of the flat-of-bottom to the Side Plane of the ship. Its bounds are, therefore, $Y_7 \in (0, \min(Y_2/2, Z_3))$ and we use the default value $3.50m$. As it is principally used to construct the cylinder-like part of the midship we employ the term "radius" for this parameter.
8. The **Flat Bottom Start** X_8 determines the aft-most point of the flat-of-bottom. We employ the ratio R_8 of X_8 to the Parallel Midship Start: $R_8 := X_8/X_5$ with the default value is $R_8 = 40\%$ which implies $X_8 = 27.60m$.
9. The **Flat Side Start** X_9 marks the aft-most point of the flat-of-side and we employ its ratio with respect to the Parallel Midship Start $R_9 := X_9/X_5$. We use the default value $R_9 = 20\%$ resulting in $X_9 = 13.80m$.
10. The **Flat Bottom End** X_{10} relates to the fore-most point of the flat-of-bottom. It has to lie between the Parallel Midship End and the FP, hence we use the ratio $R_{10} := (X_{10} - X_6)/(X_1 - X_6)$. Our default value is $R_{10} = 60\%$ which implies $X_{10} = 197.80m$.
11. Analogously, we use X_{11} and R_{11} to mark the fore-most point of the flat of side, the **Flat Side End**, and with the default value of $R_{11} = 50\%$ we have the initial length of $189.75m$.

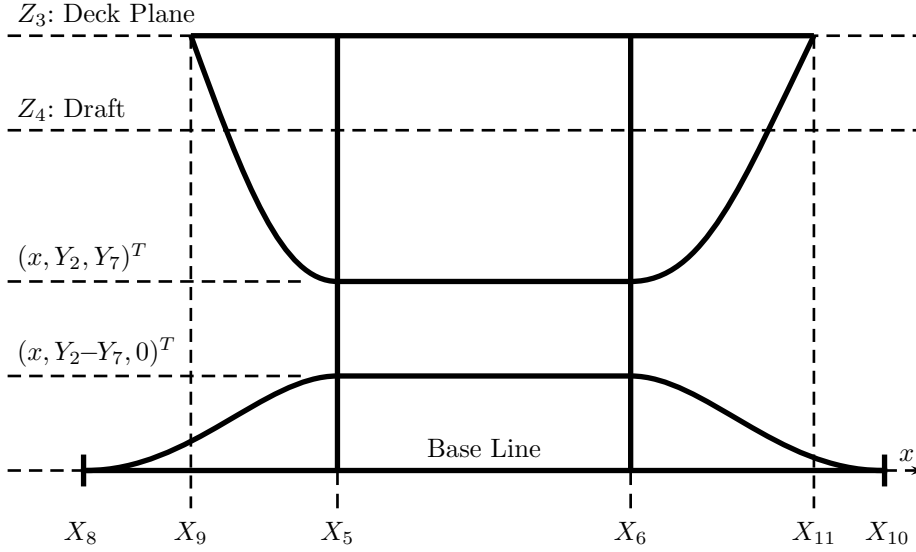


Figure II.5: Sketch of the parameters in the midship.

Note that the lengths of the start lengths of the flat-of-bottom and flat-of-side are independent from each other as no ship characteristic feature would suggest otherwise. The same holds for the end lengths.

We continue with the parameters of the bow, which are also presented in Figure II.6:

12. The **Forward Overhang Length** X_{12} marks the distance between the FP and the fore-most point on the Deck Plane. Its default value is $3.20m$.
13. The **Forward Overhang Slope** Φ_{13} is the angle with the x -axis and the ship profile curve at the draft and, therefore, at the FP. The default slope is set at 8° .
14. The **Bulb Length** X_{14} is the absolute length that the bulbous part of the bow extends in longitudinal direction beyond the FP. We use the default value $8.80m$.
15. The **Bulb Height** Z_{15} indicates the height of the foremost point of the bulb. We express this entity by R_{15} as its ratio to the Draft: $R_{15} := Z_{15}/Z_4$. The height of roughly $6m$ is implied by the default ratio $R_{15} = 33\%$. This parameter is exceptional in the sense that, a priori, Z_{15} can be negative as the bulb may drop below the base plane.
16. The **Bulb Root Height** Z_{16} is the height of the point at which the bow profile has its aft-most point between the bulb and the forward overhang. We express it through the ratio with respect to the draft: $R_{16} := Z_{16}/Z_4$. The default value is $R_{16} = 70\%$ implying roughly $Z_{16} = 12.6m$.
17. The **Bulb Width** Y_{17} is the maximum breadth of the bulbous bow at the position of the bow start at $x = X_{32}$. We give it as the dimensionless

23. In our model we only deal with a submerged transom. We choose this simplification in order to maintain the topology of the patch network. Therefore, instead of modelling the transom height, we model the **Transom Immersion** Z_{23} . We set it in relation to the Draft Z_4 , and employ the ratio $R_{23} = Z_{23}/Z_4$. The default value is $R_{23} = 3\%$ which roughly implies $Z_{23} = 0.50m$.
24. The **Shaft Exit** X_{24} is the shaft absolute longitudinal distance from the AP. The default value is $X_{24} = 5.50m$.
25. The **Shaft Height** Z_{25} is the height of the shaft's centre axis. We give it by a dimensionless parameter which puts it in relation to the transom height: $R_{25} = Z_{25}/(Z_4 - Z_{23})$. The default value is $R_{25} = 23\%$ implying roughly $Z_{25} = 4m$.
26. The **Propeller Clearance** Z_{26} gives the vertical distance at $x = X_{24}$ of the shaft centre axis to the transom profile. We put it in relation with the height of the transom and the shaft height and use the dimensionless parameter $R_{26} = Z_{26}/(Z_4 - Z_{23} - Z_{25})$. Its initial value shall be $R_{26} = 1/2$ which roughly corresponds to $6.70m$.
27. The root is the fore-most point of the transom profile. Its longitudinal position is marked by the **Root Exit** X_{27} . We express this parameter by the root's distance \hat{X}_{27} from the shaft with a default value $\hat{X}_{27} = 3.50m$ implying $X_{27} = 9m$.
28. The **Root Height** Z_{28} gives the vertical position of the root at $x = X_{27}$. The root's relative position with respect to the shaft centre axis is only reasonable as a fraction of the Propeller Clearance: $R_{28} = (Z_{28} - Z_{25})/Z_{26}$. The default value is $R_{28} = 30\%$ roughly implying $Z_{28} = 2m$.
29. The **Shaft Radius** Y_{29} is the distance from the shaft's centre axis to the four extreme points in transverse and vertical direction. If the shaft back surface is a circular disk, this parameter constitutes its radius. We use the default parameter of $0.65m$.

This concludes the parameters we expose to the user and to optimisation. The following internal parameters 30-38 hold further influence on the model but are not free to manipulate:

30. The tangent plane in the front point of the Deck Curve shall be governed by the angle of its normal vector with the z -axis, the **Forward Overhang Tilt** Φ_{30} . Its initial value is $\Phi_{30} = 110^\circ$.
31. The **Bulb Fat Parameter** R_{31} is the value which shall govern the roundness of the fore-most point of the bulb. We employ this parameter in the rescaling of the input tangent for the profile curve of the bulb at this point. Therefore, only positive values are reasonable and for $R_{31} = 1$ we neglect to rescale this tangent vector. We use the default value $R_{31} = \frac{5}{4}$.
32. A central curve in the wireframe is a section curve close to the bow root. Its location in longitudinal direction, the **Bow Start** X_{32} takes as reference the aft-most point of the profile curve between the bulb and the waterline.

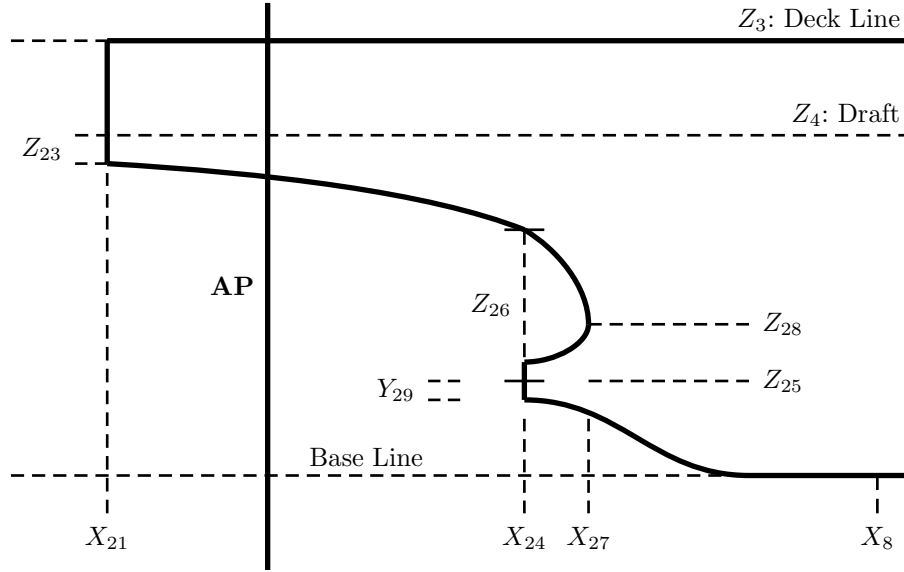


Figure II.7: Sketch of the parameters in the stern.

We find it best not to correlate this parameter to any other bow-specific parameter and determine X_{32} by a ratio of the LPP: $R_{32} = Z_{32}/X_1$. We use the default parameter $R_{32} = 98\%$ which implies $X_{32} = 225.40m$.

33. The root point of the bow section mentioned in item 32, i.e. the point between the bulb side point and the draft which is closest to the centre plane, is assigned its y -coordinate **Bow Root Beam** Y_{33} . This parameter shall be a fraction of the Bulb Width and thus we define the ratio $R_{33} := Y_{33}/Y_{17}$. The default value $R_{33} := 0.25$ implies $Y_{33} = 1.20m$.
34. The transom back plane is bounded by the top beam and a bottom curve. At the far point $(-X_{21}, Y_{22}, Z_3)^T$ in transverse direction the bottom curve parts in the angle **Transom Side Tilt** Φ_{34} from the z -axis for which we give the default value $\Phi_{34} = 20^\circ$.
35. We mark the point where the shaft profile begins to rise from the Bottom Plane by the **Shaft Rise Start** X_{35} . We introduce it as the distance \hat{X}_{35} from the Root Exit X_{27} with the default value $\hat{X}_{35} = 5m$ implying $X_{35} = 14m$.
36. The **Stern Start** X_{36} indicates where we construct the Stern Section control curve. Like above, we introduce it as the distance \hat{X}_{36} from the Root Exit and give the default value $\hat{X}_{36} = \hat{X}_{35}$ implying $X_{36} = X_{35}$.
37. We mark the **Shaft Beam** Y_{37} as the width of the Stern at $x = X_{36}$ and at the height between the Shaft and the Root: $z = 1/2(Z_{25} + Z_{28})$. For our shape of ships it is convenient to regard Y_{37} in the context of the overall beam Y_2 : $R_{37} := Y_{37}/Y_2$. As default we use $R_{37} = 25\%$, roughly implying $Y_{37} = 8m$. Note that, as opposed to other parameters for which employ the term “beam”, this does not necessarily mark the widest point of the

local hull. It serves, however, as a reference points for the construction of the Stern Section, see page 102.

38. The stern section interpolates the point which corresponds to parameter 37: $\mathbf{P} = (X_{36}, Y_{37}, 1/2(Z_{25} + Z_{28}))^T$. At this point, we impose a tangent condition which marks the shape of the section curve: the **Shaft Bend** Φ_{38} . For $\Phi_{38} = 0^\circ$, we have a vertical tangent. If the curve is parametrized to go from bottom to top, positive values of Φ_{38} imply tangents pointing towards the centre plane. Our default parameter shall be $\Phi_{38} = 20^\circ$.

II.3 Design in Catia™

A related task to the one in this thesis was materialized in the software package Catia¹ which is a fully featured hybrid parametric modeller, thus suitable for surface and solid modelling. It provides a wide range of functionalities and automation is possible through various scripting languages. In Ginnis et.al. [36] (2011) we find the documentation of this effort and the experience gained in it has contributed in a large part to the parameter selection and construction principles of this thesis. The Catia model employs a similar set of parameters as the one proposed in the previous Section II.2 and provides the geometry for an optimisation loop with the same context as we have in this thesis, i.e. iso-geometric boundary element methods to minimize the wave making resistance, compare Subsection I.3.3.

The construction of the geometric entities is governed by the functionality provided in Catia. Interpolation of points whose location is based on the input parameters and of tangent directions yields the wireframe, i.e. a control curve network (CCN). To provide continuity between patches the notion of *cross-tangent ribbons* (not to confuse with the ribbons in the Coons patch context as in Sections III.3 and III.4) is introduced and defines in combination with the curve itself a tangent plane representation at each point of each curve.

The combination of curves and ribbons is referred to as augmented CCN, compare to Figure II.8 for the bow and stern parts. A refinement of the CCN yields the final wireframe: Figure II.9.

The local construction of the patches is then performed by Hermite-type quadrilateral surface patch generation.

¹Catia™ by Dassault Systèmes since 1977: www.3ds.com/catia.

Name	PM	Value	Derived Parameters	Value
LPP	X_1	230m	-	
Beam	Y_2	32m	-	
Depth	Z_3	23m	-	
Draft	R_4	0.78	$Z_4 := R_4 Z_3$	17.94m
MS Start	R_5	0.3	$X_5 := R_5 X_1$	69m
MS End	R_6	0.5	$X_6 := X_5 + R_6(X_1 - X_5)$	149.50m
Bilge Radius	Y_7	3.50m	-	
Flat Btm Start	R_8	0.4	$X_8 := R_8 X_5$	27.60m
Flat Side Start	R_9	0.2	$X_9 := R_9 X_5$	13.80m
Flat Btm End	R_{10}	0.6	$X_{10} := X_6 + R_{10}(X_1 - X_6)$	197.80m
Flat Side End	R_{11}	0.5	$X_{11} := X_6 + R_{11}(X_1 - X_6)$	189.75m
FW Overhang	X_{12}	3.20m	-	
FW-OH Slope	Φ_{13}	0.14	-	
Bulb Length	X_{14}	8.80m	-	
Bulb Height	R_{15}	0.33	$Z_{15} := R_{15} Z_4$	$\sim 6m$
Bulb Root Height	R_{16}	0.7	$Z_{16} := R_{16} Z_4$	$\sim 12.6m$
Bulb Width	R_{17}	0.15	$Y_{17} := R_{17} Y_2$	4.8m
Bulb Side Height	R_{18}	0.28	$Y_{18} := R_{18} Y_4$	$\sim 5m$
Bulb Rise Start	R_{19}	0.75	$X_{19} := X_{10} + R_{19}(X_1 - X_{10})$	$\sim 222m$
Bulb Rise	R_{20}	0.1	$Z_{20} := R_{20} Z_{18}$	$\sim 0.5m$
Transom Behind	X_{21}	6.50m	-	
Transom Beam	R_{22}	0.95	$Y_{22} := R_{22} Y_2$	30.40m
Tr. Immersion	R_{23}	0.03	$Z_{23} := R_{23} Z_4$	$\sim 0.50m$
Shaft Exit	X_{24}	5.50m	-	
Shaft Height	R_{25}	0.23	$Z_{25} := R_{25}(Z_4 - Z_{23})$	$\sim 4m$
Prop. Clearance	R_{26}	0.5	$Z_{26} := R_{26}(Z_4 - Z_{23} - Z_{25})$	$\sim 6.70m$
Root Exit	\hat{X}_{27}	3.50m	$X_{27} := X_{24} + \hat{X}_{27}$	9m
Root Height	R_{28}	0.3	$Z_{28} := R_{28} Z_{26} + Z_{25}$	6m
Shaft Radius	Y_{29}	0.65m	-	
FW-OH Tilt	Φ_{30}	1.92	-	
Bulb Fat	R_{31}	1.25	-	
Bow Start	R_{32}	0.98	$X_{32} := X_1 R_{32}$	225.4m
Bow Root	R_{33}	0.25	$Y_{33} := R_{33} Y_{17}$	1.20m
Transom Tilt	Φ_{34}	0.35	-	
Shaft Rise Start	\hat{X}_{35}	5m	$X_{35} := X_{27} + \hat{X}_{35}$	14m
Stern Start	\hat{X}_{36}	\hat{X}_{35}	$X_{36} := X_{27} + \hat{X}_{36}$	X_{35}
Shaft Beam	R_{37}	0.25	$Y_{37} = R_{37} Y_2$	8m
Shaft Bend	Φ_{38}	0.35	-	

Table II.1: The initial set of parameters as introduced in Section II.2.

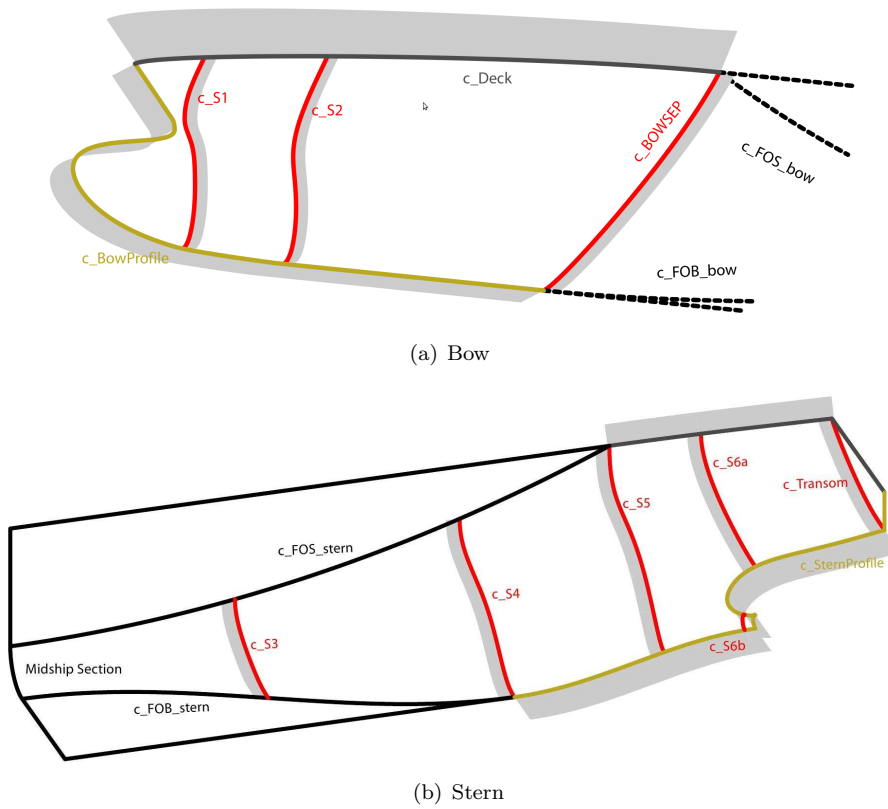
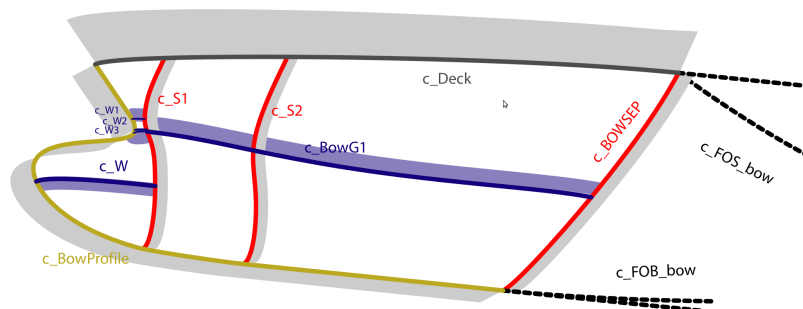
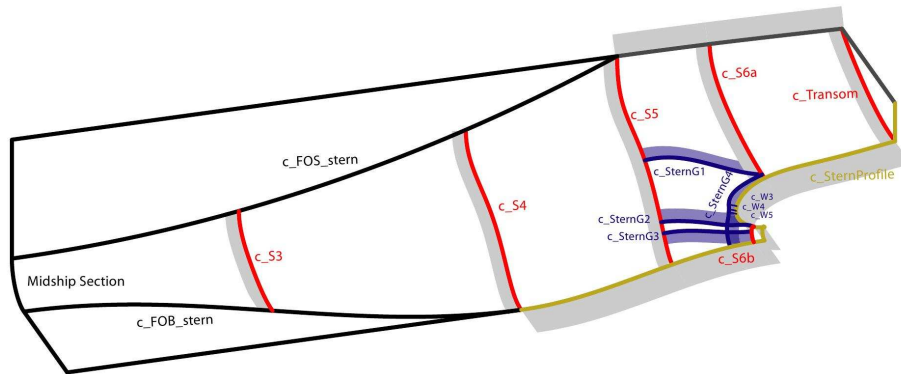


Figure II.8: Modelling in Catia: The CCN of the bow and stern with cross-tangent ribbon. See Section II.3.



(a) Bow



(b) Stern

Figure II.9: Modelling in Catia: The refined CCN of the bow and stern with cross-tangent ribbons. See Section II.3.

Chapter III

Geometric Tools for the Model

The fundamentals of curves and surfaces in Bézier, B-splines and NURBS representation, can be reviewed in the Appendix A.1 and standard literature like, e.g., Piegl and Tiller [70] (1997) or Farin [32] (2002). Based on this knowledge, we cover the essential geometric algorithms used in the creation of the PM in this section.

The ship hull produced by our parametric model is composed of a network of quadrilateral tensor product patches whose placement and shape are governed by a properly created control curves network, the *wireframe*. In order to obtain a smooth transition between patches, prior to their construction we specify the tangent plane for all wireframe curves. To this end, we present an algorithm to construct a vector field, the *frame*, which is orthogonal to the distribution of curve tangent vectors: Section III.2.

Bicubically blended Coons patches are especially suitable for the construction of quadrilateral patches as they blend positional and tangential data of opposite boundaries, see Section III.4. In this connection, a *ribbon*, which represents the cross-boundary tangent of the curve, is constructed for each boundary curve. We present an algorithm to create this ribbon in a way such that it is orthogonal to the corresponding curve frame: see Section III.3. The patch's tangent plane in the boundary, therefore, coincides with the plane implied by the frame. In this manner we obtain a continuous tangent plane transition along and across the boundaries of the patches and hereby guarantee the demanded G^1 continuity.

Since the entire model is governed by a wireframe, a routine for curve interpolation is elementary and shall be the start point in the development of our geometric routines.

III.1 Enhanced Interpolation

It is the exception in a context like this ship hull model to create a curve by its defining properties, e.g. the control points and basis of B-spline curves. Rather, the knowledge of points, tangents and curvature on the hull has to be suitably and automatically *interpolated*. It is the objective of the following paragraphs

to elaborate an interpolation algorithm which yields a curve of the common B-spline format. We employ a degree p , a set of n control points by $(\mathbf{c}_\ell)_{\ell=0}^{n-1}$ and the basis functions by $(N_\ell^p(u))_{\ell=0}^{n-1}$, which are based on the knot vector $\mathbf{k} = (k_i)_{i=0}^{n+p}$, to obtain the representation:

$$\mathbf{c}(u) = \sum_{\ell=0}^{n-1} N_\ell^p(u) \mathbf{c}_\ell,$$

compare to Appendix A.1.

We wish determine the specifications of $\mathbf{c}(u)$ such that it interpolates the following types of data:

- **Positional:** Foremost, a given set of points \mathbf{P}_i has to be interpolated.
- **Tangential:** Additionally, we want to be able to specify tangent data. In the context of this model it is not of interest to impose tangents at points whose spatial position is unknown and we, therefore, associate all tangents with points \mathbf{P}_i .
In many cases, it is useful to not interpolate tangent vectors but rather to specify a tangent direction while leaving its length to be determined in some optimal way. The two concepts of *tangent vectors*, which we denote by \mathbf{t}_i (in lower-case), and *tangent directions*, \mathbf{T}_i (in capital letters), have to be rigorously differentiated throughout this section.
- **Zero-Curvature:** For the necessities of this model, we want to be able to specify parameter values u_i^κ at which the curve has to have vanishing curvature $\kappa(u_i^\kappa) = 0$.

The interpolation problem poses as follows:

Problem III.1.1 (Interpolation). We are given the n_P points $(\mathbf{P}_i)_i$ with $i \in I = [0, \dots, n_P - 1]$. Under the conditions $j_1 \in J_1 \subset I$ and $j_2 \in J_2 \subset I$ with $J_2 \cap J_1 = \emptyset$, we are given vectors $(\mathbf{t}_{j_1})_{j_1}$ and directions $(\mathbf{T}_{j_2})_{j_2}$. Additionally, we are given a set of n_κ parameters $(u_i^\kappa)_i$ with $i = 0, \dots, n_\kappa - 1$ for which we demand that $0 \leq u_i < u_{i+1} \leq 1$.

It is the task to find a proper set of parameters u_i for $i \in I$, the B-spline basis and the n control points of a B-spline function $\mathbf{c}(u) = \sum_{\ell=0}^{n-1} N_\ell^p(u) \mathbf{c}_\ell$ such that $\mathbf{c}(u)$ is twice differentiable and for which the following holds:

$$\mathbf{c}(u_i) = \sum_{\ell=0}^{n-1} N_\ell^p(u_i) \mathbf{c}_\ell = \mathbf{P}_i, \quad i \in I, \quad (\text{III.1})$$

$$\mathbf{c}'(u_{j_1}) = \sum_{\ell=0}^{n-1} N_\ell^{p'}(u_{j_1}) \mathbf{c}_\ell = \mathbf{t}_{j_1}, \quad j_1 \in J_1, \quad (\text{III.2})$$

$$\frac{\mathbf{c}'(u_{j_2})}{\|\mathbf{c}'(u_{j_2})\|} = \frac{\sum_{\ell=0}^{n-1} N_\ell^{p'}(u_{j_2}) \mathbf{c}_\ell}{\|\sum_{\ell=0}^{n-1} N_\ell^{p'}(u_{j_2}) \mathbf{c}_\ell\|} = \mathbf{T}_{j_2}, \quad j_2 \in J_2, \quad (\text{III.3})$$

$$\kappa(u_i^\kappa) = \frac{\|\mathbf{c}'(u_i^\kappa) \times \mathbf{c}''(u_i^\kappa)\|}{\|\mathbf{c}'(u_i^\kappa)\|^3} = 0, \quad i = 0, \dots, n_\kappa - 1. \quad (\text{III.4})$$

‡

Remark III.1.2. • In our surface modelling approach we guarantee a G^1 continuous patch network by frames. As we see in Remark III.4.4 of Section III.4, this concept requires the underlying wireframe to consist of C^2 regular curves.

- We are interested in a *nice* curve, which is a rather subjective requirement, not only from a linguistic point-of-view. Different criteria for *fairness* fit different purposes; see Section III.1.4. For the time being, we state the vague demand that the curve has to comply with the intuition gained from the data.
- For given tangent directions $(\mathbf{T}_j)_j$, Condition (III.3) is replaced by the pair of one-dimensional conditions:

$$\begin{aligned} \mathbf{c}'(u_{j_2}) \cdot \mathbf{n}_{j_2}^1 &= 0, \\ \mathbf{c}'(u_{j_2}) \cdot \mathbf{n}_{j_2}^2 &= 0. \end{aligned} \quad (\text{III.5})$$

where \mathbf{n}_j^1 and \mathbf{n}_j^2 are two non-parallel directions orthogonal to \mathbf{T}_j .

- If at the parameter u_i^κ we are also given the tangent direction $\mathbf{T}_i^\kappa = \mathbf{c}'(u_i^\kappa)$, then Condition (III.4) is a linear one:

$$\mathbf{T}_i^\kappa \times \mathbf{c}''(u_i^\kappa) = \mathbf{T}_i^\kappa \times \sum_{\ell=0}^{n-1} N_\ell^{p''}(u_i^\kappa) \mathbf{c}_\ell = \mathbf{0}. \quad (\text{III.6})$$

If the tangent information is not available, we have two options to retain linear conditions. The first one is to set the second derivative zero, $\mathbf{c}''(u_i^\kappa) = 0$ which might have unwanted side-effects. Alternatively, we can estimate $\mathbf{c}'(u_i^\kappa)$ beforehand and add this tangent (vector or direction) as one further tangent condition. As we shall see, the proposed interpolation algorithm employs the construction of an auxiliary curve $\mathbf{g}(u)$, compare to Section III.1.2. The evaluation of its tangent can be used for the estimation of the tangent: $\mathbf{T}_i^\kappa = \mathbf{g}'(u_i^\kappa) / \|\mathbf{g}'(u_i^\kappa)\|$.

- If the spline space is determined, i.e. if degree p and the basis are specified, all these conditions express linear equality conditions, summarized in the following equation system:

$$\begin{pmatrix} \mathbf{P}_i \\ \vdots \\ \mathbf{t}_{j_1} \\ \vdots \\ \hline 0 \\ 0 \\ \vdots \\ \hline \mathbf{0} \\ \vdots \end{pmatrix} = E \begin{pmatrix} \mathbf{c}_0 \\ \vdots \\ \mathbf{c}_{n-1} \end{pmatrix} \quad (\text{III.7})$$

with

$$E := \begin{pmatrix} \frac{E_1}{E_2} \\ \frac{E_2}{E_3} \end{pmatrix} = \begin{pmatrix} N_0^p(u_i) & N_1^p(u_i) & \dots & N_{n-1}^p(u_i) \\ & \dots & & \\ N_0^{p'}(u_{j_1}) & N_1^{p'}(u_{j_1}) & \dots & N_{n-1}^{p'}(u_{j_1}) \\ & & \dots & \\ \hline N_0^{p'}(u_{j_2})\mathbf{n}_{j_2}^1 \cdot & N_1^{p'}(u_{j_2})\mathbf{n}_{j_2}^1 \cdot & \dots & N_{n-1}^{p'}(u_{j_2})\mathbf{n}_{j_2}^1 \cdot \\ N_0^{p'}(u_{j_2})\mathbf{n}_{j_2}^2 \cdot & N_1^{p'}(u_{j_2})\mathbf{n}_{j_2}^2 \cdot & \dots & N_{n-1}^{p'}(u_{j_2})\mathbf{n}_{j_2}^2 \cdot \\ & & \dots & \\ \hline N_0^{p''}(u_i^{\kappa})\mathbf{t}_i^{\kappa} \times & N_1^{p''}(u_i^{\kappa})\mathbf{t}_i^{\kappa} \times & \dots & N_{n-1}^{p''}(u_i^{\kappa})\mathbf{t}_i^{\kappa} \times \\ & & \dots & \end{pmatrix}. \quad (\text{III.8})$$

As we see, E_1 is composed of the evaluations of the curve's basis functions and their derivatives. Each row has, therefore, at most $p + 1$ non-zero entries and represents a linear condition which has to be read coordinate-wise. This is not the case in E_2 and E_3 where the evaluations of the derivatives of the curve's basis functions are the coefficients of the corresponding vectors. These entries have to be interpreted as scalar products, in E_2 , and as cross products, in E_3 . We employ this altered matrix form in order to simplify notation and we indicate the abbreviated operation by the \cdot and the \times symbol, respectively. Therefore, in E_2 we have one condition per row and in E_3 three.

- Note that for determination of the parameters u_i which are associated with \mathbf{P}_i , we follow the standard pertinent literature and employ centripetal, chord length or uniform parametrization, see Section III.1.3.

‡

In the subsequent sections we present minimal degrees for the constructions of frames and ribbons which depend on the degree of the respective curve. To avoid high degrees there, our target curve degree is 3 as it is the minimal degree for construction of C^2 continuous B-spline curves. The following is the outline of the Algorithm III.1.14 which we elaborate in this section and which is explicitly stated on page 54:

1. Firstly, we interpolate all pairs of data points by Bézier segments. Joining these segments in a proper manner yields a G^1 continuous piecewise Bézier curve $\mathbf{g}(u)$ which serves us as reference in later steps: Subsections III.1.1–III.1.2.
2. In the second step, we constitute a proper basis for a C^2 -continuous curve with more control points than the interpolation conditions require. We use these degrees-of-freedom to optimize the curve $\mathbf{c}(u)$ in the following senses:
 - (a) Approximate the previously constructed piecewise Bézier curve $\mathbf{g}(u)$.
 - (b) To avoid oscillating control polygons which may be caused by the above approximation, we minimize the squared side-lengths of the control polygon.

This is covered in Subsections III.1.3–III.1.4.

Our first step is the Hermite interpolation.

III.1.1 Hermite Interpolation

This subsection reviews the well-known Hermite interpolation with data consisting of two points and possibly one or two tangent vectors or directions. This data set corresponds to pairs of points $(\mathbf{P}_i, \mathbf{P}_{i+1})$ and their associated tangents of Problem III.1.1. The hereby constructed Bézier curve segments are joined to a G^1 continuous piecewise Bézier curve in the subsequent Subsection III.1.2. Note that curvature information is not regarded in this construction step.

Two Points: Two points are interpolated by a Bézier curve of degree 1, i.e. a line:

$$\mathbf{c}(u) = \mathbf{P}_0 N_0^1(u) + \mathbf{P}_1 N_1^1(u), \quad u \in [0, 1].$$

Two Points, One Tangent Vector: If we impose a tangent vector at one of the points, a Bézier curve of degree 2 is employed to interpolate the data. Let \mathbf{t}_0 be given at the point \mathbf{P}_0 , then the middle control point \mathbf{c}_1 is defined by

$$\mathbf{t}_0 = \mathbf{c}'(0) = 2(\mathbf{c}_1 - \mathbf{c}_0)$$

and the Bézier expression reads:

$$\mathbf{c}(u) = \mathbf{P}_0 N_0^2(u) + \left(\mathbf{P}_0 + \frac{1}{2}\mathbf{t}_0\right) N_1^2(u) + \mathbf{P}_2 N_2^2(u).$$

The construction is analogous for a given \mathbf{t}_1 at \mathbf{P}_1 .

Cubic Hermite Interpolation: For the interpolation of two given points \mathbf{P}_0 and \mathbf{P}_1 together with tangent vectors \mathbf{t}_0 and \mathbf{t}_1 we proceed analogously to the above approaches and employ a cubic Bézier curve $\mathbf{c}(u)$. The two inner control points \mathbf{c}_1 and \mathbf{c}_2 are determined by the given tangents:

$$\begin{aligned} \mathbf{t}_0 &= \mathbf{c}'(0) = 3(\mathbf{c}_1 - \mathbf{c}_0), \\ \mathbf{t}_1 &= \mathbf{c}'(1) = 3(\mathbf{c}_3 - \mathbf{c}_2) \end{aligned}$$

and they yield the interpolating curve $\mathbf{c}(u)$:

$$\mathbf{c}(u) = \mathbf{P}_0 N_0^3(u) + \left(\mathbf{P}_0 + \frac{1}{3}\mathbf{t}_0\right) N_1^3(u) + \left(\mathbf{P}_1 - \frac{1}{3}\mathbf{t}_1\right) N_2^3(u) + \mathbf{P}_3 N_3^3(u).$$

We can express $\mathbf{c}(u)$ in cardinal form,

$$\mathbf{c}(u) = \mathbf{P}_0 H_0^3(u) + \mathbf{t}_0 H_1^3(u) + \mathbf{t}_1 H_2^3(u) + \mathbf{P}_3 H_3^3(u),$$

with the Hermite basis functions as seen in Figure III.1(a)

$$H_0^3(u) = N_0^3(u) + N_1^3(u), \quad (\text{III.9})$$

$$H_1^3(u) = \frac{1}{3}N_1^3(u), \quad (\text{III.10})$$

$$H_2^3(u) = -\frac{1}{3}N_2^3(u), \quad (\text{III.11})$$

$$H_3^3(u) = N_2^3(u) + N_3^3(u). \quad (\text{III.12})$$

We point out the direct link of this notation to the interpolation data: At $u = 0$, all $H_i^3(0)$ are zero except $H_0^3(u)$, which is 1. Similarly, the derivatives of all $H_i^3(0)$ are zero except that of $H_1^3(u)$, which is 1. At $u = 1$, we have analogous results:

i	0	1	2	3
$H_i^3(0)$	1	0	0	0
$H_i^{3'}(0)$	0	1	0	0
$H_i^{3'}(1)$	0	0	1	0
$H_i^3(1)$	0	0	0	1

Find an example for the cubic Hermite interpolation in Figure III.1(b).

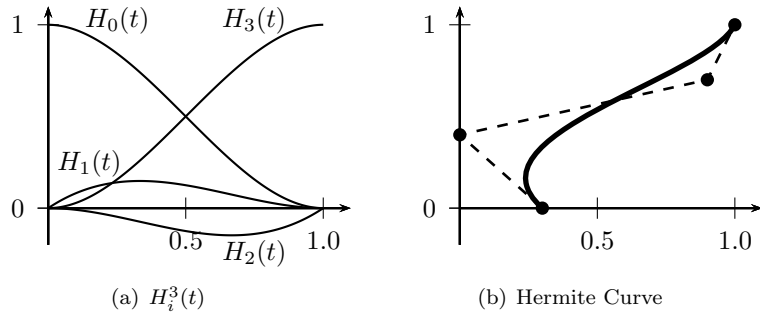


Figure III.1: (a): The cubic Hermite polynomials $H_i^3(t)$.

(b): Cubic Hermite interpolation of the locations $\mathbf{P}_0 = (0.3, 0)$ and $\mathbf{P}_1 = (1, 1)$ with the respective tangents $\mathbf{t}_0 = (-0.9, 1.2)$ and $\mathbf{t}_1 = (0.9, -0.3)$.

One Tangent Direction: Let the data for the Hermite interpolation consist of two points \mathbf{P}_0 and \mathbf{P}_1 and one tangent direction \mathbf{T}_0 which shall be imposed at the former point. Once we find a suitable length l for \mathbf{T}_0 , we can proceed like above and interpolate the three entities by a Bézier curve of degree $p = 2$.

One criterion to determine l can be to minimize the sum of the squared side lengths of the control polygon. With $L = l/2$ being the length of the first leg of the control polygon and $\mathbf{v} = \mathbf{P}_1 - \mathbf{P}_0$, we obtain its optimum L^{Min} as follows:

$$\begin{aligned}
 & \|\mathbf{c}_1 - \mathbf{c}_0\|^2 + \|\mathbf{c}_2 - \mathbf{c}_1\|^2 \rightarrow \min \\
 & \|(\mathbf{P}_0 + L\mathbf{T}_0) - \mathbf{P}_0\|^2 + \|\mathbf{P}_1 - (\mathbf{P}_0 + L\mathbf{T}_0)\|^2 \rightarrow \min \\
 & f(L) := 2L^2\mathbf{T}_0 \cdot \mathbf{T}_0 - 2L\mathbf{T}_0 \cdot \mathbf{v} + \mathbf{v} \cdot \mathbf{v} \rightarrow \min \\
 & f'(L) = 2L - 2\mathbf{T}_0 \cdot \mathbf{v} = 0 \\
 & L^{\text{Min}} := \frac{\mathbf{T}_0 \cdot \mathbf{v}}{2} \quad (\text{III.13})
 \end{aligned}$$

Since $f''(L) = 2 > 0$, L^{Min} is indeed the minimum.

L^{Min} yields reasonable results for acute angles $\alpha_0 := \angle(\mathbf{T}_0, \mathbf{v}) < \frac{\pi}{2}$. Note, however, that as α_0 becomes a right angle, L^{Min} tends to zero, and that L^{Min} is negative for obtuse angles. For this reason we consider a second choice for L :

$$L^{\text{Diag}} = \frac{\|\mathbf{v}\|}{2} \quad (\text{III.14})$$

which coincides with L^{Min} if $\alpha_0 = 0$, i.e. if the interpolation problem suggests a straight line. It is then obvious to place the middle control point in the middle of the boundary points and we denote this length by L^{Diag} .

A third option is to average $|L^{\text{Min}}|$ and L^{Diag} , e.g. with the factor $\cos \alpha_0$:

$$L^{\text{Avg}} = \cos \alpha_0 \cdot |L^{\text{Min}}| + (1 - \cos \alpha_0) \cdot L^{\text{Diag}} \quad (\text{III.15})$$

which coincides with L^{Diag} for the angles $\alpha_0 = 0, \frac{\pi}{2}, \pi$. Compare the three scalings in Figure III.2 for an acute angle and in Figure III.3 for an almost right angle. The difference in curvature plots of L^{Min} opposed to L^{Diag} and L^{Avg} is significant. While in the former case all scalings yield reasonable results with respect to curve's shape and to its curvature, the curvature for scaling by L^{Min} explodes in the latter case.

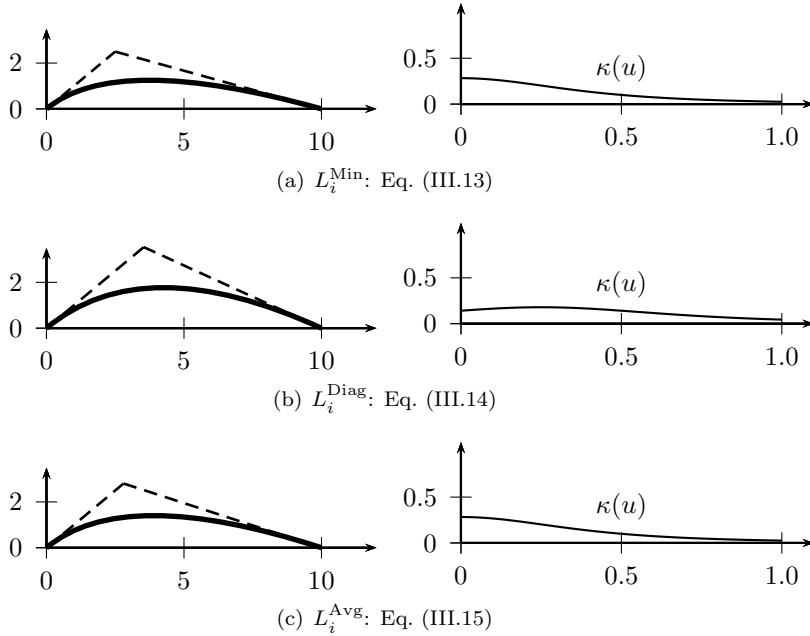


Figure III.2: Interpolation of $\mathbf{P}_0 = (0, 0)^T$, $\mathbf{T}_0 = 1/\sqrt{2}(1, 1)^T$ and $\mathbf{P}_1 = (10, 0)^T$.

Example III.1.3. We consider the interpolation problem of the points $(0, 0)$ and $(10, 0)$. The tangent direction in the former point varies as can be observed in Figure III.4: $\alpha_0 = 116.57^\circ, 90^\circ, 63.43^\circ$. \triangleleft

Two Tangent Directions: In the case of given tangent directions at both points, \mathbf{T}_0 at \mathbf{P}_0 and \mathbf{T}_1 at \mathbf{P}_1 , we employ a Bézier curve of degree $p = 3$ and take an analogous approach as above to determine the lengths l_0 and l_1 . We minimize the sum of the squared side lengths of the control polygon and with

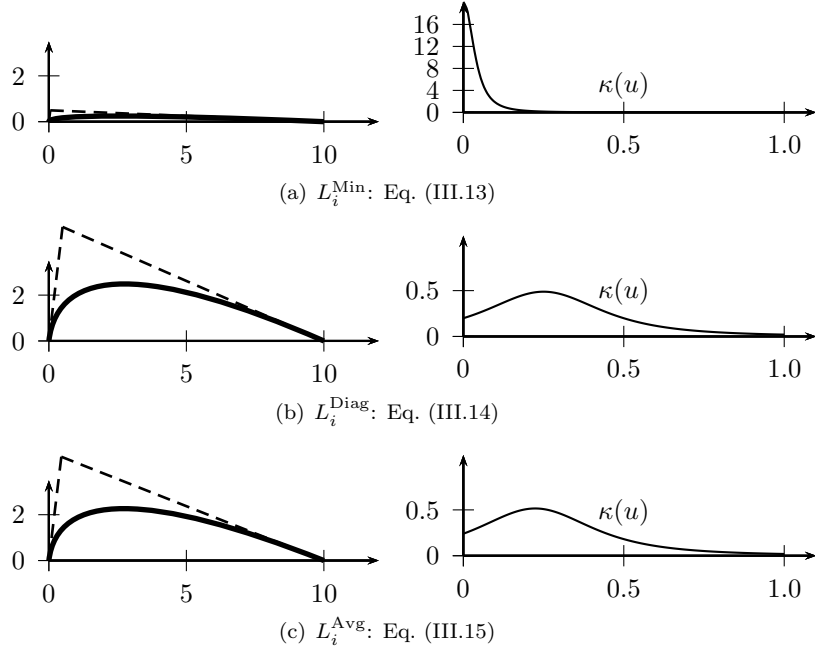


Figure III.3: Interpolation of $\mathbf{P}_0 = (0, 0)^T$, $\mathbf{T}_0 = 1/\sqrt{101}(1, 10)^T$ and $\mathbf{P}_1 = (10, 0)^T$.

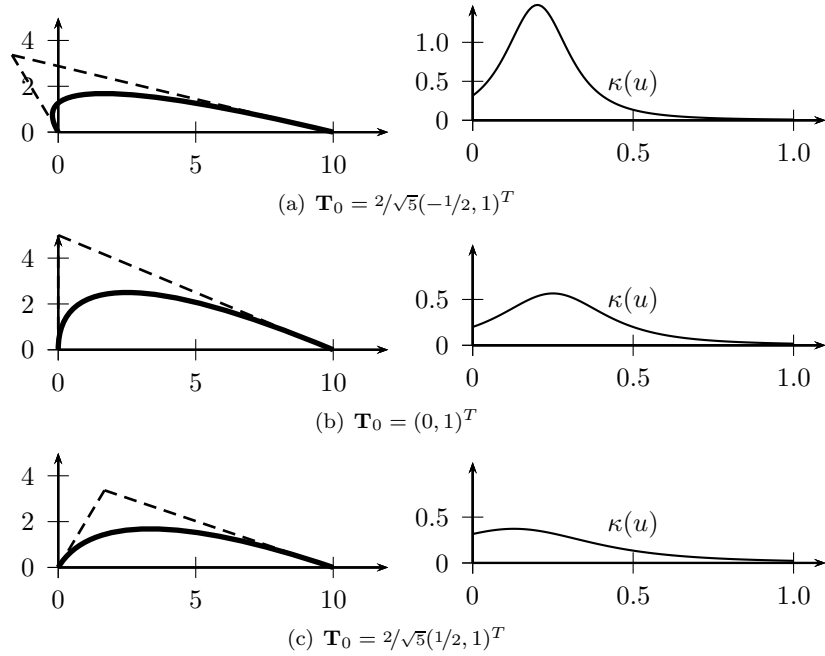


Figure III.4: Interpolation of $\mathbf{P}_0 = (0, 0)^T$, $\mathbf{P}_1 = (10, 0)^T$ and various tangent directions \mathbf{T}_0 scaled by L^{Avg} as in Eq. (III.15).

$L_0 = l_0/3$ and $L_1 = l_1/3$ and $\mathbf{v} = \mathbf{P}_1 - \mathbf{P}_0$, we obtain

$$\begin{aligned}
 & \|\mathbf{c}_1 - \mathbf{c}_0\|^2 + \|\mathbf{c}_2 - \mathbf{c}_1\|^2 + \|\mathbf{c}_3 - \mathbf{c}_2\|^2 \rightarrow \min \\
 & \quad \|(\mathbf{P}_0 + L_0\mathbf{T}_0) - \mathbf{P}_0\|^2 + \\
 & \quad \|(\mathbf{P}_1 - L_1\mathbf{T}_1) - (\mathbf{P}_0 + L_0\mathbf{T}_0)\|^2 + \\
 & \quad \|\mathbf{P}_1 - (\mathbf{P}_1 - L_1\mathbf{T}_1)\|^2 \rightarrow \min \\
 f(L_0, L_1) & := 2L_0^2\mathbf{T}_0 \cdot \mathbf{T}_0 + 2L_0L_1\mathbf{T}_0 \cdot \mathbf{T}_1 + 2L_1^2\mathbf{T}_1 \cdot \mathbf{T}_1 \\
 & \quad - 2L_0\mathbf{T}_0 \cdot \mathbf{v} - 2L_1\mathbf{T}_1 \cdot \mathbf{v} + \mathbf{v} \cdot \mathbf{v} \rightarrow \min \\
 \frac{\partial}{\partial L_0} f(L_0, L_1) & = 4L_0\mathbf{T}_0 \cdot \mathbf{T}_0 + 2L_1\mathbf{T}_0 \cdot \mathbf{T}_1 - 2\mathbf{T}_0 \cdot \mathbf{v} = 0. \\
 \frac{\partial}{\partial L_1} f(L_0, L_1) & = 2L_0\mathbf{T}_0 \cdot \mathbf{T}_1 + 4L_1\mathbf{T}_1 \cdot \mathbf{T}_1 - 2\mathbf{T}_1 \cdot \mathbf{v} = 0
 \end{aligned}$$

This minimum is obtained by solving the following linear system of equations:

$$\begin{pmatrix} 2 & \mathbf{T}_0 \cdot \mathbf{T}_1 \\ \mathbf{T}_0 \cdot \mathbf{T}_1 & 2 \end{pmatrix} \begin{pmatrix} L_0 \\ L_1 \end{pmatrix} = \begin{pmatrix} \mathbf{T}_0 \cdot \mathbf{v} \\ \mathbf{T}_1 \cdot \mathbf{v} \end{pmatrix}.$$

Note, that the involved matrix is positive definite for all \mathbf{T}_0 and \mathbf{T}_1 and the optimal solution $(L_0^{\text{Min}}, L_1^{\text{Min}})$ is the following:

$$\begin{pmatrix} L_0^{\text{Min}} \\ L_1^{\text{Min}} \end{pmatrix} = \frac{1}{4 - (\mathbf{T}_0 \cdot \mathbf{T}_1)^2} \begin{pmatrix} 2\mathbf{T}_0 \cdot \mathbf{v} - (\mathbf{T}_0 \cdot \mathbf{T}_1)(\mathbf{T}_1 \cdot \mathbf{v}) \\ 2\mathbf{T}_1 \cdot \mathbf{v} - (\mathbf{T}_0 \cdot \mathbf{T}_1)(\mathbf{T}_0 \cdot \mathbf{v}) \end{pmatrix}. \quad (\text{III.16})$$

Depending on $\alpha_0 := \angle(\mathbf{T}_0, \mathbf{v})$ and $\alpha_1 := \angle(\mathbf{T}_1, \mathbf{v})$, we possibly obtain zero or negative values for L_0 or L_1 also in this case and we consider the following alternative scaling:

$$L^{\text{Diag}} = \frac{\|\mathbf{v}\|}{3}. \quad (\text{III.17})$$

A possible average of $|L_i^{\text{Min}}|$ and L^{Diag} employs the weights $\cos \alpha_0$ and $\cos \alpha_1$:

$$\begin{pmatrix} L_0^{\text{Avg}} \\ L_1^{\text{Avg}} \end{pmatrix} = \begin{pmatrix} \cos \alpha_0 \cdot |L_0^{\text{Min}}| + (1 - \cos \alpha_0) \cdot L^{\text{Diag}} \\ \cos \alpha_1 \cdot |L_1^{\text{Min}}| + (1 - \cos \alpha_1) \cdot L^{\text{Diag}} \end{pmatrix}. \quad (\text{III.18})$$

In Figure III.5, all three choices of scalings can be compared for the case of nearly right angles. We can see the significant advantage of L^{Diag} and L^{Avg} over $|L^{\text{Min}}|$ with respect to the curvature changes.

Example III.1.4. We pose a similar interpolation problem as above with the points $\mathbf{P}_0 = (0, 0)$ and $\mathbf{P}_1 = (10, 0)$. Firstly, we vary the tangent \mathbf{T}_0 at \mathbf{P}_0 and, secondly, \mathbf{T}_1 at \mathbf{P}_1 :

$$\begin{aligned}
 (a) : \quad & \mathbf{T}_0 = \sqrt{5}/2 \begin{pmatrix} -1/2 \\ 1 \end{pmatrix} & \mathbf{T}_1 = \sqrt{5}/2 \begin{pmatrix} -1/2 \\ 1 \end{pmatrix} \\
 (b) : \quad & \mathbf{T}_0 = \sqrt{5}/2 \begin{pmatrix} -1/2 \\ 1 \end{pmatrix} & \mathbf{T}_1 = \sqrt{5}/2 \begin{pmatrix} -1/2 \\ -1 \end{pmatrix} \\
 (c) : \quad & \mathbf{T}_0 = \sqrt{5}/2 \begin{pmatrix} -1/2 \\ 1 \end{pmatrix} & \mathbf{T}_1 = \sqrt{5}/2 \begin{pmatrix} 1/2 \\ 1 \end{pmatrix} \\
 (d) : \quad & \mathbf{T}_0 = \sqrt{5}/2 \begin{pmatrix} 1/2 \\ 1 \end{pmatrix} & \mathbf{T}_1 = \sqrt{5}/2 \begin{pmatrix} 1/2 \\ -1 \end{pmatrix} \\
 (e) : \quad & \mathbf{T}_0 = \begin{pmatrix} 0 \\ 1 \end{pmatrix} & \mathbf{T}_1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}
 \end{aligned}$$

Compare to Figure III.6. ◁

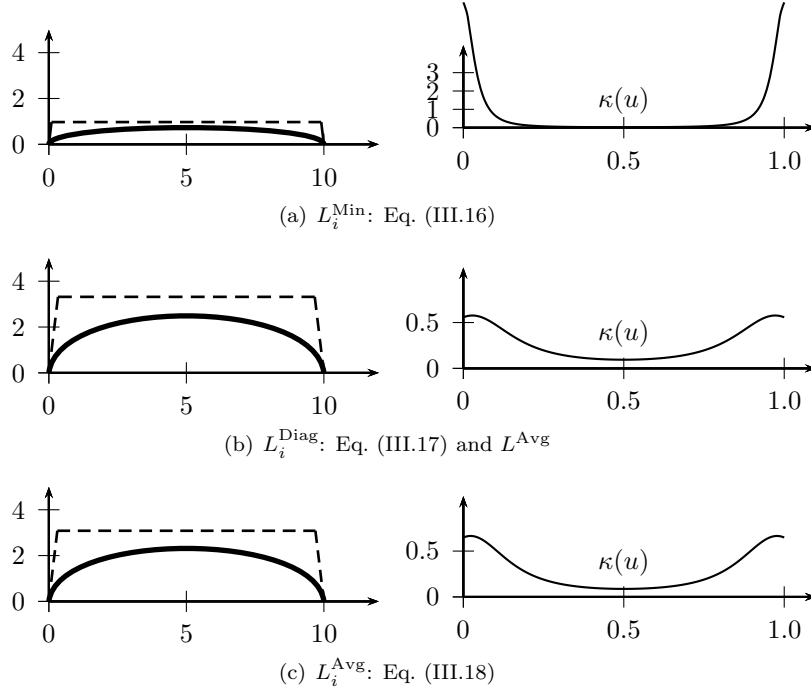


Figure III.5: Interpolation of $\mathbf{P}_0 = (0, 0)^T$, $\mathbf{T}_0 = 1/\sqrt{101}(1, 10)^T$, $\mathbf{P}_1 = (10, 0)^T$ and $\mathbf{T}_1 = 1/\sqrt{101}(-1, 10)^T$.

One Tangent Vector, One Tangent Direction: Finally, we consider the mixed case, i.e. where we are given one tangent *direction* \mathbf{T}_0 at \mathbf{P}_0 and one tangent *vector* \mathbf{t}_1 at \mathbf{P}_1 . This is a special case of the previous one as, with the formulation of that minimisation problem, the length $L_1 = \|\mathbf{t}_1\|/3$ is not subject to optimisation. The minimization of the lengths of the control polygon sides yields the problem

$$2L_0^2 \mathbf{T}_0 \cdot \mathbf{T}_0 + 2L_0 \mathbf{T}_0 \cdot (\mathbf{t}_1 - \mathbf{v}) + \dots \rightarrow \min$$

with an analogous solution L_0 to the one in Equation (III.13):

$$L_0^{\text{Min}} = \frac{\mathbf{T}_0 \cdot (\mathbf{t}_1 - \mathbf{v})}{2}.$$

III.1.2 G^1 Piecewise Bézier Curve Construction

Applying the above Hermite interpolation to all pairs of points yields a continuous curve whose segments $\mathbf{g}_i(u)$ join G^1 continuously if there are tangents associated with all points. At interior data points \mathbf{P}_i with no tangent information we apply the following manipulation of the two curve segments $\mathbf{g}_{i-1}(u)$ and $\mathbf{g}_i(u)$ which connect in \mathbf{P}_i . Let \mathbf{T}_i^- and \mathbf{T}_i^+ be the tangent directions of these curves at \mathbf{P}_i . We define \mathbf{T}_i as follows:

$$\mathbf{T}_i = \frac{1}{2}(\mathbf{T}_i^- + \mathbf{T}_i^+).$$

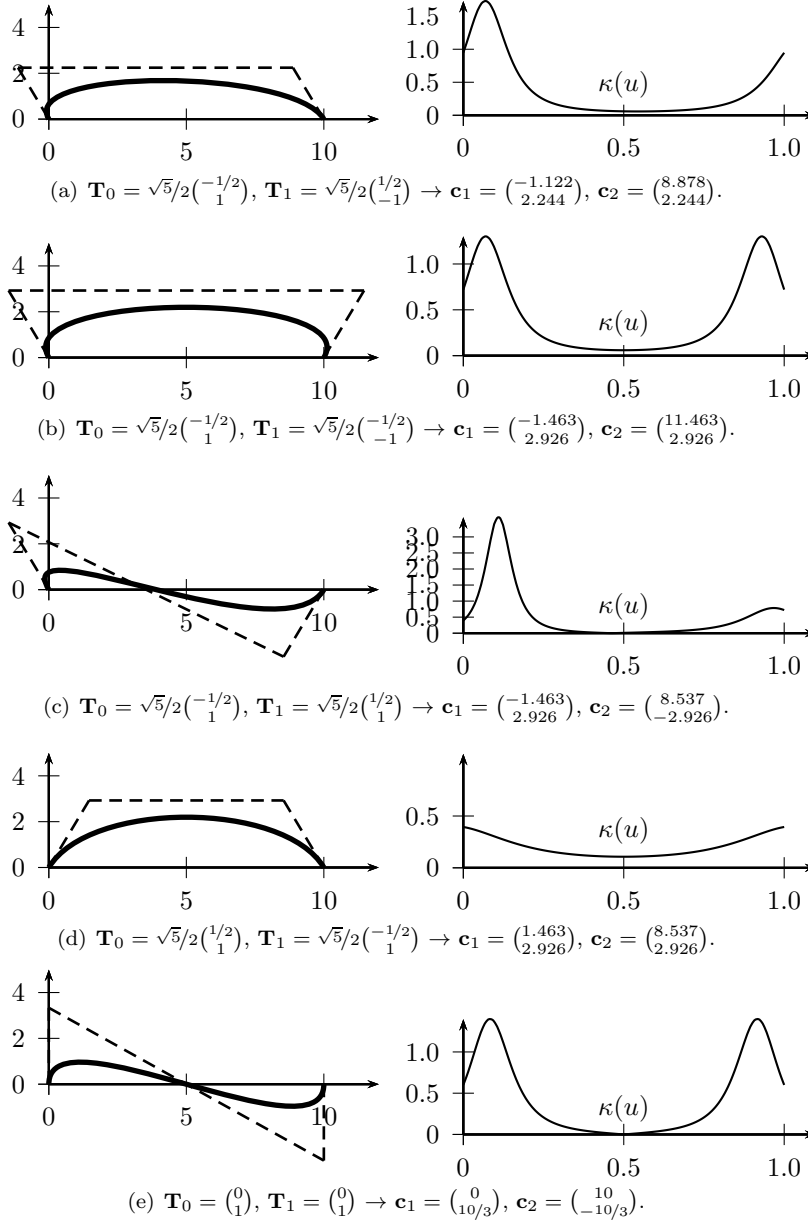


Figure III.6: Interpolation of $\mathbf{P}_0 = (0, 0)^T$ and $\mathbf{P}_1 = (10, 0)^T$ and varying tangent directions \mathbf{T}_0 and \mathbf{T}_1 scaled by L_0^{Avg} and L_1^{Avg} , respectively, as in Eq. (III.18).

Then, another Hermite interpolation of all points, each of which has now an associated tangent, yields a set of curves which join G^1 continuously. Note that in the above convex combination the weights of the tangent directions \mathbf{T}_i^- and \mathbf{T}_i^+ are both $1/2$. In our application it proves advantageous to equally weigh the influence of both curves, $\mathbf{g}_{i-1}(u)$ and $\mathbf{g}_i(u)$, onto \mathbf{T}_i , regardless of their respective arc length.

Example III.1.5. We consider two curve segments $\mathbf{g}_{i-1}(u)$ and $\mathbf{g}_i(u)$ with specification as in Figure III.7 and which do not connect smoothly.

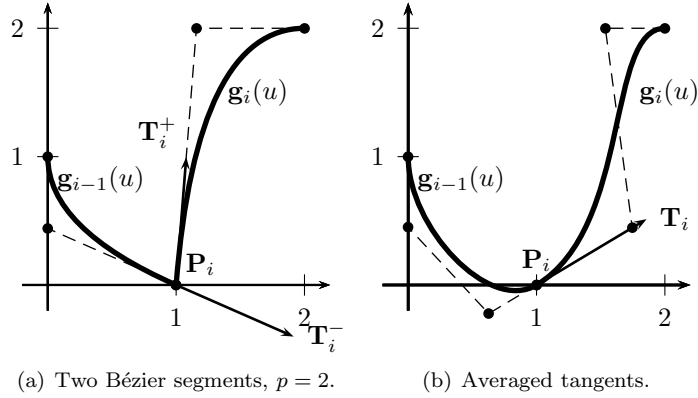


Figure III.7: G^1 joint of Bézier segments $\mathbf{g}_{i-1}(u)$ and $\mathbf{g}_i(u)$ as in example III.1.5.

They are the result of the local Hermite interpolation of the data

$$\mathbf{P}_{i-1} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \mathbf{T}_{i-1} = \begin{pmatrix} 0 \\ -1 \end{pmatrix}, \quad \mathbf{P}_i = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

and

$$\mathbf{P}_i = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad \mathbf{P}_{i+1} = \begin{pmatrix} 2 \\ 2 \end{pmatrix}, \mathbf{T}_{i+1} = \begin{pmatrix} 1 \\ 0 \end{pmatrix},$$

respectively. The tangent directions of $\mathbf{g}_{i-1}(u)$ and $\mathbf{g}_i(u)$ at \mathbf{P}_i are $\mathbf{T}_i^- = (0.916, -0.402)^T$ and $\mathbf{T}_i^+ = (0.079, 0.997)^T$, respectively. Their average is $\mathbf{T}_i = (0.858, 0.513)^T$. \triangleleft

The resulting curve can be represented straight forward as a B-spline curve of degree 3 (the degree of the segments $\mathbf{g}_i(u)$) with interior knots of multiplicity 3:

$$\mathbf{k} = \underbrace{(0, 0, 0, 0)}_4, \underbrace{(u_1, u_1, u_1)}_3, \dots, \underbrace{(u_{n_P-1}, u_{n_P-1}, u_{n_P-1})}_3, \underbrace{(1, 1, 1, 1)}_4.$$

The G^1 continuity is then ensured by the control point arrangement and the positions of the interior knots are determined by the parameters u_i which are associated with the interpolation points \mathbf{P}_i . If those are not provided by the input, the next section provides machinery to choose them adequately.

III.1.3 Interpolation Parameters and Basis

In typical situations, the interpolation points \mathbf{P}_i are not associated with a position u_i in the parametric domain. This correspondence and the specification of a basis are essential to render the conditions of Problem III.1.1 linear.

Parameter Choices: It is not customary to choose parameter values u_i for the interpolation points \mathbf{P}_i by uniformly spaced abscissae but rather by chord length or centripetal parametrization:

1. Uniform parametrization:

$$u_i = \frac{i}{n_P - 1} \quad i = 0, \dots, n_P - 1. \quad (\text{III.19})$$

This parametrization disregards the data arrangement in the geometry space and is, therefore, prone to producing unwanted results.

2. Chord length parametrization:

$$\begin{aligned} \Delta_i &= \|\mathbf{P}_i - \mathbf{P}_{i-1}\| \quad i = 1, \dots, n_P - 1, \\ \Delta &= \sum_{i=1}^{n_P-1} \Delta_i. \\ u_0 &= 0, \\ u_i &= u_{i-1} + \frac{\Delta_i}{\Delta} \quad i = 1, \dots, n_P - 1. \end{aligned} \quad (\text{III.20})$$

With this parametrization, the parameter spacing is proportional the distance between the data points.

3. Centripetal parametrization:

$$\begin{aligned} \Delta_i &= \sqrt{\|\mathbf{P}_i - \mathbf{P}_{i-1}\|} \quad i = 1, \dots, n_P - 1, \\ \Delta &= \sum_{i=1}^{n_P-1} \Delta_i. \\ u_0 &= 0, \\ u_i &= u_{i-1} + \frac{\Delta_i}{\Delta} \quad i = 1, \dots, n_P - 1. \end{aligned} \quad (\text{III.21})$$

This parametrization smooths out the variation in the centripetal force of a point moving along the curve, thus its denomination.

As different situations may suggest one method over another, there is no general rule about which one to employ. In this work, all three methods have been implemented with a default choice for the centripetal parametrization as it gives better results when the data takes sharp turns. For details to these parametrizations we refer to the pertinent literature, e.g. Lee [56] (1989). Compare the influences of the these three parametrizations on the results of the interpolation in Example III.1.11 on page 50.

Basis Determination: A B-spline basis of degree p and for n control points is determined by the locations of the $n - p - 1$ interior knots k_i^I which can be chosen uniformly:

$$k_i^I = \frac{i+1}{n-p} \quad i = 0, \dots, n-p-2. \quad (\text{III.22})$$

The alternative of *averaging* the parametrization values u_i is recommended in the global interpolation algorithm in Piegl [70] (1997):

$$\begin{aligned} k_i^I &= \frac{1}{p} \sum_{j=0}^{p-1} u_{i+1+j} \quad i = 0, \dots, n_P - p - 2. \\ \bar{\mathbf{k}}^0 &= (0, k_0^I, \dots, k_{n_P-p-2}^I, 1). \end{aligned} \quad (\text{III.23})$$

In this manner, we produce knot values $\bar{\mathbf{k}}^0$ which reflect the distribution of the parameter values u_i . However, this averaging is designed for an algorithm that interpolates only point data and our may include tangent and curvature information as well.

As we are also interested in a basis which produces more control points than the data demands, namely n , we regard the values of Eq. (III.23) as the “control values” of a one-dimensional Bézier expression of degree $\bar{n}_P - p$. Degree elevation of this expression increases the number of its control points while retaining their distribution, compare to Equation (A.1) of the Appendix. With the formula of multiple degree elevation by $d_k = n - (n_P - p + 1)$ we produce the vector of knot values $\bar{\mathbf{k}}$:

$$\bar{\mathbf{k}} = (\bar{k}_i)_{i=0}^{n-p} = \left(\sum_{j=\max(0, i-d_k)}^{\min(p, i)} \frac{\binom{p}{j} \binom{d_k}{i-j} \bar{k}_j^0}{\binom{p+d_k}{i}} \right)_{i=0}^{n-p} \quad (\text{III.24})$$

The overall knot vector is then obtained by inserting the missing boundary multiplicities:

$$\mathbf{k} = (\underbrace{0, \dots, 0}_p, \bar{k}_0, \bar{k}_1, \dots, \bar{k}_{n-p}, \underbrace{1, \dots, 1}_p).$$

Remark III.1.6 (Solvability). Note, that a basis constructed in this way can only guarantee solutions to Equation System (III.7) if only point data is given: [70]. In other cases, data can be produced such that the interpolation fails, compare to Example III.1.8. Note that in the construction of our model such cases are unlikely as our data is typically rather evenly spread in space. However, if such cases arise, enriching the basis by inserting proper knots is an effective work-around. ‡

Example III.1.7. In this example, we construct the basis in the following situation. We assume that the interpolation entities are given at the parameters $\mathbf{u} = (0, 0.2, 0.6, 0.75, 1)$. With tangent and curvature data given at positions 0, 1, 4 and 2, 4, respectively, we introduce the vector of multiplicities $\mathbf{m} = (2, 3, 1, 1, 3)$ associated with \mathbf{u} . Following Equation (III.23), we obtain the initial knots values, 3 of which are interior:

$$\bar{\mathbf{k}}^0 = (0, 0.267, 0.517, 0.783, 1).$$

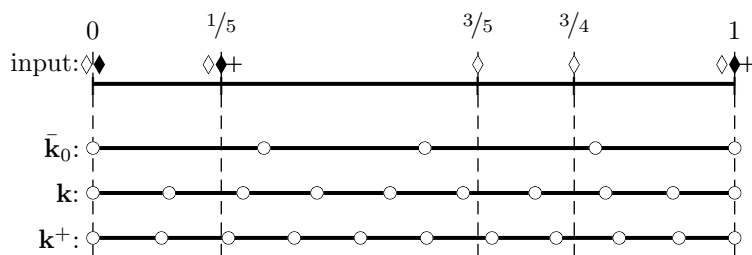


Figure III.8: Distribution of knots as in Example III.1.7.

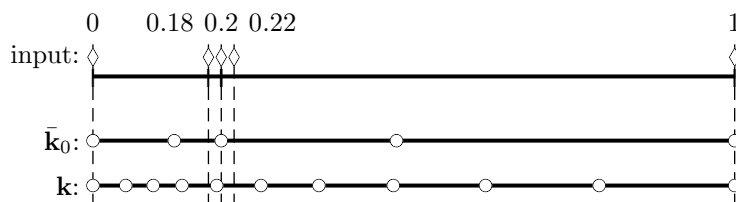


Figure III.9: Distribution of knots as in Example III.1.8.

The multiplicities \mathbf{m} determine the minimal number 12 of control point which implies that our basis has to have $8 = 12 - (3+1)$ interior knots. By elevating the degree by 5, Equation (III.24), we can comply with this constraint and obtain the knot vector

$$\mathbf{k} = (0, 0, 0, 0, 0.119, 0.234, 0.349, 0.463, 0.577, 0.689, 0.799, 0.904, 1, 1, 1, 1).$$

An illustration of this example can be found in Figure III.8 where we compare the input parameter values \mathbf{u} , the result of the averaging $\bar{\mathbf{k}}_0$, the knots after degree elevation \mathbf{k} and those of another degree elevation \mathbf{k}^+ . \triangleleft

Example III.1.8. To exhibit the fact that the distribution of knots retains that of the parametrization parameters we let the following be our given parameters:

$$\mathbf{u} = (0, 0.18, 0.2, 0.22, 1).$$

A basis of degree 3 and for 13 control point is produced by:

$$\begin{aligned} \bar{\mathbf{k}}^0 &= (0, 0.127, 0.2, 0.473, 1) \\ \mathbf{k} &= (0, 0, 0, 0, \\ &\quad 0.051, 0.094, 0.139, 0.193, 0.262, 0.352, 0.468, 0.612, 0.789, \\ &\quad 1, 1, 1, 1). \end{aligned}$$

Compare to Figure III.9.

We observe that two additional conditions, as could be tangential and curvature ones, at each of the parameters u_2 and u_3 would leave the knot span $[k_7, k_8)$ with six conditions for four involved control points. Interpolation is then not possible and the proper insertion of another knot is one way to work around this problem, see Remark III.1.6. \triangleleft

Example III.1.10 compares the results of the interpolation using the bases induced by (III.22) and by (III.23).

III.1.4 Approximation by a C^2 Curve

The equations in (III.7) tell us that we have n_P given interpolation points, n_t given interpolation tangents vectors, n_T given interpolation tangents directions and n_κ the number of given parameters with zero curvature which yield the minimal number of control points:

$$n_E := n_P + n_t + \frac{2}{3}n_T + n_\kappa. \quad (\text{III.25})$$

Note that here we suppose that the zero curvature parameter locations coincide with given tangents.

The basis determines the conditions which the resulting B-spline curve $\mathbf{c}(u)$ has to fulfil. If we choose more control points than needed for $\mathbf{c}(u)$, $n \geq n_E$, we gain degrees of freedom which are used to optimize the curve. At this point, we incorporate to the previously created piecewise Bézier curve $\mathbf{g}(u)$, see Subsection III.1.2, which we consider to be the optimum from a fairness point-of-view and which we, therefore, approximate in the least-squares sense for $m + 1$ samples $(\hat{u}_j)_{j=0}^m$:

$$e^g = \sum_{j=0}^m (e_j^g)^2 \rightarrow \min.$$

While this functional measures the fidelity of the curve under construction with $\mathbf{g}(u)$, we are also concerned with the “fairness” of the control polygon. Minimizing the sum of its squared lengths penalises longer control polygons than needed and thus avoids oscillations. It yields the following objective functional:

$$\begin{aligned} \Delta_i &:= \|\mathbf{c}_i - \mathbf{c}_{i-1}\| \\ e^\Delta &= \sum_{i=1}^{n-1} \Delta_i^2 \rightarrow \min. \end{aligned}$$

We shall use a convex combination of e^Δ and e^g , namely

$$\begin{aligned} Q(w^C) &:= w^C e^g + (1 - w^C) e^\Delta \\ Q(w^C) &\rightarrow \min, \end{aligned} \quad (\text{III.26})$$

where $w^C \in [0, 1]$. This functional is to be minimized under the conditions (III.7)-(III.8), which leads to a quadratic programming problem with linear constraints, which is solved by methods of linear algebra. Compare to the literature, e.g. Collatz and Wetterling [20] (1975) or Bertsekas [13] (2003).

Example III.1.9. We interpolate the following positions and tangent directions:

$$\begin{aligned} (\mathbf{P}_i)_i &= \left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 5 \\ 0 \end{pmatrix}, \begin{pmatrix} 20 \\ 5 \end{pmatrix}, \begin{pmatrix} 13 \\ 14 \end{pmatrix}, \begin{pmatrix} 15 \\ 15 \end{pmatrix} \right) \\ (\mathbf{T}_j)_j &= \left(\begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, -, \begin{pmatrix} 2/\sqrt{5} \\ 1/\sqrt{5} \end{pmatrix} \right) \end{aligned}$$

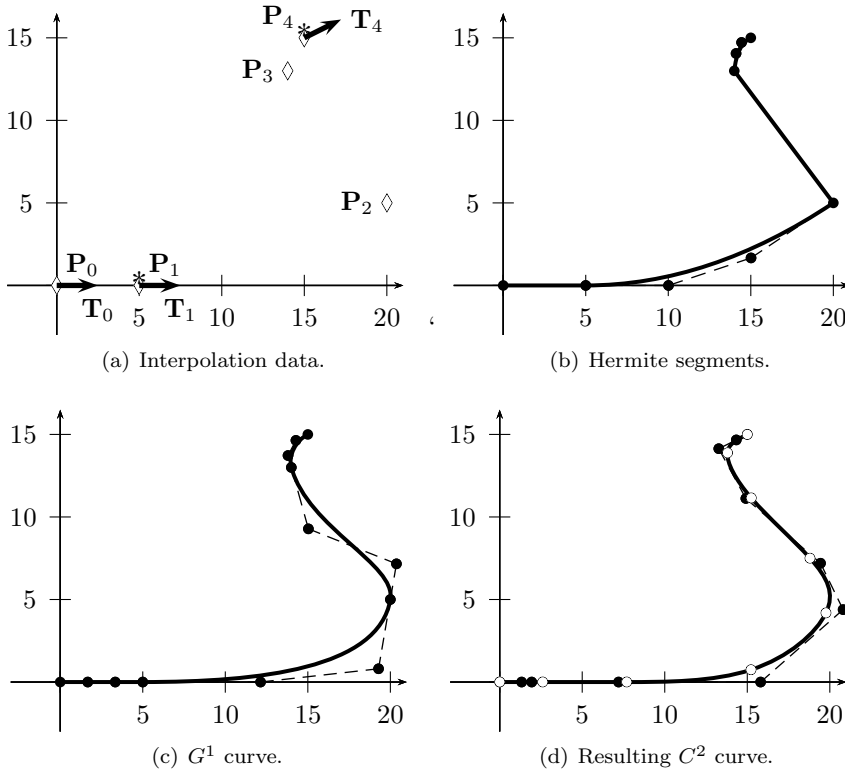


Figure III.10: Illustration of the interpolation in Example III.1.9.

Also, we want to impose a zero-curvature condition at \mathbf{P}_1 and \mathbf{P}_4 . To do this, we first find the appropriate parameter values $(u_i)_{i=0}^4$ of these points, as in Section III.1.3, and then impose $\kappa(u_1) = 0$ and $\kappa(u_4) = 0$. This example is shown in Figure III.10: (a) illustrates the data, (b) shows the calculated Hermite segments, in (c) we see the G^1 continuously joined reference curve $\mathbf{g}(u)$ for our optimisation and finally we obtain the resulting C^2 curve in (d).

To produce these results we use the centripetal parametrization of the data, Eq. (III.21), and the knot values as in Eq. (III.24) with 7 interior knots:

$$\mathbf{u} = (0, 0.206, 0.572, 0.862, 1)$$

$$\mathbf{k} = (0, 0, 0, 0, 0.13, 0.265, 0.403, 0.54, 0.672, 0.795, 0.906, 1, 1, 1, 1)$$

Note that the data demands the minimum of $10 = \lceil \frac{29}{3} \rceil$ control points, thus 6 interior knots. In order to create degrees of freedom for the optimisation, we produce the basis \mathbf{k} with 7 interior knots. This combination of data and basis leaves 4 degrees of freedom more than constraints which we use to minimize $Q(\frac{1}{2})$ as in Equation (III.26). \triangleleft

Example III.1.10. We consider the input of the above Example III.1.9 and compare the knot vector constructed there with the uniform knot vector:

$$\mathbf{k} = (0, 0, 0, 0, 1/8, 1/8, 1/4, 3/8, 1/2, 5/8, 3/4, 7/8, 1, 1, 1, 1).$$

With all the other specifications left unaltered, the results for both are shown in Figure III.11 where we compare the curvature distribution, the approximation error and the ratio of two consecutive control polygon side lengths:

Knots	κ		e_j^g		Δ_{i+1}/Δ_i		
	max	mean	max	mean	max	mean	min
uniform	1.41	0.178	0.79	0.257	5.475	1.513	0.181
adaptive	1.296	0.17	0.869	0.327	2.895	1.453	0.116

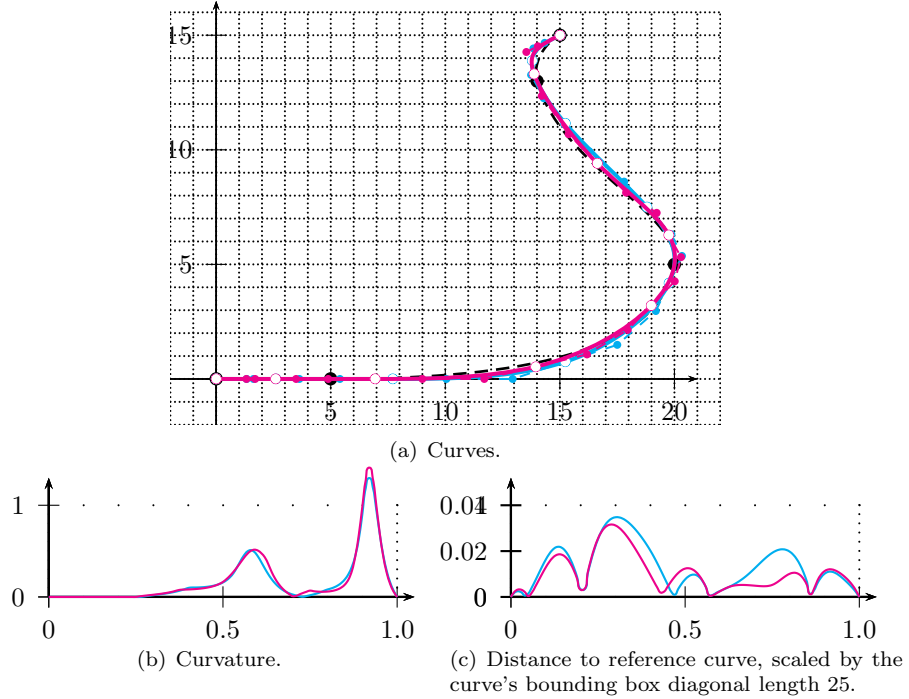


Figure III.11: Comparison of knots distributions as in Example III.1.10: adaptive (cyan) vs. uniform (magenta). The reference curve $\mathbf{g}(u)$ is marked dashed black.

We observe that for this example, the curvature distributions are similar while the uniform knot distribution seems to have slightly better performance with respect to the approximation of $\mathbf{g}(u)$. A comparison of the ratio of consecutive control polygon side lengths, however, favours the adaptive knot distribution. \triangleleft

Example III.1.11. Based on the original Example III.1.9 we compare with different choices for the parametrization while leaving the other specifications unchanged. Additionally to the above centripetal parameters \mathbf{u}^C , we consider the chord length parameters \mathbf{u}^L and uniform abscissae \mathbf{u}^U :

$$\mathbf{u}^U = (0, 0.25, 0.5, 0.75, 1)$$

$$\mathbf{u}^L = (0, 0.151, 0.63, 0.932, 1)$$

$$\mathbf{u}^C = (0, 0.206, 0.572, 0.862, 1)$$

These parameter choices imply the respective knot vectors, obtained by averaging as in (III.23):

$$\mathbf{k}^U = (0, 0, 0, 0, 0.125, 0.25, 0.375, 0.5, 0.625, 0.75, 0.875, 1, 1, 1, 1)$$

$$\mathbf{k}^L = (0, 0, 0, 0, 0.13, 0.271, 0.417, 0.563, 0.701, 0.825, 0.927, 1, 1, 1, 1)$$

$$\mathbf{k}^C = (0, 0, 0, 0, 0.13, 0.265, 0.403, 0.54, 0.672, 0.795, 0.906, 1, 1, 1, 1)$$

The results for all three are shown in Figure III.12.

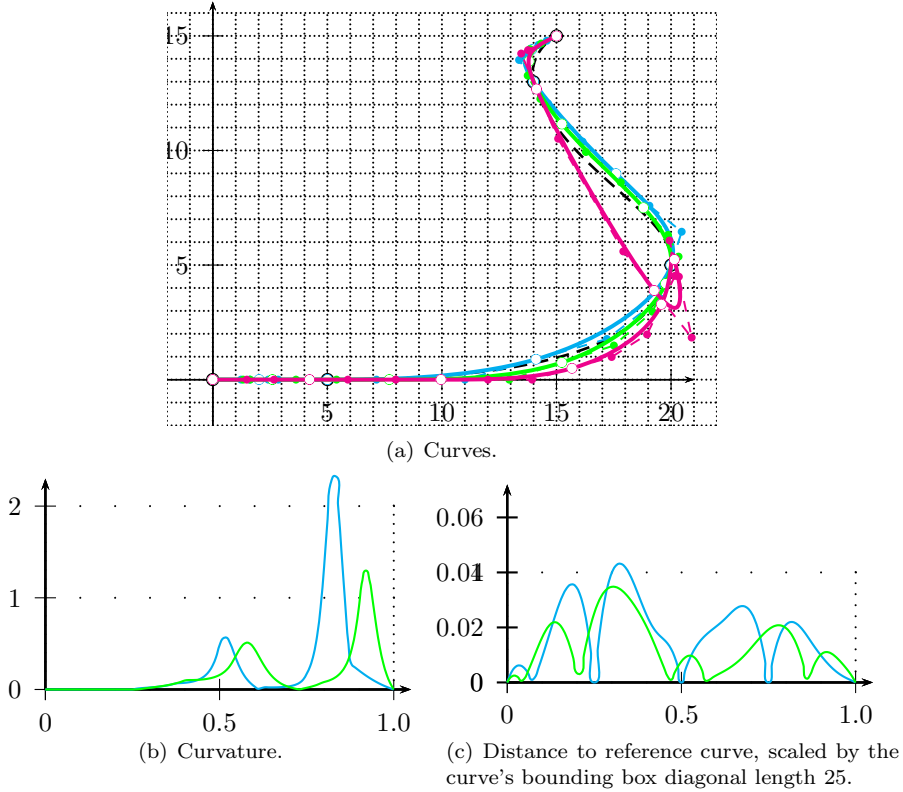


Figure III.12: Comparison of parametrizations as in Example III.1.11: uniform (cyan), chord-length (magenta), centripetal (green). The reference curve $\mathbf{g}(u)$ is marked dashed black.

We compare the curvature distribution, the approximation error and the ratio of two consecutive control polygon side lengths:

P'zation	κ		e_j^g		Δ_{i+1}/Δ_i		
	max	mean	max	mean	max	mean	min
\mathbf{u}^U	2.329	0.225	1.079	0.448	6.6	1.737	0.111
\mathbf{u}^L	56.179	0.959	6.568	1.174	20.804	2.972	0.402
\mathbf{u}^C	1.296	0.17	0.869	0.327	2.895	1.453	0.116

Note the unacceptable shape of the curve employing of the chord length parametrization, a problem associated with this parametrization which has been addressed in, e.g., [41] and [56]. \triangleleft

Example III.1.12. We consider the original Example III.1.9 and compare different choices for the approximation weight $w^C = 0, 1/2, 1$. Again, all the other specifications are retained. Results for all three are shown in Figure III.13.

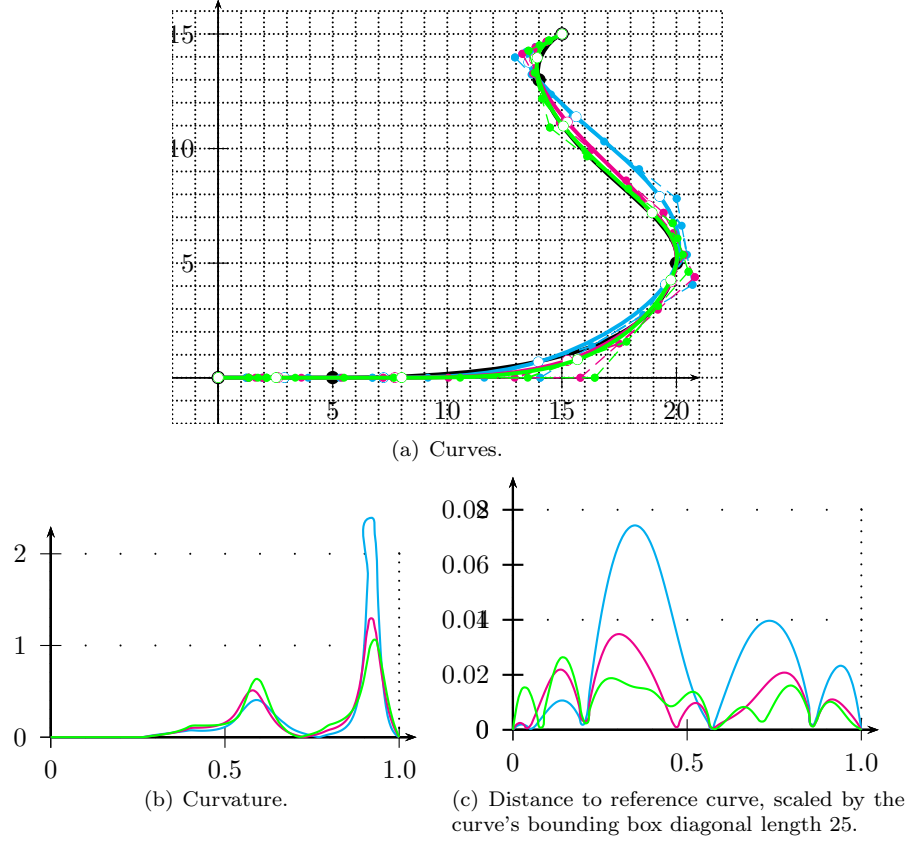


Figure III.13: Comparison of approximation weights as in Example III.1.12: $w^C = 0$ (cyan), $w^C = 1/2$ (magenta), $w^C = 1$ (green). The reference curve $\mathbf{g}(u)$ is marked dashed black.

We compare the curvature distribution, the approximation error and the ratio of two consecutive control polygon side lengths:

w^C	κ		e_j^g		Δ_{i+1}/Δ_i		
	max	mean	max	mean	max	mean	min
0	2.329	0.187	1.858	0.668	3.539	1.264	0.331
1/2	1.296	0.17	0.869	0.327	2.895	1.453	0.116
1	1.064	0.175	0.658	0.268	3.496	1.676	0.124

As expected, the quality of the approximation and that of the control polygon length minimization behave inverse proportionally: For increasing approximation weight w^C the approximation error decreases and the ratio of the lengths of two consecutive control polygon sides increases. \triangleleft

Example III.1.13. At last, we compare the results for different degrees of freedom. Therefore, we consider the data and parameters of Example III.1.9 with the following knot vectors, all of which are produced by Equation (III.24):

$$\mathbf{k}^1 = (0, 0, 0, 0, 0.148, 0.304, 0.463, 0.617, 0.763, 0.892, 1, 1, 1, 1),$$

$$\mathbf{k}^2 = (0, 0, 0, 0, 0.13, 0.265, 0.403, 0.54, 0.672, 0.795, 0.906, 1, 1, 1, 1),$$

$$\mathbf{k}^3 = (0, 0, 0, 0, 0.115, 0.235, 0.357, 0.479, 0.599, 0.713, 0.82, 0.916, 1, 1, 1, 1).$$

In these case we have 5, 8, 11 degrees of freedom, respectively, and \mathbf{k}^2 corresponds to the knot vector \mathbf{k} in the original example.

Results for all three are shown in Figure III.14.

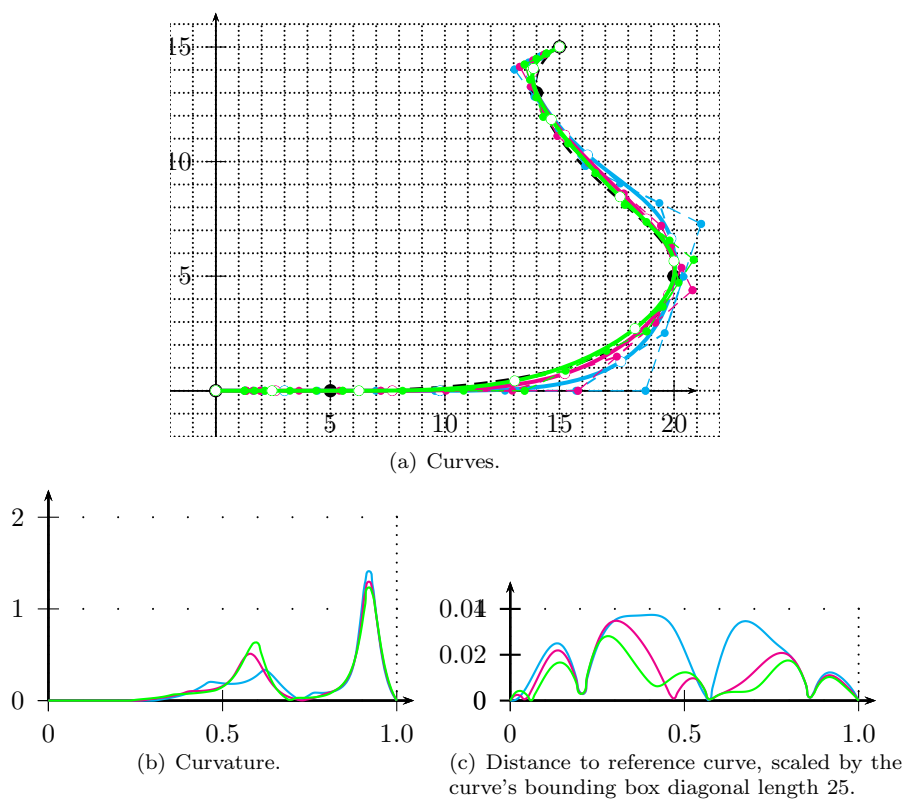


Figure III.14: Comparison of degrees of freedom as in Example III.1.13: 5 (cyan), 8 (magenta) and 11 (green).

We compare the curvature distribution and the approximation error:

\mathbf{k}	d.o.f.	κ		e_j^g		Δ_{i+1}/Δ_i		
		max	mean	max	mean	max	mean	min
\mathbf{k}^1	5	1.411	0.167	0.935	0.51	4.1	1.466	0.099
\mathbf{k}^2	8	1.296	0.17	0.869	0.327	2.895	1.453	0.116
\mathbf{k}^3	11	1.234	0.173	0.703	0.255	2.261	1.285	0.262

For higher degrees of freedom, the approximation quality improves and the lengths of the control polygon arms are evened out. \triangleleft

III.1.5 The Interpolation Algorithm

We summarize the procedures of this section in one algorithm:

Algorithm III.1.14 (Interpolation). Regard the input as in the stated Problem III.1.1 in the beginning of the section and the target degree p , the number n^+ of extra control points and the approximation parameter w^C :

input	Points $(\mathbf{P}_i)_i$	$i \in I = (0, \dots, n_P)$
	Tangent vectors $(\mathbf{t}_{j_1})_{j_1}$	$j_1 \in J_1 \subset I$
	Tangent directions $(\mathbf{T}_{j_2})_{j_2}$	$j_2 \in J_2 \subset I, J_2 \cap J_1 = \emptyset$
	Zero-curvature parameters $(u_i^c)_i$	$i \in (0, \dots, n_\kappa - 1)$
	Degree p	$p \geq 3$
	Extra control points n^+	$n^+ \geq 0$
	Approximation weight w^C	$0 \leq w^C \leq 1$
output	Knot vector \mathbf{k}	
	Control points $(\mathbf{c}_i)_i$	

- 1: Determine centripetal parameters $(u_i)_i$. ▷ Eq. (III.21)
 - 2: **for** $i = 0, \dots, n_P - 1$ **do**
 - 3: Create Bézier curves $\mathbf{H}_i(u)$, $u \in [u_i, u_{i+1}]$. ▷ S. III.1.1
 - 4: Join $(\mathbf{H}_i(u))_{i=0}^{n_P-1}$ into $\mathbf{G}(u)$. ▷ S. III.1.2
 - 5: Number of control points: $n = n_P + |J_1| + \frac{2}{3}|J_2| + n_\kappa + n^+$.
 - 6: Average $(u_i)_i$ to obtain $\bar{\mathbf{k}}_0$. ▷ Eq. (III.23)
 - 7: “Degree elevate” $\bar{\mathbf{k}}_0$ to obtain $\bar{\mathbf{k}}$ and the knot vector \mathbf{k} . ▷ Eq. (III.24)
 - 8: Generate equality conditions with E . ▷ Eq. (III.8)
 - 9: Minimize $Q(w^C)$ under conditions E . ▷ Eq. (III.8), (III.26)
 - 10: **return** \mathbf{k} and $(\mathbf{c}_i)_{i=0}^{n-1}$.
-

A-Posteriori Manipulation: Since in most applications we have no clear picture of the tangent lengths a-priori, we usually interpolate tangent directions. The result is likely to suit our purposes as we can confirm in Section IV. However, situation may arise where we wish to control the tangent lengths based on the result of the interpolation with tangent directions. In such a case, we replace the original tangent directions \mathbf{T}_j by tangent vectors \mathbf{t}_j which are obtained by the evaluation of $\mathbf{c}'(u_j)$. By construction, these tangent vectors are parallel to the input direction. By rescaling them appropriately and executing a new interpolation problem with tangent *vectors* instead of the direction, we gain valuable control over the curve. Note, that such this produces a denser basis for the resulting curve as tangent vectors demand more degrees of freedom than tangent directions.

III.2 Enhancing Curves with Frames

As we have discussed in Section II.1.2, we lay the groundwork for a G^1 continuous patch network by enhancing each of the wireframe’s curves with a tangent plane distribution. In Subsection III.2.1, we formally define the *frame* as the distribution of normal vectors to the curve for Bézier and B-spline curves and

we constitute its characteristic conditions. We also regard all cases of possible boundary conditions which might occur in this work. Finally, we review the construction of the rotation-minimizing frame (Section III.2.2), which we consider the optimum frame and which we, therefore, employ to optimise our frame: Section III.2.3.

III.2.1 Frame Conditions

We can define a distribution of tangent planes of a curve $\mathbf{c}(u)$ by a distribution of normal vectors $\mathbf{n}(u)$. The minimal condition for this vector field is the orthogonality to the curve's distribution of (unit) tangent vectors $\mathbf{T}(u)$; compare to Figure III.15.

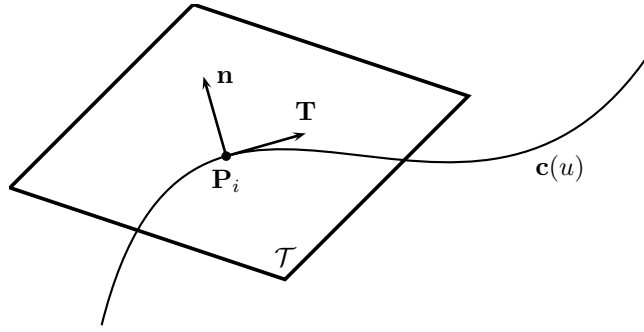


Figure III.15: The tangent plane \mathcal{T} of a curve $\mathbf{c}(u)$ at the point $\mathbf{P}_i = \mathbf{c}(u_i)$.

Definition III.2.1. For a given curve $\mathbf{c}(u)$ we regard an associated vector field $\mathbf{f}(u)$. If it is perpendicular to the distribution of tangent vectors of the curve, i.e. if

$$\forall u \in [0, 1] : \quad \mathbf{f}(u) \perp \mathbf{c}'(u), \quad (\text{III.27})$$

then we shall denominate $\mathbf{f}(u)$ as a *frame* of $\mathbf{c}(u)$. $\#$

Remark III.2.2. In this denomination we follow the example of the Frenet frame which is an orthonormal basis, constituted by the tangent, the (main) normal and the bi-normal. For sufficiently smooth curves with arc length parametrization the tangent is uniquely defined and the main normal is the unit vector in the direction of the tangent's derivative. The bi-normal is defined as the cross product of the tangent and the main normal. In the case of our frame $\mathbf{f}(u)$, we find ourselves with an orthonormal basis as well if we regard the triplet of vectors

$$\left(\frac{\mathbf{c}'(u)}{\|\mathbf{c}'(u)\|}, \frac{\mathbf{f}(u)}{\|\mathbf{f}(u)\|}, \frac{\mathbf{c}'(u) \times \mathbf{f}(u)}{\|\mathbf{c}'(u) \times \mathbf{f}(u)\|} \right).$$

Like for the Frenet frame, the first vector is the tangent and the third one is the cross product of the first two vectors. Due to this analogy, we adopt the term *frame* and for the sake of simplicity we employ it to refer to $\mathbf{f}(u)$ instead of the above triplet of vectors. $\#$

The orthogonality in Equation (III.27) can be expressed with the scalar product of the corresponding fields, $\mathbf{c}'(u) \cdot \mathbf{f}(u) = 0$, whose explicit form highly depends on the representations of $\mathbf{c}(u)$ and $\mathbf{f}(u)$.

The Bézier Case: We consider a curve in Bézier form of degree p ,

$$\mathbf{c}(u) = \sum_{i=0}^p N_i^p(u) \mathbf{c}_i,$$

and construct a frame $\mathbf{f}(u)$ with an analogue representation of degree d and with the control vectors \mathbf{f}_i . We refer to Appendix A.2 to see that a scalar product of vector fields in Bézier form can be written as a scalar Bézier expression $X(u)$ whose degree is the sum of the degrees of the two fields:

$$0 = \mathbf{c}'(u) \cdot \mathbf{f}(u) =: X(u) = \sum_{i=0}^{p-1+d} N_i^{p-1+d}(u) x_i.$$

To provide orthogonality between the fields, the $p+d$ control values x_i of the scalar product have to be zero. With $\boldsymbol{\tau}_i := p(\mathbf{c}_{i+1} - \mathbf{c}_i)$ being the p control vectors of $\mathbf{c}'(u)$, we obtain a system of $p+d$ linear equations for the $d+1$ unknown control vectors of $\mathbf{f}(u)$ as we can compare in Equation (A.3):

$$0 = x_i = \sum_{j=\max(0, i-d)}^{\min(p-1, i)} \binom{p-1}{j} \binom{d}{i-j} \boldsymbol{\tau}_j \cdot \mathbf{f}_{i-j}$$

$$\stackrel{k:=i-j}{=} \sum_{k=\max(i-(p-1), 0)}^{\min(i, d)} \binom{p-1}{i-k} \binom{d}{k} \boldsymbol{\tau}_{i-k} \cdot \mathbf{f}_k, \quad 0 \leq i \leq d+p.$$

With

$$B_{ik} := \binom{p-1}{i-k} \binom{d}{k} \quad \text{for } \begin{cases} 0 \leq i \leq d+p \\ \max(i-(p-1), 0) \leq k \leq \min(i, d) \end{cases}$$

we obtain the equation system

$$E^\perp (\mathbf{f}_0, \dots, \mathbf{f}_d)^T = 0. \quad (\text{III.28})$$

where the system matrix E^\perp is of banded structure with maximum band width $\min(p-1, d) + 1$,

$$E^\perp := \begin{pmatrix} \boldsymbol{\tau}_0 \cdot & & & & 0 \\ B_{10} \boldsymbol{\tau}_1 \cdot & B_{11} \boldsymbol{\tau}_0 \cdot & & & \\ B_{20} \boldsymbol{\tau}_2 \cdot & B_{21} \boldsymbol{\tau}_1 \cdot & B_{22} \boldsymbol{\tau}_0 \cdot & & \\ & & \ddots & & \\ & B_{d+p-3, d-2} \boldsymbol{\tau}_{p-1} \cdot & B_{d+p-3, d-1} \boldsymbol{\tau}_{p-2} \cdot & B_{d+p-3, d} \boldsymbol{\tau}_{p-3} \cdot & \\ & & B_{d+p-2, d-1} \boldsymbol{\tau}_{p-1} \cdot & B_{d+p-2, d} \boldsymbol{\tau}_{p-2} \cdot & \\ 0 & & & & \boldsymbol{\tau}_{p-1} \cdot \end{pmatrix}, \quad (\text{III.29})$$

and where we interpret all vector multiplications as scalar products.

The B-spline Case: In the case of B-spline curves with interior knots we can obtain similar conditions as found in Equation System (III.28). However, their derivation becomes quite complicated, even if the curve basis and the frame

basis have the same knot positions. Due to its elevated complexity, we avoid dealing with orthogonality of the full B-spline case and rather construct the frame as a piecewise Bézier curve in which proper conditions provide the wanted continuity. The vector of $\nu + 1$ different knot positions $\bar{\mathbf{k}} := (0, \bar{k}_1, \dots, \bar{k}_{\nu-1}, 1)$ yields ν Bézier segments $(\mathbf{c}^0(u), \dots, \mathbf{c}^{\nu-1}(u))$ and for the frame $\mathbf{f}^i(u)$ of each of these segments we obtain a matrix E_i of orthogonality conditions, like the above (III.29), such that $E_i(\mathbf{f}_0^i, \dots, \mathbf{f}_d^i)^T = 0$.

For the incorporation of proper continuity conditions, it is suitable to compose *one* equation system comprising the conditions for all frame parts $\mathbf{f}^i(u)$. Therefore, we compose from all E_i a matrix of block diagonal structure and for $\nu(d + 1)$ control vectors. However, for all $i \in (1, \dots, \nu - 1)$, the first equation of E_i and the last one of E_{i-1} express the same orthogonality due to the curve continuity, compare to Matrix (III.29). A C^0 continuous frame is, therefore, provided if $\mathbf{f}_d^{i-1} = \mathbf{f}_0^i$ holds. This condition can be fulfilled on the level of the basis by introducing the following knot vector,

$$\mathbf{k} := (k_i)_{i=0}^{d(\nu+1)} = \underbrace{(0, \dots, 0)_{d+1}} \underbrace{(\bar{k}_1, \dots, \bar{k}_1)_d} \dots \underbrace{(\bar{k}_{\nu-1}, \dots, \bar{k}_{\nu-1})_d} \underbrace{(1)_{d+1}}, \quad (\text{III.30})$$

yielding a basis for the $\nu d + 1$ control vectors \mathbf{f}_j of $\mathbf{f}(u)$ where

$$\mathbf{f}_{id} = \mathbf{f}_d^{i-1} = \mathbf{f}_0^i, \quad i = 1, \dots, \nu - 1,$$

compare to Figure III.16.

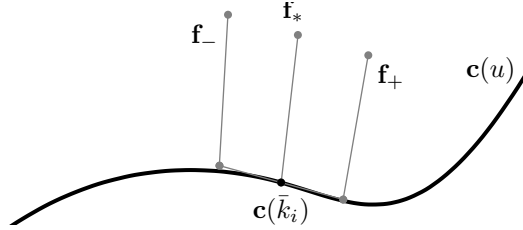


Figure III.16: The frame control vectors $\mathbf{f}_- := \mathbf{f}_{d-1}^{i-1}$, $\mathbf{f}_* := \mathbf{f}_{id} = \mathbf{f}_d^{i-1} = \mathbf{f}_0^i$ and $\mathbf{f}_+ := \mathbf{f}_1^i$ around the interior knot \bar{k}_i of the $\mathbf{c}(u)$.

The overall system matrix is affected in two ways: Firstly and for all $i > 0$, the redundant first equation of E_i is erased yielding \tilde{E}_i . Secondly and also due to the introduced basis, we have to align the first column of \tilde{E}_{i+1} with the last one of \tilde{E}_i to obtain the equation system

$$E^\perp (\mathbf{f}_0, \dots, \mathbf{f}_{d\nu})^T = 0$$

with the system matrix E^\perp of the following structure:

$$E^\perp = \begin{pmatrix} (\cdot & E_0 & \cdot) & & & & 0 \\ & (\cdot & \tilde{E}_1 & \cdot) & & & \\ & & (\cdot & \tilde{E}_2 & \cdot) & & \\ & & & & \ddots & & \\ 0 & & & & & (\cdot & \tilde{E}_{\nu-1} & \cdot) \end{pmatrix}. \quad (\text{III.31})$$

Remark III.2.4 (Requirement of C^2 Continuous Curves). In general, for a given B-spline curve $\mathbf{c}(u)$ a frame $\mathbf{f}(u)$ can only be constructed C^1 continuously and in B-spline representation if $\mathbf{c}(u)$ is at least C^2 continuous.

To prove this, we represent $\mathbf{c}(u)$ and $\mathbf{f}(u)$ as piecewise Bézier expressions with the help of proper insertion of knots and regard, henceforth, the joint of the two Bézier segments $\mathbf{c}^{i-1}(u)$ and $\mathbf{c}^i(u)$. We abbreviate the involved control vectors of the curve's derivative by $\boldsymbol{\tau}_- := \boldsymbol{\tau}_{d-1}^{i-1}$, $\boldsymbol{\tau}_* := \boldsymbol{\tau}_d^{i-1} = \boldsymbol{\tau}_0^i$ and $\boldsymbol{\tau}_+ := \boldsymbol{\tau}_1^i$. The control vectors of the frame parts $\mathbf{f}^{i-1}(u)$ and $\mathbf{f}^i(u)$ are abbreviated analogously by $\mathbf{f}_- := \mathbf{f}_{d-1}^{i-1}$, $\mathbf{f}_* := \mathbf{f}_d^{i-1} = \mathbf{f}_0^i$ and $\mathbf{f}_+ := \mathbf{f}_1^i$. The orthogonality conditions of Eq. (III.28) which of exclusively involve these vectors then read:

$$B_{d+p-2,d-1} \boldsymbol{\tau}_* \cdot \mathbf{f}_- + B_{d+p-2,d} \boldsymbol{\tau}_- \cdot \mathbf{f}_* = 0, \quad (\text{e1})$$

$$\boldsymbol{\tau}_* \cdot \mathbf{f}_* = 0, \quad (\text{e2})$$

$$B_{1,1} \boldsymbol{\tau}_* \cdot \mathbf{f}_+ + B_{1,0} \boldsymbol{\tau}_+ \cdot \mathbf{f}_* = 0. \quad (\text{e3})$$

With $\Delta_- := \Delta_{i-1}$ and $\Delta_+ := \Delta_i$ and with Eq. (e2), the C^1 condition (III.32) reduces Equation (e3) to:

$$B_{1,1} \boldsymbol{\tau}_* \cdot \mathbf{f}_- - \frac{\Delta_-}{\Delta_+} B_{1,0} \boldsymbol{\tau}_+ \cdot \mathbf{f}_* = 0. \quad (\text{e3})$$

Since $B_{d+p-2,d-1} = B_{1,1}$ and $B_{d+p-2,d} = B_{1,0}$, Equations (e1) and (e3) express the same equality if the curve is C^2 continuous as then, for $\boldsymbol{\tau}_+$ and $\boldsymbol{\tau}_-$ the following holds

$$\frac{p-1}{\Delta_+} (\boldsymbol{\tau}_+ - \boldsymbol{\tau}_*) = \frac{p-1}{\Delta_-} (\boldsymbol{\tau}_* - \boldsymbol{\tau}_-).$$

If the curve is C^1 continuous but only G^2 continuous, we have that $\boldsymbol{\tau}_+ = \boldsymbol{\tau}_* + c \frac{\Delta_+}{\Delta_-} (\boldsymbol{\tau}_* - \boldsymbol{\tau}_-)$ for a constant $c \neq 1$. Then, Equation (e3) becomes

$$B_{1,1} \boldsymbol{\tau}_* \cdot \mathbf{f}_- + c B_{1,0} \boldsymbol{\tau}_- \cdot \mathbf{f}_* = 0$$

which does not express the same equality as Eq. (e1). In fact, these two equations represent an intersection problem of two parallel planes. $\#$

Proposition III.2.5. *Be ν the number of Bézier segments of $\mathbf{c}(u)$. For the construction of a frame we have the following sufficient conditions:*

For $\nu > 2$:

$$d \geq \frac{p}{2} + 1.$$

For $\nu = 2$:

$$d \geq \frac{p}{2}.$$

For $\nu = 1$ (the Bézier case):

$$d \geq \frac{p-1}{2} - 1.$$

Proof. The result follows by comparing the number of unknowns and equations in the homogeneous Equation System (III.34). For each Bézier segment, we have $3(d+1)$ unknowns for $p+d$ equations stemming from the orthogonality conditions. Interior Bézier segments of the curve, i.e. the restriction of the curve

to an interval of two consecutive interior knots, are the most constrained cases as both of their bounds carry C^1 conditions which account for 3 additional equations each. As proven in Remark III.2.4, we can remove one equation from the orthogonality conditions of each interior Bézier segment and thus obtain that $3(d+1) \geq p+d-1+6$.

If we have only two Bézier segments, both of them constitute a boundary segment. In this case, we have a redundant equation on only one of the two segments and, thus, on the more constrained part we remain to have $p+d$ equations from the orthogonality conditions and 3 equations from the continuity conditions.

In the case of a Bézier curve, we have no continuity conditions present and, therefore, d has to fulfil $3(d+1) \geq p-1+d+1$. \blacksquare

Given Boundary Frames: In typical situations the construction of a frame is performed with frame data posed on the end points of the curve. Such data, i.e. the boundary conditions in the frame construction, affect the first and second control vectors of the respective boundary and are always assumed to comply with respective first and last two equations of E^\perp . However, the resulting linear system of equations is rendered non-homogeneous and solvability is no longer implied by providing more degrees of freedom than equations.

In the first case, we regard the frame construction with the knowledge of the frame \mathbf{F}_0 at the start and the frame \mathbf{F}_{-0} at the end of the curve. It is obvious that these two frames are as well the first and last control vector $\mathbf{f}_0 = \mathbf{F}_0$ and $\mathbf{f}_{\nu d} = \mathbf{F}_{-0}$ for ν Bézier segments. With the conventions of Remark III.2.3 we have the following equation system of frame constraints:

$$\begin{aligned} E(\mathbf{f}_0, \dots, \mathbf{f}_{d\nu})^T &= \begin{pmatrix} E^\perp \\ E^C \\ E^B \end{pmatrix} (\mathbf{f}_0, \dots, \mathbf{f}_{d\nu})^T \\ &= \begin{pmatrix} \frac{E^\perp}{E^C} \\ \frac{1}{1} & 0 & \dots \\ \dots & & 0 & 1 \end{pmatrix} (\mathbf{f}_0, \dots, \mathbf{f}_{d\nu})^T = \begin{pmatrix} 0 \\ 0 \\ \mathbf{F}_0 \\ \mathbf{F}_{-0} \end{pmatrix} \quad (\text{III.35}) \end{aligned}$$

The number n of columns and the number m of rows of E and depend on the frame degree d :

Proposition III.2.6. *We regard a spatial B-spline curve $\mathbf{c}(u)$ of degree p and the frame-control vectors \mathbf{F}_0 and \mathbf{F}_{-0} , which are perpendicular to the curve at the parameters $u = 0$ and $u = 1$, respectively, as given. We guarantee that $m \geq n$ for $\nu = 1$ (the Bézier case) if*

$$d \geq \frac{p+1}{2}$$

and in the B-spline case with interior knots if

$$d \geq \frac{p}{2} + 1.$$

Proof. For Bézier curves, we have $2 \cdot 3$ more equations from the given data, which eliminate 2 equations from the orthogonality condition.

For $\nu > 1$, we have at least one Bézier segment at the boundary with 3 equations from the C^1 continuity and 3 from the given boundary vector, which, in turn, leaves 1 of the $p + d$ equations from the orthogonality condition as solved. ■

Remark III.2.7. In order to provide a statement about the solvability of System (III.35), the rank of E has to be determined which is a central topic of linear algebra and not trivial for general matrices. The experience gained in this work indicates that a frame constructed with minimal degree is, in general, not satisfactorily fair and, thus, a higher degree is customarily employed. To provide acceptable frames, it suffices, in most cases, to employ a degree which is higher by 1 than the minimum. ‡

Given Compatibility Conditions: We are going to encounter the case where the boundary control vectors are given and we have an extra condition on the second control vectors. These conditions originate from the demands of compatible ribbons, see Section III.3, in particular Proposition III.3.3 and Remark III.3.4. In this case, the equation system reads as follows:

$$\begin{aligned}
 E(\mathbf{f}_0, \dots, \mathbf{f}_{d\nu})^T &= \begin{pmatrix} \frac{E^\perp}{E^C} \\ \frac{E^{B_0}}{E^{b_1}} \end{pmatrix} (\mathbf{f}_0, \dots, \mathbf{f}_{d\nu})^T \\
 &= \begin{pmatrix} \frac{E^\perp}{E^C} \\ 1 & 0 & \dots \\ & \dots & 0 & 1 \\ 0 & \mathbf{a} & 0 & \dots \\ & \dots & 0 & \mathbf{b} & 0 \end{pmatrix} (\mathbf{f}_0, \dots, \mathbf{f}_{d\nu})^T = \begin{pmatrix} 0 \\ 0 \\ \mathbf{F}_0 \\ \mathbf{F}_{-0} \\ x_a \\ x_b \end{pmatrix}
 \end{aligned} \tag{III.36}$$

Again, we follow the above conventions, interpret the entries of E^{b_1} as scalar products and examine the relation of rows m and columns n of E (the comments of Remark III.2.7 hold also here):

Corollary III.2.8. *We are given the situation of the above Proposition III.2.6 and the additional vectors \mathbf{a} and \mathbf{b} , which are not parallel to $\boldsymbol{\tau}_0$ and $\boldsymbol{\tau}_{p-1}$, respectively, and the corresponding right-hand-side values x_a and x_b such that $\mathbf{a} \cdot \mathbf{f}_1 = x_a$ and $\mathbf{b} \cdot \mathbf{f}_{\nu d-1} = x_b$. The inequality $m \geq n$ holds if*

$$d \geq \frac{p+3}{2}. \tag{III.37}$$

Proof. For a Bézier curve we have two additional equations to fulfil with respect to Proposition III.2.6. As for the B-spline setting, the most constrained cases are the boundary segments in each of which we have one additional equation. ■

Two Given Boundary Frames: An even more demanding case is the following one for the explicitly given first two and last two control vectors. In this

case, we have the following equation system:

$$\begin{aligned}
 E(\mathbf{f}_0, \dots, \mathbf{f}_{d\nu})^T &= \begin{pmatrix} \frac{E^\perp}{E^C} \\ \frac{E^{B_0}}{E^{B_1}} \end{pmatrix} (\mathbf{f}_0, \dots, \mathbf{f}_{d\nu})^T \\
 &= \begin{pmatrix} \frac{E^\perp}{E^C} \\ 1 & 0 & \dots \\ & \dots & 0 & 1 \\ 0 & 1 & 0 & \dots \\ & \dots & 0 & 1 & 0 \end{pmatrix} (\mathbf{f}_0, \dots, \mathbf{f}_{d\nu})^T = \begin{pmatrix} 0 \\ 0 \\ \mathbf{F}_0 \\ \mathbf{F}_{-0} \\ \mathbf{F}_1 \\ \mathbf{F}_{-1} \end{pmatrix}
 \end{aligned} \tag{III.38}$$

The same conventions as proposed in Remark III.2.3 hold and we with Remark III.2.7 the dimensions (m, n) of E depend on the frame degree d :

Corollary III.2.9. *We are given the situation of Proposition III.2.6 and, additionally, the second and second-to-last control vectors \mathbf{F}_1 and \mathbf{F}_{-1} . We assume them to be valid, i.e. they solve the second and second-to-last orthogonality condition of E^\perp : Eq. (III.29). For $m \geq n$ to hold, d has to fulfil*

$$d \geq \frac{p+5}{2} \tag{III.39}$$

for Bézier curves and for curves with interior knots:

$$d \geq \frac{p}{2} + 2. \tag{III.40}$$

Proof. In the Bézier case we have 12 additional constraints. These, however, render four orthogonality conditions to tautologies.

As for the B-spline case, the most constrained parts of the curve are the boundary segments in which we have 6 additional constraints from the boundary data eliminating 2 equations from the orthogonality condition and 3 additional equation from the C^1 continuity. ■

III.2.2 The Rotation-Minimizing Frame

For a given curve and with $\mathbf{T}(u)$ being the distribution of its unit-tangent, we regard a triplet of orthonormal vectors $(\mathbf{T}(u), \phi(u), \mathbf{T}(u) \times \phi(u))$ which is called a *rotation minimizing frame* (RMF) if it does not rotate about the instantaneous tangent, i.e. its angular velocity satisfies $\omega \mathbf{T} \equiv 0$, compare to Bishop [14] (1975) and Wang et.al. [85] (2008). Since the RMF is defined by $\phi(u)$, we again abbreviate notation and simply refer to $\phi(u)$ as the RMF.

RMF is defined for any regular C^1 continuous curve and its minimal-twist property and stable behaviour in the presence of inflection points (unlike the Frenet frame) make it the preferred frame in many applications in computer graphics. Figure III.17 shows a comparison to the Frenet frame for a B-spline curve and clearly exhibits its intrinsic property. We, therefore, declare the RMF as the optimal frame for our applications.

Introduced as the solution of an ordinary differential equation [14], it is generally not possible to calculate the RMF in closed form. In [85], we find a

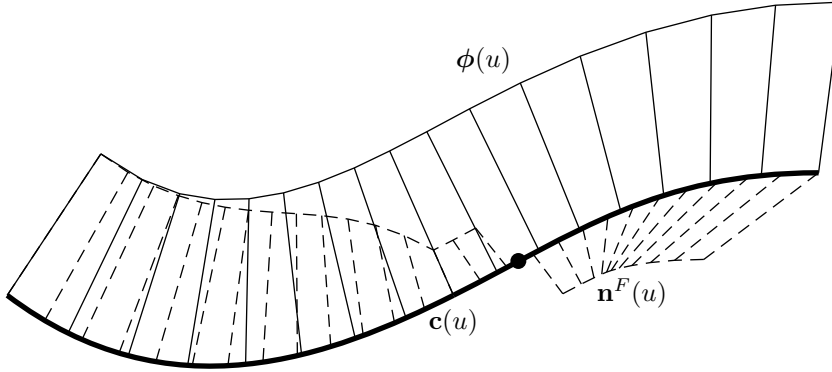


Figure III.17: The RMF $\phi(u)$ (solid) is shown with the main-normal $\mathbf{n}^F(u)$ (dashed) of the Frenet frame. Note the jump of $\mathbf{n}^F(u)$ at the inflection point of the curve.

stable and efficient method, the double reflection method, to accurately compute the RMF at a sequence of sampled points.

Double Reflection Method: We regard the sample parameters $(u_i)_{i=0}^m$, at which we calculate the $m + 1$ curve points \mathbf{x}_i and unit tangents \mathbf{T}_i , and the initial frame ϕ_0 with $\phi_0 \perp \mathbf{c}'(u_0)$

For every i , we first construct the bisecting plane \mathcal{P}_1 of two consecutive points \mathbf{x}_i and \mathbf{x}_{i+1} : Fig. III.18(a). We use this plane to reflect \mathbf{T}_i and ϕ_i from \mathbf{x}_i onto \mathbf{x}_{i+1} to obtain \mathbf{T}_i^L and ϕ_i^L , respectively, Fig. III.18(b).

In the next step, we construct another plane, \mathcal{P}_2 , which bisects the points $\mathbf{x}_{i+1} + \mathbf{T}_i^L$ and $\mathbf{x}_{i+1} + \mathbf{T}_{i+1}$, Fig. III.18(c). In this plane, which clearly contains \mathbf{x}_{i+1} , we reflect ϕ_i^L to obtain ϕ_{i+1} : Fig. III.18(d).

We leave the proof of this algorithm's validity to the mentioned literature but point out that since reflections preserve angles, we ensure that $\mathbf{T}_i^L \perp \phi_i^L$ and $\mathbf{T}_{i+1} \perp \phi_{i+1}$. A set of RMF samples $(\phi_i)_{i=0}^m$ is created.

RMF with Boundary Conditions: The double reflection method is developed for a given frame at the start of the curve but can be adapted for data on both ends: Let a given end frame $\hat{\phi}_m$ deviate from the above constructed ϕ_m by a certain angle. We do this by adding rotation linearly proportionally to the arc length parameter as this choice for the distribution of the additional rotation minimizes the total squared angular speed, see [85].

The illustration of an RMF construction can be found in Figure III.19 whose input is that of Example III.2.10.

III.2.3 Optimisation of the Frame

In Section III.2.1 we have seen suggestions for the lower bounds of the degree of the frame in all relevant cases. If these bounds are not integral numbers or if we choose a higher degree for the frame, we have degrees of freedom which we use for optimisation with respect to two criteria.

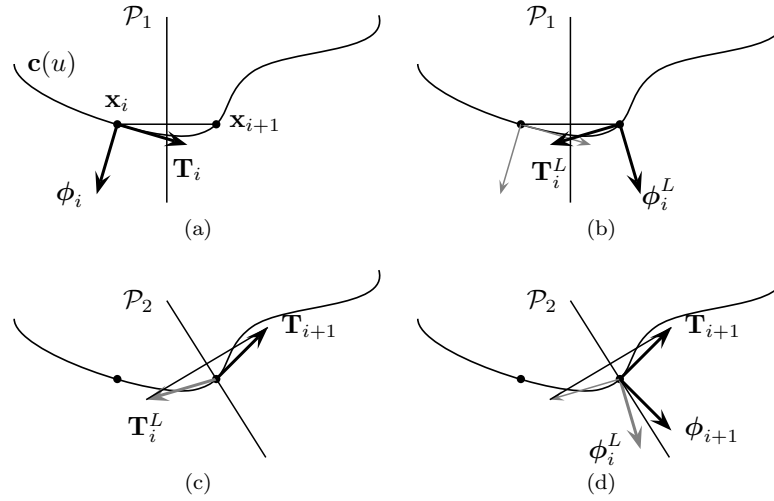


Figure III.18: The double reflection method to calculate RMF-samples ϕ_i of $\mathbf{c}(u)$.

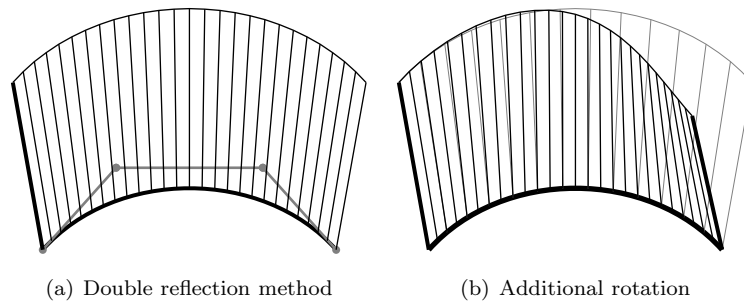


Figure III.19: The calculation of the RMF: (a) Double reflection method. (b) Additional rotation to fulfil boundary frame at the end.

The first is the approximation of the above defined RMF in the least square sense which reads as follows for $m + 1$ samples $(u_j)_{j=0}^m$:

$$\begin{aligned} e_j^\phi &:= \|\phi_j - \mathbf{f}(u_j)\| = \|\phi_j - \sum_{i=0}^{\nu d} \mathbf{f}_i N_i^d(u_j)\|, \\ e^\phi &= \sum_{j=0}^m (e_j^\phi)^2 \rightarrow \min. \end{aligned} \quad (\text{III.41})$$

The second criterion is the minimization of the frame control polygon for which we interpret the frame as a curve and write the expression for the νd frame control “points” \mathbf{f}_i . This criterion penalises longer control polygons than needed and avoids oscillations:

$$\begin{aligned} \Delta_i &:= \|\mathbf{f}_i - \mathbf{f}_{i-1}\|, \\ e^\Delta &:= \sum_{i=1}^{\nu d} \Delta_i^2 \rightarrow \min. \end{aligned} \quad (\text{III.42})$$

A convex combination of (III.41) and (III.42) with the parameter w^F constitutes the overall objective functional:

$$\begin{aligned} Q(w^F) &:= w^F e^\phi + (1 - w^F) e^\Delta, \\ Q(w^F) &\rightarrow \min. \end{aligned} \quad (\text{III.43})$$

Minimisation of $Q(w^F)$ under the linear conditions comprised in the corresponding matrix E leads to a quadratic programming problem with equality constraints, see e.g. [20, 13].

Example III.2.10. We regard the Bézier curve of degree $p = 3$ and with the control points:

$$\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 3 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 4 \\ 0 \\ -1 \end{pmatrix}.$$

We want to construct a frame with the boundary vectors $\mathbf{F}_0 = 1/\sqrt{6}(-1, -1, 2)^T$ and $\mathbf{F}_{-0} = 1/\sqrt{6}(-1, -2, 1)^T$ which are orthogonal to curve at the respective end points.

The following is a comparison of the maximum and the mean e_j^ϕ and of the maximum and mean Δ_i as we alter the parameters d and w^F . While Proposition III.2.6 suggests that the degree d of the frame has to be greater or equal to 2, we regard degrees 3 and 4 in order to obtain degrees of freedom for the optimisation. For the purpose of illustrating a limit case, we compare the results

to those of degree 10.

Bézier		e_j^ϕ		Δ_i	
d	w^F	max	mean	max	mean
3	0	0.148	0.121	0.577	0.555
	1/2	0.132	0.093	0.577	0.563
	1	0.133	0.089	0.577	0.569
4	0	0.148	0.121	0.408	0.364
	1/2	0.132	0.091	0.418	0.37
	1	0.132	0.086	0.509	0.459
10	0	0.155	0.126	0.172	0.12
	1/2	0.052	0.036	0.165	0.143
	1	2.38×10^{-05}	9.32×10^{-06}	0.229	0.164

For a fixed w^F higher degrees yield better results and for a fixed degree by increasing the approximation parameter w^F , the error of the RMF approximation decreases but cause the control vectors to vary more. In Figure III.20 we see this construction with the approximation weight $w^F = 1/2$ and the degrees $d = 3$ and $d = 4$. ◁

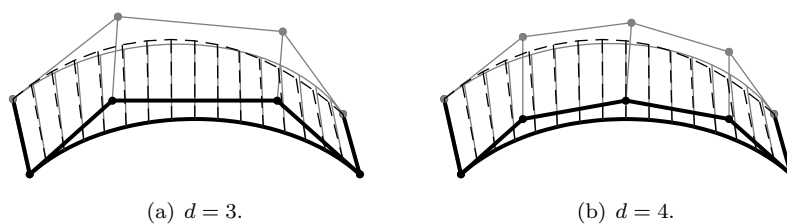


Figure III.20: Shown is Example III.2.10 for enhancing a Bézier curve with a frame of degree 3 and degree 4. respectively. In both cases we observe samples of the constructed frame and its frame control polygon in solid grey and samples of the RMF in dashed black.

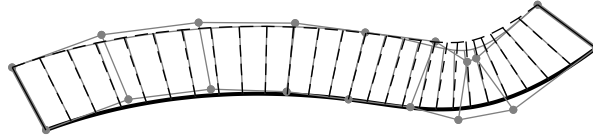
Example III.2.11. We are given the B-spline curve of degree $p = 3$, with one interior knot $k_4 = 1/2$ and the control points:

$$\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 3 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 4 \\ 0 \\ -1 \end{pmatrix}, \begin{pmatrix} 5 \\ 0 \\ 0 \end{pmatrix}.$$

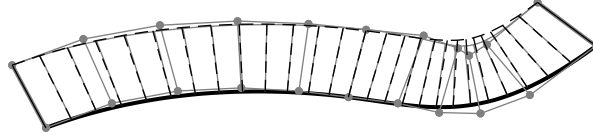
As above we are given boundary frame vectors which are orthogonal to curve at the respective end points: $\mathbf{F}_0 = 1/\sqrt{6}(-1, -1, 2)^T$ and $\mathbf{F}_{-0} = 1/\sqrt{2}(-1, 0, 1)^T$

We compare the same measures as in Example III.2.10:

B-spline		e_j^ϕ		Δ_i	
d	w^F	max	mean	max	mean
4	0	0.358	0.241	0.445	0.193
	1/2	0.178	0.062	0.596	0.242
	1	0.043	0.022	1.006	0.317
5	0	0.354	0.241	0.349	0.152
	1/2	0.142	0.046	0.411	0.191
	1	0.038	0.011	0.68	0.278
10	0	0.349	0.243	0.166	0.074
	1/2	0.069	0.019	0.217	0.093
	1	0.0029	0.0004	0.653	0.291



(a) $d = 4$.



(b) $d = 5$.

Figure III.21: Shown is Example III.2.11 for enhancing a B-Spline curve with frames of degrees 4 and 5. We compare both cases to the RMF samples in black and observe the constructed frame and its control polygon in solid grey.

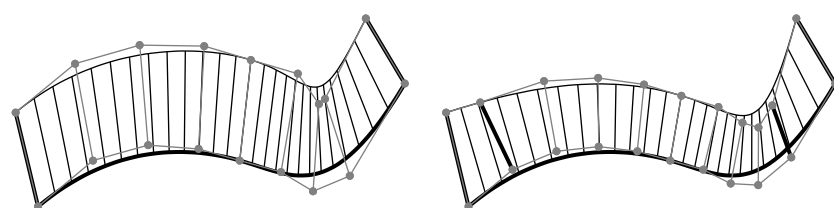
We obtain similar results as above. For $w^F = 1/2$, the cases of $d = 4$ and $d = 5$ can be compared to the RMF in Figure III.21. \triangleleft

Example III.2.12. We consider the same data as in the previous Example III.2.11 and additionally impose compatibility conditions as in Equation (III.36):

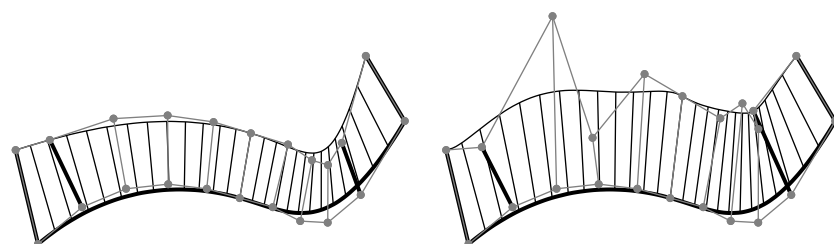
$$\mathbf{f}_1 \cdot \begin{pmatrix} -4.518 \\ 0.266 \\ -2.126 \end{pmatrix} = 1.362, \quad \mathbf{f}_3 \cdot \begin{pmatrix} -0.648 \\ 0 \\ 0.762 \end{pmatrix} = -0.340275.$$

In Figure III.22 we compare the constructed frames with and without regard of these extra boundary conditions.

In general, b_1 conditions can be given in a way such that the RMF is no longer optimal in the neighbourhood of the boundaries. In such cases, emphasis on the approximation of the RMF by elevating w^F yields bad results, as can be observed in this example. We compare again the measures mentioned in



(a) No compatibility conditions, $w^F = 1/2$. (b) With compatibility conditions, $w^F = 0$.



(c) With compatibility conditions, $w^F = 1/2$. (d) With compatibility conditions, $w^F = 1$.

Figure III.22: Shown is Example III.2.12 for enhancing a B-Spline curve with frames of degree 5 and different approximation weights under b_1 boundary conditions. We compare the cases of $w^F = 0, 1/2, 1$ to the case with no b_1 boundary conditions and $w^F = 1/2$.

Example III.2.10 and conclude with similar observations as above:

d	Comp. Cond. w^F	e_j^ϕ		Δ_i	
		max	mean	max	mean
5	0	0.394	0.312	0.465	0.247
	1/2	0.188	0.074	0.494	0.268
	1	0.114	0.034	1.266	0.582
6	0	0.363	0.275	0.382	0.189
	1/2	0.158	0.057	0.368	0.208
	1	0.078	0.02	1.53	0.545
10	0	0.33	0.231	0.225	0.095
	1/2	0.100	0.030	0.218	0.11
	1	0.037	0.003	5.893	1.488

◁

III.2.4 The Framing Algorithm

Algorithm III.2.13 (Framing). Given the B-spline curve $\mathbf{c}(u)$ of degree p and frames \mathbf{F}_0 and \mathbf{F}_{-0} which are orthogonal to the curve at its start and end, respectively: $\mathbf{F}_0 \perp \mathbf{c}'(0)$ and $\mathbf{F}_{-0} \perp \mathbf{c}'(1)$. We may be given additional boundary conditions as in Corollary III.2.8:

$$\text{cond}_a = (\mathbf{a}, x_a), \quad \text{cond}_b = (\mathbf{b}, x_b). \quad (\text{III.44})$$

Alternatively to these conditions and under the condition of validity with respect to E^\perp , we may be given the second control vectors from the boundaries \mathbf{F}_1 and \mathbf{F}_{-1} as in Corollary III.2.8.

The algorithm to construct the frame $\mathbf{f}(u)$ of $\mathbf{c}(u)$ expects this data, the frame-degree elevation d^+ and the averaging parameter w as inputs:

input	$\mathbf{c}(u)$ \mathbf{F}_0 \mathbf{F}_{-0} d^+ w^F	$\mathbf{F}_0 \perp \mathbf{c}(0)$ $\mathbf{F}_{-0} \perp \mathbf{c}(1)$ $d^+ \geq 0$ $w^F \in [0, 1]$
optional input	cond_a or \mathbf{F}_1 cond_b or \mathbf{F}_{-1}	
output	degree d Control vectors $(\mathbf{f}_i)_{i=0}^{\nu d}$	

- 1: Determine d^{Min} . ▷ Prop. III.2.6, Cor. III.2.8
- 2: $d = \lceil d^{\text{Min}} \rceil + d^+$.
- 3: Frame knot vector: \mathbf{k} . ▷ Eq. (III.30)
- 4: Split $\mathbf{c}(u)$ at $(\bar{k}_i)_{i=1}^{\nu-1}$: $(\mathbf{c}^i(u))_{i=0}^{n^k-1}$
- 5: **for** $i = 0, \dots, n^k - 1$ **do**
- 6: Create coefficient matrix E_i . ▷ Eq. (III.28)
- 7: Join $(E_i)_i$: $E^\perp = \text{diag}((E_i)_i)$. ▷ Eq. (III.31)
- 8: Continuity conditions: E^C . ▷ Eq. (III.34)
- 9: Boundary conditions: E^{B_0} . ▷ Eq. (III.35)
- 10: **if** $\exists \text{cond}_a$ **and** $\exists \text{cond}_b$ **then**

- | | |
|---|----------------|
| 11: Boundary conditions: E^{b_1} . | ▷ Eq. (III.36) |
| 12: else if $\exists \mathbf{F}^1$ and $\exists \mathbf{F}^{n-1}$ then | |
| 13: Boundary conditions: E^{B_1} . | ▷ Eq. (III.38) |
| 14: Join $E^\perp, E^C, E^{B_0}, E^{b_1}, E^{B_1}$ to E . | ▷ Eq. (III.34) |
| 15: Create RMF samples: $(\phi_j)_{j=0}^m$. | ▷ S. III.2.2 |
| 16: Minimize $Q(w^F)$ under conditions E . | ▷ Eq. (III.43) |
| 17: return d and control vectors $(\mathbf{f}_i)_{i=0}^d$. | |
-

III.3 From Frames to Ribbons

The previous section has provided us with an algorithm to specify tangent planes along a curve, i.e. along the boundaries of patches. It is the purpose of the following paragraphs to prepare this information for the bicubically blended Coons patch construction. The behaviour of these patches in the neighbourhood of their boundaries is governed by a set of boundary ribbons whose construction is the next methodical step.

Ribbons are constructed analogously to frames as they too constitute a vector field which is orthogonal to another given one. The difference to frames is that ribbons always occur in the context of a loop of curves. While the scale of frames was immaterial, that of ribbons is central.

Firstly, we define ribbons for the Bézier- and B-Spline-case and establish their characteristic conditions in Section III.3.1. Secondly, we provide the conditions for compatibility of ribbons in Section III.3.2. We conclude the construction of ribbons with their optimisation, Section III.3.4, for which we previously introduce the notion of *rotation-minimizing ribbon* in Section III.3.3.

III.3.1 Cross-Boundary Ribbon Conditions

For a given curve $\mathbf{c}(u)$, we demand its *boundary ribbon* $\mathbf{r}(u)$ to be perpendicular to its given frame $\mathbf{f}(u)$:

$$\forall u \in [0, 1]: \quad \mathbf{r}(u) \perp \mathbf{f}(u). \quad (\text{III.45})$$

As in the case of the frames, this orthogonality can be expressed in terms of the scalar product $0 = \mathbf{f}(u) \cdot \mathbf{r}(u)$ and we also impose similar boundary conditions.

The Bézier Case: We start, as above, in the Bézier setting and regard the frame of degree d for which we construct a ribbon of degree δ . The scalar Bézier function $X(u)$ expresses the orthogonality of the two fields and is of degree $d + \delta$:

$$0 = \mathbf{f}(u) \cdot \mathbf{r}(u) =: X(u) = \sum_{i=0}^{d+\delta} N_i^{d+\delta}(u) x_i.$$

To ensure $X(u) = 0$ for all $u \in [0, 1]$, all $d + \delta + 1$ control values x_i have to be zero which yields a system of $d + \delta + 1$ linear equations for the $\delta + 1$ unknown ribbon control vectors \mathbf{r}_i , analogously to the Equation system (III.28) for frames:

$$E^\perp (\mathbf{r}_0, \mathbf{r}_1, \dots, \mathbf{r}_\delta)^T = 0, \quad (\text{III.46})$$

For the determination of the ribbon's minimal degree we take the following details into account:

- As opposed to frames, the notion of boundary ribbons is always linked to a patch and their first and last (control) vectors are fully determined by the neighbouring curves' tangents, see Section III.4.
- In the context of compatibility, see below in Subsection III.3.2, we also determine the second and last-but-one ribbon control vector by minimizing the twist vector in the vertex.

The four control vectors \mathbf{R}_0 , \mathbf{R}_1 , \mathbf{R}_{-1} and \mathbf{R}_{-0} are, therefore, determined beforehand and we assume that they comply with the above orthogonality conditions:

$$\begin{aligned}
 0 &= \mathbf{R}_0 \cdot \mathbf{f}_0, \\
 0 &= \binom{d}{1} \binom{\delta}{0} \mathbf{R}_0 \cdot \mathbf{f}_1 + \binom{d}{0} \binom{\delta}{1} \mathbf{R}_1 \cdot \mathbf{f}_0, \\
 0 &= \binom{d}{1} \binom{\delta}{0} \mathbf{R}_{-0} \cdot \mathbf{f}_{\nu d-1} + \binom{d}{0} \binom{\delta}{1} \mathbf{R}_{-1} \cdot \mathbf{f}_{\nu d}, \\
 0 &= \mathbf{R}_{-0} \cdot \mathbf{f}_{\nu d}.
 \end{aligned}$$

We express all conditions on the ribbon as in Eq. III.38 and mark the matrix of the boundary conditions by E^B and the matrix from the C^1 -conditions, as in Equation (III.49), by E^C :

$$E(\mathbf{r}_i)_i = \begin{pmatrix} E^\perp \\ E^C \\ E^B \end{pmatrix} (\mathbf{r}_i)_i = \begin{pmatrix} \frac{E^\perp}{E^C} \\ \hline 1 & 0 & \dots \\ 0 & 1 & \dots \\ & & \dots & 1 & 0 \\ & & \dots & 0 & 1 \end{pmatrix} (\mathbf{r}_i)_{i=0}^{\nu\delta} = \begin{pmatrix} 0 \\ 0 \\ \mathbf{R}_0 \\ \mathbf{R}_1 \\ \mathbf{R}_{-1} \\ \mathbf{R}_{-0} \end{pmatrix} \quad (\text{III.51})$$

The same conventions as proposed in Remark III.2.3 hold and in this case, the first and last two equations become tautologies with the given frame vectors. Comparing the number of rows m and columns n of E yields the following lower bound for δ :

Proposition III.3.2. *For $m \geq n$ to hold, the ribbon degree δ has to fulfil*

$$\delta \geq \frac{d}{2} + 2$$

in the case of $\nu > 1$ and

$$\delta \geq \frac{d}{2} + 3$$

in the case of $\mathbf{f}(u)$ in Bézier form (if $\nu = 1$).

Proof. The proof is analogous to Corollary III.2.9 with the roles of $p - 1$ and d taken here by d and δ , respectively. \blacksquare

III.3.2 Compatibility and Vertex Twist Vectors

The behaviour of a bicubically blended Coons patch in its vertices is governed by *two* ribbons, one for each curve. This can result in possibly conflicting ways and only if the ribbons have *compatible* behaviour, a Coons patch can be constructed to comply with both ribbons.

The conditions of orthogonality between ribbons and frames do not always allow the construction of such compatible ribbons and we shall see in Proposition III.3.3 that we have to be aware of compatibility already from the point of frame construction by imposing appropriate conditions then.

The Compatibility Condition: We go into detail and state the conditions for the ribbons of the bicubically blended Coons patch which we elaborate in Section III.4.2. To this end we consider the vertex where the curves $\mathbf{a}(u)$ and $\mathbf{d}(v)$ meet at $u = 0$ and $v = 0$, respectively. The respective given frames and unknown ribbons are $\mathbf{f}^a(u)$, $\mathbf{f}^d(v)$ and $\mathbf{r}^a(u)$, $\mathbf{r}^d(v)$. We also assume that $\mathbf{f}^a(0) = \mathbf{f}^d(0)$.

1. The start- and end-vectors of each ribbon have to match the *tangents* of the respective neighbour curves:

$$\mathbf{r}^d(0) = \mathbf{a}'(0) \quad (\text{III.52})$$

$$\mathbf{r}^a(0) = \mathbf{d}'(0). \quad (\text{III.53})$$

2. We formally define compatibility by the matching of the derivatives of two ribbons in a vertex:

$$\lim_{u \rightarrow 0} \frac{\partial}{\partial u} \mathbf{r}^a(u)|_{u=0} = \lim_{v \rightarrow 0} \frac{\partial}{\partial v} \mathbf{r}^d(v)|_{v=0}. \quad (\text{III.54})$$

which reads in B-Spline representation as follows

$$\frac{\delta^a}{\Delta^a} (\mathbf{r}_1^a - \mathbf{r}_0^a) = \frac{\delta^d}{\Delta^d} (\mathbf{r}_1^d - \mathbf{r}_0^d) \quad (\text{III.55})$$

where we assume bases with δ^a and δ^d interior knots, respectively. Δ^a and Δ^d are the respective lengths of the knot intervals at the common vertex of $\mathbf{a}(u)$ and $\mathbf{d}(v)$.

We examine the ribbon's orthogonality condition (III.46) in the context of both of these two curves $\mathbf{a}(u)$ or $\mathbf{d}(v)$ and see that its first two equations are fully determined by the terms involved above in (III.55). With the given \mathbf{r}_0^a and \mathbf{r}_0^d we can disregard the first equations. It is the respective second equation of (III.46) which requires close attention if linked with the linear relation (III.55) between \mathbf{r}_1^a and \mathbf{r}_1^d . We rewrite this equation for $\mathbf{a}(u)$ and for $\mathbf{d}(v)$ with the explicit coefficients $B_{10} = d$ and $B_{11} = \delta$, compare to Section A.2:

$$\mathbf{a}(u) : \quad 0 = d^a \mathbf{f}_1^a \cdot \mathbf{r}_0^a + \delta^a \mathbf{f}_0^a \cdot \mathbf{r}_1^a, \quad (\text{III.56})$$

$$\mathbf{d}(v) : \quad 0 = d^d \mathbf{f}_1^d \cdot \mathbf{r}_0^d + \delta^d \mathbf{f}_0^d \cdot \mathbf{r}_1^d. \quad (\text{III.57})$$

Since this triplet of equations, (III.55)-(III.57), is constituted of 5 equations for 6 unknowns, the question of solvability might not be obvious. However,

substituting \mathbf{r}_1^d in (III.57) by (III.55) reveals the difficulty:

$$-d^a \mathbf{f}_1^a \cdot \mathbf{r}_0^a = \delta^a \mathbf{f}_0^a \cdot \mathbf{r}_1^a \quad (\text{III.58})$$

$$-\frac{\Delta^a}{\Delta^d} d^d \mathbf{f}_1^d \cdot \mathbf{r}_0^d = \delta^a \mathbf{f}_0^d \cdot \mathbf{r}_1^a - \delta^a \underbrace{\mathbf{r}_0^a \cdot \mathbf{f}_0^d}_{=0} + \frac{\Delta^a}{\Delta^d} \delta^a \underbrace{\mathbf{r}_0^d \cdot \mathbf{f}_0^d}_{=0}. \quad (\text{III.59})$$

These two equations express the intersection problem of two planes. However, the frames in the vertex are identical, i.e. $\mathbf{f}_0^a = \mathbf{f}_0^d$, and therefore these “planes” are parallel. The solvability is, therefore, only granted if they are identical. This yields the following condition:

$$\frac{1}{\Delta^a} d^a \mathbf{f}_1^a \cdot \mathbf{r}_0^a = \frac{1}{\Delta^d} d^d \mathbf{f}_1^d \cdot \mathbf{r}_0^d. \quad (\text{III.60})$$

We can eliminate all ribbon terms from this equation if we take Eq. (III.52) and (III.53) into account, i.e. with $\mathbf{t}_0^d := \mathbf{d}'(0) = \mathbf{r}_0^d$ and $\mathbf{t}_0^a := \mathbf{a}'(0) = \mathbf{r}_0^a$:

$$\frac{d^a}{\Delta^a} \mathbf{f}_1^a \cdot \mathbf{t}_0^d = \frac{d^d}{\Delta^d} \mathbf{f}_1^d \cdot \mathbf{t}_0^a. \quad (\text{III.61})$$

By this, we have proven the already anticipated proposition:

Proposition III.3.3. *Given are two curves with a common vertex at their respective start points and their frames, which are identical in their common vertex, in B-Spline representation of the respective degrees d and \hat{d} with interior knot multiplicity d and \hat{d} , respectively. The ribbons for a bicubically blended Coons patch can be constructed compatibly if*

$$\frac{d}{\Delta} \mathbf{f}_1 \cdot \hat{\mathbf{t}}_0 = \frac{\hat{d}}{\hat{\Delta}} \hat{\mathbf{f}}_1 \cdot \mathbf{t}_0. \quad (\text{III.62})$$

holds where Δ and $\hat{\Delta}$ mark their respective knot interval lengths at their common vertex, \mathbf{t}_0 and $\hat{\mathbf{t}}_0$ are the respective tangents at the vertex and \mathbf{f}_1 and $\hat{\mathbf{f}}_1$ their respective second frame control vectors.

Remark III.3.4. We refer to Corollary III.2.8 where we already analysed the bound on the frame degree d and the resulting equation system for this case. We would consider \hat{d} , $\hat{\Delta}$, $\hat{\mathbf{t}}_0$, $\hat{\mathbf{f}}_1$ to be given from the adjacent curve. With d and \mathbf{t}_0 , we obtain the condition on \mathbf{f}_1 . $\#$

With this proposition, we obtain the possibility to construct compatible frames and ribbons and we fully determine \mathbf{r}_1^a and \mathbf{r}_1^d by minimization of the vertex twist:

Minimal Vertex Twist: In accordance with Section III.4, we present the notion of the *twist vector* as the patch’s mixed derivative at a determined point. In each of the patch vertices, the patch has to have the same behaviour as the ribbons. The twist vector can, therefore, be defined by ribbons:

$$\mathbf{T}^w = \frac{\delta}{\Delta} (\mathbf{r}_1 - \mathbf{r}_0). \quad (\text{III.63})$$

An essential property of the Coons patch is that it minimizes the twist on the surface, see [32]. We adopt this principle and construct the ribbons such that \mathbf{T}^w is minimal:

$$\|\mathbf{T}^w\|^2 \rightarrow \min$$

under the conditions (III.58) and (III.59). This can be expressed as a quadratic programming problem in the coordinates of \mathbf{r}_1 and solved by linear algebra methods, see e.g. [20, 13]. We show in the following lemma, that the minimal twist vector is always parallel to the frame vector in the vertex:

Lemma III.3.5. *Under the conditions of orthogonality $\mathbf{r}_0 \cdot \mathbf{f}_0 = 0$ and (III.56) (III.55) and the compatibility (III.62) the minimal vertex twist vector is given by*

$$\mathbf{T}^w = -d \frac{\mathbf{r}_0 \cdot \mathbf{f}_1}{\|\mathbf{f}_0\|} \mathbf{f}_0. \quad (\text{III.64})$$

Proof. A geometric proof can be found in Figure III.23.

We determine \mathbf{r}_1 such that \mathbf{T}^w has minimal length. Condition III.56 reads as follows:

$$\begin{pmatrix} r_1^x \\ r_1^y \\ r_1^z \end{pmatrix}^T \cdot \begin{pmatrix} \xi \\ \psi \\ \zeta \end{pmatrix} = k := -\frac{d}{\delta \|\mathbf{f}_0\|} \mathbf{r}_0 \cdot \mathbf{f}_1$$

with $(\xi, \psi, \zeta) = \mathbf{f}_0 / \|\mathbf{f}_0\|$. Henceforth, we write the coordinates of \mathbf{r}_i as (x_i, y_i, z_i) and we minimize $\|\mathbf{T}^w\|^2$ with the substitution $r_1^z = z_1 = \frac{k - \xi x_1 - \psi y_1}{\zeta}$:

$$\begin{aligned} \|\mathbf{T}^w\|^2 &\rightarrow \min \\ f(x_1, y_1) &:= \frac{\delta^2}{\Delta^2} (x_1 - x_0)^2 + (y_1 - y_0)^2 + \left(\frac{k - \xi x_1 - \psi y_1}{\zeta} - z_0 \right)^2 \\ &f(x_1, y_1) \rightarrow \min. \end{aligned}$$

The extremum can be found by setting the gradient of $f(x_1, y_1)$ zero:

$$\begin{aligned} \nabla_{(x_1, y_1)} f(x_1, y_1) &= \\ 2 \frac{\delta^2}{\Delta^2} \begin{pmatrix} x_1 - x_0 - \xi/\zeta^2 (k - \xi x_1 - \psi y_1 + \zeta z_0) \\ y_1 - y_0 - \psi/\zeta^2 (k - \xi x_1 - \psi y_1 + \zeta z_0) \end{pmatrix} &= \begin{pmatrix} 0 \\ 0 \end{pmatrix} \\ \begin{pmatrix} (1 - \psi^2) & -\xi\psi \\ -\xi\psi & (1 - \xi^2) \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} &= \begin{pmatrix} -\zeta^2 x_0 + \xi k - \zeta \xi z_0 \\ -\zeta^2 y_0 + \psi k - \zeta \psi z_0 \end{pmatrix} \end{aligned}$$

We obtain $(x_1, y_1) = (k\xi + x_0, k\psi + y_0)$ in which we have repeatedly used the two properties: $\xi^2 + \psi^2 + \zeta^2 = 1$ and $\mathbf{f}_0 \cdot \mathbf{r}_0 = \xi x_0 + \psi y_0 + \zeta z_0 = 0$ and in the same manner we substitute z_1 and obtain:

$$\mathbf{r}_1 = \begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} = \begin{pmatrix} k\xi + x_0 \\ k\psi + y_0 \\ k\zeta + z_0 \end{pmatrix} = k \frac{\mathbf{f}_0}{\|\mathbf{f}_0\|} + \mathbf{r}_0.$$

The eigenvalues of the above matrix are 1 and $1 - \xi^2 - \zeta^2$ and, therefore, imply the matrix' positive definiteness. The lemma follows by substituting for k . ■

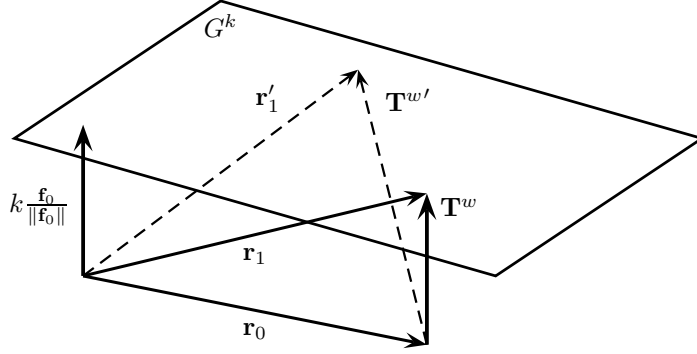


Figure III.23: A geometric proof for Lemma III.3.5. We regard the set of planes $\frac{1}{\|\mathbf{f}_0\|}\mathbf{f}_0 \cdot \mathbf{x} = c$, denoted by G^c . \mathbf{r}_0 is given to lie in the plane with $c = 0$ and Eq. (III.56) dictates that \mathbf{r}_1 lies in the one with $c = k = -d/\delta\mathbf{r}_0 \cdot \mathbf{f}_1/\|\mathbf{f}_0\|$. The minimal distance between these two “points” is the distance k between the parallel planes.

III.3.3 The Rotation-Minimizing Ribbon

We return to the concept of the rotation-minimizing frame reviewed in Section III.2.2 and apply the algorithm of the double reflection method to the initial *frame* ϕ_0 and *ribbon* ρ_0 at $m + 1$ parameter values $(u_j)_{j=0}^m$ to produce $(\phi_j)_{j=0}^m$ and $(\rho_j)_{j=0}^m$, compare to Figure III.24.

At this point we are not only given a set of points $(\mathbf{x}_j)_{j=0}^m$ but also an already existing set of frame samples $(\mathbf{f}_j)_{j=0}^m$. If those are in fact samples of the rotation-minimizing frame, then the double reflection method yields ϕ_j and ρ_j such that $\phi_j = \mathbf{f}_j$ and $\rho_j \perp \mathbf{f}_j$. This is not the case if \mathbf{f}_j differ from the rotation-minimizing frame. Then, a rotation is determined by ϕ_j and \mathbf{f}_j which we apply to ρ_j . The result is a set of vectors $(\rho_j)_{j=0}^m$ which are orthogonal to our given frame samples $(\mathbf{f}_j)_{j=0}^m$.

In the second step, we have to apply another rotation to ρ_j as we are given the boundary condition of the last ribbon vector. We continue to follow the routine from Section III.2.2 on the RMF and add this rotation linearly as in this way the squared angular speed is reduced. The axis of this rotation is, in this case, the frame direction \mathbf{f}_j . In this way, the constructed rotation minimizing ribbon (RMR) samples lie in the planes defined by the frame.

Finally, a proper scaling is applied to the RMR samples such that the ribbon lengths at the start and end match the tangent vector lengths of the neighbour curves. The difference between the start and end length is interpolated proportionally to the arc length.

III.3.4 Optimisation of the Ribbon

We have established all requirements to construct a ribbon orthogonally to the frame and compatibly to its neighbours. By following the above approach to minimize the vertex twist, we have determined 4 of the $n + 1$ control vectors \mathbf{r}_1 , \mathbf{r}_0 and \mathbf{r}_{n-1} , \mathbf{r}_n . Like in the case of the frame construction, we possibly

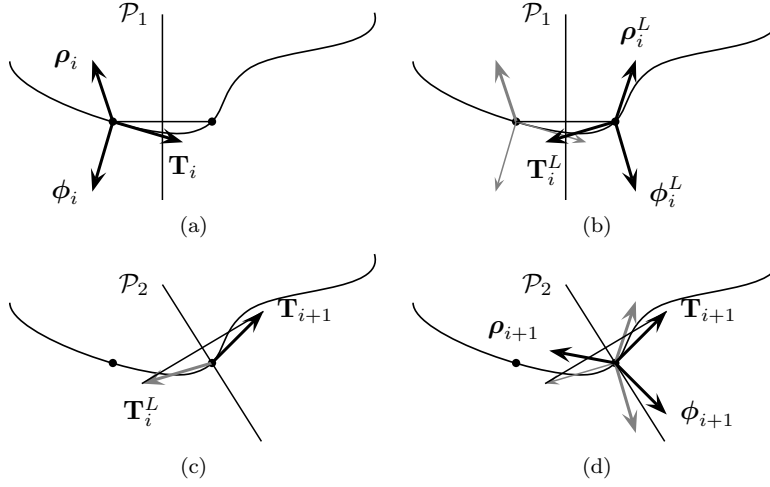


Figure III.24: The double reflection method to calculate a curve's RMF-samples ϕ_i and RMR-samples ρ_i .

produce degrees of freedom, especially if we choose a higher degree than the lower bounds in Proposition III.3.2 suggest.

As in the case of frames, we employ a convex combination of two objective functions. The first shall be the approximation of the rotation minimizing ribbon in the least-squares sense, Equation (III.65), while the second shall minimize the distances between ribbon control “points”, Equation (III.66):

$$e_j^\rho := \|\rho_j - \mathbf{r}(u_j)\| = \left\| \rho_j - \sum_{i=0}^{\nu\delta} N(u_j) \mathbf{r}_i \right\|$$

$$e^\rho = \sum_{j=1}^m (e_j^\rho)^2 \rightarrow \min, \quad (\text{III.65})$$

$$\Delta_i = \|\mathbf{r}_i - \mathbf{r}_{i-1}\|$$

$$e^\Delta := \sum_{i=1}^{\nu\delta} \Delta_i^2 \rightarrow \min. \quad (\text{III.66})$$

The full objective function reads

$$Q(w^R) := w^R e^\rho + (1 - w^R) e^\Delta, \quad (\text{III.67})$$

$$Q(w^R) \rightarrow \min$$

where the blend parameter $w^R \in [0, 1]$ represents the weight of the RMR approximation in this optimisation. Under the linear conditions (III.51) this results in a quadratic programming problem with equality constraints, see e.g. [20, 13].

Example III.3.6. We regard the following Bézier curve $a(u)$ of degree $p = 3$

with frame $\mathbf{f}^a(u)$ of degree $d = 4$:

$$\begin{aligned}
 (\mathbf{a}_i)_i &= \left(\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 3 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 5 \\ 0 \end{pmatrix} \right), \\
 (\mathbf{f}_i^a)_i &= \left(\frac{1}{\sqrt{21}} \begin{pmatrix} 1 \\ -2 \\ 4 \end{pmatrix}, \begin{pmatrix} 0.094 \\ -0.38 \\ 0.971 \end{pmatrix}, \begin{pmatrix} -0.1 \\ 0.005 \\ 0.784 \end{pmatrix}, \begin{pmatrix} -0.29 \\ 0.377 \\ 0.957 \end{pmatrix}, \frac{1}{\sqrt{6}} \begin{pmatrix} -1 \\ 1 \\ 2 \end{pmatrix} \right)
 \end{aligned}$$

At \mathbf{a}_0 , the curve connects to the neighbour curve $\mathbf{d}(v)$, which is also of degree 3. Its tangent in the vertex is $\mathbf{t}^d = (-3, 3/2, 0)^T$ and its frame $\mathbf{f}^d(v)$ of degree 5 has the second to last control vector $\mathbf{f}_4^d = (0.325, -0.476, 0.876)^T$. The last control vector, which is placed in the vertex, coincides with the first one of $\mathbf{f}^a(u)$. As for the other boundary, we have similar specifications for the neighbour $\mathbf{b}(v)$: $\mathbf{t}^b = (3, 0, 3/2)^T$ and $\mathbf{f}_1^b = (-0.369, 0.376, 0.901)^T$. We can confirm that these frames are compatible by evaluating Equation (III.55) and we calculate the minimal twist vector at \mathbf{a}_0 , $\mathbf{W}^{da} = (0.247, -0.494, 0.988)^T$, and at \mathbf{a}_3 , $\mathbf{W}^{ab} = (0.919, -0.919, -1.838)^T$. Note that these vectors are indeed multiples of \mathbf{f}_0^a and \mathbf{f}_5^a , respectively.

We compare the calculated ribbons with different ribbon degrees δ and RMR approximation weights w^R by the means of two criteria. The first is the error e^ρ in least squares sense to the RMR and the second one is the distance between two consecutive control ‘‘points’’. For both, we regard the maximum and mean values:

δ	Bézier w^R	e^ρ		Δ_i	
		max	mean	max	mean
6	0	0.532	0.321	5.816	2.741
	1/2	0.218	0.142	11.733	3.698
	1	0.119	0.063	17.226	4.718
7	0	0.531	0.321	5.358	2.534
	1/2	0.159	0.112	11.761	3.393
	1	0.064	0.039	23.453	4.97

Find the constructed ribbon for $\delta = 6$ and $w^R = \frac{1}{2}$ in Figure III.25. \triangleleft

Example III.3.7. We consider a similar example as above with the B-Spline curve $\mathbf{a}(u)$ which has one interior knot $k_4 = 1/2$ and the following 5 control points:

$$(\mathbf{a}_i)_i = \left(\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ \frac{1}{2} \end{pmatrix}, \begin{pmatrix} 0 \\ \frac{5}{2} \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 4 \\ \frac{1}{2} \end{pmatrix}, \begin{pmatrix} 0 \\ 5 \\ 0 \end{pmatrix} \right).$$

The following are the explicit boundary control vectors and the vector at the interior knot:

$$(\mathbf{f}_i^a)_i = \left(\frac{1}{\sqrt{21}} \begin{pmatrix} 1 \\ -2 \\ 4 \end{pmatrix}, \begin{pmatrix} 0.157 \\ -0.378 \\ 0.914 \end{pmatrix}, \dots, \begin{pmatrix} -0.126 \\ 0 \\ 0.957 \end{pmatrix}, \dots, \begin{pmatrix} -0.39 \\ 0.421 \\ 0.842 \end{pmatrix}, \frac{1}{\sqrt{6}} \begin{pmatrix} -1 \\ 1 \\ 2 \end{pmatrix} \right).$$

With the same neighbour curves as above but different second frame vectors, $\mathbf{f}_3^d = (0.351, -0.484, 0.907)^T$ and $\mathbf{f}_1^b = (-0.397, 0.463, 0.998)^T$, we obtain the

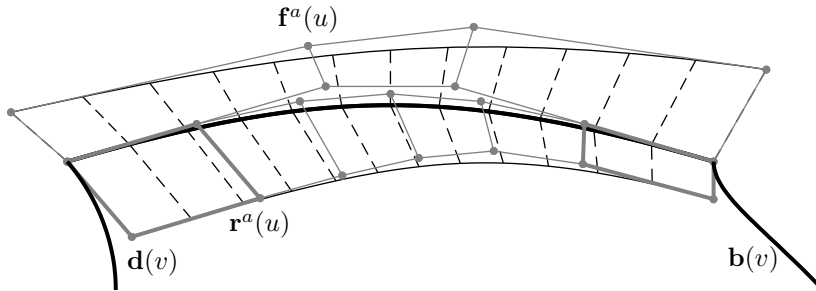


Figure III.25: We present is the constructed ribbon of Example III.3.6 with $\delta = 6$ and $w^R = 1/2$. The samples of the frame and ribbon, which is scaled by $1/3$, of $\mathbf{a}(u)$ are marked in dashed black, their control polygons in grey.

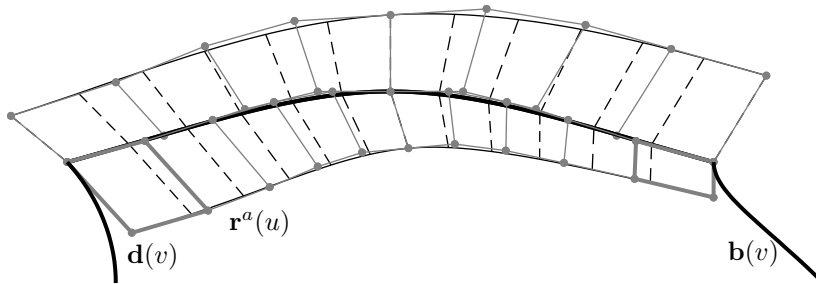


Figure III.26: We present is the constructed ribbon of Example III.26 with $\delta = 5$ and $w^R = 1/2$. The samples of the frame and ribbon, which is scaled by $1/3$, of $\mathbf{a}(u)$ are marked in dashed black, their control polygons in grey.

twist vectors $\mathbf{W}^{da} = (0.158, -0.316, 0.631)^T$ and $\mathbf{W}^{ab} = (0.354, -0.354, -0.707)^T$ which yield the visualised ribbon control vectors in the boundary.

For the construction of the interior ribbon vectors of $\mathbf{r}^a(u)$ we employ the same measures as in the above example and compare the degrees $\delta = 5, 6$ and the RMR approximation weights $w^R = 0, 1/2, 1$:

B-Spline		e^ρ		Δ_i	
δ	w^R	max	mean	max	mean
5	0	0.779	0.47	3.189	2.183
	1/2	0.321	0.186	3.537	2.232
	1	0.254	0.137	4.393	2.226
6	0	0.796	0.47	3.59	2.169
	1/2	0.256	0.147	3.9	2.177
	1	0.151	0.066	6.178	2.274

A visualisation of the 11 control vectors of the frame $\mathbf{f}^a(u)$ of degree 5 can be found in Figure III.26. \triangleleft

III.3.5 Ribbon Construction Algorithm

Algorithm III.3.8 (Ribbon Construction). The input of this algorithm is composed by the curves $\mathbf{a}(u)$, $\mathbf{b}(v)$ and $\mathbf{d}(v)$ (with $\mathbf{a}(0) = \mathbf{d}(0)$ and $\mathbf{a}(1) = \mathbf{b}(0)$), the respective compatible frames $\mathbf{f}^a(u)$, $\mathbf{f}^b(v)$ and $\mathbf{f}^d(v)$, the degree elevation parameter δ^+ and the RMR approximation weight w^R . We obtain a B-spline expression for the boundary ribbon.

input	$\mathbf{a}(u)$, $\mathbf{f}^a(u)$	$\mathbf{b}(0) = \mathbf{a}(1)$, $\mathbf{f}^b(v)$ compatible with $\mathbf{f}^a(u)$ $\mathbf{d}(0) = \mathbf{a}(0)$, $\mathbf{f}^d(v)$ compatible with $\mathbf{f}^a(u)$ $\delta^+ \geq 0$ $w^R \in [0, 1]$
	$\mathbf{b}(v)$, $\mathbf{f}^b(v)$	
	$\mathbf{d}(v)$, $\mathbf{f}^d(v)$	
	δ^+	
	w^R	
output	degree δ	
	Control vectors $(\mathbf{r}_i)_{i=0}^{\nu\delta}$	
1: Determine δ^{Min} . \triangleright Prop. III.3.2		
2: $\delta = \lceil \delta^{\text{Min}} \rceil + \delta^+$		
3: Ribbon knot vector: \mathbf{k} . \triangleright Eq. (III.48)		
4: Split $\mathbf{f}(u)$ at $(\bar{k}_i)_{i=1}^{\nu-1}$: $(\mathbf{f}^i(u))_{i=0}^{\nu-1}$		
5: for $i = 0, \dots, N^k - 1$ do		
6: Create coefficient matrix E_i . \triangleright Eq. (III.46)		
7: Join $(E_i)_i$: $E^\perp = \text{diag}((E_i)_i)$. \triangleright Eq. (III.46)		
8: Continuity conditions matrix: E^C . \triangleright Eq. (III.49)		
9: $\mathbf{r}^0 = \mathbf{d}'(0)$, $\mathbf{r}^{\nu\delta} = \mathbf{b}'(0)$.		
10: Minimize Twist in the vertices to obtain \mathbf{r}^1 and $\mathbf{r}^{\nu\delta-1}$. \triangleright Eq. (III.3.5)		
11: Boundary conditions matrix: E^B . \triangleright Eq. (III.51)		
12: Join E^\perp , E^C , E^B in E . \triangleright Eq. (III.51)		
13: Create RMR samples: $(\rho_j)_{j=0}^m$. \triangleright S. III.3.3		
14: Minimize $Q(w^R)$ under the conditions of E . \triangleright Eq. (III.67).		
15: return δ and control vectors $(\mathbf{r}_i)_{i=0}^{\nu\delta}$.		

III.4 Coons Patch Construction

The Coons patch construction offers a straight-forward way to blend between opposite curves of a quadrilateral curve loop and it is, therefore, especially well suited for our requirements. We distinguish the types of *bilinear* and *bicubical* blends. In both cases, the denomination is due to the degree of the blending functions rather than the resulting surface degree. While the former constructs a ruling between opposite boundaries, see Subsection III.4.1, in the latter the boundary curves are required to be equipped with cross-tangent information, the boundary ribbons, see Subsection III.4.2. In the previous Section III.3 we have provided a construction for ribbons such that they comply with the tangent plane information which we impose a-priori by the means of the frames.

III.4.1 Bilinearly Blended Coons Patches

To interpolate the loop of boundary curves $\mathbf{a}(u)$, $\mathbf{b}(v)$, $\mathbf{c}(u)$ and $\mathbf{d}(v)$, we assume that each curve touches its neighbours at either both respective start-parameters or both respective end-parameters, not a combination of those, e.g. $\mathbf{b}(v)$ connects $\mathbf{a}(0)$ and $\mathbf{c}(0)$. We define three surfaces:

$$\mathbf{s}^u(u, v) := (1 - v)\mathbf{a}(u) + v\mathbf{c}(u) \quad (\text{III.68})$$

$$\mathbf{s}^v(u, v) := (1 - u)\mathbf{d}(v) + u\mathbf{b}(v) \quad (\text{III.69})$$

$$\mathbf{s}^{uv}(u, v) := (1 - u \quad u) \begin{pmatrix} \mathbf{a}(0) & \mathbf{d}(1) \\ \mathbf{b}(0) & \mathbf{c}(1) \end{pmatrix} \begin{pmatrix} 1 - v \\ v \end{pmatrix} \quad (\text{III.70})$$

$$(\text{III.71})$$

and refer to Example III.4.5 and Figure III.28 for an illustration. The surfaces $\mathbf{s}^u(u, v)$ and $\mathbf{s}^v(u, v)$ are ruled surfaces of opposite input curves which employ the linear blending functions $(1 - u), u, (1 - v), v$. The surface $\mathbf{s}^{uv}(u, v)$ bilinearly interpolates the vertices. Its boundaries coincide with $\mathbf{s}^u(0, v)$ and $\mathbf{s}^u(1, v)$ and with $\mathbf{s}^v(u, 0)$ and $\mathbf{s}^v(u, 1)$, respectively, i.e. those boundaries of $\mathbf{s}^u(u, v)$ and $\mathbf{s}^v(u, v)$ which do not interpolate the data curves. The Coons patch is then defined by the following Boolean sum of these three surfaces:

$$\mathbf{s}(u, v) := \mathbf{s}^u(u, v) + \mathbf{s}^v(u, v) - \mathbf{s}^{uv}(u, v). \quad (\text{III.72})$$

Straight forward evaluations of $\mathbf{s}(u, v)$ at its boundaries verify that this interpolates indeed the given boundary data:

$$\mathbf{s}(0, v) = (1 - v)\mathbf{a}(0) + v\mathbf{c}(0) + \mathbf{d}(v) - [(1 - v)\mathbf{a}(0) + v\mathbf{c}(0)] = \mathbf{d}(v).$$

Algorithm III.4.1 (Bilinear Blend). We summarize the construction of the bilinearly blended Coons patch:

input	$\mathbf{a}(u)$ $\mathbf{b}(v)$ $\mathbf{c}(u)$ $\mathbf{d}(v)$
output	surface representation $\mathbf{s}(u, v)$

1: sort curves to loop.

- 2: reverse directions of third and fourth curve $\mathbf{c}(u)$ & $\mathbf{d}(v)$.
 - 3: create $\mathbf{s}^u(u, v)$. ▷ Eq. (III.68)
 - 4: create $\mathbf{s}^v(u, v)$. ▷ Eq. (III.69)
 - 5: create $\mathbf{s}^{uv}(u, v)$. ▷ Eq. (III.70)
 - 6: $\mathbf{s}(u, v) = \mathbf{s}^u(u, v) + \mathbf{s}^v(u, v) - \mathbf{s}^{uv}(u, v)$. ▷ Eq. (III.72)
 - 7: **return** $\mathbf{s}(u, v)$.
-

III.4.2 Bicubically Blended Coons Patches

In this type of patch construction we consider boundary curves $\mathbf{a}(u)$, $\mathbf{b}(v)$, $\mathbf{c}(u)$ and $\mathbf{d}(v)$ with provided ribbons as introduced in Section III.3. They specify the tangent with respect to “the other parameter” and, therefore, represent the patch’s cross-boundary derivatives, compare to Figure III.29:

$$\mathbf{r}^a(u), \quad \mathbf{r}^b(v), \quad \mathbf{r}^c(u), \quad \mathbf{r}^d(v).$$

The curves are parametrized as above and the ribbons adopt the parametrizations of their corresponding curve. Additionally, at its end points each ribbon has to coincide with the direction and length of the tangent vector of the respective neighbour curve at the corresponding point. All ribbons are oriented towards the interior of the patch.

Like in the bilinear case, we blend opposite data. This time, however, we interpolate positional and tangential data at the start and end. Cubic Hermite polynomials $H_i^3(t)$ are known to be suited well for this application, as we have reviewed in Section III.1.1, and thus we define

$$\mathbf{s}^u(u, v) := H_0^3(v)\mathbf{a}(u) + H_1^3(v)\mathbf{r}^a(u) + H_2^3(v)\mathbf{r}^c(u) + H_3^3(v)\mathbf{c}(u) \quad (\text{III.73})$$

$$\mathbf{s}^v(u, v) := H_0^3(u)\mathbf{d}(v) + H_1^3(u)\mathbf{r}^d(v) + H_2^3(u)\mathbf{r}^b(v) + H_3^3(u)\mathbf{b}(v) \quad (\text{III.74})$$

As in the bilinear method, we define the a third surface interpolating those boundaries of $\mathbf{s}^u(u, v)$ and $\mathbf{s}^v(u, v)$ which do not coincide with the data curves:

$$\mathbf{s}^{uv}(u, v) := (H_0^3(u), H_1^3(u), H_2^3(u), H_3^3(u)) \cdot \begin{pmatrix} \mathbf{a}(0) & \mathbf{r}^d(0) & \mathbf{r}^d(1) & \mathbf{d}(1) \\ \mathbf{r}^a(0) & \cdot & \cdot & \mathbf{r}^c(0) \\ \mathbf{r}^a(1) & \cdot & \cdot & \mathbf{r}^c(1) \\ \mathbf{b}(0) & \mathbf{r}^b(0) & \mathbf{r}^b(1) & \mathbf{c}(1) \end{pmatrix} \begin{pmatrix} H_0^3(v) \\ H_1^3(v) \\ H_2^3(v) \\ H_3^3(v) \end{pmatrix} \quad (\text{III.75})$$

The bicubically blended Coons patch is then defined as a Boolean sum:

$$\mathbf{s}(u, v) := \mathbf{s}^u(u, v) + \mathbf{s}^v(u, v) - \mathbf{s}^{uv}(u, v). \quad (\text{III.76})$$

The missing values in the matrix in (III.75) are the patch’s mixed derivatives, or *twist vectors*, at the four corners and form a delicate part in this type of construction. To see this, we regard the corner $\mathbf{S}(0, 0)$ where the curves $\mathbf{a}(u)$ and $\mathbf{d}(v)$ touch. Both curves are equipped with a ribbon, which is why the derivative w.r.t. u and the derivative w.r.t. v are both provided by *both* pairs of curves/ribbons.

In the above Section III.3.2 we required the ribbons to match the neighbour curves’ tangents and in addition we constructed the frames in a particular way,

such that the ribbons could be constructed compatibly, see Proposition III.3.3. As a result, we can define the twist vector in the vertices analogously to this:

$$\frac{\partial^2}{\partial u \partial v} \mathbf{s}(u, v)|_{u=0, v=0} := \lim_{u \rightarrow 0} \frac{\partial}{\partial u} \mathbf{r}^a(u)|_{u=0} = \lim_{v \rightarrow 0} \frac{\partial}{\partial v} \mathbf{r}^d(v)|_{v=0}. \quad (\text{III.77})$$

Definition III.4.2. If the equalities of (III.77) hold, we declare the ribbons $\mathbf{r}^a(u)$ and $\mathbf{r}^d(v)$ as “compatible“. $\#$

For incompatible data, the literature offers the construction of Gregory patches which employ variable twists but yield patches with discontinuous twists and they have zero corner weights, if written in rational Bézier form, see [31].

As we have provided the necessary machinery to provide compatible ribbons, in this thesis the input for the bicubically blended Coons patch construction is always guaranteed to be compatible. We can explicitly write the above matrix of (III.75):

$$\begin{pmatrix} \mathbf{a}(0) & \mathbf{r}^d(0) & \mathbf{r}^d(1) & \mathbf{d}(1) \\ \mathbf{r}^a(0) & \frac{\partial}{\partial u} \mathbf{r}^a(u)|_{u=0} & -\frac{\partial}{\partial v} \mathbf{r}^d(v)|_{v=1} & \mathbf{r}^c(0) \\ \mathbf{r}^a(1) & \frac{\partial}{\partial v} \mathbf{r}^b(v)|_{v=0} & -\frac{\partial}{\partial u} \mathbf{r}^c(u)|_{u=1} & \mathbf{r}^c(1) \\ \mathbf{b}(0) & \mathbf{r}^b(0) & \mathbf{r}^b(1) & \mathbf{c}(1) \end{pmatrix}$$

and, therefore, fully define $\mathbf{s}^{uv}(u, v)$. By this construction we have proven the following corollary to Proposition III.3.3:

Corollary III.4.3. *Given a loop of 4 curves equipped with 4 frames in B-spline representation. A bicubically blended Coons patch which is orthogonal to the frames at the corresponding boundaries can be constructed if this condition:*

$$\frac{d}{k_1} \mathbf{f}_1 \cdot \hat{\mathbf{t}}_0 = \frac{\hat{d}}{\hat{k}_1} \hat{\mathbf{f}}_1 \cdot \mathbf{t}_0. \quad (\text{III.78})$$

holds in every vertex for the respective curves and frames, where their parametrization directions are assumed to be away from the vertex, the knot values k_1 and \hat{k}_1 are the first interior knot values of the respective curves and the entities \mathbf{t}_0 and \mathbf{f}_1 are the tangent vector and frame vector of the curve’s restriction to the interval $[0, k_1]$; analogously for $\hat{\mathbf{t}}_0, \hat{\mathbf{f}}_1$.

Remark III.4.4. We note that the blending functions do not influence the patches continuity which, therefore, depends only on the blended entities themselves. A loop of C^2 continuous curves and C^1 continuous and compatible ribbons yields a C^1 continuous surface patch: Remarks III.2.4 and III.3.1. $\#$

Example III.4.5. We construct a surface between the loop of four Bézier curves of degree 3. Figure III.27 shows them with their respective frames which are of degrees 3 and 4 in u and v direction, respectively.

Figures III.28(a), (b) and (c) of Figure III.28(d) exhibit the respective *linear* blends $\mathbf{S}^u(u, v)$, $\mathbf{S}^v(u, v)$ and $\mathbf{S}^{uv}(u, v)$ of the Coons patch $\mathbf{S}(u, v)$, shown in (d), which disregards the frame data.

The computed ribbons, scaled by the factor $1/3$, can be found in Figure III.29 and the bicubical blends in Figure III.30: (a), (c) and (e) show the blends $\mathbf{S}^u(u, v)$, $\mathbf{S}^v(u, v)$ and $\mathbf{S}^{uv}(u, v)$, respectively. The resulting Coons patch $\mathbf{S}(u, v)$ is shown in (g). \triangleleft

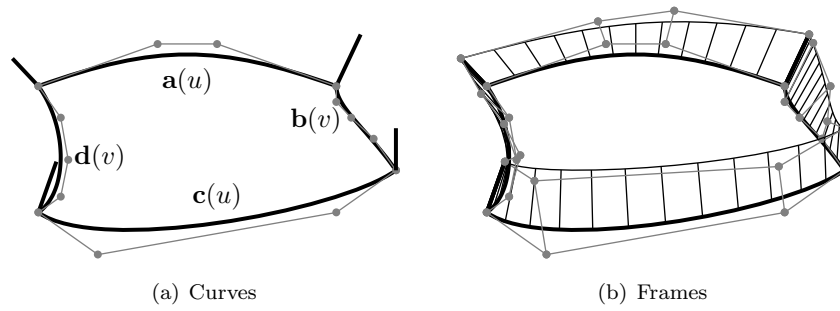


Figure III.27: Curves and frames as posed Example III.4.5.

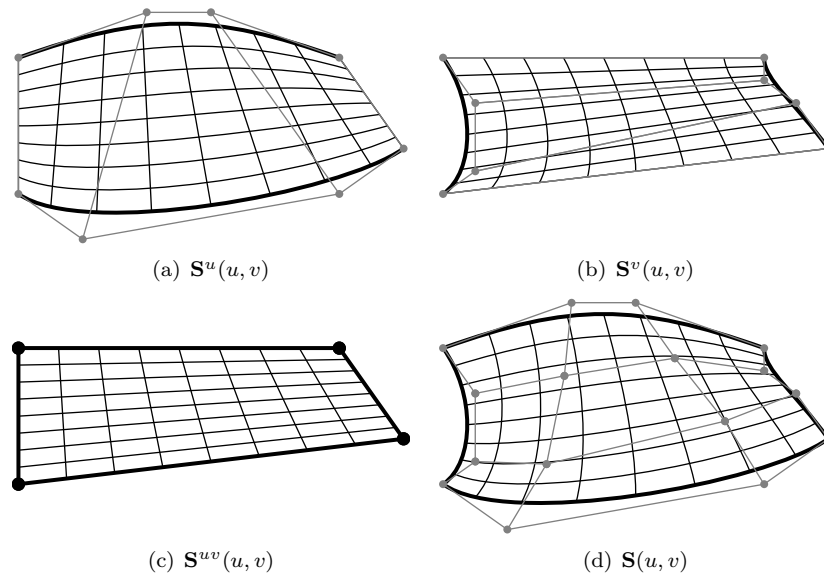


Figure III.28: Bilinearly blended Coons patch illustration as in Example III.4.5.

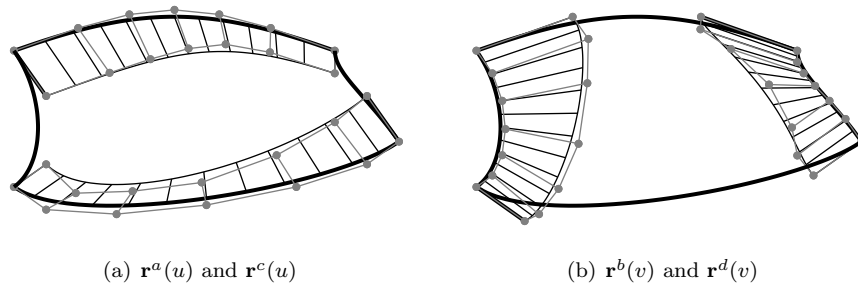


Figure III.29: The constructed ribbons of Example III.4.5 scaled by $\frac{1}{3}$.

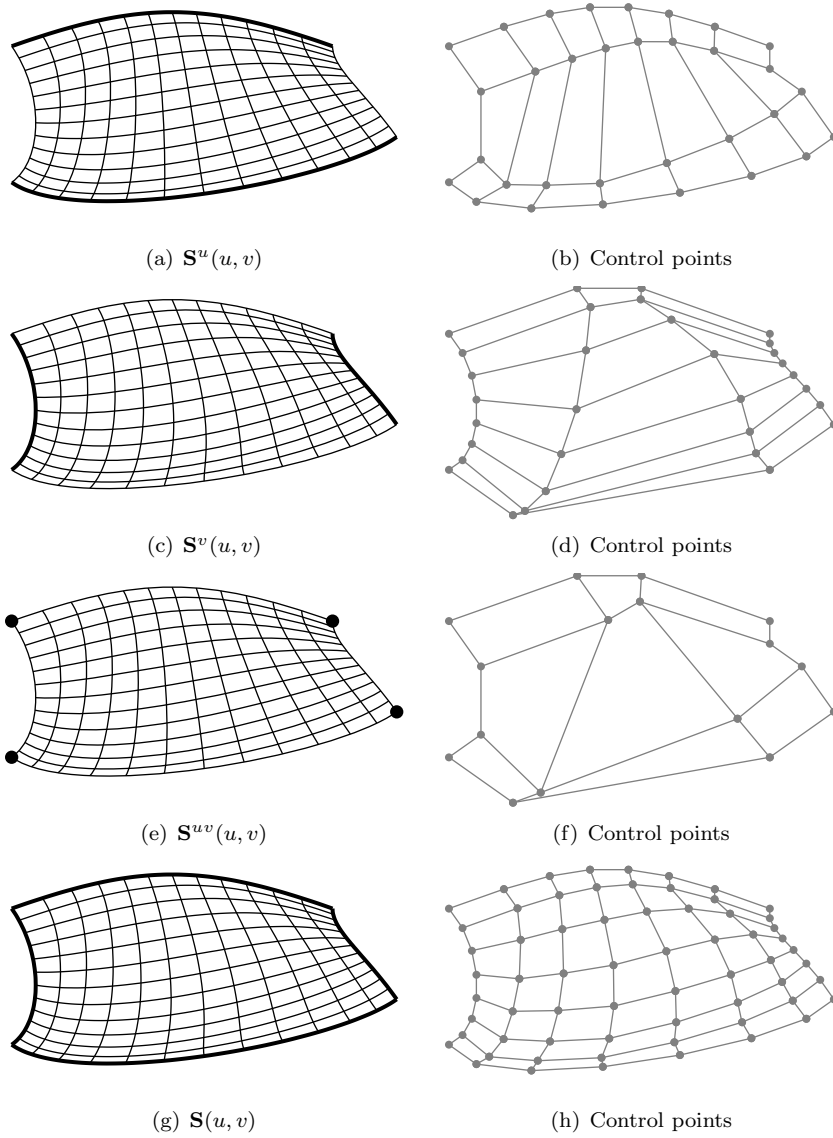


Figure III.30: Bicubically blended Coons patch for Example III.4.5.

Example III.4.6 (Patch Construction). This example represents a typical situation in this parametric model. We are given two primary curves $\mathbf{a}(u)$ and $\mathbf{c}(u)$ in B-spline representation. Each of them have one interior knot at $k_4^a = k_4^c = 1/2$. Two curves $\mathbf{b}(v)$ and $\mathbf{d}(v)$ intersect the primary ones at their respective parameters $u_{ab} = 0.8$, $u_{cb} = 0.4$ and $u_{cd} = 0.9$, $u_{ad} = 0.2$. They, in turn, are given in Bézier representation, i.e. with no interior knots. Compare to Figure III.31. In our model, such situations occur frequently due to the wireframe strict hierarchical order.

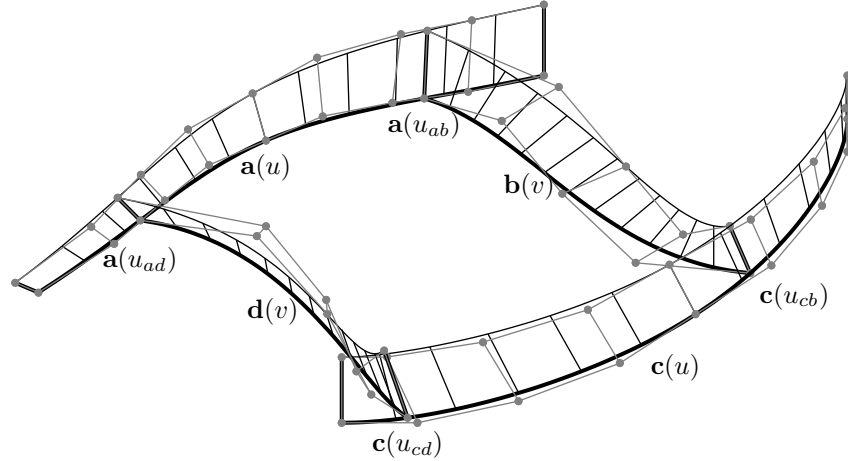


Figure III.31: Frames for example III.4.6.

The construction of the frames $\mathbf{f}^b(v)$ and $\mathbf{f}^d(v)$ takes into account the already existing primary curves and their frames $\mathbf{f}^a(u)$ and $\mathbf{f}^c(u)$. For their compatibility, we regard the four sub-curves of $\mathbf{a}(u)$ which result from splitting at $u = u_{ad}, k_4^a, u_{ab}$ and those of $\mathbf{c}(u)$ by splitting at $u = u_{cb}, k_4^c, u_{cd}$. The curves $\mathbf{a}(u)|_0^{u_{ad}}$ and $\mathbf{d}(v)$ determine the twist vector in their common vertex and analogue curve pairs yield the other ones.

The curves and the compatibly constructed frames of degrees $d = 5$ yield the uniquely defined twist vectors in all vertices and the calculation of the ribbons is performed with the parameters of the degree δ^+ and the approximation weight w^R . In Figure III.32 we find these ribbons for $\delta^a = \delta^c = 7$, $\delta^b = \delta^d = 8$ and $w^R = 1/2$ and scaled by $1/3$.

The bicubical blends can be found in Figure III.33 with the finally patch $\mathbf{S}(u, v)$ being of degree 7×8 . ◁

Algorithm III.4.7 (Bicubical Blend). We summarize the construction of the bicubically blended Coons patch:

input	$\mathbf{a}(u), \mathbf{f}^a(u)$ $\mathbf{b}(v), \mathbf{f}^b(v)$ $\mathbf{c}(u), \mathbf{f}^c(u)$ $\mathbf{d}(v), \mathbf{f}^d(v)$ δ^+, w^R	$\mathbf{f}^a(u), \mathbf{f}^b(v), \mathbf{f}^c(u), \mathbf{f}^d(v)$ compatible
output	surface representation $\mathbf{s}(u, v)$	

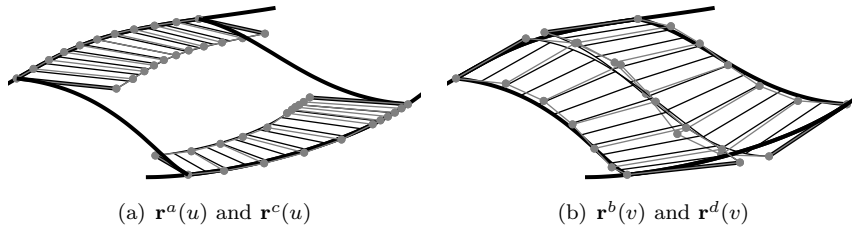


Figure III.32: The constructed ribbons of Example III.4.6 scaled by $1/3$.

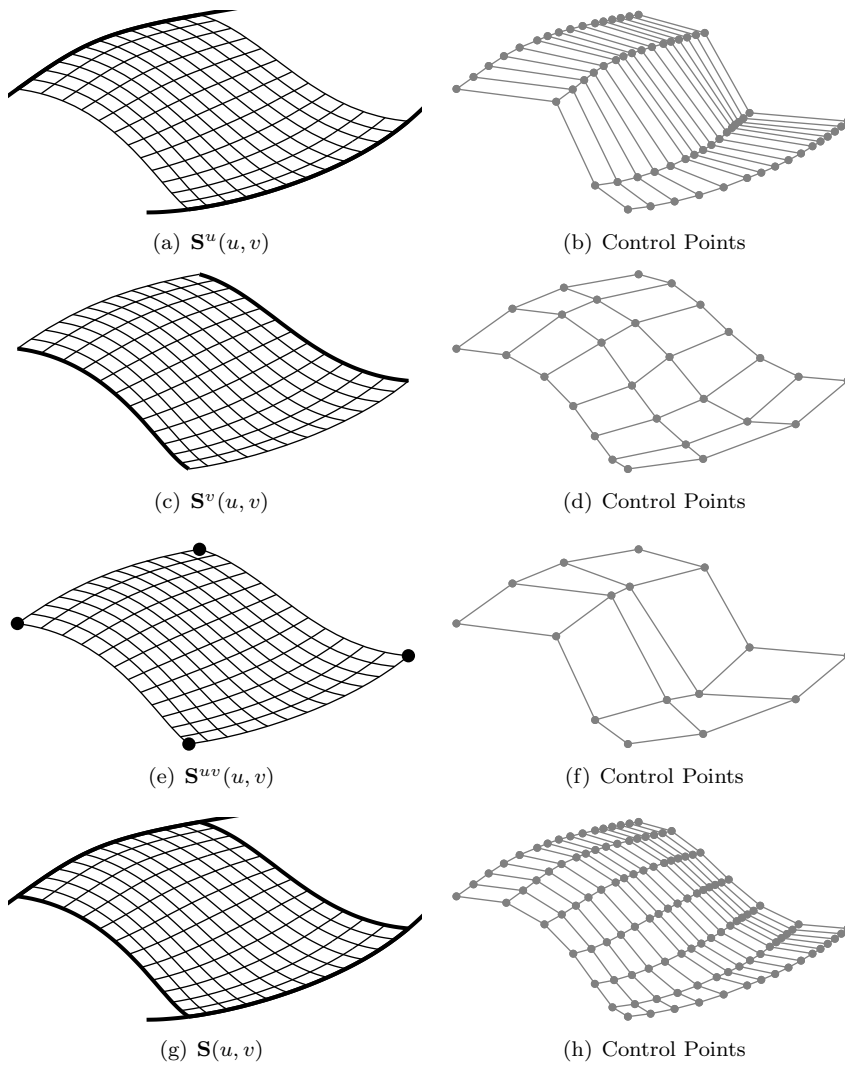


Figure III.33: Cubical Blends for example III.4.6.

- 1: sort curves to loop.
 - 2: reverse parameter directions of third and fourth curve: $\mathbf{c}(u)$, $\mathbf{d}(v)$.
 - 3: reverse parameter directions of third and fourth frame: $\mathbf{f}^c(u)$, $\mathbf{f}^d(v)$.
 - 4: construct ribbons $\mathbf{r}^a(u)$, $\mathbf{r}^b(u)$, $\mathbf{r}^c(u)$, $\mathbf{r}^d(u)$. ▷ Algorithm III.3.8
 - 5: create $\mathbf{s}^u(u, v)$. ▷ Eq. (III.73)
 - 6: create $\mathbf{s}^v(u, v)$. ▷ Eq. (III.74)
 - 7: create $\mathbf{s}^{uv}(u, v)$. ▷ Eq. (III.75)
 - 8: $\mathbf{s}(u, v) = \mathbf{s}^u(u, v) + \mathbf{s}^v(u, v) - \mathbf{s}^{uv}(u, v)$. ▷ Eq. (III.76)
 - 9: **return** $\mathbf{s}(u, v)$.
-

Chapter IV

Construction Steps

Chapter III has supplied us with the geometric tools to construct a network of curves, to equip each of them with a tangent plane representation and finally to fill quadrilateral curve loops with surfaces. It is the purpose of the following two sections to design a valid ship hull by employing these geometric algorithms and by following the design strategy outlined in Section II.1. We go through the design steps in a hierarchical order.

In Section IV.1 we construct the wireframe geometry using the input parameters which we introduced in Section I.5. We present the structure and necessary hierarchy of the curves.

In Section IV.2 we build, based on the wireframe's neat structure, a network of surface patches and discuss their construction and topological arrangement. We also discuss the particular cases of non-quadrilateral patches there.

IV.1 Three Stages of the Wireframe

The design of a ship takes its start in the determination of the principal parameters, which we have already provided in Section II.2 and which are summarized in Table II.1 on page 30. They should, in combination with the ship's characteristics, give an intuitive idea of the ship's eventual shape. Before we give a detailed description of all the curves of the wireframe, we shall firstly distinguish the characteristic differences of the curves by the categorization into *three stages* of the wireframe, compare to Figure IV.1.

1. Curves which depend exclusively on the given input parameters constitute Stage I, see Subsection IV.1.1. No co-dependences between curves or frames occur and the set of curves is, therefore, entirely unordered. This stage includes the majority of the curves in the midship area since they are either straight and axis-parallel lines or simple Hermite curves. Stage I of the bow and stern includes the respective profile curves which lie in the Centre Plane and, therefore, bound the semi hull of the ship.
2. Stage II are curves which, in addition to the input parameters, depend on the existence of the curves and frames of Stage I. This stage is unordered within itself as its curves do not depend on other entities of Stage II. The

construction of these curves is covered in Subsection IV.1.2 and the section curves in the bow and stern area are typical examples for this stage.

3. We classify the rest of the curves as Stage III: Subsection IV.1.3. They depend on at least one entity of Stage II and can depend on one another. Therefore, a strict order of curves within this stage is required.

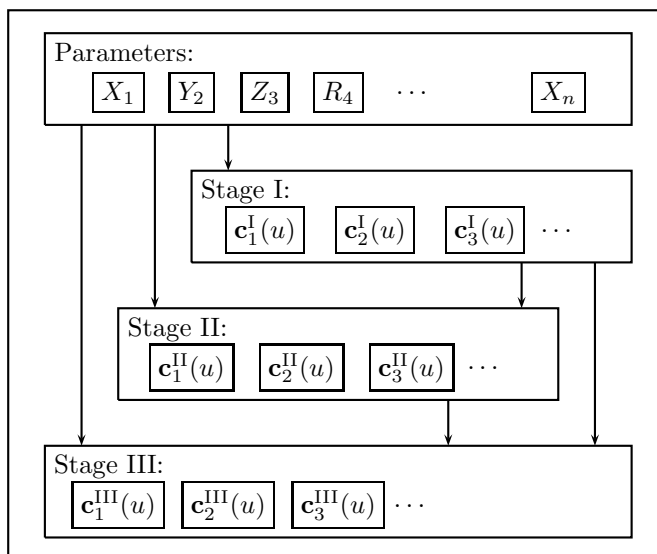


Figure IV.1: Diagram of the stages of the wireframe.

Curve Orientation: Note that we try to keep a general orientation of the ship’s curves. To facilitate the reference to the “start” and “end” of curves, every curve shall be parametrized in a way such that its start point is closer to the origin than its end point.

Curve Construction: For all curves, the interpolation of the given data is done by the interpolation Algorithm III.1.14 on page 54. It is obvious that for Hermite data, i.e. two points and up to two tangents, we prefer to employ the Hermite interpolation. In this case, Algorithm III.1.14 terminates in step 3 and returns the Bézier curve $\mathbf{H}_0(u)$. In all other situations, we continue to execute the algorithm with $p = 3$ as this is the minimal degree for the construction of twice continuously differentiable curves.

For the blend parameter w^C between the approximation of the curve $\mathbf{G}(u)$ and the minimization of the control polygon’s length, see Eq. (III.26), we always choose $w^C = 1/2$.

The choice to increase the number of additional control points n^+ , which produce additional degrees of freedom for the optimisation, proportionally to the minimal number of needed control points n_{\min} has proven to produce satisfactorily nice curves:

$$n^+ = \lfloor \frac{n_{\min}}{p} \rfloor.$$

In this manner, e.g. we have one additional control point for 0, 1 or 2 interior knots since n_{\min} is at least 3 for all data with more than two points.

We retain the option to rescale the resulting tangent lengths and employ Algorithm III.1.14 with tangent *vectors*, as described in Subsection III.1.5. For the affected curves, we explicitly mention its usage.

Frame Construction: Concerning the construction of frames, we always employ Algorithm III.2.13 except for cases where a constant frame suffices. For all curves we specify the minimal input of the boundary frame vectors and determine the minimal degree by Proposition III.2.6. If at a boundary there already exists a curve with a frame, we have to consider compatibility for which we refer to Corollary III.2.8. However, at no point do we fully predetermine two boundary frame control vectors as in the case of Corollary III.2.9.

If we are given interior frame vectors, we split the curve at the corresponding parameter and construct piecewise frames by connecting the parts in a C^1 manner as we do at all interior knots, compare to Eq. (III.32).

The design parameters d^+ and w^F determine the degree of the frame and the weight of the RMF approximation in the optimisation, respectively. As for the former, we choose to raise the degree by 1, i.e. $d^+ = 1$. The latter shall be fixed at $w^F = 1/2$, thus equally weighing the RMF-approximation with the minimization of the sum of the squared lengths of the frame control polygon sides. The constructed frames which result from these parameter choices are satisfactory.

IV.1.1 Stage I

Stage I is the most fundamental set of curves. Their construction and that of the attributed frames depend only on the input parameters.

Stage I, Midship: Most of the midship curves are part of Stage I as they are straight lines or Hermite curves connecting pairs of points with possibly associated tangents. Their construction is, therefore, straight-forward and possible coincidences or interactions occurs in only obvious manners which are governed by the input parameters, compare to Figure IV.2.

The first two curves lie in y - z plane at $x = X_5$, i.e. the start length of the parallel midship, and represent the base and stem of such a section curve, respectively.

1. The **Parallel Midship Back Base** $\mathbf{c}^1(u)$ is a straight line in transverse direction from the Centre Plane to far-most expansion of the flat-of-bottom. With the beam Y_2 and the bilge radius Y_7 we get the data points:

$$\mathbf{P}_0 = \begin{pmatrix} X_5 \\ 0 \\ 0 \end{pmatrix}, \quad \mathbf{P}_1 = \begin{pmatrix} X_5 \\ Y_2/2 - Y_7 \\ 0 \end{pmatrix}.$$

2. The **Parallel Midship Back Stem** $\mathbf{c}^2(u)$ is the straight line in vertical direction that connects the lowest point of the flat-of-side with the deck

curve. With the depth Z_3 and the bilge radius Y_7 we have:

$$\mathbf{P}_0 = \begin{pmatrix} X_5 \\ Y_2/2 \\ Y_7 \end{pmatrix} \quad \mathbf{P}_1 = \begin{pmatrix} X_5 \\ Y_2/2 \\ Z_3 \end{pmatrix}.$$

The following two curves are constructed identically to the previous two with the location $x = X_6$:

3. **Parallel Midship Front Base:** $\mathbf{c}^3(u)$.
4. **Parallel Midship Front Stem:** $\mathbf{c}^4(u)$.

We continue with four straight lines which expand in longitudinal direction connecting the end points of the Curves 1 and 2 with their respective pendant points on Curves 3 and 4.

5. The **Parallel Midship Centre Bar** $\mathbf{c}^5(u)$ connects the points $(X_5, 0, 0)^T$ and $(X_6, 0, 0)^T$ and lies, thus, on the base line.
6. The **Parallel Midship Bottom Bar** $\mathbf{c}^6(u)$ expands from $(X_5, Y_2/2 - Y_7, 0)^T$ to $(X_6, Y_2/2 - Y_7, 0)^T$ and, therefore, lies in the Bottom Plane at the far-most expansion of the Flat Bottom.
7. The **Parallel Midship Side Bar** $\mathbf{c}^7(u)$ lies in the Side Plane at the Flat Side's lowest point and connects $(X_5, Y_2/2, Y_7)^T$ with $(X_6, Y_2/2, Y_7)^T$.
8. The **Parallel Midship Top Bar** $\mathbf{c}^8(u)$ connects $(X_5, Y_2/2, Z_3)^T$ with $(X_6, Y_2/2, Z_3)^T$ and lies in the Side Plane as much as in the Deck Plane.

Note, that no curve is needed at the Deck Plane since it has no influence on the wetted hull. For the sake of completeness we include also $\mathbf{c}^8(u)$ in the model.

We have created quadrilateral loops for the two patches of the parallel midship which are of planar nature. Therefore, we have not enhanced any of these curves with frames. We continue with those curves which complete the triangular boundary loops of the patches of the flat-of-bottom and flat-of-side:

9. The **Back Flat Bottom Base** $\mathbf{c}^9(u)$ connects through a straight line the start point of the flat-of-bottom $\mathbf{P}_0 = (X_8, 0, 0)^T$ with the parallel midship $\mathbf{P}_1 = (X_5, 0, 0)^T$. No frame is required for this line as it bounds only one surface which is a plane.
10. The **Back Flat Bottom Diagonal** $\mathbf{c}^{10}(u)$ has the same start point as the base curve $\mathbf{c}^9(u)$. Its end point lies at the end of Parallel Midship Back Base $\mathbf{c}^1(1)$:

$$\mathbf{P}_0 = \begin{pmatrix} X_8 \\ 0 \\ 0 \end{pmatrix}, \quad \mathbf{P}_1 = \begin{pmatrix} X_5 \\ Y_2/2 - Y_7 \\ 0 \end{pmatrix}.$$

With two points and two tangent directions, both in x -axis, $\mathbf{T}_0 = \mathbf{T}_1 = (1, 0, 0)^T$, we construct Curve $\mathbf{c}^{10}(u)$ through Hermite interpolation. Since one of the patches which are bounded by this curve is the flat patch Back Flat Bottom, the curve's frame has to be constant in the direction of the plane's normal: $\mathbf{f}^{10}(u) = \mathbf{F} = (0, 0, -1)^T$.

11. The **Back Flat Side Base** $\mathbf{c}^{11}(u)$ connects through a straight line the $\mathbf{P}_0 = (X_9, Y_2/2, Z_3)^T$ with $\mathbf{P}_1 = (X_5, Y_2/2, Z_3)^T$. Similarly to the Midship Stern Curve $\mathbf{c}^8(u)$, this is not a part of the wetted ship hull and we construct this straight line merely for the sake of completeness.
12. The **Back Flat Side Diagonal** $\mathbf{c}^{12}(u)$ has the same start point as the base curve $\mathbf{c}^{11}(u)$. Its end point lies at the start of Parallel Midship Back Stem $\mathbf{c}^2(0)$.

$$\mathbf{P}_0 = \begin{pmatrix} X_9 \\ Y_2/2 \\ Z_3 \end{pmatrix}, \quad \mathbf{P}_1 = \begin{pmatrix} X_5 \\ Y_2/2 \\ Y_7 \end{pmatrix}.$$

With two points and only the tangent direction in x -axis being specified at the end of the curve, $\mathbf{T}_1 = (1, 0, 0)^T$, we construct Curve $\mathbf{c}^{12}(u)$ through Hermite interpolation from Section III.1.1. Similarly to $\mathbf{c}^{10}(u)$, this curve bounds the Back Flat Side and we construct a constant frame $\mathbf{f}^{12}(u) = \mathbf{F} = (0, 1, 0)^T$.

13. The **Front Flat Bottom Base** $\mathbf{c}^{13}(u)$ is the pendant to $\mathbf{c}^9(u)$ and expands from $(X_6, 0, 0)^T$ to $(X_{10}, 0, 0)^T$. No frame needs to be constructed; see above.
14. The **Front Flat Bottom Diagonal** $\mathbf{c}^{14}(u)$ is constructed analogously to Curve 10 and connects $(X_6, Y_2/2 - Y_7, 0)^T$ with $(X_{10}, 0, 0)^T$ where at both points the tangent direction $\mathbf{T} = (1, 0, 0)^T$ is imposed. As in the above case of diagonal curves, we construct a constant frame $\mathbf{f}^{14}(u) = \mathbf{F} = (0, 0, -1)^T$.
15. As for the **Front Flat Side Base** $\mathbf{c}^{15}(u)$, we have the same analogy as above and connect $(X_6, Y_2/2, Z_2)^T$ with $(X_{11}, Y_2/2, Z_2)^T$.
16. The **Front Flat Side Diagonal** $\mathbf{c}^{16}(u)$ connects $(X_6, Y_2/2, Z_7)^T$ with $(X_{11}, Y_2/2, Z_2)^T$. At the former point we impose the tangent direction $(1, 0, 0)^T$ and construct this curve by Hermite interpolation. The constant frame is given by $\mathbf{f}^{16}(u) = \mathbf{F} = (0, 1, 0)^T$.

Stage I, Bow: We continue with two curves of the bow whose positional and tangential data is governed by the input parameters, compare to Figure IV.5 on page 97.

17. The **Bow Deck Curve** $\mathbf{c}^{17}(u)$ connects the end of the flat-of-side $\mathbf{P}_0 = (X_{11}, Y_2/2, Z_3)^T$ with the foremost point of the forward overhang. Tangential information is only provided at the start, $\mathbf{T}_0 = (1, 0, 0)^T$, and the curve is constructed by Hermite interpolation. Since this curve does not bound any patch, compatibility can be overlooked in the frame construction. However, the existence of this frame $\mathbf{f}^{17}(u)$ serves as guidance to the construction of the Bow Start Section $\mathbf{c}^{31}(u)$. The start frame vector \mathbf{F}_0 is simply the positive transverse direction. The end frame \mathbf{F}_1 , on the other hand, is the rotation of the Forward Overhang Tilt Vector $(\cos \Phi_{30}, 0, \sin \Phi_{30})^T$ into the normal plane of the end tangent. Let

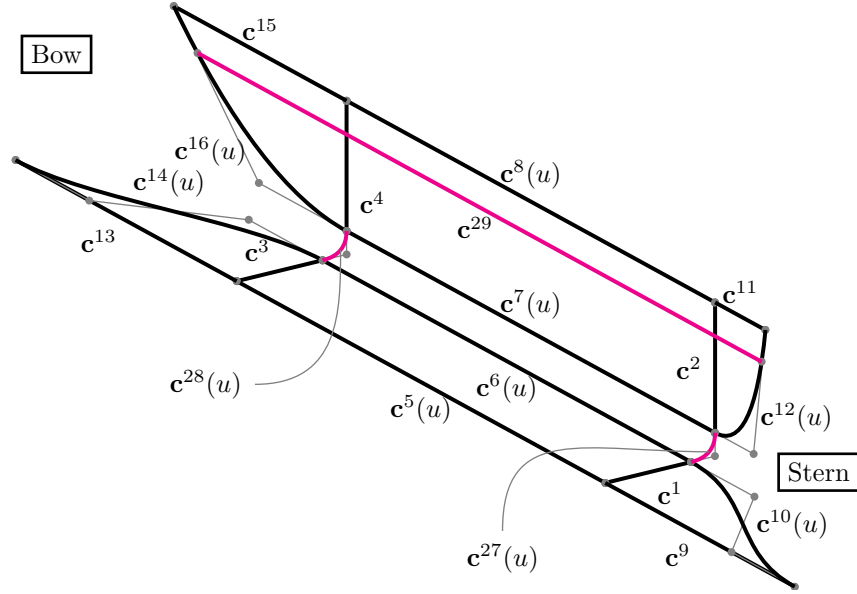


Figure IV.2: The curves of the midship.

$\mathbf{t}_1 = \mathbf{c}^{17'}(u)$ and α be its angle with the x -axis, then \mathbf{F}_1 is given by

$$\mathbf{F}_1 = \begin{pmatrix} \cos(\frac{\pi}{2} - \alpha) \cos \Phi_{30} \\ \sin(\frac{\pi}{2} - \alpha) \\ \sin \Phi_{30} \end{pmatrix}.$$

18. The planar **Bow Profile Curve** has one of the most complicated geometries of the model. It connects the end point of the flat-of-bottom with the front tip of the forward overhang. On its interior it interpolates various points and tangent directions all of which are specified by the input data, also found in Figure IV.3:

- (a) Starting point is the end of the flat of bottom with the tangent direction of the x -axis:

$$\mathbf{P}_0 = \begin{pmatrix} X_{10} \\ 0 \\ 0 \end{pmatrix}, \quad \mathbf{T}_0 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}.$$

- (b) The point of the start of the bulb rise is the next:

$$\mathbf{P}_1 = \begin{pmatrix} X_{19} \\ 0 \\ 0 \end{pmatrix}, \quad \mathbf{T}_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}.$$

Note that the tangents are the same as the interpolating line: $\mathbf{T}_0 = \mathbf{T}_1 = \frac{\mathbf{P}_1 - \mathbf{P}_0}{\|\mathbf{P}_1 - \mathbf{P}_0\|}$.

- (c) The bulb rise point lies on the forward perpendicular and has z -coordinate of the Bulb Rise parameter: $\mathbf{P}_2 = (X_1, 0, Z_{20})^T$.

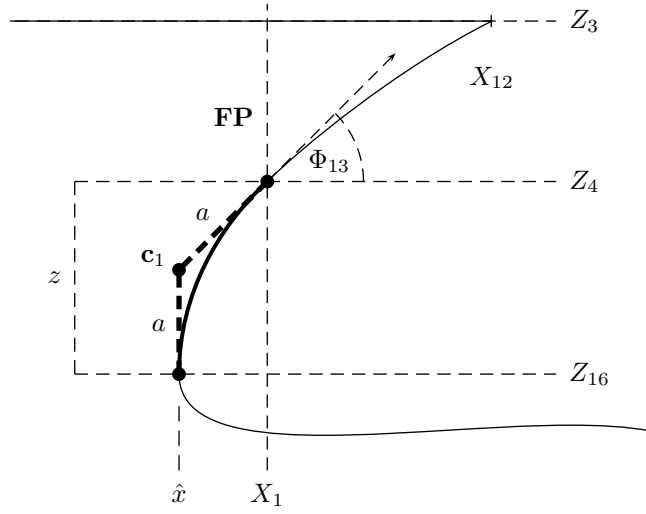


Figure IV.4: Sketch of the determination of the Bulb Root Point, item 18e of $\mathbf{c}^{18}(u)$. With $z := Z_4 - Z^{16}$, $\mathbf{c}_1 = (\hat{x}, 0, Z^{16} + a)^T$ being the location of the middle control point and $\sin \Phi_{13} = \frac{z-a}{a}$, we get $a = \frac{z}{1 + \sin \Phi_{13}}$. With $\cos \Phi_{13} = \frac{X_1 - \hat{x}}{a}$ we obtain $\hat{x} = z \frac{\cos \Phi_{13}}{1 + \sin \Phi_{13}}$.

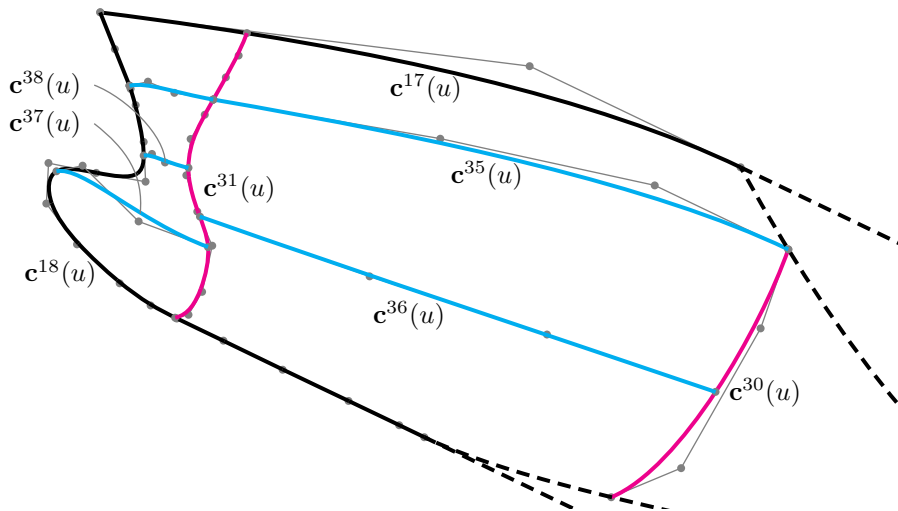
- (g) At last, we interpolate the tip of the forward-overhang in the Deck Plane: $\mathbf{P}_6 = (X_1 + X_{12}, 0, Z_3)^T$.

In order to guarantee a straight line segment between \mathbf{P}_0 and \mathbf{P}_1 , we construct two curve parts: The line $(1-u)\mathbf{P}_0 + u\mathbf{P}_1$ is the first and an interpolation of the points $\mathbf{P}_1, \dots, \mathbf{P}_6$ with their associated tangents is the second. In this interpolation we incorporate the condition of $\kappa = 0$ at \mathbf{P}_1 and, thereby, ensure the two parts to connect curvature continuously. After a linear reparametrization of the first part, the two parts can be connected into one B-spline curve $\mathbf{c}^{18}(u)$. However, the bulbs of the parent ships tend to exhibit a rounder, less pointy bulb, i.e. a longer tangent at the bulb front point. To this end we have introduced the Bulb Fat parameter R_{31} which we use to rescale $\mathbf{t}_3 = \mathbf{c}(u_3)$ and employ again Algorithm III.1.14 with the tangent vector \mathbf{t}_3 instead of the original tangent direction \mathbf{T}_3 .

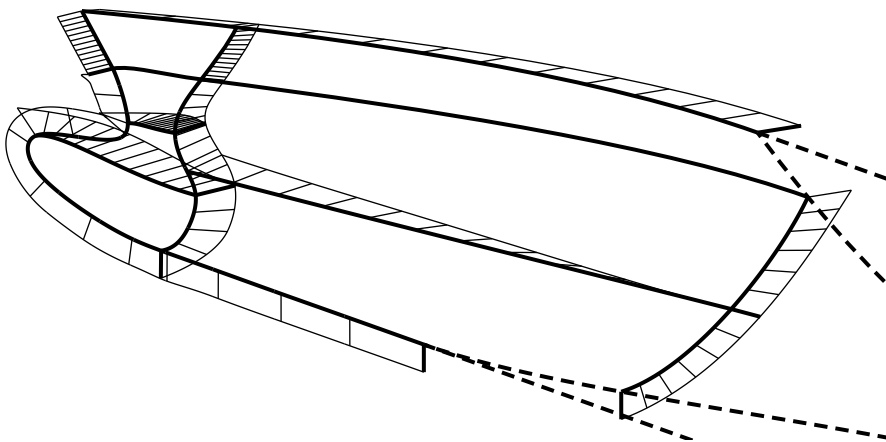
As for the frame, we keep in mind the mirror of the hull at the Centre Plane. A smooth transition requires the entire frame $\mathbf{f}^{18}(u)$ up to the waterline to lie in the Centre Plane. To this end, we provide frame vectors as the straight-forward curve normals in this plane at \mathbf{P}_0 and \mathbf{P}_6 and employ Algorithm III.2.13.

Stage I, Stern: As for the stern, we have eight more curves belonging to Stage I, compare to Figure IV.6 on page 100.

19. The **Stern Deck Curve** $\mathbf{c}^{19}(u)$ is constructed by a simple Hermite curve as we have two points and two tangents to interpolate. Our start point



(a) Curves and control points.



(b) Curves and frames.

Figure IV.5: The stages of the wireframe in the bow. Stage I is coloured in black, stage II in magenta and cyan curves mark stage III.

is the widest point of the transom back plane, which lies also in the Deck Plane, and we impose the tangent direction of the $(1, 0, 0)^T$. The end point is given by the start point of the flat-of-side, which lies in the Deck Plane as well, and we impose the same tangent direction.

As in the bow area, the Deck Curve's frame $\mathbf{f}^{19}(u)$ is constructed straight forward as we have a Bézier curve and no compatibility to consider. At the start, the frame direction is determined by Φ_{34} , $(0, \sin \Phi_{34}, \cos \Phi_{34})^T$, and at the end the frame direction is $(0, 1, 0)^T$.

20. The **Shaft Profile** $\mathbf{c}^{20}(u)$ connects the lowest point of the back shaft surface, $\mathbf{P}_0 = (X_{25}, 0, Z_{25} - Z_{29})$, with the aft-most point of the flat-of-bottom, $\mathbf{P}_2 = (X_8, 0, 0)^T$, and at both points the tangent direction is the x -axis. On the interior we interpolate another point which depends on the internal parameter of the shaft rise $\mathbf{P}_1 = (X_{35}, 0, 0)^T$. Similarly to the Bow Profile $\mathbf{c}^{18}(u)$, we construct two separate parts. In this way we guarantee a straight line segment from \mathbf{P}_1 to \mathbf{P}_2 . The condition of $\kappa = 0$ at \mathbf{P}_1 for the first part ensures that parts connect curvature continuously. We impose the frame vector $(0, 0, -1)^T$ at all the three interpolation points in order to obtain a planar frame $\mathbf{f}^{20}(u)$.

21. The **Shaft Side** $\mathbf{c}^{21}(u)$ is the circle-like curve that bounds the back plane of the shaft. We can construct this curve as a true circle in rational B-Spline representation. However, we opt to employ our standard interpolation routine since the Coons patch blending is not implemented for rational curves. Its construction is straight forward as we interpolate the two extreme points in vertical direction with tangents in transverse direction, $\mathbf{P}_0 = (X_{24}, 0, Z_{25} - Y_{29})^T$, $\mathbf{P}_2 = (X_{24}, 0, Z_{25} + Y_{29})^T$. Additionally, we ensure the distance of Y_{29} in transverse direction and impose the tangent in vertical direction there: $\mathbf{P}_1 = (X_{24}, Y_{29}, Z_{25})^T$.

Since this corresponds to a boundary where no G^1 continuity has to be provided, we neglect to construct a frame.

22. The **Shaft Stem** connects the two extreme points in vertical direction of the shaft: $\mathbf{P}_0 = (X_{24}, 0, Z_{25} - Z_{29})^T$, $\mathbf{P}_1 = (X_{24}, 0, Z_{25} + Z_{29})^T$.

This line bounds only a planar patch and, therefore, no framing is necessary.

23. The **Transom Profile** $\mathbf{c}^{23}(u)$ connects the top point of the shaft with the bottom of the transom. We interpolate the following four points:

- (a) Our initial point with the corresponding tangent direction is

$$\mathbf{P}_0 = \begin{pmatrix} X_{24} \\ 0 \\ Z_{25} + Z_{29} \end{pmatrix}, \quad \mathbf{T}_0 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}.$$

- (b) Next, this curve has to pass through the stern root:

$$\mathbf{P}_1 = \begin{pmatrix} X_{27} \\ 0 \\ Z_{28} \end{pmatrix}, \quad \mathbf{T}_1 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.$$

(c) The propeller clearance is an essential design parameter:

$$\mathbf{P}_2 = \begin{pmatrix} X_{24} \\ 0 \\ Z_{25} + Z_{26} \end{pmatrix}.$$

(d) Finally, we connect to the transom bottom point:

$$\mathbf{P}_3 = \begin{pmatrix} X_{21} \\ 0 \\ Z_4 - Z_{23} \end{pmatrix}.$$

Our interpolation routine has proven to yield a satisfactory curve.

Since this is the ship's profile, the frame has to be planar in the Centre Plane. The necessary frame vector directions are, therefore, given by the intersection of normal plane of the curve with the x - z plane. No compatibility has to be considered in this frame construction.

24. The **Transom Side** $\mathbf{c}^{24}(u)$ connects the bottom point of the transom with its side point by a Hermite curve: $\mathbf{P}_0 = (X_{21}, 0, Z_4 - Z_{23})^T$, $\mathbf{P}_1 = (X_{21}, Y_{22}, Z_3)^T$. While at the former point the curve parts with transverse direction, we use the fixed tilt parameter Φ_{34} to determine that of the end point: $\mathbf{T}_1 = (0, \sin \Phi_{34}, \cos \Phi_{34})^T$.

Note, that no frame has to be constructed since no cross-boundary G^1 continuity is required.

25. The **Transom Stem** connects the two extreme points in vertical direction of the transom:

$$\mathbf{P}_0 = \begin{pmatrix} X_{21} \\ 0 \\ Z_4 - Z_{23} \end{pmatrix}, \quad \mathbf{P}_1 = \begin{pmatrix} X_{24} \\ 0 \\ Z_3 \end{pmatrix}.$$

This line bounds only a planar patch and, therefore, no framing is necessary.

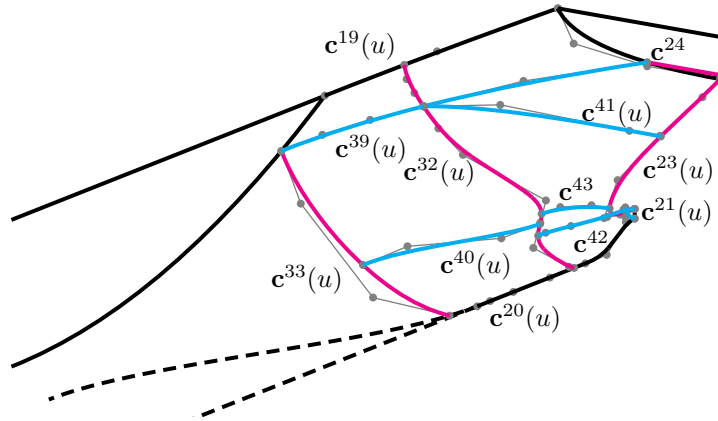
26. The **Transom Bar** $\mathbf{c}^{26}(u)$ is connects the two extreme points of the transom on the Deck Plane. Since this curve has no influence on the wetted surfaces of the hull, we construct it merely for the sake of completeness.

IV.1.2 Stage II

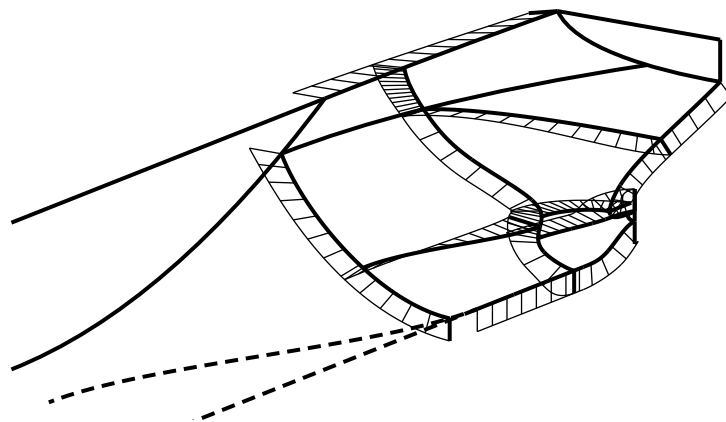
Based on the input parameters and the curves and frames constructed in Stage I, we build the curves of this Stage II. As above, we create an unordered set of curves as they do not depend on each other.

Stage II, Midship:

27. The **Parallel Midship Back Corner** $\mathbf{c}^{27}(u)$ is the section curve of the bilge at $x = X_5$ and provides a smooth transition between the Base $\mathbf{c}^1(1)$ and Stem $\mathbf{c}^2(0)$. It is, thus, the result of a Hermite interpolation as in



(a) Curves and Control Points



(b) Curves and Frames

Figure IV.6: The stages of the wireframe in the stern. Stage I is coloured in black, stage II in magenta and cyan curves mark stage III.

Section III.1.1 with two points and two tangents. Optimally, this curve is a circle for which we would have to employ rational B-splines. Since the framing algorithm is not implemented for those, we opt to employ a cubic Hermite curve.

As for the frame $\mathbf{f}^{27}(u)$, the basic ship characteristic tells us that this curve bounds an extruded surface, the bilge. Therefore the frame $\mathbf{f}^{27}(u)$ has to lie in the y - z plane. Its input boundary frames are obviously $(0, 0, -1)^T$ and $(0, 1, 0)^T$, respectively.

Note that it is the nature of the bilge that forces the frame to lie in the y - z plane. However, this is not guaranteed a-priori if we construct the frame to be compatible with the Flat-of-Bottom Diagonal and the Flat Side Diagonal, respectively. Compatibility is, in turn, imperative for this curve and reads as follows for the joint of this Corner Curve with the Flat Bottom Diagonal $\mathbf{c}^{10}(u)$:

$$d^{27}\mathbf{f}_1^{27} \cdot \mathbf{t}_0^{10} = d^{10}\mathbf{f}_1^{10} \cdot \mathbf{t}_1^{27}. \quad (\text{IV.1})$$

$$d^{27}\mathbf{f}_1^{27} \cdot \mathbf{t}_0^{10} = d^{10}(0, 0, -1)^T \cdot (0, 1, 0)^T = 0. \quad (\text{IV.2})$$

We see that \mathbf{f}_1^{27} has to be orthogonal to the tangent of the Diagonal curve. This is, in fact, fulfilled for any vector \mathbf{f}_1^{27} in the y - z plane if $\mathbf{t}_0^{10} = (1, 0, 0)^T$ as is the case in our construction. If we perturb this tangent, the condition for \mathbf{f}_1^{27} changes and the frame is not orthogonal to the extruded Bilge surface.

If we changed the hierarchy between the Diagonal Curves and this Corner Curve, we would obtain at a similar condition for the Flat Bottom Patch and we would need $\mathbf{t}_0^{27} = (0, 1, 0)^T$ to produce a constant frame $\mathbf{f}^{10}(u)$.

28. The **Parallel Midship Front Corner** $\mathbf{c}^{28}(u)$ and its frame $\mathbf{f}^{28}(u)$ are constructed analogously to $\mathbf{c}^{27}(u)$ at $x = X_6$.
29. The **Midship Waterline** $\mathbf{c}^{29}(u)$ is the straight line connecting the intersection point of the Front and Back Flat Side Diagonals with the Draft Plane. Since all the patches that are bounded by this curve are planar, we have no need to construct a frame.

Stage II, Bow: We start in the bow, where there are only two curves, both of which suggest section-like curves of the hull:

30. The **Midship Bow Curve** $\mathbf{c}^{30}(u)$ finds its longitudinal position \hat{x} at the point where the Front Flat Side Diagonal $\mathbf{c}^{16}(u)$ intersects the Draft Plane: $\mathbf{P}_1 = (\hat{x}, Y_2/2, Z_4)^T$. As we try to parametrize away from the origin, this shall be the curve's end point.

As for the start point, we find the point on the bottom which has the same longitudinal position: $\mathbf{P}_0 = (\hat{x}, y, 0)^T$. Depending on the relative position of \hat{x} to that of the Flat Bottom End Point X_{10} , \mathbf{P}_0 can lie on the Bulb Profile Curve $\mathbf{c}^{18}(u)$ or on the Front Flat Bottom Diagonal $\mathbf{c}^{14}(u)$. In the both cases, another intersection problem yields the y -coordinate \hat{y} and, thus, the start point $\mathbf{P}_0 = (\hat{x}, \hat{y}, 0)^T$.

The tangent planes at both points are given by the constant frames $(0, 0, -1)$ and $(0, 1, 0)$, respectively. Within these planes, we construct the tangents

\mathbf{T}_0 and \mathbf{T}_1 : \mathbf{T}_0 is the normal vector of $\mathbf{c}^{14}(u)$ or $\mathbf{c}^{18}(u)$ at \mathbf{P}_0 . At the end point \mathbf{P}_1 we average the waterline's inverse tangent (which we know a-priori by the existing frame at \mathbf{P}_1 and the waterline's placement in the x - y plane) with the tangent direction of the Diagonal $\mathbf{c}^{16}(u)$. This yields \mathbf{T}_1 and $\mathbf{c}^{30}(u)$ is then obtained by Hermite interpolation.

As for the determination of the frame $\mathbf{f}^{30}(u)$, we know the boundary frames from the existing curve frames and we emphasize that compatibility with $\mathbf{f}^{14}(u)$ and $\mathbf{f}^{16}(u)$ which has to be considered at the respective ends.

31. The **Bow Front Section** $\mathbf{c}^{31}(u)$ is a planar curve at $x = X_{32}$, i.e. a constant shift from the root of the Bow Profile $\mathbf{c}^{18}(u)$.

- (a) Its start point \mathbf{P}_0 is the intersection of the Bulb Profile Curve with the y - z plane at $x = X_{32}$. Since we construct a planar curve, the start tangent has to be $\mathbf{T}_0 = (0, 1, 0)^T$ which is orthogonal to the local frame.
- (b) The next point to interpolate is the bulb side-point whose coordinates are fully given by the input parameters. The nature of this point forces the tangent to be in vertical direction:

$$\mathbf{P}_1 = \begin{pmatrix} X_{32} \\ Y_{17} \\ Z_{18} \end{pmatrix}, \quad \mathbf{T}_1 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.$$

- (c) Since the shape of this curve is similar to that of the Bow Profile, the next point is the root, i.e. that point between the bulb side point and the waterline which is closest to the Centre Plane. Since we are close to the root of the Profile in longitudinal direction, we take the same root height as in the Profile: Z_{16} . The y -coordinate is determined by the fixed parameter $Y_{33}/2$:

$$\mathbf{P}_2 = \begin{pmatrix} X_{32} \\ Y_{33}/2 \\ Z_{16} \end{pmatrix}, \quad \mathbf{T}_2 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.$$

- (d) Finally, we intersect the Deck Curve at $x = X_{32}$ to obtain \hat{y} and interpolate $\mathbf{P}_3 = (X_{32}, \hat{y}, Z_3)^T$. The tangent is determined by the intersection of the Deck Curve's tangent plane, which is defined by its frame, and the y - z plane of this curve: \mathbf{T}_3 .

The frames of the existing curves at the start and end constitute the specifications for the frame $\mathbf{f}^{31}(u)$ of the current curve. It is left to mention, that we have to regard the compatibility condition at the start since it is part of the wetted ship hull.

Stage II, Stern: As for the stern, we have two section-like curves which belong to Stage II of the stern-wireframe, compare to Figure IV.6 on page 100:

32. The **Stern Back Section** $\mathbf{c}^{32}(u)$, compare to Figure IV.7, is the section curve in the y - z plane $x = X_{35}$ which represents the constant shift of the root point in longitudinal direction. We interpolate three points

IV.1. THREE STAGES OF THE WIREFRAME

and tangents. The first and last are obtained by the intersection of the $y-z$ plane with the Shaft Profile $\mathbf{c}^{20}(u)$ and the Deck Curve $\mathbf{c}^{19}(u)$, respectively: $\mathbf{P}_0, \mathbf{P}_2$. At those points, the existing frames in combination with the plane of construction mark the tangent directions: $\mathbf{T}_0, \mathbf{T}_2$.

The location in vertical direction of the middle point \mathbf{P}_1 is given by the intermediate of the height of shaft and that of the root and its transverse expansion is given by the Shaft Beam: $y = Y_{37}/2$. The tangent \mathbf{T}_1 at this point marks the shape of the curve. It is governed by the parameter Φ_{38} which is the angle that \mathbf{T}_1 shall have with the vertical axis:

$$\mathbf{P}_1 = \begin{pmatrix} X_{35} \\ Y_{37}/2 \\ 1/2(Z_{25} + Z_{28}) \end{pmatrix}, \quad \mathbf{T}_1 = \begin{pmatrix} 0 \\ -\sin \Phi_{38} \\ \cos \Phi_{38} \end{pmatrix}.$$

Therefore \mathbf{T}_1 points towards the ship Centre Plane for positive Φ_{38} .

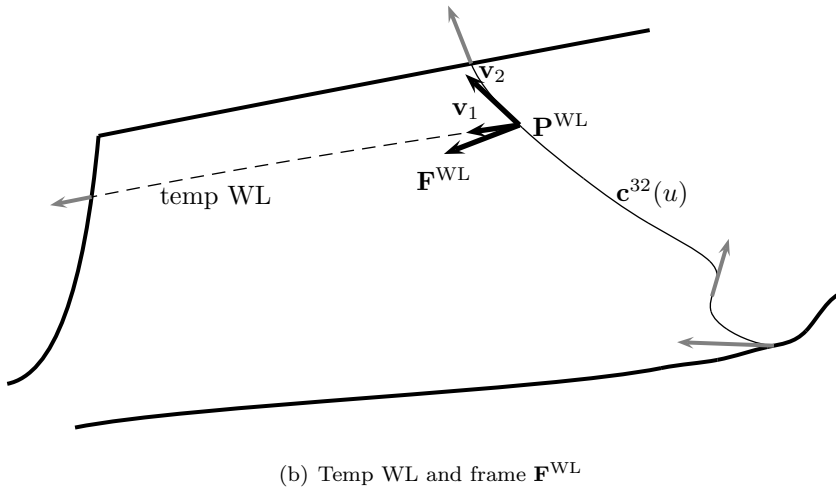
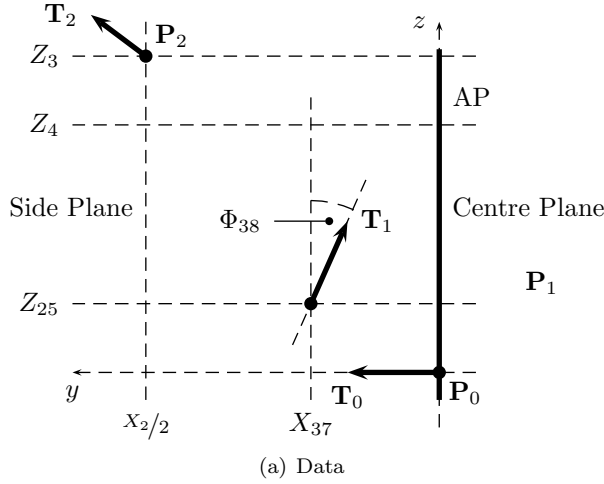


Figure IV.7: The construction of the Stern Back Section $\mathbf{c}^{32}(u)$.

While the curve is planar, this is not necessarily true for its frame $\mathbf{f}^{32}(u)$. Its boundary vectors are determined by the frames $\mathbf{f}^{20}(u)$ and $\mathbf{f}^{19}(u)$, respectively. The determination of the frame at the height of the Draft significantly improves the quality of the involved curves and patches if done in the following way. We construct an auxiliary curve which we shall refer to as Temp Waterline, interpolating two points in the Draft Plane. The first is the point \mathbf{P}^{WL} of this very Stern Back Section $\mathbf{c}^{32}(u)$ at $z = Z_4$ while the second point lies on the Back Flat Side Diagonal $\mathbf{c}^{12}(u)$ at $z = Z_4$. With the tangent direction $\mathbf{T} = (1, 0, 0)^T$ at the latter point, the Temp Waterline is created by Hermite interpolation. The evaluation of its unit tangent at \mathbf{P}^{WL} yields the vector \mathbf{v}_1 which shall lie in the tangent plane of the Stern Back Section. With \mathbf{v}_2 being the unit tangent of $\mathbf{c}^{32}(u)$ at \mathbf{P}^{WL} , we obtain the frame direction $\mathbf{F}^{\text{WL}} = \mathbf{v}_2 \times \mathbf{v}_1$ at \mathbf{P}^{WL} .

33. The **Midship Stern Section** $\mathbf{c}^{33}(u)$ is the section-like curve constructed analogously to that of the bow $\mathbf{c}^{30}(u)$. The intersection of the Back Flat Side Diagonal $\mathbf{c}^{12}(u)$ with the Draft Plane gives the position of the end point to interpolate: \mathbf{P}_1 . We find the point at the bottom (Back Flat Bottom Diagonal $\mathbf{c}^{10}(u)$ or Bow Profile $\mathbf{c}^{18}(u)$) with the same x -coordinate as \mathbf{P}_1 and obtain \mathbf{P}_0 . The tangent directions are determined as above as the normals to the existing curves in their tangent planes: $\mathbf{T}_0, \mathbf{T}_1$. At \mathbf{P}_1 we average this vector with the known direction $(1, 0, 0)^T$ of the waterline which itself has yet to be constructed. The curve is finally obtained by Hermite interpolation.

By ensuring the compatibility at the start and end of the curve, we can construct the frame with the frame vectors provided by the existing curves' frames.

34. The **Transom Back Waterline** $\mathbf{c}^{34}(u)$ is the straight line in the Draft Plane, at $z = Z_4$, and the transom's back plane. \mathbf{P}_0 and \mathbf{P}_1 are the result of intersections of this plane with the transom's stem $\mathbf{c}^{22}(u)$ and the Transom Side $\mathbf{c}^{24}(u)$, respectively.

We can neglect the construction of a frame since this curve bounds a planar patch.

IV.1.3 Stage III

In Stage III we construct the last set of curves whose hierarchical order has to be, in general, respected as they can depend on each other, which is the principal characteristic of this stage.

Stage III, Bow: We start with the last control curves of the bow, compare to Figure IV.5 on page 97.

35. The most essential curve of these is the **Bow Waterline** $\mathbf{c}^{35}(u)$ as it is the missing curve to bound the bow's wetted surface. It is constructed as the interpolation of three points with three tangent directions.
 - \mathbf{P}_0 is the point of the Front Flat Side Diagonal $\mathbf{c}^{16}(u)$ at the Draft, which is also the start point of the Bow Midship Section $\mathbf{c}^{30}(u)$. Obviously, \mathbf{T}_0 is in longitudinal direction.

- The point of the Bow Front Section $\mathbf{c}^{31}(u)$ at the height of the Draft is the second interpolation point: \mathbf{P}_1 . \mathbf{T}_1 is given by the intersection of the Draft Plane, i.e. at $z = Z_4$, with the tangent plane given by the existing frame $\mathbf{f}^{31}(u)$ at \mathbf{P}_1 .
- Finally, \mathbf{P}_2 is the point of the Bow Profile Curve in the Draft Plane. The tangent is, like above, the intersection of the Draft Plane with the plane given by the frame at \mathbf{P}_2 .

Since \mathbf{P}_1 marks the vertex of two patches and \mathbf{T}_1 guarantees G^1 continuity of $\mathbf{c}^{35}(u)$, this curve can be constructed in two parts, each interpolating a pair of points and tangents.

The construction of a frame is not necessary since no cross-boundary G^1 continuity is needed.

36. The **Bow Horizontal Curve** $\mathbf{c}^{36}(u)$ is the curve which expands over the transitional region between the Midship and the Bow in principally longitudinal direction. We connect the Midship Bow Section $\mathbf{c}^{30}(u)$ with the Bow Front Section $\mathbf{c}^{31}(u)$ with a Hermite curve. In detail, we choose an interior point of the former, $\mathbf{P}_0 = \mathbf{c}^{30}(0.5)$, and a point of the latter curve which lies between the Bulb Side Point and root point: \mathbf{P}_1 . Since at both points we are already given frames, we project the vector $\mathbf{v} = (1, 0, 0)^T$ into the these tangent planes to obtain \mathbf{T}_0 and \mathbf{T}_1 , respectively.

The given frames with the compatibility conditions at both points determine the frame for this Bézier curve.

37. The **Bulb Curve** $\mathbf{c}^{37}(u)$ connects the Bulb Side Point with Bulb Front Point. Therefore, we have

$$\mathbf{P}_0 = \begin{pmatrix} X_{32} \\ Y_{17} \\ Z_{18} \end{pmatrix}, \quad \mathbf{P}_1 = \begin{pmatrix} X_1 + X_{14} \\ 0 \\ Z_{15} \end{pmatrix}.$$

The tangents \mathbf{T}_0 & \mathbf{T}_1 are the projections of the respective vectors $\mathbf{v} = (1, 0, 0)^T$ and $\mathbf{v} = (0, -1, 0)^T$ into the local tangent planes at \mathbf{P}_0 and \mathbf{P}_1 , respectively.

The existing frames and the compatibility condition at both points provide the input to the frame construction.

38. The **Bulb Root Curve** $\mathbf{c}^{38}(u)$ connects the Root of the Bow Section with that of Bulb Profile Curve:

$$\mathbf{P}_0 = \begin{pmatrix} X_{32} \\ Y_{33} \\ Z_{16} \end{pmatrix}, \quad \mathbf{P}_1 = \begin{pmatrix} \hat{x} \\ 0 \\ Z_{16} \end{pmatrix},$$

where \hat{x} is the x -coordinate calculated in item 18e.

The tangents and frames are calculated analogously to those of the above Bulb Curve $\mathbf{c}^{37}(u)$. As for all curves in this stage, compatibility has to be ensured at both points.

This concludes the set of curves for the bow. Note that we have produced one triangular patch, the Bulb Bottom Patch $\mathbf{S}^{13}(u, v)$, which is bounded by the Bow Profile Curve $\mathbf{c}^{18}(u)$, the Bow Section $\mathbf{c}^{31}(u)$ and the Bulb Curve $\mathbf{c}^{37}(u)$. We see in Section IV.2 below how to we construct this patch.

The Bow Horizontal Curve $\mathbf{c}^{36}(u)$ and Bulb Root Curve $\mathbf{c}^{38}(u)$ subdivide already quadrilateral curve loops. From a topological point-of-view their existence is, therefore, not secondary. However, these curves prove to be fundamental for the control over the shape of the affected patches. Since they are also the only curves with this characteristic, we can regard this network of control curve as minimal.

Stage III, Stern: We continue with Stage III of the stern, the last set of curves. Compare to Figure IV.6 on page 100.

39. Like in the bow area, we are missing one curve to completely bound the wetted hull in the stern: the **Stern Waterline** $\mathbf{c}^{39}(u)$. It is constructed as the interpolation of three points with two tangent directions.

- The first point \mathbf{P}_0 is the intersection of the Transom Side Curve with the Draft Plane, where we do not specify a tangent.
- The section point is the point of the Stern Back Section $\mathbf{c}^{32}(u)$ at the height of the Draft: \mathbf{P}_1 . \mathbf{T}_1 is again given by the intersection of the Draft Plane with the tangent plane given by the existing frame at \mathbf{P}_1 .
- The last point, \mathbf{P}_2 , is given by the end point of the Midship Stern Section $\mathbf{c}^{33}(u)$. It is also the intersection of the Draft Plane with the Back Flat Side Diagonal and \mathbf{T}_2 is the longitudinal direction.

This curve can be constructed in two parts, each interpolating a pair of points and tangents which is done by Hermite interpolation.

No frame is constructed here since no cross-boundary G^1 continuity is needed.

40. The **Stern Horizontal Curve** connects the Stern Back Section $\mathbf{c}^{32}(u)$ and Midship Stern Section $\mathbf{c}^{33}(u)$. We let \mathbf{P}_0 be the point \mathbf{P}_1 from the construction of $\mathbf{c}^{32}(u)$ and for \mathbf{P}_1 we evaluate $\mathbf{c}^{33}(u)$ at an interior parameter. For the tangents \mathbf{T}_0 and \mathbf{T}_1 , we project the vector $\mathbf{v} = (1, 0, 0)^T$ into the tangent planes at \mathbf{P}_0 and \mathbf{P}_1 , respectively.

The frame data is determined by the existing frame vectors at \mathbf{P}_0 and \mathbf{P}_1 . The consideration of the compatibility condition is again fundamental and we obtain $\mathbf{f}^{32}(u)$.

41. The **Transom Diagonal** $\mathbf{c}^{41}(u)$ is to create quadrilateral curve loops in the transom area. We interpolate the point \mathbf{P}_0 of the propeller clearance (of the Transom Profile Curve $\mathbf{c}^{23}(u)$) with the point \mathbf{P}_1 of the Stern Section $\mathbf{c}^{32}(u)$ in the Draft Plane. The tangent directions are $\mathbf{T}_0 = (0, 1, 0)^T$ and the projection of $\mathbf{v} = (1, 0, 1)$ into the local tangent plane at \mathbf{P}_1 .

In this way, also the frame data is directly determined by the existing frames at \mathbf{P}_0 and \mathbf{P}_1 , respectively, and we have to consider the compatibility conditions with $\mathbf{f}^{23}(u)$ and $\mathbf{f}^{32}(u)$, respectively.

42. The **Shaft Curve** $\mathbf{c}^{42}(u)$ is a curve with principally horizontal expansion from the start point \mathbf{P}_0 of the Transom Profile $\mathbf{c}^{23}(u)$ to that point of the Stern Section $\mathbf{c}^{32}(u)$ that has $z = Z_{25}$: \mathbf{P}_2 . At these points, we give the tangent directions by units of the projections of the vectors $(1, 10, 0)^T$ and $(1, 0, 0)^T$, respectively, into the local tangent planes: \mathbf{T}_0 & \mathbf{T}_2 .

However, this data is not enough to create a reasonable Shaft Curve. We employ the following local auxiliary Temp Curve, compare to Figure IV.8, for which we interpolate the side point of Shaft Side with \mathbf{P}_2 . The Hermite interpolation of those points with the projection of $\mathbf{v} = (1, 0, 0)$ into the respective tangent planes as the corresponding tangents yields the Temp Curve, which can be considered a smooth and reasonable transition between the Shaft Side and the Stern Back Section. Its frame is obtained by the natural boundary frames which are constructed to be orthogonal to this Temp Curve and the existing local curves.

The Temp Curve shall serve as a guide for the Shaft Curve from point \mathbf{P}_1 which is determined by the intersection of Temp Curve with the y - z plane at the Root Exit $x = X_{27}$. Therefore, we give \mathbf{P}_1 as the second interpolation point together the tangent \mathbf{T}_1 of Temp Curve at \mathbf{P}_1 .

The frame data for this Shaft Curve at \mathbf{P}_0 and \mathbf{P}_2 is determined by the existing frame vectors of $\mathbf{c}^{23}(u)$ and $\mathbf{c}^{32}(u)$, respectively: \mathbf{F}_0 and \mathbf{F}_2 . At \mathbf{P}_1 , we impose the frame vector of Temp Curve at this point, \mathbf{F}_1 .

Compatibility has to be ensured at the start and the end, with respect to $\mathbf{f}^{23}(u)$ and $\mathbf{f}^{32}(u)$, respectively.

Note that \mathbf{P}_0 is also the end point of the Shaft Side Curve $\mathbf{c}^{21}(u)$. Since that curve is not framed, compatibility with respect to $\mathbf{c}^{21}(u)$ is not an issue.

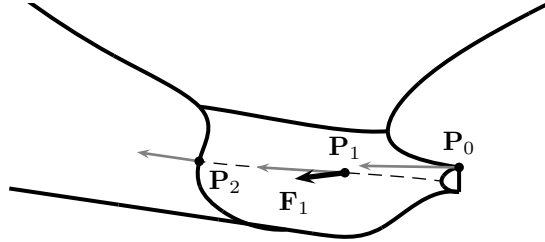


Figure IV.8: Construction of the Shaft Curve $\mathbf{c}^{42}(u)$.

43. The **Stern Root Curve** $\mathbf{c}^{43}(u)$ is a curve with principal expansion in longitudinal direction. It interpolates the root point \mathbf{P}_0 of the Transom Profile Curve and the point \mathbf{P}_1 of the Stern Back Section of equal height. At the former we impose $\mathbf{T}_0 = (0, 1, 0)$ which is orthogonal to the local frame vector. At the latter we project the vector $\mathbf{v}_0 = (1, 0, 0)$ into the tangential plane defined by the local frame and obtain \mathbf{T}_1 .

Since we are given frame data at \mathbf{P}_0 and \mathbf{P}_1 , the construction of the frame is straight forward with the consideration of the compatibility condition.

Having constructed all curves of the stern (and therefore all 43 curves of the wireframe) we refer again to Figure IV.6 and point out the complicated

topology behind the Stern Back Section. Six curves bound this region and by the insertion of the Transom Diagonal and the Shaft Curve, we obtain three quadrilateral loops. In this sense the Root Curve is secondary as it subdivides an already quadrilateral loop. The same holds for the Stern Horizontal Curve. Both considerably help to guide the patch geometry in their respective areas and we can, therefore, regard this as a minimal curve network for the Stern area.

IV.2 Ship Hull Patch Network

The existence of a neatly constructed wireframe allows the construction of a surface patch network constituting the ship hull. The wireframe was defined and constructed in a way such that intersections exist at reasonable locations and facilitate the construction of surfaces. Therefore, we interpret the above constructed control curves as the union of curve loops.

Boundary Loops: The majority of these loops are quadrilateral and in these cases, the preparation of the boundary curves for the Coons patch construction takes place in two steps:

1. Typically, each boundary curve is only a subcurve of a member of the wireframe. We can find the start parameter a and the end parameter b by proper intersection problems. The restriction of the curve to the parameter interval $[a, b]$ with $0 \leq a < b \leq 1$ yields the boundary curve. With a linear reparametrization we obtain the same curve geometry on the parameter interval $[0, 1]$. As for the frames, obviously the same reparametrization has to be performed as for their respective curves.
2. Once all boundary curves have parameter intervals $[0, 1]$, we have to reorient them appropriately in order for corresponding end vertices to meet as demanded by the Coons patch construction. We do this in two steps in which all operations apply to the respective frames as well:
 - (a) Firstly, we generate a loop, i.e. we orient the curve parametrizations such that the end point of one curve is the start point of the next. We choose one curve $\mathbf{a}(u)$ and denominate the following curves as $\mathbf{b}(v)$, $\mathbf{c}(u)$ and $\mathbf{d}(v)$, respectively. We refer to the set of curves with this ordering and these parametrizations as the patch's *boundary loop*.
 - (b) In the second step, we invert the parameter directions of $\mathbf{c}(u)$ and $\mathbf{d}(v)$ and obtain a set of curves, which is prepared for the Coons patch construction of Algorithms III.4.1 and III.4.7.

For triangular loops, we see below how to artificially generate a quadrilateral loop to which these two steps can be applied.

Ribbon Construction: The next step of the patch construction is to create cross-tangent ribbons in the manner of Algorithm III.3.8 and we choose the parameters δ^+ and w^R analogously to d^+ and w^F for the frame construction, see Section IV.1.

For boundary curves with no defined frame, the ribbon construction is performed without orthogonality conditions resulting in a ribbon in Bézier representation of degree $\delta = 3$ as it is the minimal degree for ribbons to comply with the boundary data and compatibility.

Patch Construction: The majority of the patches are constructed by bicubically blended Coons patches, Algorithm III.4.7. However, we also have patches for which, due to their simplicity, a bilinearly blended Coons patch construction, as obtained by Algorithm III.4.1, suffices. The patches of the midship are principal examples for bilinear patches in which the G^1 continuity is fulfilled automatically. The Transom Back Patch $\mathbf{S}_{21}(u, v)$ (pg. 121) and the Shaft Back Patch $\mathbf{S}_{22}(u, v)$ (pg. 122) are the other planar patches which differ from the rest in the sense that they are bounded by sharp edges and, therefore, not G^1 continuous to its neighbours by construction.

IV.2.1 Patches With Singularities

Algorithms III.4.1 and III.4.7 are both designed for quadrilateral topologies. However, the wireframe construction of Section IV.1 also yields curve loops of non-quadrilateral nature. In particular, the following patches are bounded by *triangular* loops:

- Back Flat Bottom: $\mathbf{S}^4(u, v)$ on page 110,
- Back Flat Side: $\mathbf{S}^5(u, v)$ on page 111,
- Front Flat Bottom: $\mathbf{S}^6(u, v)$ on page 111,
- Front Flat Side: $\mathbf{S}^7(u, v)$ on page 111,
- Transom Back: $\mathbf{S}^{21}(u, v)$ on page 121,
- Bulb Bottom: $\mathbf{S}^{13}(u, v)$ on page 116.

We opt to represent these triangular patches as degenerate quadrilateral patches, thus retaining the general format of tensor product patches and enabling consistent methods in the analysis of hull. We can do this, by e.g. one of the following two options:

1. We split one of the three boundaries at an interior point and, therefore, formally obtain four curves. With this construction, we produce a singularity at this artificial vertex as the tangents of the adjacent boundary curves are collinear.
2. The alternative is to interpret one of the three vertices as a degenerate boundary curve. In this case, a singularity is introduced as one entire boundary curve is shrunk to a point and thus all tangents vanish.

Only the last one of the above six patches is of non-planar nature. We compare its construction according to these two techniques below, see Figure IV.11 on page 117.

The Shaft Back $\mathbf{S}^{22}(u, v)$ is bounded by only *two* curves. For its construction we can employ both of the above mentioned strategies and produce two

singularities. However, this patch is planar and the bilinearly blended Coons patch construction is applied.

For the discussion of singularly parametrized geometry in the context of isogeometric analysis, we refer to the literature, see e.g. Takacs and Jüttler [81, 82].

IV.2.2 Midship Patches

All of the following patches of the midship are obtained by the bilinearly blended Coons patch construction. Compare their construction to Figure IV.9 on page 112.

1. The **Parallel Midship Bottom** $\mathbf{S}^1(u, v)$ is the rectangular patch in the Bottom Plane. The boundary curves and neighbours are the following:
 - The Parallel Midship Back Base $\mathbf{c}^1(u)$ connects to the Back Flat Bottom Patch $\mathbf{S}^4(u, v)$.
 - The Front Base $\mathbf{c}^3(u)$ is the common boundary curve with the Front Flat Bottom Patch $\mathbf{S}^6(u, v)$,
 - The Centre Bar $\mathbf{c}^5(v)$ lies in the Centre Plane, thus there is no neighbour patch.
 - The Bottom Bar $\mathbf{c}^6(v)$ connects to the Bilge $\mathbf{S}^3(u, v)$.
2. Like the previous patch, the **Parallel Midship Side** $\mathbf{S}^2(u, v)$ is a rectangular one and lies in the Side Plane. It is bounded by the following curves and neighbour patches:
 - The Parallel Midship Back Stem $\mathbf{c}^2(u)$ with $u \in [0, U_2]$ connects to the Back Flat Side Patch $\mathbf{S}^5(u, v)$ where U_2 is obtained by the intersection of $\mathbf{c}^2(u)$ with the Waterline.
 - The Front Stem $\mathbf{c}^4(u)$ connects to the Front Flat Side Patch $\mathbf{S}^7(u, v)$ on $u \in [0, U_4]$ with U_4 like above.
 - At the Side Bar $\mathbf{c}^7(v)$ we have the Bilge $\mathbf{S}^3(u, v)$ as the neighbour.
 - The Waterline $\mathbf{c}^{29}(v)$ is the upper bound of the ship hull's wetted surface. Therefore, no neighbour patch is located there. For this patch, we restrict this curve to $v \in [V_{29,1}, V_{29,2}]$ with bounds obtained by the intersection with the first two boundary curves.
3. The **Bilge** $\mathbf{S}^3(u, v)$ is the cylinder-like part of the parallel midship that results from a bilinear blend of the curves Parallel Midship Back and Front Corner, $\mathbf{c}^{27}(u)$ and $\mathbf{c}^{28}(u)$, and the Side and Bottom Bar, $\mathbf{c}^6(v)$ and $\mathbf{c}^7(v)$. The respective neighbour patches are Stern and Bow Midship Joint Patches, $\mathbf{S}^{14}(u, v)$ and $\mathbf{S}^8(u, v)$, and the Parallel Midship Bottom and Side, $\mathbf{S}^1(u, v)$ and $\mathbf{S}^2(u, v)$.

The geometry can also be constructed equivalently by extruding $\mathbf{c}^{27}(u)$ in longitudinal direction from X_5 to X_6 since $\mathbf{S}^v(u, v) = \mathbf{S}^{uv}(u, v)$.
4. The **Back Flat Bottom** $\mathbf{S}^4(u, v)$, is the triangular patch between three boundary curves:

- The Parallel Midship Back Base $\mathbf{c}^1(u)$ connecting to the already presented Midship Bottom Patch $\mathbf{S}^1(u, v)$.
- The Flat Bottom Base $\mathbf{c}^9(u)$ at which there is no neighbour.
- The Flat Bottom Diagonal $\mathbf{c}^{10}(v)$ is the last boundary of this patch. Its neighbours depend on the position of the start point of Midship Stern Section $\mathbf{c}^{33}(u)$. If that lies on the Diagonal, at the parameter $v = V_{10}$, for $v \in [0, V_{10}]$ the corresponding neighbour is the Stern Midship Bottom $\mathbf{S}^{16}(u, v)$ and for $v \in [V_{10}, 1]$ the Stern Midship Joint $\mathbf{S}^{14}(u, v)$. In the opposite case, $\mathbf{S}^{14}(u, v)$ is the neighbour on the totality of this curve.

A bilinear blend suffices since all these curves lie in the Bottom Plane. Note, that the frame $\mathbf{f}^{10}(u)$ of the Diagonal is constructed to be constant and orthogonal to this plane, thus consistent with $\mathbf{S}^4(u, v)$. It is relevant only for the construction of the neighbour patches.

We employ the bilinearly blended Coons patch also for this patch with the three mentioned curves and the degenerate curve $\hat{\mathbf{c}}(v) = \mathbf{c}^1(0) = \mathbf{c}^9(1)$, i.e. the opposite point of the Diagonal $\mathbf{c}^{10}(v)$.

5. The **Back Flat Side** $\mathbf{S}^5(u, v)$ is another triangular patch between the following curves:

- The Parallel Midship Back Stem $\mathbf{c}^2(u)$ with $u \in [0, U_2]$ connects to the Parallel Midship Side Patch $\mathbf{S}^5(u, v)$, see above.
- The Midship Waterline $\mathbf{c}^{29}(u)$ with $u \in [0, V_{29,1}]$, where $V_{29,1}$ is its intersection with the Back Stem, bounds the wetted part of the ship hull. Therefore, we have no neighbour here.
- The Flat Side Diagonal $\mathbf{c}^{12}(v)$ with $v \in [0, V_{12}]$ bounds this patch to the Stern Midship Joint $\mathbf{S}^{14}(u, v)$.

The construction is analogous to that of the pendant Bottom Patch $\mathbf{S}^4(u, v)$ through the insertion of a degenerate curve $\hat{\mathbf{c}}(v) = \mathbf{c}^2(U_2) = \mathbf{c}^{29}(V_{29,1})$.

6. The **Front Flat Bottom** $\mathbf{S}^6(u, v)$ is the pendant of $\mathbf{S}^4(u, v)$ in the Front. Its bounding curves are the following:

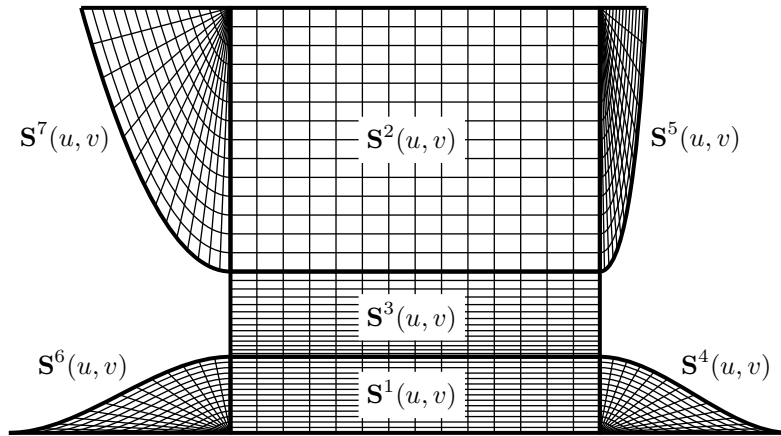
- The Parallel Midship Front Base $\mathbf{c}^3(u)$ is the common boundary with the Parallel Midship Bottom Patch $\mathbf{S}^1(u, v)$.
- At the Flat Bottom Base $\mathbf{c}^{13}(u)$ there is no neighbour patch.
- The Flat Bottom Diagonal $\mathbf{c}^{14}(v)$ is the boundary with respect to two patches if the start point of the Midship Bow Curve $\mathbf{c}^{30}(u)$ lies on it. In that case, the intersection parameter is V_{14} and for $v \in [0, V_{14}]$, the neighbour is the Bow Midship Joint $\mathbf{S}^8(u, v)$ while for $v \in [V_{14}, 1]$ it is the Bow Midship Bottom $\mathbf{S}^{10}(u, v)$. If $\mathbf{c}^{30}(u)$ does not start on $\mathbf{c}^{14}(v)$, for $v \in [0, 1]$ the neighbour patch is $\mathbf{S}^8(u, v)$.

This patch is constructed like the previous triangular patches by inserting the degenerate curve $\hat{\mathbf{c}}(v) = \mathbf{c}^3(0) = \mathbf{c}^{13}(0)$.

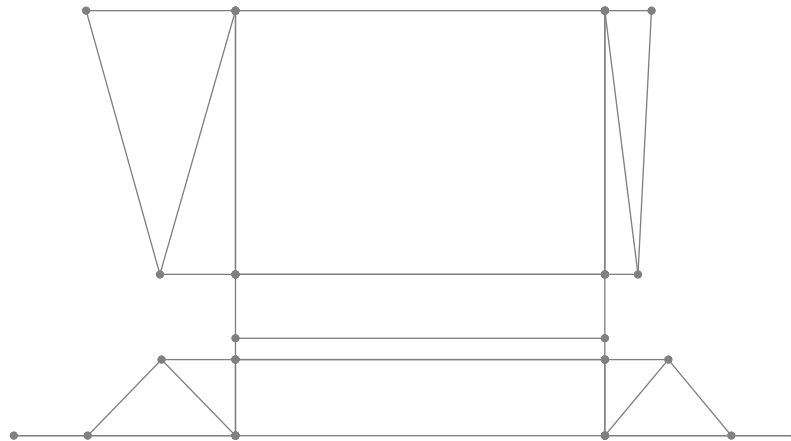
7. The **Front Flat Side** $\mathbf{S}^7(u, v)$ is the triangular patch bounded by the following curves:

- The Parallel Midship Front Stem $\mathbf{c}^4(u)$ for $u \in [0, U_4]$ with the intersection of this curve with the waterline at U^4 . The neighbour is the Parallel Midship Side Patch $\mathbf{S}^2(u, v)$.
- At the Midship Waterline $\mathbf{c}^{29}(u)$ for $u \in [V_{29,2}, 1]$ with intersection parameter $V_{29,2}$ with the Front Stem we have no neighbour.
- The Flat Side Diagonal $\mathbf{c}^{16}(v)$ with $v \in [0, V_{16}]$ where V_{16} is the intersection parameter with the Waterline bounds the patch to the Bow Midship Joint Patch $\mathbf{S}^8(u, v)$.

Like in the previous triangular patches, we insert a degenerate curve $\hat{\mathbf{c}}(v) = \mathbf{c}^{16}(V_{16}) = \mathbf{c}^{29}(1)$ in order to construct a quadrilateral loop.



(a) Isoparametric curves.



(b) Control points.

Figure IV.9: The patches of the midship.

IV.2.3 Bow Patches

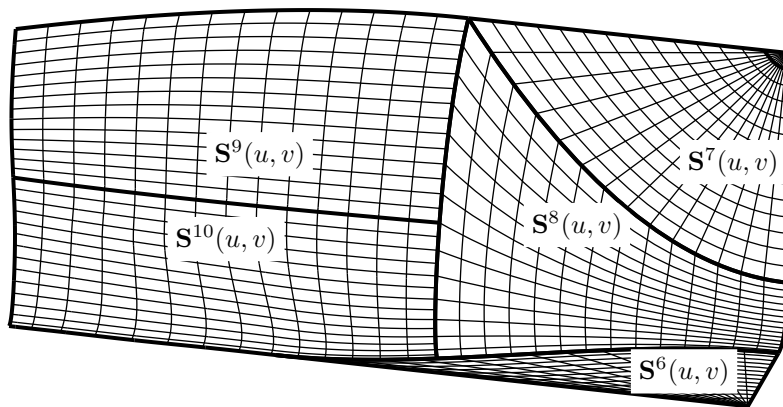
The following patches of the bow are shown in Figure IV.10 on page 115 and in Figure IV.12 on page 118, respectively.

8. The **Bow Midship Joint** $\mathbf{S}^8(u, v)$ connects the Midship with the Bow:
 - The Parallel Midship Front Corner $\mathbf{c}^{28}(u)$ is the connection to the Bilge $\mathbf{S}^3(u, v)$.
 - At the Front Flat Side Diagonal $\mathbf{c}^{16}(v)$ for $v \in [0, V_{16}]$ we have the neighbour of the Front Flat Side $\mathbf{S}^7(u, v)$.
 - The Midship Bow Section $\mathbf{c}^{30}(u)$ is the connection to two patches: With U_{30} being the parameter of the start point of the Bow Horizontal Curve $\mathbf{c}^{36}(v)$, for $[0, U_{30}]$, the neighbour is the Bow Midship Bottom Patch $\mathbf{S}^{10}(u, v)$ and for $[U_{30}, 1]$, it is the Bow Midship Top Patch $\mathbf{S}^{10}(u, v)$.
 - If the start point of the Midship Bow Curve $\mathbf{c}^{30}(u)$ lies on the Front Flat Bottom Diagonal $\mathbf{c}^{14}(v)$ we have determined the parameter V_{14} and obtain the boundary by trimming the Diagonal to $[0, V_{14}]$.
Otherwise, this start point lies on the Bow Profile Curve $\mathbf{c}^{18}(v)$ at parameter V_{18} . In that case the $\mathbf{c}^{14}(v)$ is, in its totality, the first part of this boundary while the second part is $\mathbf{c}^{18}(v)$ for $v \in [0, V_{18,1}]$. Through a linear reparametrization of the Profile we can join these curve C^1 continuously and we remark that the frames of both curves are constant in the neighbourhood of the joint.
The patch's neighbour at the Diagonal is the Front Flat Bottom Patch $\mathbf{S}^6(u, v)$ while at the Profile there is no neighbour.
9. The **Bow Midship Top** $\mathbf{S}^9(u, v)$ is the upper one of the two transitional patches between the Midship and the Bow. It is bounded by the following curves:
 - The Midship Bow Section $\mathbf{c}^{30}(u)$ bounds this and the above Bow Midship Joint Patch $\mathbf{S}^8(u, v)$ for $u \in [U_{30}, 1]$.
 - At the Bow Waterline $\mathbf{c}^{35}(v)$ for $v \in [0, V_{35}]$ with V_{35} being this curve's parameter of intersection with the Bow Front Section $\mathbf{c}^{31}(u)$, this patch has no neighbour.
 - The Bow Front Section $\mathbf{c}^{31}(u)$ bounds the patch with respect to two neighbours. The end point of the Bow Horizontal Curve $\mathbf{c}^{36}(v)$, the start point of the Bulb Root Curve $\mathbf{c}^{38}(v)$ and the Waterline $\mathbf{c}^{35}(v)$ determine three parameters on this curve: $U_{31,2}$, $U_{31,3}$ and $U_{31,4}$. On $[U_{31,2}, U_{31,3}]$ this patch's neighbour is the Bulb Top $\mathbf{S}^{12}(u, v)$ and on $[U_{31,3}, U_{31,4}]$ it is the Forward Overhang Patch $\mathbf{S}^{11}(u, v)$.
 - The Bow Horizontal Curve $\mathbf{c}^{36}(v)$ is, in its totality, the common boundary with the Bow Midship Bottom Patch $\mathbf{S}^{10}(u, v)$.
10. The **Bow Midship Bottom** $\mathbf{S}^{10}(u, v)$ is the lower one the two transitional patches between the Midship and the Bow. It is bounded by the following framed curves:

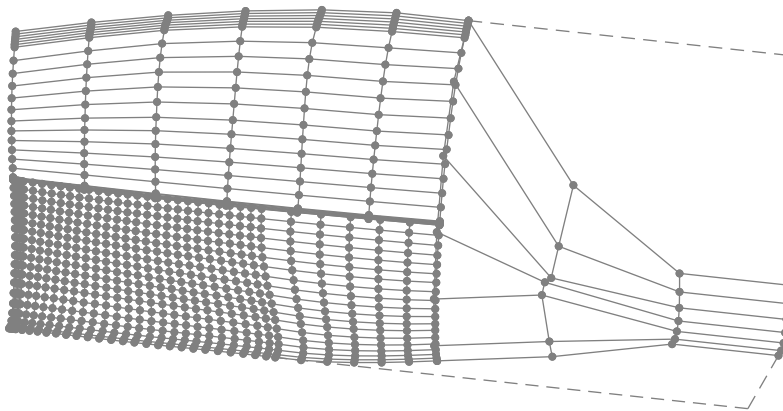
- Like for the Top Patch, the Midship Bow Curve $\mathbf{c}^{30}(u)$ bounds this and the above Bow Midship Joint Patch $\mathbf{S}^8(u, v)$ for $u \in [0, U_{30}]$.
- Also analogously to above, the Bow Horizontal Curve $\mathbf{c}^{36}(v)$ is, in its totality, the common boundary with the Bow Midship Top Patch $\mathbf{S}^9(u, v)$.
- The start point of the Bulb Curve $\mathbf{c}^{37}(u)$ determines the parameter $U_{31,1}$ on the Bow Front Section $\mathbf{c}^{31}(u)$ which is the boundary of this patch with respect to the Bulb Bottom Patch $\mathbf{S}^{13}(u, v)$ on $[0, U_{31,1}]$ and with respect to the Bulb Top Patch $\mathbf{S}^{12}(u, v)$ on $[U_{31,1}, U_{31,2}]$.
- The location of the start point of the Midship Bow Curve $\mathbf{c}^{30}(u)$ determines this boundary curve. If this point lies on the Front Flat Bottom Diagonal $\mathbf{c}^{14}(v)$ at $v = V_{14}$, $\mathbf{c}^{14}(v)$ for $v \in [V_{14}, 1]$ forms the first part of this boundary curve with the neighbour Front Flat Bottom Patch $\mathbf{S}^6(u, v)$. The second part is constituted by the Bow Profile Curve $\mathbf{c}^{18}(v)$ for $v \in [0, V_{18,2}]$ where there is no neighbour and where $V_{18,2}$ is the parameter where $\mathbf{c}^{31}(u)$ intersects $\mathbf{c}^{18}(v)$.

In the case that the start point of the Midship Bow Section lies on $\mathbf{c}^{18}(v)$, at $v = V_{18,1}$, we obtain the boundary curve by limiting it to $[V_{18,1}, V_{18,2}]$; compare to the Bow Midship Joint Patch $\mathbf{S}^8(u, v)$.

11. The **Forward Overhang Patch** $\mathbf{S}^{11}(u, v)$ is the top one of the three patches beyond the Bow Front Section $\mathbf{c}^{31}(u)$. It is constituted by the following curves:
 - The Bow Front Section $\mathbf{c}^{31}(u)$ for $[U_{31,3}, U_{31,4}]$ it has as a neighbour the Midship Bow Top Patch $\mathbf{S}^9(u, v)$.
 - The Bow Waterline $\mathbf{c}^{35}(v)$ on $[V_{35}, 1]$ represents the upper boundary where no neighbour is present.
 - For the Bow Profile Curve $\mathbf{c}^{18}(u)$ we determine the parameters $V_{18,4}$ and $V_{18,5}$ which correspond to the end points of the Bulb Root Curve $\mathbf{c}^{38}(u)$ and the Bow Waterline $\mathbf{c}^{35}(u)$, respectively. The curve limited to $[V_{18,4}, V_{18,5}]$ represents this patch's front boundary where we have no neighbour.
 - The Bulb Root Curve $\mathbf{c}^{38}(u)$ is, in its totality, the last boundary and connects this patch to the Bulb Top Patch $\mathbf{S}^{12}(u, v)$.
12. The **Bulb Top** $\mathbf{S}^{12}(u, v)$ is placed below the Forward Overhang Patch $\mathbf{S}^{11}(u, v)$ and is bounded by the following curves:
 - The Bulb Root Curve $\mathbf{c}^{38}(u)$ bounds this patch with respect to Forward Overhang Patch $\mathbf{S}^{11}(u, v)$.
 - The Bow Profile Curve $\mathbf{c}^{18}(u)$, limited to the parameters $V_{18,3}$ and $V_{18,4}$, represents the front boundary with no neighbour. While the latter parameter was introduced in the Forward Overhang Patch $\mathbf{S}^{11}(u, v)$, the former correspond to the end points of the Bulb Curve $\mathbf{c}^{37}(u)$.
 - The Bulb Curve $\mathbf{c}^{37}(u)$ in its totality bounds this patch to the Bulb Bottom Patch $\mathbf{S}^{13}(u, v)$.



(a) Isoparametric curves.



(b) Control points.

Figure IV.10: The patches of the joint between midship and bow.

- The Bow Front Section $\mathbf{c}^{31}(u)$ is the common boundary curve with the Bow Midship Bottom Patch $\mathbf{S}^{10}(u, v)$ and the Bow Midship Top Patch $\mathbf{S}^{10}(u, v)$, for $u \in [U_{31,1}, U_{31,2}]$ and $u \in [U_{31,2}, U_{31,3}]$, respectively.
13. The **Bulb Bottom** $\mathbf{S}^{13}(u, v)$ concludes the bow patch network. It is constituted by only three boundary curves as we already mentioned in the categorization of the patches in Section IV.2.1:
- The Bow Front Section $\mathbf{c}^{31}(u)$ for $u \in [0, U_{31,1}]$ connects this patch to the Bow Midship Bottom Patch $\mathbf{S}^{10}(u, v)$.
 - The Bulb Curve $\mathbf{c}^{37}(u)$ in its totality bounds this patch to the Bulb Top Patch $\mathbf{S}^{12}(u, v)$.
 - The Bow Profile $\mathbf{c}^{18}(t)$, for $t \in [V_{18,2}, V_{18,3}]$ is front boundary with no neighbour.

We consider two ways to formally produce a quadrilateral loop, compare to Subsection IV.2.1. The first is to interpret the vertex of the intersection of the Bow Front Section and the Bulb Curve as a degenerate curve. Alternatively, we propose to split the Bulb Profile Curve into two parts at a suitable interior point.

In either case, the constructed surface is singular at the manipulation point as it does not have a well-defined normal. In the former case the curve tangent is zero and in the latter case we have collinear tangents at the inserted points. In both cases, however, geometric smoothness is guaranteed by construction since all ribbons are generated in an unaltered manner and do not cease to comply with existing frames. Therefore, also the constructed surface geometry is perpendicular to the frames in its boundary curves, including the singular point.

In Figure IV.11, we find a comparison of the control polygons of the patch for both cases.

IV.2.4 Stern Patches

We find examples for their constructions of the following stern patches in Figure IV.13 on page 120 and in Figure IV.14 on page 123.

14. The **Stern Midship Joint** $\mathbf{S}^{14}(u, v)$ connects the Midship with the Stern and is bounded by the following curves:
- The Parallel Midship Back Corner $\mathbf{c}^{27}(u)$ is the connection to the Bilge $\mathbf{S}^3(u, v)$.
 - At the Back Flat Side Diagonal $\mathbf{c}^{12}(v)$ for $v \in [0, V_{12}]$ we have the neighbour of the Back Flat Side $\mathbf{S}^7(u, v)$.
 - The Stern Midship Section $\mathbf{c}^{30}(u)$ is the connection to two patches and we consider the parameter U_{33} which corresponds to the end point of the Stern Horizontal Curve $\mathbf{c}^{40}(v)$. For $[0, U_{33}]$, the neighbour is the Stern Midship Bottom Patch $\mathbf{S}^{16}(u, v)$ and for $[U_{33}, 1]$, it is the Stern Midship Top Patch $\mathbf{S}^{16}(u, v)$.

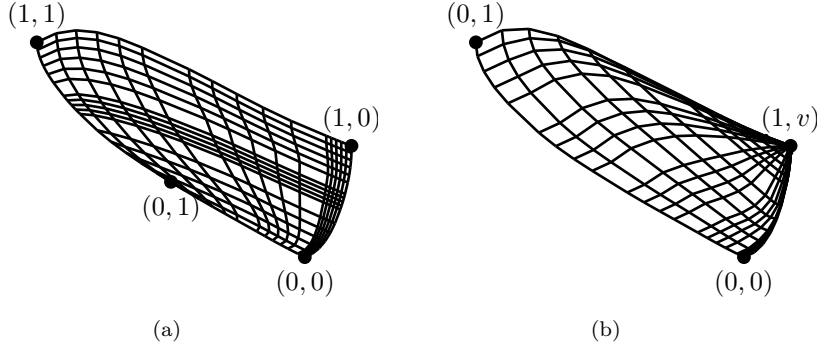


Figure IV.11: The net of control points of Bulb Bottom patch $\mathbf{S}^{13}(u, v)$. A quadrilateral boundary loop can be produced by splitting one curve as in (a) where we split the Bulb Profile $\mathbf{c}^{18}(u)$. The alternative is to regard a vertex as a degenerate curve as we do in (b) where the intersection of the Bow Front Section and Bow Curve is interpreted as the fourth boundary curve.

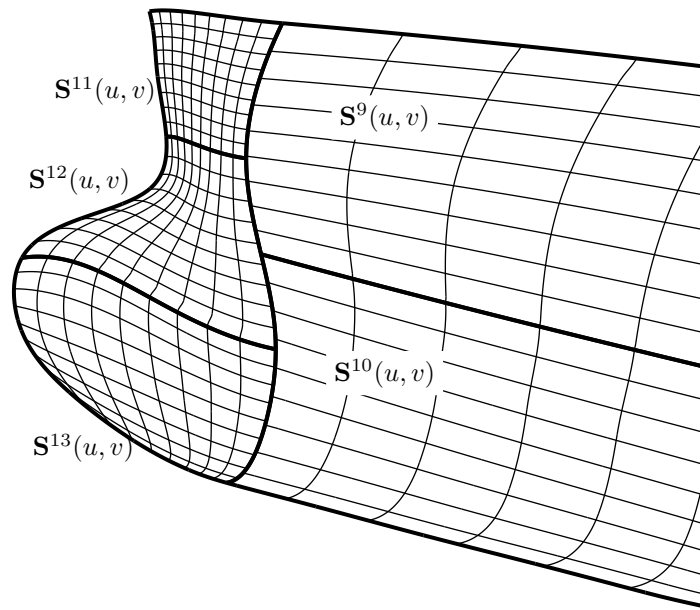
- If the start point of the Midship Stern Section $\mathbf{c}^{33}(u)$ lies on the Back Flat Bottom Diagonal $\mathbf{c}^{10}(v)$ we have determined the parameter V_{10} and obtain the boundary by trimming the Diagonal to $[V_{10}, 1]$.

Otherwise, this start point lies on the Shaft Profile Curve $\mathbf{c}^{20}(v)$ at parameter $V_{20,2}$. In that case the Diagonal is, in its totality, the second part of this boundary while the first part is the Profile Curve for $v \in [V_{20,2}, 1]$. Through a linear reparametrization of the Shaft Profile Curve we can join these curve C^1 continuously and we remark that the frames of both curves are constant in the neighbourhood of the joint.

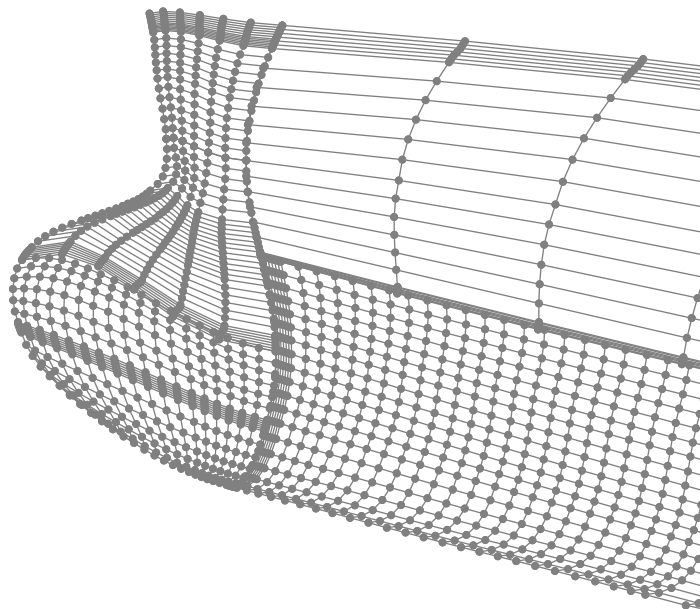
The patch's neighbour at the Diagonal is the Back Flat Bottom Patch $\mathbf{S}^4(u, v)$ while at the Profile there is no neighbour.

15. The **Stern Midship Top** $\mathbf{S}^{15}(u, v)$ is the upper one of the transitional patches between the Stern and the Midship. Its boundary loop is constituted by the following curves:

- The Midship Bow Section $\mathbf{c}^{30}(u)$ bounds this and the above Bow Midship Joint Patch $\mathbf{S}^8(u, v)$ for $u \in [U_{30}, 1]$.
- At the Stern Waterline $\mathbf{c}^{39}(v)$, for $v \in [V_{39}, 1]$ with V_{39} being this curve's parameter of intersection with the Stern Back Section $\mathbf{c}^{32}(u)$, this patch has no neighbour.
- The Stern Back Section $\mathbf{c}^{32}(u)$ bounds the patch with respect to two neighbours. The start point of the Stern Horizontal Curve $\mathbf{c}^{40}(v)$, the end point of the Shaft Root Curve $\mathbf{c}^{43}(v)$ and the Stern Waterline $\mathbf{c}^{39}(v)$ determine three parameters on this curve: $U_{32,2}$, $U_{32,3}$ and $U_{32,4}$. On $[U_{32,2}, U_{32,3}]$ this patch's neighbour is the Shaft Top $\mathbf{S}^{19}(u, v)$ and on $[U_{32,3}, U_{32,4}]$ it is the Stern Centre Patch $\mathbf{S}^{18}(u, v)$.



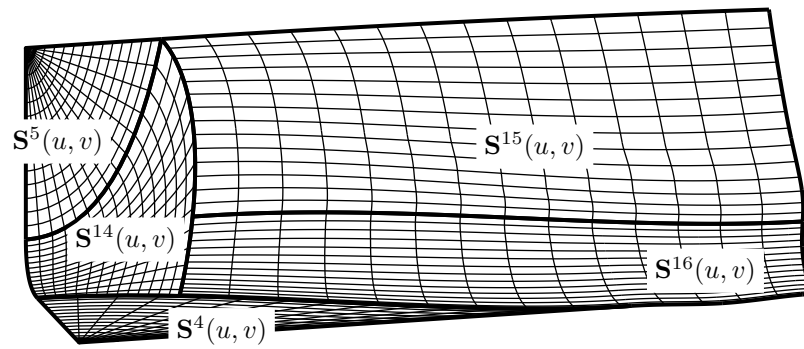
(a) Isoparametric curves.



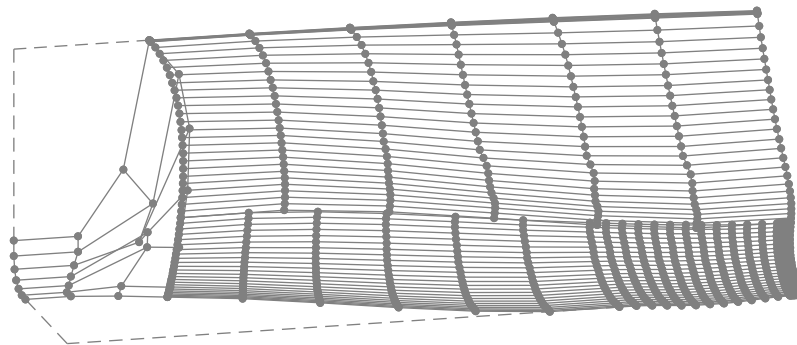
(b) Control points.

Figure IV.12: The patches of the bow.

- The Stern Horizontal Curve $\mathbf{c}^{40}(v)$ is, in its totality, the common boundary with the Bow Midship Bottom Patch $\mathbf{S}^{16}(u, v)$.
16. The **Stern Midship Bottom** $\mathbf{S}^{16}(u, v)$ is the lower one of the two transitional patches between the Midship and the Bow. It is bounded by the following framed curves:
- Like in the Top Patch, the Midship Stern Section $\mathbf{c}^{33}(u)$ bounds this and the above Stern Midship Joint Patch $\mathbf{S}^{14}(u, v)$ for $u \in [0, U_{33,1}]$.
 - Also analogously to above, the Stern Horizontal Curve $\mathbf{c}^{40}(v)$, in its totality, is the common boundary with the Stern Midship Top Patch $\mathbf{S}^{15}(u, v)$.
 - The end point of the Shaft Curve $\mathbf{c}^{42}(v)$ and start point of Stern Horizontal Curve $\mathbf{c}^{40}(v)$ determine the parameters $U_{32,1}$ and $U_{32,2}$ on the Stern Back Section $\mathbf{c}^{32}(u)$. We limit, thus, this curve to $[0, U_{32,2}]$. This bounds $\mathbf{S}^{16}(u, v)$ to the Shaft Bottom Patch $\mathbf{S}^{20}(u, v)$ on $[0, U_{32,1}]$ and the Shaft Top Patch $\mathbf{S}^{19}(u, v)$ on $[U_{32,1}, U_{32,2}]$.
 - The location of the start point of the Midship Stern Section $\mathbf{c}^{33}(u)$ determines this boundary. If this point lies on the Back Flat Bottom Diagonal $\mathbf{c}^{10}(v)$, at V_{10} , a composition of the Shaft Profile Curve $\mathbf{c}^{20}(v)$ and the Diagonal forms this boundary. The start parameter of the former curve is determined by the start point of the Stern Back Section $\mathbf{c}^{32}(u)$: $V_{20,1}$. While on the part of the Diagonal, which we limit to $[0, V_{10}]$, we have the Back Flat Bottom Patch as a neighbour $\mathbf{S}^4(u, v)$, the part of the Profile has no neighbour.
In the case that the start point of the Midship Stern Section lies on the Shaft Profile, at $v = V_{20,2}$, we obtain the boundary curve by limiting it to $[V_{20,1}, V_{20,2}]$; compare to the Stern Midship Joint Patch $\mathbf{S}^{14}(u, v)$.
17. The **Transom Patch** $\mathbf{S}^{17}(u, v)$ is the top one of the patches before the Stern Back Section. It is bounded by the following curves:
- The Stern Waterline $\mathbf{c}^{39}(u)$ for $u \in [0, V_{39}]$, where V_{39} is determined by the end point of the Transom Diagonal $\mathbf{c}^{41}(v)$, is the patch's top bound with no neighbour.
 - The Transom Diagonal $\mathbf{c}^{41}(v)$ bounds this patch with respect to the Stern Centre Patch $\mathbf{S}^{18}(u, v)$.
 - Transom Profile Curve $\mathbf{c}^{23}(u)$ bounds this patch to the back where no neighbour is present. For this patch we limit the curve to $[U_{23,2}, 1]$ where $U_{23,2}$ corresponds to the start point of the Transom Diagonal.
 - The Transom Side $\mathbf{c}^{24}(v)$ bounds this curve with respect to the Transom Back Patch for $v \in [0, V^{24}]$ where V^{24} corresponds to the start point of the Waterline.
18. The **Stern Centre** $\mathbf{S}^{18}(u, v)$ is the patch between the Transom Patch $\mathbf{S}^{17}(u, v)$ and the Stern Midship Top Patch $\mathbf{S}^{15}(u, v)$. It is bounded by the following curves:



(a) Isoparametric curves.



(b) Control points.

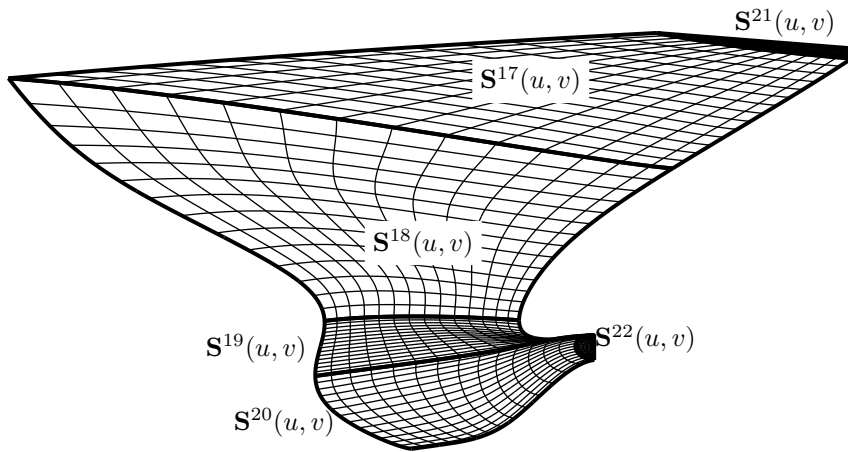
Figure IV.13: The Patches of the joint between the Mid-Ship and the Stern.

- The Transom Profile Curve $\mathbf{c}^{23}(u)$ bounds this patch to the back where no neighbour is present. For this patch we limit the curve to $[U_{23,1}, U_{23,2}]$ where $U_{23,1}$ corresponds to the start point of the Stern Root Curve and $U_{23,2}$ to that of the Transom Diagonal.
 - The Transom Diagonal $\mathbf{c}^{41}(v)$ bounds this patch to the Transom Patch $\mathbf{S}^{17}(u, v)$.
 - The Stern Back Section $\mathbf{c}^{32}(u)$, limited to $u \in [U_{32,2}, U_{32,3}]$ with $U_{32,2}$ corresponding to the end point of the Stern Root Curve and $U_{32,3}$ to intersection with the Waterline, limits this patch to the Stern Midship Top.
 - The Stern Root Curve $\mathbf{c}^{43}(v)$ bounds this patch with respect to the Shaft Top $\mathbf{S}^{19}(u, v)$.
19. The **Shaft Top** $\mathbf{S}^{19}(u, v)$ is located below the Stern Centre Patch $\mathbf{S}^{18}(u, v)$. Its bounds are the following:
- The Transom Profile $\mathbf{c}^{23}(u)$ for $u \in [0, U_{23,1}]$ with $U_{23,1}$ corresponding to the start point of the Stern Root Curve is the first boundary. We have no neighbour here.
 - The Stern Root Curve $\mathbf{c}^{43}(v)$ bounds this patch with respect to the Stern Centre $\mathbf{S}^{18}(u, v)$.
 - The end point of the Shaft Curve $\mathbf{c}^{42}(v)$ determines $U_{32,1}$, that start point of the Stern Horizontal Curve $\mathbf{c}^{40}(v)$ determines the parameter $U_{32,2}$ and the end point of the Stern Root Curve $\mathbf{c}^{43}(v)$ determines $U_{32,3}$ of the Stern Back Section $\mathbf{c}^{32}(u)$.
The Stern Back Section $\mathbf{c}^{32}(u)$ connects this patch to the Stern Midship Bottom $\mathbf{S}^{16}(u, v)$ for $[U_{32,1}, U_{32,2}]$ and to the Stern Midship Top $\mathbf{S}^{16}(u, v)$ for $[U_{32,2}, U_{32,3}]$.
 - The Shaft Curve $\mathbf{c}^{42}(v)$ connects this patch to the Shaft Bottom Patch $\mathbf{S}^{20}(u, v)$.
20. The **Shaft Patch** $\mathbf{S}^{20}(u, v)$ is the last non-planar patch of the hull. It is bounded by the following curves:
- The Shaft Side Curve $\mathbf{c}^{21}(u)$ bounds this patch to the Shaft Back Patch $\mathbf{S}^{22}(u, v)$.
 - The Shaft Curve $\mathbf{c}^{42}(v)$ connects this patch to the Shaft Top Patch $\mathbf{S}^{19}(u, v)$.
 - The Stern Back Section $\mathbf{c}^{32}(u)$ connects this patch to the Stern Midship Bottom $\mathbf{S}^{16}(u, v)$ for $u \in [0, U_{32,1}]$.
 - The Shaft Profile $\mathbf{c}^{20}(v)$ for $v \in [0, V_{20,1}]$ with $V_{20,1}$ corresponding to the start point of the Stern Back Section, bounds this patch to the bottom.
21. The **Transom Back** $\mathbf{S}^{21}(u, v)$ is the triangular planar patch on the back plane of the transom. For the parameter U_{25} corresponding to the height of the Draft, the Transom Stem $\mathbf{c}^{25}(u)$ is the first curve for $v \in [0, U_{25}]$ with no neighbour. Similarly, for the parameter V_{24} corresponding to

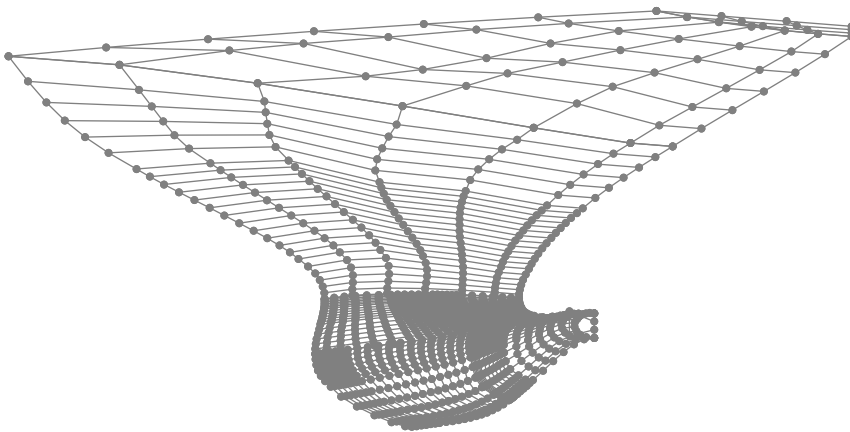
the height of the Draft, the Transom Side $\mathbf{c}^{24}(v)$ is the second curve for $v \in [0, V_{24}]$ with the Transom Patch \mathbf{S}^{17} as its neighbour. The Transom Back Waterline $\mathbf{c}^{34}(t)$ with no neighbour completes the loop. Due to its planar nature, a bilinear blend constructs the proper patch if we insert the vertex of the intersection of the Stem and the Waterline as a degenerate curve.

22. The **Shaft Back** $\mathbf{S}^{22}(u, v)$ concludes the patch network. It is a particular patch since it has only two boundary curves, the Shaft Stem $\mathbf{c}^{22}(u)$ with no neighbour and the Shaft Side Curve $\mathbf{c}^{21}(u)$ where it connects to the Shaft Patch $\mathbf{S}^{20}(u, v)$.

Like for triangular patches, we can either split both curves or insert two degenerate curves at the current vertices. Since the patch is planar, we leave this patch without further analysis and state that in this model we have opted for the latter approach.



(a) Isoparametric curves.



(b) Control points.

Figure IV.14: The patches of the stern.

Chapter V

Model Instances

The difficulty for a parametric model to guarantee valid instances is illustratively underlined in Hoffmann and Kim [47] 2000. In this chapter we present hull instances for parameter changes in all areas of the ship. The selected alterations of the default parameter set seek to demonstrate the flexibility of the developed model and we examine each of these ship hulls instances through a rendered plot and through isophote curves.

Isophote Curves: To exhibit the G^1 continuity we plot in black and white stripes the isophote lines on the surface, see [74, 39, 40, 4]. This method yields lines of equal light intensity and forms, therefore, a useful tool to detect surface irregularities and discontinuities. In particular, their continuity indicates G^1 continuity of the geometry. To produce all of these plots, we have employed the software package Rhinoceros 3DTM (see item 3 of the list on page 142).

View Points: The following is a list of the employed view points (v_1)–(v_6) for the model, each of which comprises the position \mathbf{P} of a hypothetical camera and the target point \mathbf{T} of its focus. Due to the large difference between the ship principal dimensions, in particular that between the LPP and the Depth, we have neglected a strict side view, i.e. in direction $(0, -1, 0)^T$. Instead, we have opted to employ the View Points (v_1) and (v_2) which, in combination, capture the totality of the ship's geometry. For a detailed examination of the bow we introduce the View Points (v_3) and (v_4) and for the stern area we employ the View Points (v_5), and (v_6):

- (v_1) The **Bow Midship View**, as can be found in Figure V.1(a), exhibits the entire ship from a point near the front point of the bulbous bow, $\mathbf{P} = (X_1 + 60, 80, -15)^T$, focusing backwards on the point $\mathbf{T} = (X_1 - 45, 5, 10)^T$, which roughly corresponds to the joint of the bow with the midship.
- (v_2) To capture the entity of the ship from the stern, we introduce the **Stern Midship View**, as can be found in Figure V.1(b). Here we place the camera in the point $\mathbf{P} = (-60, 80, 15)^T$ and focus on the point $\mathbf{T} = (45, 5, 10)^T$.
- (v_3) By placing the camera in the point $\mathbf{P} = (X_1 + 35, 32, -17)^T$ and by focusing on the point $\mathbf{T} = (X_1 - 20, 0, 13)^T$ we obtain the **Bow Bottom View**

which, therefore, exhibits the bow part of the ship from a point below the base plane of the ship. An example can be found in Figure V.2(a).

- (v₄) The **Bow Top View**, as can be found in Figure V.2(b), exhibits the bow part of the ship from a point above the deck plane of the ship, $\mathbf{P} = (X_1 + 35, 32, 27)^T$. The focus lies on the point $\mathbf{T} = (X_1 - 25, 0, 5)^T$, i.e. the bulb.
- (v₅) The **Stern Back View**, as can be found in Figure V.3(a), displays the stern area by placing the camera in the point $\mathbf{P} = (-35, 32, 0)^T$ and by focusing on the point $\mathbf{T} = (17, 4, 12)^T$.
- (v₆) The **Stern Side View** places the camera in the point $\mathbf{P} = (0, 60, -15)^T$ and focuses on the target point $\mathbf{T} = (17, 4, 10)^T$. It, therefore, exhibits the stern part of the ship from a point below the base plane of the ship. An example for this view can be found in Figure V.3(b).

Note that the Views (v₁), (v₃) and (v₄), i.e. those views which correspond to the bow part of the ship, are relative to the ship's LPP X_1 . This choice provides consistent views onto the bow regardless of the LPP.

V.1 The Default Parameter Set

The initial set of parameters was introduced in Section II.2 and summarised in Table II.1 on Page 30. We present the instance of this parameter set as our first result and exhibit all view points in Figures V.1, V.2 and V.3.

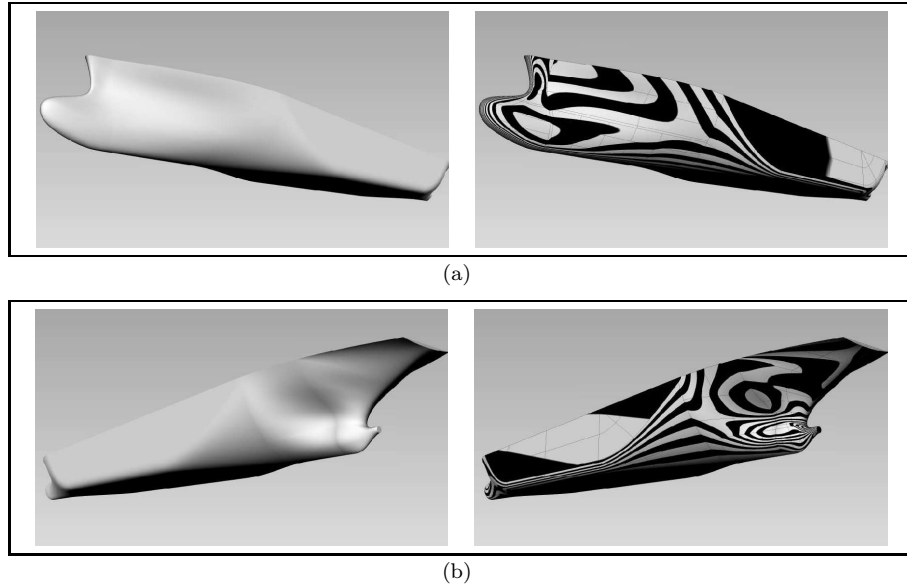
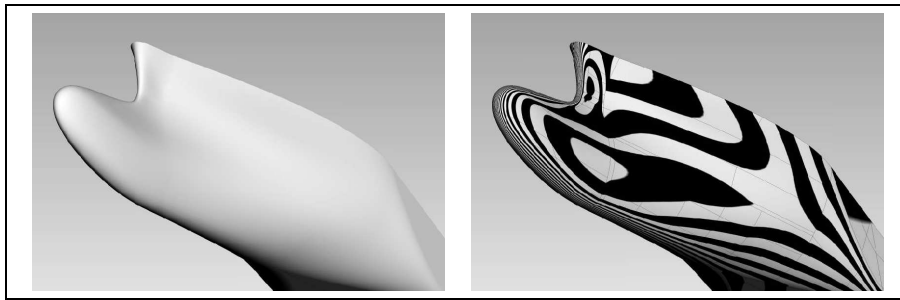
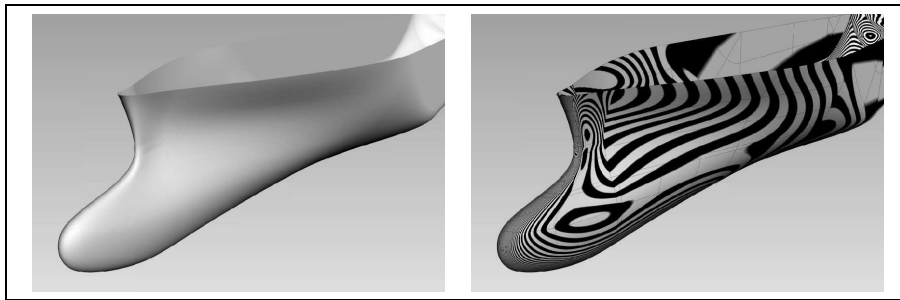


Figure V.1: The initial parameter set as found in Table II.1 displayed in the global Views (v₁) and (v₂).

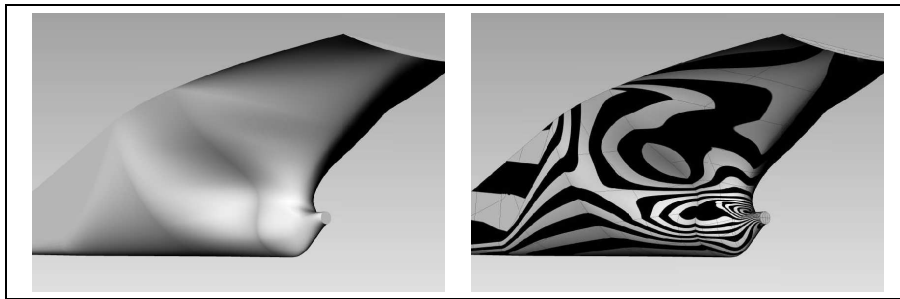


(a)

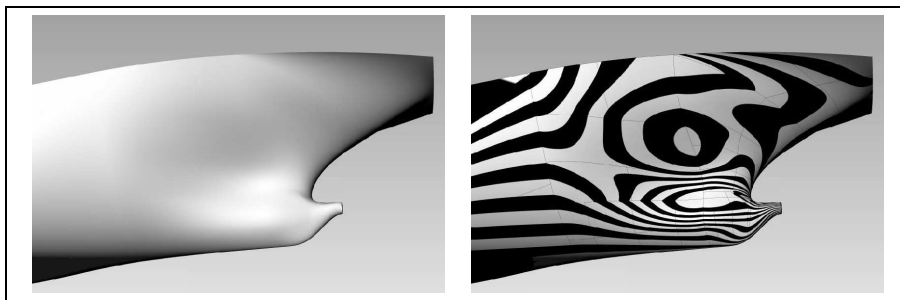


(b)

Figure V.2: The initial parameter set as found in Table II.1 is displayed in the Views (v_3) and (v_4) focusing on the bow.



(a)



(b)

Figure V.3: The initial parameter set as found in Table II.1 is displayed in the Views (v_5) and (v_6) focusing on the stern.

V.2 Global Parameter Changes

The instances presented in this section correspond to alterations of the global parameters LPP X_1 , Beam Y_2 and Depth Z_3 .

Parameter Set “Long”

In the first alteration of the initial parameter set we raise the LPP to $300m$, see Figure V.4:

Name	PM	default value	absolute change	relative change	current value
LPP	X_1	$230m$	$70m$	130.43%	$300m$

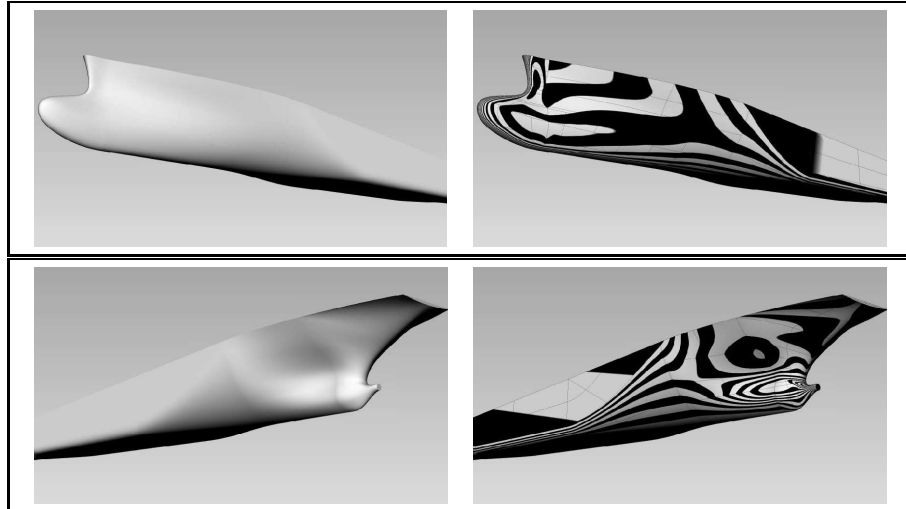
 (V.1)


Figure V.4: The parameter set “Long” as found in the table in Equation (V.1) is displayed in the global views (v_1) and (v_2).

Parameter Set “Short”

We continue with another alteration of the ship’s LPP and introduce the parameter set “Short” where we reduce the LPP to the lower value of $150m$:

Name	PM	default value	absolute change	relative change	current value
LPP	X_1	$230m$	$-80m$	65.22%	$150m$

 (V.2)

Find the corresponding plots in Figure V.5.

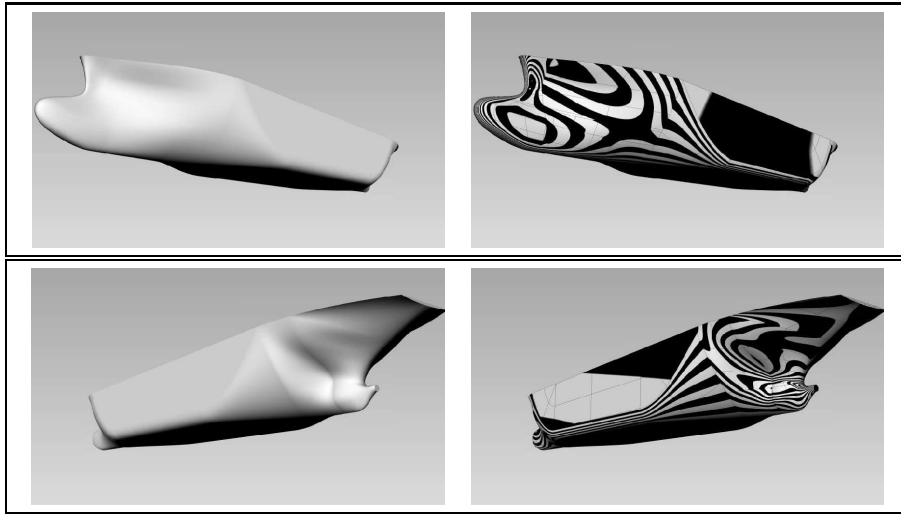


Figure V.5: The parameter set “Short” as found in the table in Equation (V.2) is displayed in the global views (v_1) and (v_2).

Parameter Set “Wide”

The next alteration regards the Beam which we raise to $50m$:

Name	PM	default value	absolute change	relative change	current value
Beam	Y_2	$32m$	$18m$	156.25%	$50m$

(V.3)

Figure V.6 exhibits the corresponding plots.

Parameter Set “Narrow”

As the counterpart to the above, we regard the small Beam of $15m$, compare to Figure V.6:

Name	PM	default value	absolute change	relative change	current value
Beam	Y_2	$32m$	$-17m$	46.875%	$15m$

(V.4)

Parameter Set “High”

We increase the depth to $30m$ to raise the overall height of the ship:

Name	PM	default value	absolute change	relative change	current value
Depth	Z_3	$23m$	$7m$	130.43%	$30m$

(V.5)

Find the corresponding plots in Figure V.8.

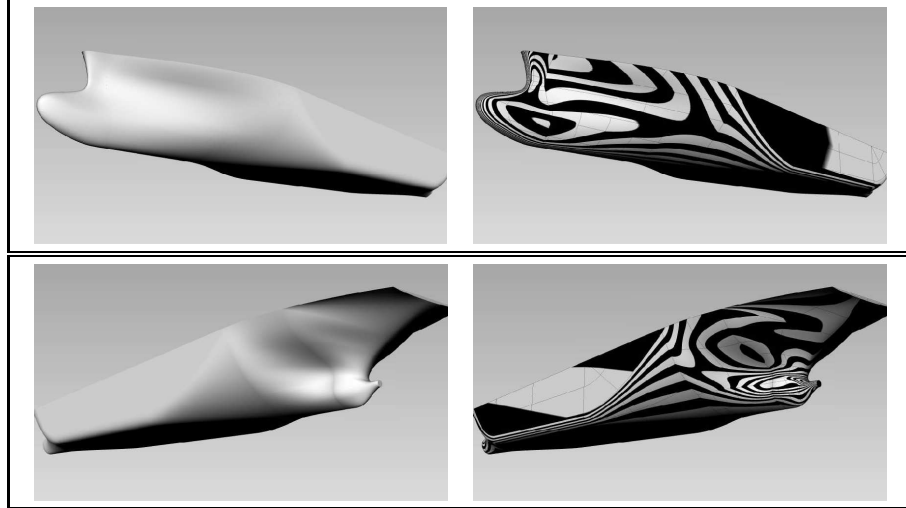


Figure V.6: The parameter set “Wide” as found in the table in Equation (V.3) is displayed in the global views (v_1) and (v_2).

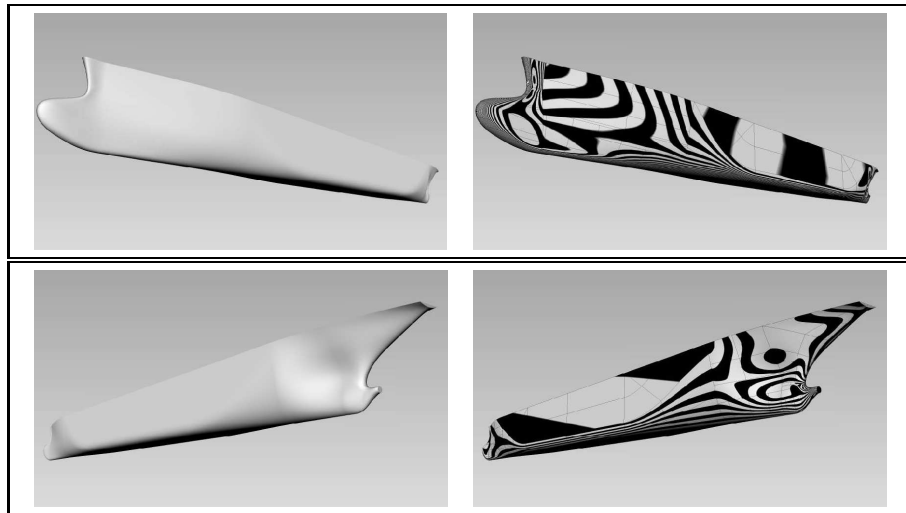


Figure V.7: The parameter set “Narrow” as found in the table in Equation (V.4) is displayed in the global views (v_1) and (v_2).

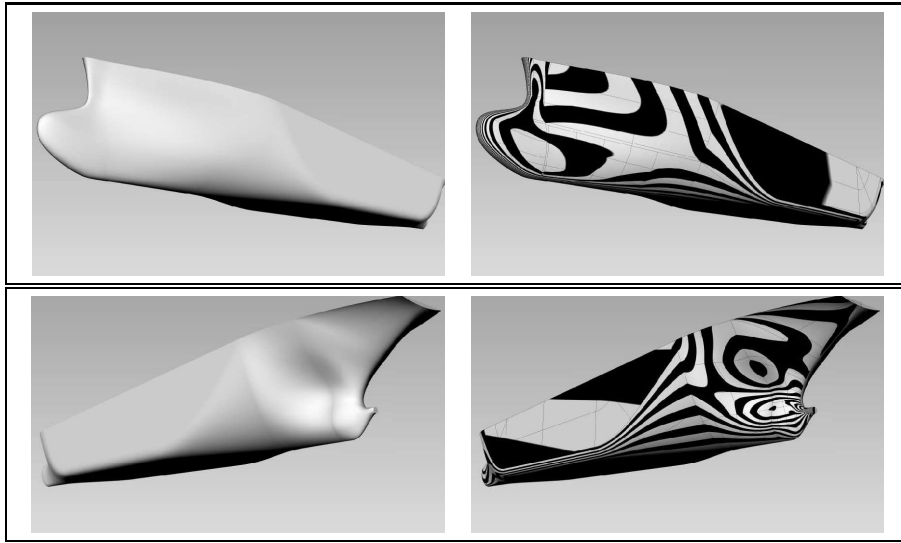


Figure V.8: The parameter set “High” as found in the table in Equation (V.5) is displayed in the global views (v_1) and (v_2) .

Parameter Set “Low”

By reducing the depth to $15m$, we decrease the overall height of the ship. Find the corresponding plots in Figure V.9:

Name	PM	default value	absolute change	relative change	current value
Depth	Z_3	$23m$	$-8m$	65.22	$15m$

(V.6)

V.3 Midship Parameter Changes

The parameter changes applied in this section are related to the parallel midship.

Parameter Set “Long Parallel Midship”

In order to create a rather large parallel midship, we change its start and the end appropriately:

Name	PM	default value	absolute change	relative change	current value
MS Start	R_5	0.3	-0.1	66.67%	0.2
	X_5	$69m$	$-23m$	66.67%	$46m$
MS End	R_6	0.5	0.3	160%	0.8
	X_6	$150m$	$66.2m$	144.13%	$216.2m$

(V.7)

where $X_6 = X_5 + R_6(X_1 - X_5)$. Note that the resulting length of the midship has increased by $89.2m$, a relative elevation of 210.12%. Find the corresponding plots in Figure V.10.

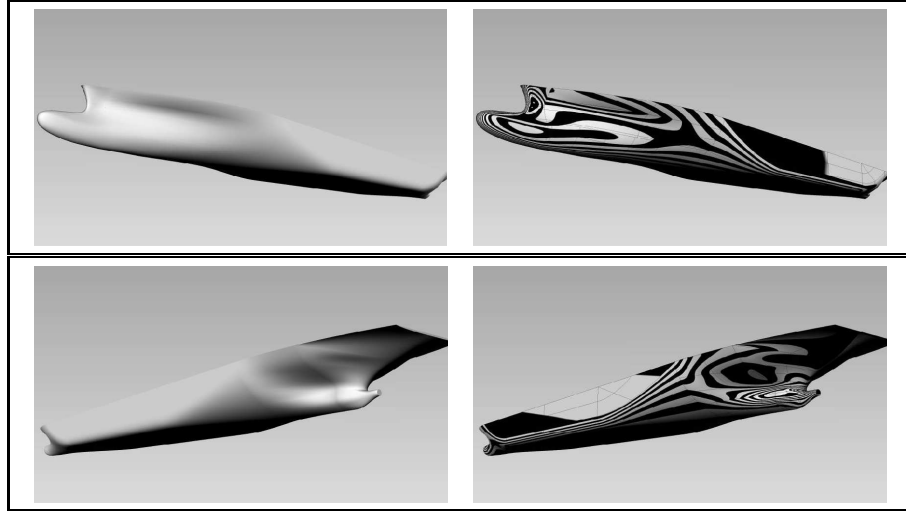


Figure V.9: The parameter set “Low” as found in the table in Equation (V.6) displayed in the global views (v_1) and (v_2).

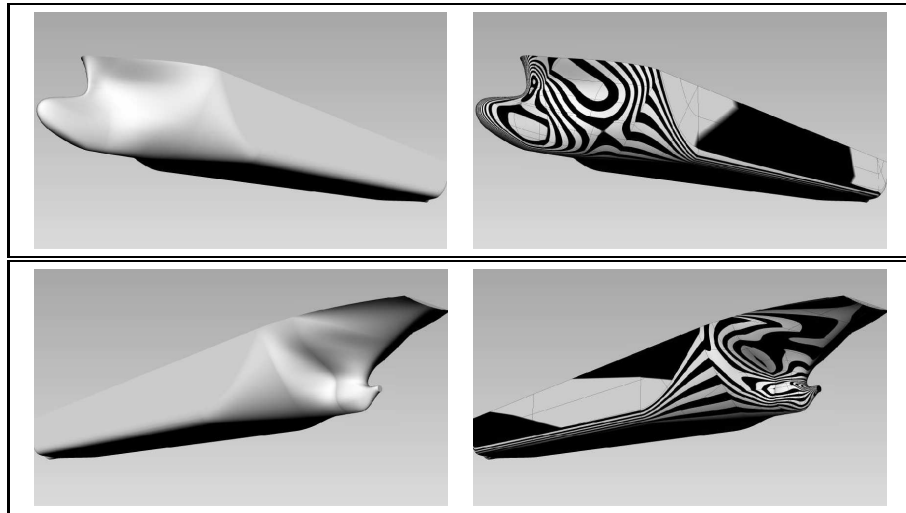


Figure V.10: The parameter set “Long Parallel Midship” as found in the table in Equation (V.7) is displayed in the global views (v_1) and (v_2).

Parameter Set “Short Parallel Midship”

Analogously to above, a short parallel midship results from the following changes, as can be compared to in Figure V.11:

Name	PM	default value	absolute change	relative change	current value	
MS Start	R_5	0.3	0.18	160%	0.48	(V.8)
	X_5	69m	41.4m	160%	110.4m	
MS End	R_6	0.5	-0.45	10%	0.05	
	X_6	150m	-33.62m	77.59%	116.4m	

where $X_6 = X_5 + R_6(X_1 - X_5)$. With these changes we reduce the length of the parallel midship to 6m, which corresponds to 7.4% of the original length.

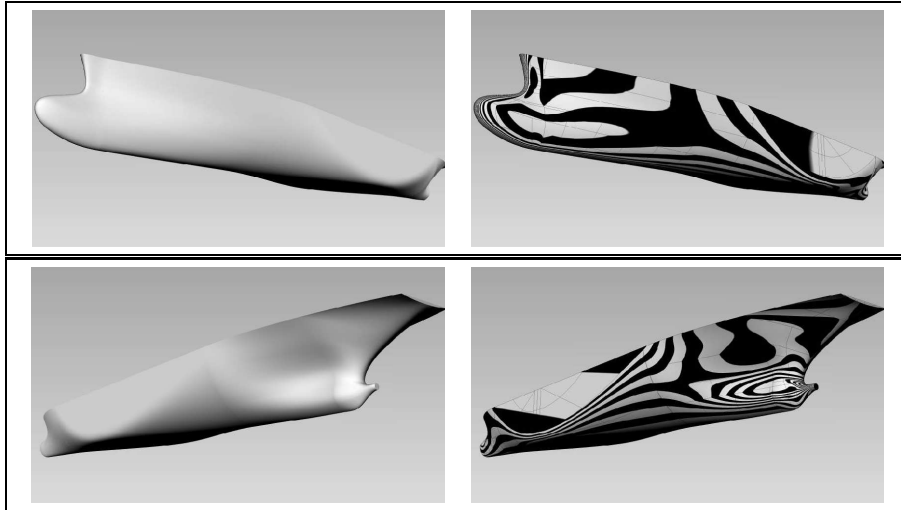


Figure V.11: The parameter set “Short Parallel Midship” as found in the table in Equation (V.8) is displayed in the global views (v_1) and (v_2).

V.4 Bow Parameter Changes

We focus on changes of parameters which are related exclusively to the bow area. Therefore, we subsequently alter the bulb’s length, its height and width.

Parameter Set “Long Bulb”

To enlarge the bulb we raise the value of the bulb length to 15m:

Name	PM	default value	absolute change	relative change	current value	
Bulb Length	X_{14}	8.8m	6.2m	170.45%	15m	(V.9)

Find the corresponding plots in Figure V.12.

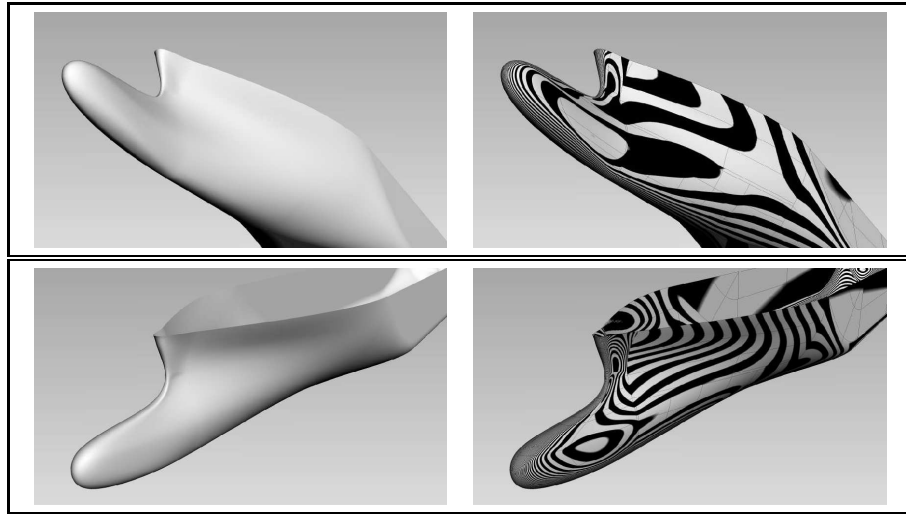


Figure V.12: The parameter set “Long Bulb” as found in the table in Equation (V.9) is displayed in the bow views (v_3) and (v_4).

Parameter Set “Short Bulb”

Figure V.13 shows the result of decreasing the bulb’s length as follows:

Name	PM	default value	absolute change	relative change	current value
Bulb Length	X_{14}	$8.8m$	-4.8	45.45%	4

(V.10)

Parameter Set “Broad Bulb”

To create a broad bulb we increase the value of Bulb Width and obtain:

Name	PM	default value	absolute change	relative change	current value
Bulb Width	R_{17}	0.15	0.25	266.67%	0.4
	Y_{17}	$4.8m$	$8m$	266.67%	$12.8m$

(V.11)

Find the corresponding plots in Figure V.14.

Parameter Set “High Bulb”

We elevate the front point of bulb and bulb rise by the following parameter alterations:

Name	PM	default value	absolute change	relative change	current value
Bulb Height	R_{15}	0.28	0.2	178%	0.5
	Z_{15}	$5m$	$4m$	178%	$9m$
Bulb Rise	Z_{20}	$0.1m$	$0.3m$	400%	$0.4m$

(V.12)

Find the corresponding plots in Figure V.15.

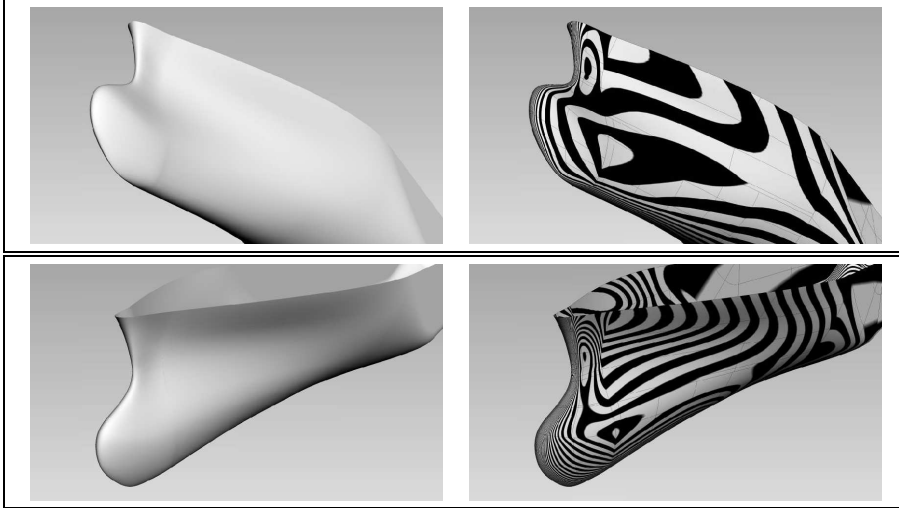


Figure V.13: The parameter set “Short Bulb” as found in the table in Equation (V.10) displayed in the bow views (v_3) and (v_4) .

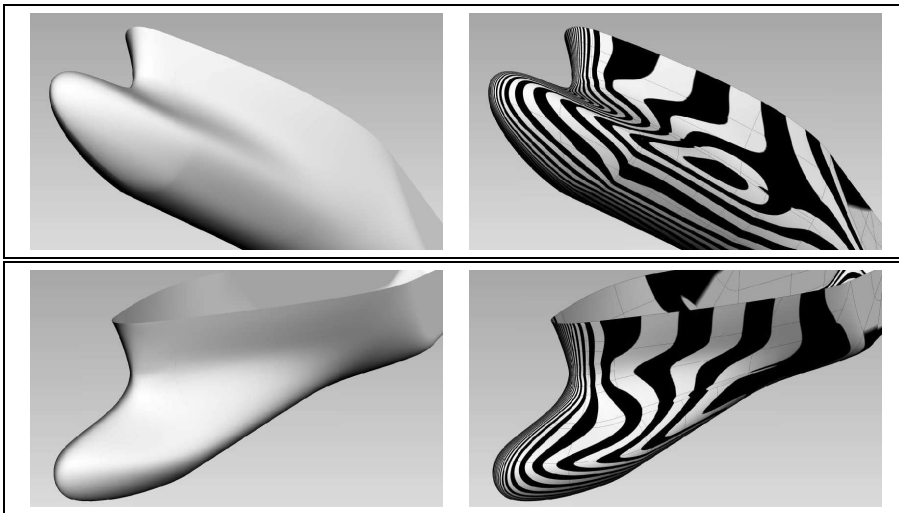


Figure V.14: The parameter set “Broad Bulb” as found in the table in Equation (V.11) displayed in the bow views (v_3) and (v_4) .

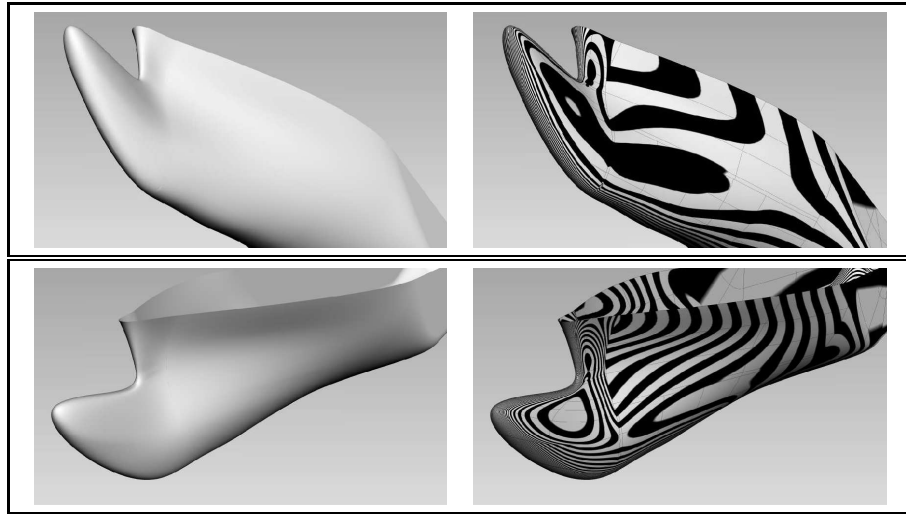


Figure V.15: The parameter set “High Bulb” as found in the table in Equation (V.12) is displayed in the bow views (v_3) and (v_4).

Parameter Set “Low Bulb”

Lowering the front point of bulb to the setting below parameter requires a change of the bulb rise in order to obtain a reasonable shape. The following alterations yield Figure V.16:

Name	PM	default value	absolute change	relative change	current value
Bulb Height	R_{15}	0.28	-0.23	17%	0.05
	Z_{15}	5m	-4.1m	17%	0.9m
Bulb Rise	Z_{20}	0.1m	-0.1m	0.0%	0.0m

(V.13)

V.5 Stern Parameter Changes

We conclude this chapter with instances resulting from parameter changes whose influence is restricted to the stern area. In this context we modify the length of the transom and the position of the shaft.

Parameter Set “Long Transom”

The plots of an instance with a long transom, as achieved by the following changes, is found in Figure V.17.

Name	PM	default value	absolute change	relative change	current value
Tr. Behind	X_{21}	6.50m	3.5m	153.85%	10.0m

(V.14)

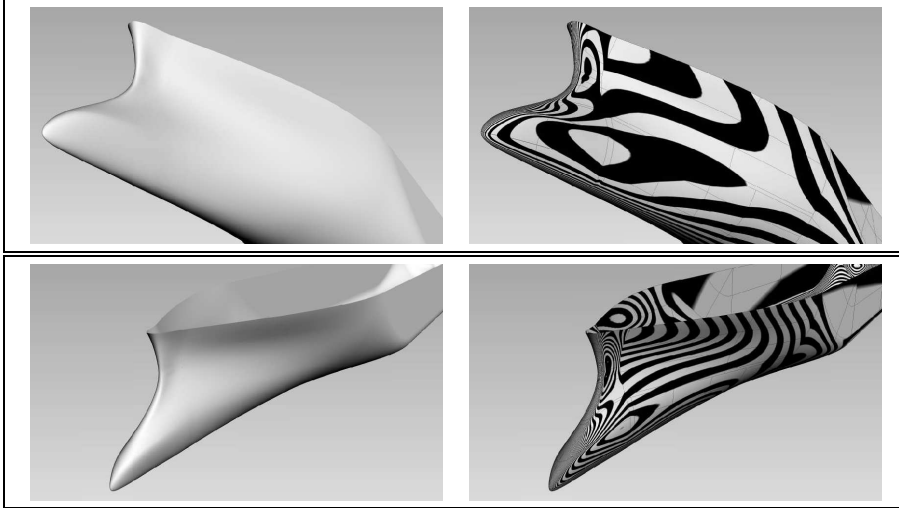


Figure V.16: The parameter set “Low Bulb” as found in the table in Equation (V.13) is displayed in the bow views (v_3) and (v_4).

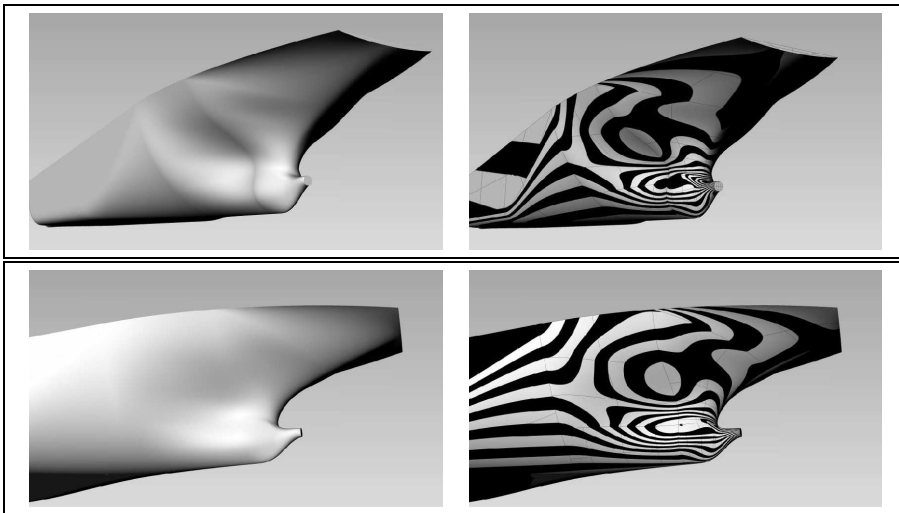


Figure V.17: The parameter set “Long Transom” as found in the table in Equation (V.14) is displayed in the stern views (v_5) and (v_6).

Parameter Set “Long Shaft”

The introduction of a long shaft involves more parameters if we wish to retain certain characteristics. In order to keep a convex shape of the Transom Profile $\mathbf{c}^{23}(u)$ and to keep the positions of the shaft and the propeller clearance, we increase the transom immersion and adjust the ratios of the shaft height and propeller clearance to yield absolute heights as obtained by the default parameter set. Increasing the root exit to the setting below can now be done without altering the convexity the profile curve. We also raise the root height for a less flat transition between the shaft and the root.

Name	PM	default value	absolute change	relative change	current value
Root Exit	\hat{X}_{27}	3.50m	3.5m	200%	7.0m
	X_{27}	9m	3.5m	133.33%	12m
Tr. Immersion	R_{23}	0.03	0.31	1111.11%	0.33
	Z_{23}	0.54m	5.46m	1111.11%	6m
Shaft Height	R_{25}	0.23	0.1	142.86%	0.33
	Z_{25}	4m	0m	0%	4m
Prop. Clear.	R_{26}	0.5	0.35m	168%	0.84
	Z_{26}	6.72	0m	0%	6.72m
Root Height	R_{28}	0.3	0.1m	133.33%	0.4
	Z_{28}	2.016m	0.672m	133.33%	2.688m

(V.15)

As can be found in Table II.1, with the Shaft Exit X_{24} we have

$$X_{27} = X_{24} + \hat{X}_{27}$$

and with the Draft Z_4 , we have

$$\begin{aligned} Z_{23} &= R_{23}Z_4, \\ Z_{25} &= R_{25}(Z_4 - Z_{23}), \\ Z_{26} &= R_{26}(Z_4 - Z_{23} - Z_{25}), \\ Z_{28} &= R_{28} * Z_{26}. \end{aligned}$$

Find the corresponding plots in Figure V.18.

Parameter Set “High Shaft”

To evaluate the position of the shaft, we introduce the following change and find the corresponding plots in Figure V.19:

Name	PM	default value	absolute change	relative change	current value
Shaft Height	R_{25}	0.23	0.17	217.39%	0.5
	Z_{25}	4.08m	4.67m	214.46%	8.75m

(V.16)

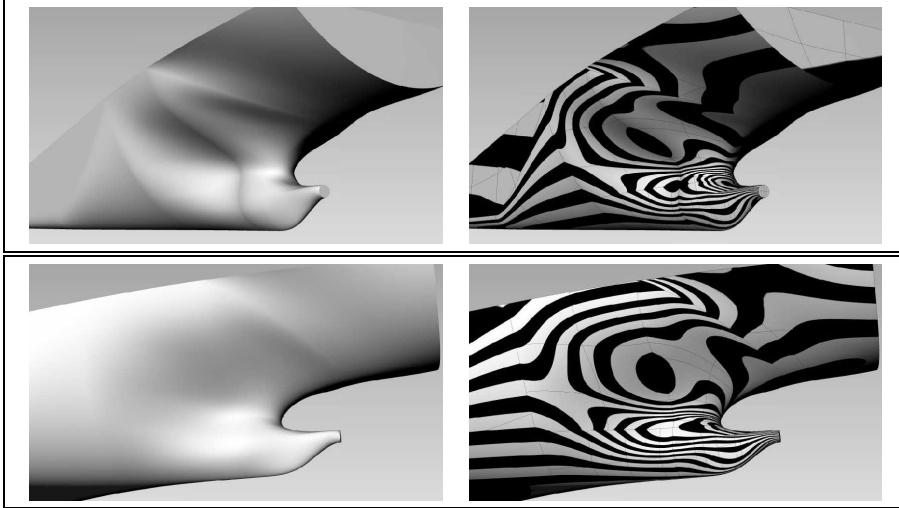


Figure V.18: The parameter set “Long Shaft” as found in the table in Equation (V.15) is displayed in the stern views (v_5) and (v_6).

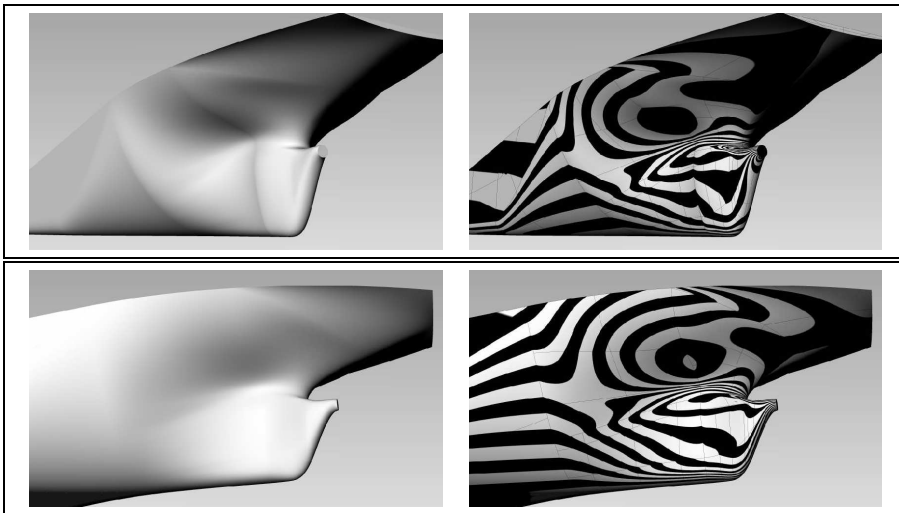


Figure V.19: The parameter set “Long Transom” as found in the table in Equation (V.16) is displayed in the stern views (v_5) and (v_6).

Parameter Set “Low Shaft”

To lower the position of the shaft, we apply the following alteration:

Name	PM	default value	absolute change	relative change	current value
Shaft Height	\tilde{R}_{25}	0.23	-0.23	43.48%	0.1
	Z_{25}	4.08m	-2.33m	42.89%	1.75m

(V.17)

Find the corresponding plots in Figure V.20.

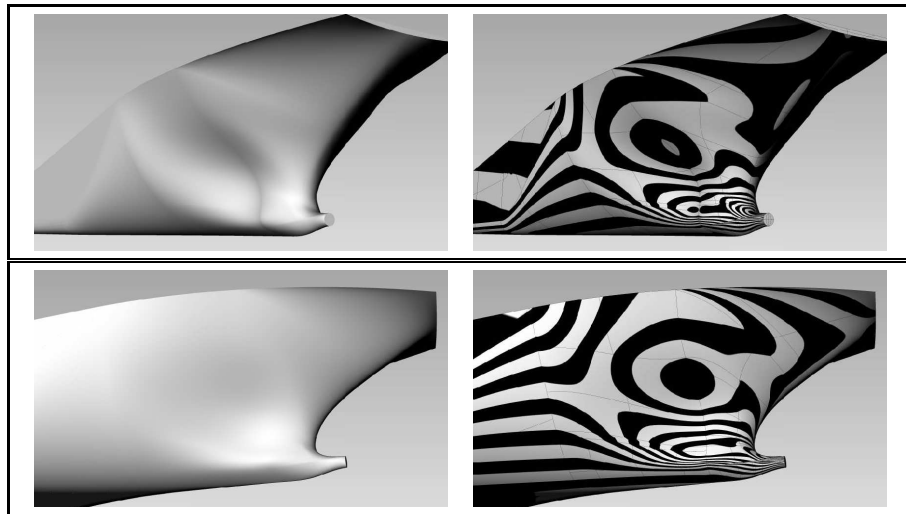


Figure V.20: The parameter set “Low Shaft” as found in the table in Equation (V.17) is displayed in the stern views (v_5) and (v_6).

Chapter VI

Implementation Notes

VI.1 Data Manipulations

Input points and tangents for the curve interpolation of Algorithm III.1.14 can be altered if all of the following conditions are fulfilled:

- the data are not direct derivations from design parameters in Section II.2,
- the data are governed by another curve, e.g. they are points on a curve or directions in the tangent plane of a curve,
- the data's location in its generating curve's parameter space is in the neighbourhood of a knot.

Then, we retain the option to replace the original data point with the point that corresponds to the closest knot. This often proves to be a convenient manipulation to avoid close knot values. Note, that the first two condition have to be fulfilled in order to modify any input design parameters while the third is to avoid large shifts.

VI.2 The Environment for the Task

To implement a parametric model of a container ship hull, as proposed in Section I.2, a proper environment has to be determined. The following are three possible categories of environments for the realisation of the task:

1. A general-purpose programming language like C++¹ or Java² provides the necessary transparency as the development takes place on source code level. This approach is only reasonable if specific toolboxes providing stable algorithms for basic problems in the fields of linear algebra and geometry are available. Possible libraries for geometry and, particularly, for splines include CGAL³, IRIT⁴, opennurbs⁵ and GoTools⁶, all of which

¹C++TM by Bjarne Stroustrup since 1979: www.stroustrup.com/C++.html.

²JavaTM by Sun Microsystems since 1991: www.java.com.

³CGALTM by the CGAL Open Source Project since 1986: www.cgal.org.

⁴IRITTM by Gershon Elber: www.cs.technion.ac.il/~irit.

⁵opennurbsTM by Robert McNeel Associates since January 2000: www.opennurbs.org.

⁶GoToolsTM by Sintef: www.sintef.no/Projectweb/Geometry-Toolkits/GoTools.

include implementations for the necessary basic linear algebra algorithms.

2. The use of a software package designed specifically for mathematical applications provides the necessary control over geometric entities. Possibilities for this choice include Matlab¹, Octave² and packages with focus on symbolic computation as are Maple³, Mathematica⁴ and Maxima⁵.
3. As many CAD software packages provide interfaces for scripting, their suitability for our purposes particularly depending on their transparency. The use of 3D geometric modelling software like Rhinoceros 3D⁶ or FreeCad⁷ in combination with a scripting language like Python⁸ is an example in this category.

Since implementation as in item 1 is done on a source level, it is associated by an elevated work-load. However, it is superior to the others in terms of genericity and efficiency. In contrast, environments in item 2 provide more native functionality, but the adaption to the development of a PM in such environments can become problematic in the case of close-source environments. Both approaches are platform-independent solutions as opposed to solutions of the third family: item 3. In fact, very few non-proprietary software packages are available to create 3D geometric models. FreeCad, based on the geometry kernel Open Cascade Technology⁹ is one of these but its development is still at an early stage.

Alltogether, we regard the *implementation in C++* with the use of appropriate B-spline libraries as most suitable as its advantages outweigh those of the other approaches. While all the libraries mentioned in item 1 present valid possibilities, the choice was to employ the GoTools C++ geometric libraries, see below. Apart from GoTools, only C++ standard libraries and the Boost C++ libraries¹⁰, which are also needed by GoTools, are employed. This chosen environment allows a cross-platform implementation of the code and has full open-source character.

The GoTools Geometric Library: GoTools by Sintef is a collection of C++ interdependent libraries for geometry which are partly based on the SISL library¹¹. The open nature of this library provides the possibility to manually manipulate elementary specifications of its data structures which include points, B-spline bases, curves and surfaces.

The following is a list of those algorithms which have been employed in this thesis:

¹MatlabTM by Mathworks since 1984: www.mathworks.com/products/matlab.

²GNU OctaveTM by John W. Eaton since 1988: www.gnu.org/software/octave/.

³MapleTM by Waterloo Maple since 1980: www.maplesoft.com/products/maple.

⁴MathematicaTM by Wolfram Research since 1988: www.wolfram.com/products/mathematica.

⁵MaximaTM by Macsyma group since 1982: maxima.sourceforge.net.

⁶Rhinoceros 3DTM by Robert McNeel & Associates since 1992: www.rhino3d.com.

⁷FreeCadTM by Juergen Riegel since 2002: free-cad.sourceforge.net.

⁸PythonTM by the Python Software Foundation since 1991: python.org.

⁹Open Cascade TechnologyTM by Open Cascade since 1999: www.opencascade.org.

¹⁰BoostTM C++ libraries since 2003: www.boost.org.

¹¹SISLTM by Sintef since 1980: www.sintef.no/sisl.

- basic algorithms for vector manipulation like rotating and scaling,
- evaluations of points on curves and surfaces,
- *bounding box* calculations,
- *length* and *area* measures of curves and surfaces,
- *partitioning* of curves,
- *appending* curves to others,
- basic B-spline related algorithms like *knot insertion* and *degree elevation*,
- *curve orientation* for the generation of curve loops,
- *closest point* computation, which is employed to calculate the intersection of curves with planes, based on Newton iteration.

In addition to these geometric algorithms, GoTools provide machinery of the field of linear algebra. In particular, we make use of the LU decomposition algorithm to solve linear systems of equations.

It is left to mention that we employ the functionality of GoTools to read and write IGES¹ files, a common format in CAD software to store geometric data. This offers the possibility to visually examine the output of the model in CAD environments.

VI.3 SmoothPM: An Implementation in C++

The development of this parametric ship hull model has been realised in the environment of C++ resulting in the program “SmoothPM”. We provide an outline over the general structure of the implementation and present the employed classes and their fundamental members and functions.

The following main objects, whose description is found in Section VI.3.1, store the geometric elements for the model:

- **Ship** holds parameters, curves and patches.
- **Enhanced Curve** is the format of the control curves of the ship.
- **Boundary Segment** provides topological information between patches and curves.
- **Patch** is the object for the surface patches of the hull.

As for the employed geometry generation tools, we point out the following, see Section VI.3.2:

- **Interpolator** provides the principle functionality for curve generation.
- **Orthogonal Vector Field** is used to construct frames and ribbons.

We conclude this chapter in Section VI.3.3 with remarks on the ship hull design with SmoothPM.

¹Initial Graphics Exchange SpecificationTM (IGES), since 1980 by the U.S. National Bureau of Standards: a data format for describing product design and manufacturing information: [66].

VI.3.1 The Main Objects

Four objects can be regarded as principle throughout the generation of the ship hull, compare to Fig. VI.1.

Ship: The base class contains all parameters, curves and surfaces of the ship hull. It governs the steps of the ship design:

```
class Ship:
    ▷ float          ext_pm_1, ext_pm_2, ...;
    ▷ float          int_pm_1, int_pm_2, ...;
    ▷ EnhancedCurve curve_1, curve_2, ...;
    ▷ Patch          patch_1, patch_2, ...;
    ◁ void   create_wireframe();
    ◁ void   assign_patch_topology();
    ◁ void   build_patches();
```

Enhanced Curve: This class holds two B-spline expressions in the format SplineCurve of the GoTools library. The first corresponds to the geometry of the curve itself and the second one represents the curve frame. To interpolate given data and to enhance the curve with a frame, Enhanced Curve employs the functionalities provided by the construction classes below.

```
class EnhancedCurve:
    ▷ SplineCurve  curve;
    ▷ SplineCurve  frame;
    ◁ void   interpolate( /* interpolation data */ );
    ◁ void   enhance_with_frame( /* frame data */ );
```

Boundary Segment: The purpose of this class is to store the topology information between patches. For two adjacent patches, an object of this class is created and holds pointers to both patches. The object is completed by another pointer to the inheriting curve of the patches' common boundary and by the parameters to which the curve has to be trimmed.

```
class BoundarySegment:
    ▷ Patch          *left_parent, *right_parent;
    ▷ EnhancedCurve *curve;
    ▷ float          start, end;
    ◁ EnhancedCurve  get_geometry(bool invert);
```

Patch: The main contents of this class is, of course, the surface geometry, which is stored in GoTools format SplineSurface. For its construction, we firstly assign to the patch a vector of pointers to Boundary Segment objects. Together with information about each Boundary Segment's orientation we obtain loop of curves, compare to Fig. VI.1 where we see an example of a typical patch arrangement. With a properly constructed control curve network, it is possible to obtain a quadrilateral curve loop and, therefore, the Algorithms III.4.1 and III.4.7 can be applied to generate the surface geometry.

class Patch:

```

    ▷ Ship                *parent;
    ▷ bool                linear_build;
    ▷ SplineSurface      geometry;
    ▷ vector<BoundarySegment*> loop;
    ▷ vector<bool>       orientations;

    < void    add_boundary_segment(BoundarySegment*);
    < void    make_loop(void);
    < vector<EnhancedCurve> make_coons_loop(void);
    < vector<SplineCurve>  make_ribbons(vector<EnhancedCurve>);
    < void    bicubical_blend( /* curves and ribbons */ );
    < void    bilinear_blend( /* curves */ );
    
```

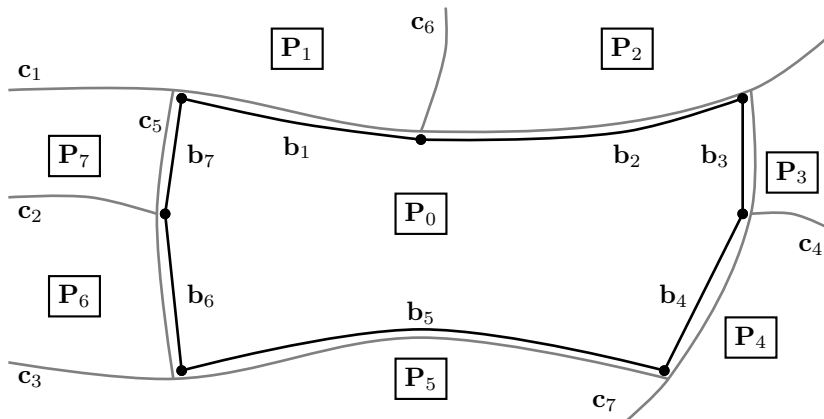


Figure VI.1: The topological map of the patch P_0 : The Boundary Segments b_1, \dots, b_7 provide the link to the adjacent Patches P_1, \dots, P_7 , respectively. Trimming the corresponding inheriting Enhanced Curves (c_1, c_3, c_5 and c_7) to appropriate parameters results in a quadrilateral loop. Compare to Section VI.3.1.

VI.3.2 The Construction Classes

Interpolator: The following structure stores a point in the GoTools format Point, an associated parameter value and information whether this point should be regarded a direction rather than a point or vector:

```
struct DataPoint:  
    ▷ Point    pt;  
    ▷ float    pm;  
    ▷ bool     unit;
```

With the help of Data Point, we gather the interpolation points and tangents (vectors or directions) in the following class. Additionally, we can include parameter values where the curve's curvature has to be zero. Once all data is populated, Algorithm III.1.14 is employed to construct a piecewise Hermite curve, a basis and, finally, a C^2 continuous curve. The solution of the linear equation system is obtain by a LU decomposition provided in the GoTools library.

```
class Interpolator:  
    ▷ vector<DataPoint>  points, tangents;  
    ▷ vector<float>      k0_pms;  
    ▷ vector<float>      knots;  
    ▷ SplineCurve        pw_hermite_curve, curve;  
    ◁ void    create_pw_hermite();  
    ◁ void    make_basis( /* degree, number of extra CP's */ );  
    ◁ void    interpolate( /* approximation weight */ );
```

Orthogonal Vector Field: In preparation of the construction of frames, we introduce the following structure storing information about the compatibility, see Section III.3.2. It also includes a flag which indicates whether this object has to be taken into account or not.

```
struct CompCond:  
    ▷ bool    is_relevant;  
    ▷ Point   tangent, frame_vector_0, frame_vector_1;  
    ▷ int     degree;
```

The construction of frames and ribbons is, in principle, the same. We have, therefore, implemented the following class which computes an vector field orthogonal to an existing one. It stores the given field, end vectors and compatibility information. Additionally, a flag indicates whether the construction refers to a frame or a ribbon. We solve the resulting system of linear equations by a LU decomposition provided in the GoTools library.

```
class OrthoField:  
    ▷ SplineCurve *given_curve;
```



```
▷ SplineCurve *given_field;
▷ Point      start_vector, end_vector;
▷ CompCond   start_cc, end_cc;
▷ bool       is_frame;
▷ SplineCurve calculated_field;

◁ void create_rmf();
◁ void create_field( /* degree, rmf weight */ );
```

VI.3.3 Design with SmoothPM

Performance: The quadratic programming problems which arise in curve interpolation and in the construction of frames and ribbons result in linear equation systems. Their solution, which we obtain by an LU decomposition provided by GoTools, becomes costly as the matrices grow. In this connection, Algorithm III.3.8 for the creation of ribbons is particularly costly since the involved fields are usually of elevated degrees. The generation of all 22 patches, therefore, consumes over 95% of the overall time, which is approximately one minute on a personal computer with a 2.5 Ghz processor.

To accelerate the generation of a model instances, we have parallelised the construction of the patches with the help of Open MPI¹. This is possible due to the hierarchical equality of the patches as we proposed in the design principles in Section II.1. With sufficiently many processor kernels at SmoothPM's disposition, the overall time can be, therefore, reduced to the time of the costliest patch construction plus the almost negligible time of the generation of the wireframe.

Communication with SmoothPM: At the current stage of development, the designer's communication with SmoothPM is elementary. The input parameters are passed to the program via a text file and another file with the ship hull geometry in IGES format is returned upon successful completion. Alterations of the design process have to be done on source code-level.

¹Open MPI™ by the MPI Forum since 1994: www.open-mpi.org.

Chapter VII

Outlook and Future Work

Future work for this model includes the three aspects of optimisation, implementation and methodology:

Embedding into Optimisation: This PM has been designed for the specific purpose of an isogeometric optimisation loop. It is, therefore, the foremost future task to embed the program SmoothPM into an optimisation loop as in Section I.3.3.

Implementation: For the C++ program SmoothPM, see Section VI.3, several improvements are pending:

1. To replace the piecewise Bézier expressions of frames and ribbons, compare to Sections III.2 and III.3, by B-spline expressions, the linear equation systems which stem from the product of B-Spline expressions have to be implemented.
2. The developed libraries of frames and ribbon construction through orthogonal vector-fields can be used in the environment of an existing scriptable CAD-modeller for the creation of geometric and parametric models of different objects. In this way a more visual design process is possible.

Methodology: From a methodological point-of-view we propose to explore the following:

1. Reasonable a-priori choices for the design parameters of the algorithms of interpolation, framing and ribboning have to be established, i.e. w^C and n^+ in Algorithm III.1.14, w^F and d^+ in Algorithm III.2.13 and w^R and δ^+ in Algorithm III.3.8.
2. In the current design process no a-posteriori surface fairing is employed. However, while the first two rows of control points at each of the patch's boundaries have to remain untouched in order not to jeopardise the cross-boundary continuity, all other control points of the patch can be subject to fairing.

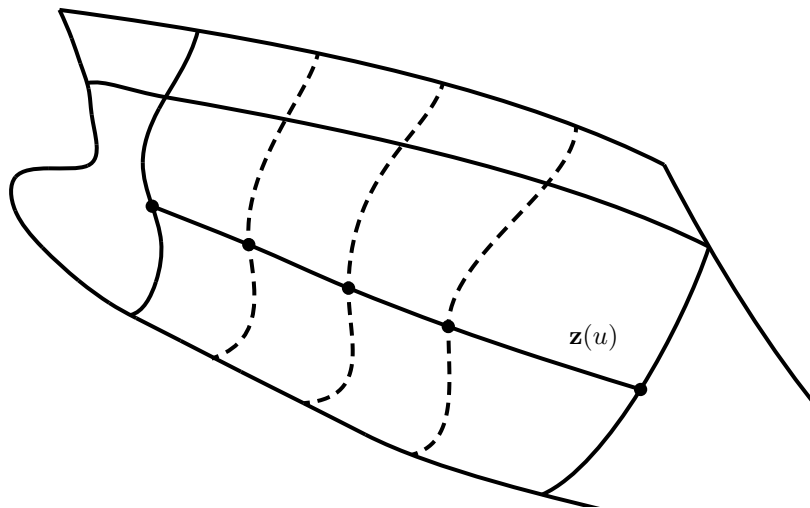
3. The construction process is designed for Coons patches only in the sense that the ribbons have to be constructed compatibly and suitably for the Coons patch construction. However, any patch construction capable of respecting existing boundary information can be employed. An alternative, like e.g. Gregory patches which employ variable twist, may provide an advantageous alternative.
4. In the line of the previous item, we may adopt the principle of Coons patches to fulfil cross-boundary conditions of higher order. (Bi-)quintic blends can be used to blend between curvature data at opposite boundary in addition to the positional and tangential conditions. Such machinery opens the possibility for the below described zero sectional-curvature curve approach.

A Zero Sectional-Curvature Curve Approach: We conclude the outlook section and, therefore, this dissertation with comments on this alternative design approach. We examine the transitional areas between the midship and the bow by intersecting the hull with planes which are parallel to $y-z$ plane. The obtained curves, the so-called section curves, are characterised by a particular shape, compare to the ship characteristics in Section I.5. For a given range, they all exhibit one inflection point (points where the sign of the curvature κ changes) and by a moving intersection plane we obtain a collection of such inflection points forming a smooth curve, the *zero sectional-curvature curve* (ZSCC), compare to Figures VII.1(a). In the stern area, the behaviour is similar but the section curves exhibit two inflection points and we, therefore, obtain two ZSCC, see Figure VII.1(b).

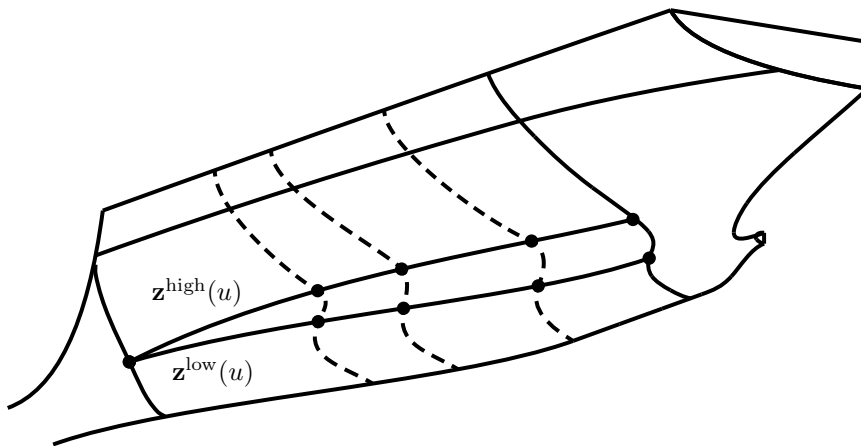
A possible design approach is to define these curves already in the process of wireframe construction and to a-posteriori employ a patch construction which complies with the information of the ZSCCs. A Biquintically blended Coons patch construction, which rule between curves and respecting first and second-order vector-fields at the boundaries, may be the missing piece to follow such an approach, compare to item 4 above.

In the bow area the ZSCC $\mathbf{z}(u)$ would be similar to the current Bow Horizontal Curve $\mathbf{c}^{36}(u)$ whereas in the stern, we would employ two such curves, $\mathbf{z}^{\text{high}}(u)$ and $\mathbf{z}^{\text{low}}(u)$, substituting the current Stern Horizontal Curve $\mathbf{c}^{40}(u)$. For their construction we have to guarantee that the end points are indeed inflection points of their respective section curves. This can be done by finding the points on the section curves where $\kappa = 0$ or by constructing them with Interpolation Algorithm III.1.14 and imposing parameters with $\kappa = 0$.

Conclusion: We have provided novel geometric algorithms to construct orthogonal vector fields for B-spline expressions and employ them to enhance given curves with the distribution of their moving tangent plane, the frame. In the design process of a container ship, the wireframe geometry has been created by constructing also a corresponding frame for each curve. To fill the principally quadrilateral curve loops of the wireframe, we employ the bicubically blended Coons path construction and we use the boundary ribbons of this construction to comply with the tangent plane information provided by the frames. An exact G^1 continuous ship hull is obtained providing an approximation free geometry



(a) The zero sectional-curvature curve $\mathbf{z}(u)$ of the Bow.



(b) The zero sectional-curvature curves $\mathbf{z}^{\text{high}}(u)$ and $\mathbf{z}^{\text{low}}(u)$ of the Stern.

Figure VII.1: Zero Sectional-Curvature Curves.

of the ship hull, which is, therefore, particularly well suited for isogeometric analysis.

The model is governed 29 + 9 parameters which reflect principal ship characteristics enabling the automated creation of hull instances. This parametric aspect is fundamental in an optimisation loop for geometry of the ship hull and to this end we have presented the results of 18 different parameter sets.

Appendix A

B-Splines

A.1 Curves and Surfaces

To provide quick reference and notation, we summarize the principal definitions and theorems for spline curves and surfaces. A detailed description can be found in, e.g. in Piegl and Tiller [70] (1997) or Farin [32] (2002).

Bézier Curves: For $p > 0$, a set of $p + 1$ basis functions $(N_i^p)_{i=0}^p(u)$ for $u \in [0, 1]$ is defined as follows:

$$N_i^p(u) = \binom{p}{i} u^i (1-u)^{p-i}.$$

The corresponding Bézier curve is a parametric curve defined by its degree p and its $p + 1$ control points $(\mathbf{c}_i)_{i=0}^p$:

$$\mathbf{c}(u) = \sum_{i=0}^p N_i^p(u) \mathbf{c}_i.$$

B-Spline Curves: A generalization of Bézier curves are B-spline curves. For $p > 0$, a set of $n \geq p + 1$ basis functions is defined by a vector of $n + p + 1$ non-decreasing abscissae, the knots $(k_i)_{i=0}^{n+p}$. We consider only the case of knot vectors with boundary multiplicities $p + 1$ and where no more than $p + 1$ consecutive knots coincide. The basis functions are defined recursively:

$$N_i^0(u) = \begin{cases} 1 & k_i < u < k_{i+1} \\ 0 & \text{else} \end{cases},$$
$$N_i^p(u) = \frac{u - k_i}{k_{i+p} - k_i} N_i^{p-1}(u) + \frac{k_{i+1} - u}{k_{i+p+1} - k_{i+1}} N_{i+1}^{p-1}(u)$$

where we define quotients of the form $0/0$ as 0. If $n = p + 1$, these coincide with the Bézier basis functions.

For a given degree p and $n + p + 1$ knot values, a set of n control points $(\mathbf{c}_i)_{i=0}^{n-1}$ defines the curve:

$$\mathbf{c}(u) = \sum_{i=0}^{n-1} N_i^p(u) \mathbf{c}_i.$$

NURBS Curves: Employing the same specifications as in the B-spline case and with a set of n positive weights $(w_i)_{i=0}^{n-1}$ we define the Non-Uniform Rational B-spline as follows:

$$\mathbf{c}(u) = \frac{\sum_{i=0}^{n-1} N_i^p(u) w_i \mathbf{c}_i}{\sum_{i=0}^{n-1} N_i^p(u) w_i}.$$

Derivatives: The continuity of a B-spline curve depends on its degree p and the multiplicities of its interior knots. B-spline and NURBS are infinitely many times differentiable on every non-degenerate knot interval $[k_i, k_{i+1})$. At the knots, they are $p - m$ times differentiable where m denotes the multiplicity of the interior knot.

To obtain the derivate of a B-spline curve we have to derive its basis functions:

$$N_i^{p'}(u) = \frac{p}{k_{i+p} - k_i} N_i^{p-1}(u) - \frac{p}{k_{i+p+1} - k_{i+1}} N_{i+1}^{p-1}(u).$$

Therefore, the first derivative of a B-spline curve is another B-spline expression of degree $p - 1$ with the same interior knots and knot multiplicity p at the boundaries:

$$\begin{aligned} \mathbf{c}'(u) &= \sum_{i=0}^{n-2} N_i^{p-1}(u) \mathbf{d}_i \\ &= \sum_{i=0}^{n-2} N_i^{p-1}(u) \frac{p}{k_{i+p+1} - k_{i+1}} (\mathbf{c}_{i+1} - \mathbf{c}_i). \end{aligned}$$

Knot Insertion: A B-spline curve can be represented with denser knot abscissae than initially defined. Inserting the knot \hat{k} with $k_j < \hat{k} < k_{j+1}$ into the interval (k_j, k_{j+1}) , the $n + 1$ control points \mathbf{Q}_i to obtain the initial curve are defined as follows:

$$\begin{aligned} \mathbf{Q}_i &= \mathbf{c}_i & i \leq j - p \\ \alpha_i &= \frac{\hat{k} - k_j}{k_{j+p} - k_j} \\ \mathbf{Q}_i &= \alpha_i \mathbf{c}_i + (1 - \alpha_i) \mathbf{c}_{i-1} & j - p + 1 \leq i \leq j \\ \mathbf{Q}_{i+1} &= \mathbf{c}_i & k + 1 \leq i \end{aligned}$$

Note, that the curve parametrization is not altered by knot insertion.

Degree Elevation: The representation of a B-spline curve by a degree $\hat{p} > p$ is called degree elevation and can be done without modifying the parametrization of the curve. As for the basis, we raise the multiplicity of every knot value by $\hat{p} - p$. With the number of interior knot values s we obtain the $\hat{n} = n + s + 1$ new control points \mathbf{Q}_i by solving for them the following system of equations:

$$\sum_{i=0}^{\hat{n}-1} N_i^{\hat{p}}(u) \mathbf{Q}_i = \sum_{i=0}^{n-1} N_i^p(u) \mathbf{c}_i. \quad (\text{A.1})$$

Tensor Product Surfaces: A tensor product B-spline surface is defined by the degrees p and q and knot values with respect to two parameter directions u and v . They define two sets of basis function $(N_i^p(u))_{i=0}^{m-1}$ and $(N_j^q(v))_{j=0}^{n-1}$ and with $m * n$ control points \mathbf{S}_{ij} we define the B-spline surface

$$\mathbf{S}(u, v) = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} N_i^p(u) N_j^q(v) \mathbf{S}_{ij}.$$

Derivatives, degree elevation and knot insertion with respect to both parameter directions are analogous to in the curve case.

A.2 Product of Bézier Expressions

While addition and subtraction are well-defined for curves, this is not the case for their product. However, for one-dimensional Bézier expression, there is the natural interpretation of the product of real numbers. Be $x(u)$ and $y(u)$ two Bézier expressions of degree p and q , respectively, we define their product $z(u)$:

$$\begin{aligned} z(u) &:= x(u)y(u) \\ &= \left(\sum_{i=0}^p N_i^p(u) x_i \right) \left(\sum_{i=0}^q N_i^q(u) y_i \right) = \\ &= \left(\sum_{i=0}^p \binom{p}{i} (1-u)^{p-i} u^i x_i \right) \left(\sum_{i=0}^q \binom{q}{i} (1-u)^{q-i} u^i y_i \right) = \\ &= \sum_{i=0}^p \sum_{i=0}^q \binom{p}{i} \binom{q}{i} (1-u)^{p+q-i-j} u^{i+j} x_i y_i \end{aligned}$$

which can then be written as a Bézier expression of degree $p+q$, see e.g. Farouki and Rajan [33] (1988):

$$z(u) = \sum_{k=0}^{p+q} \binom{p+q}{k} (1-u)^{p+q-k} u^k z_k$$

The explicit coefficients z_k are, therefore:

$$z_k = \sum_{j=\max(0, k-q)}^{\min(p, k)} \frac{\binom{p}{j} \binom{q}{k-j}}{\binom{p+q}{k}} x_j y_{k-j}.$$

Scalar Product: As for the product of two vectors, we can define their scalar product, also referred to as inner product, in 3D Euclidean space as follows:

$$\mathbf{v}_1 \cdot \mathbf{v}_2 = v_1^x v_2^x + v_1^y v_2^y + v_1^z v_2^z.$$

The scalar product of two Bézier curves results, therefore, in a scalar Bézier expression:

$$\begin{aligned} z(u) = \mathbf{c}(u) \cdot \mathbf{d}(u) &= \left(\sum_{i=0}^p N_i^p(u) \mathbf{c}_i \right) \cdot \left(\sum_{i=0}^q N_i^q(u) \mathbf{d}_i \right) = \\ &= \sum_{k=0}^{p+q} \binom{p+q}{k} (1-u)^{p+q-k} u^k z_k \end{aligned} \quad (\text{A.2})$$

whose coefficients z_k are the following:

$$\begin{aligned} z_k &= \sum_{j=\max(0,k-q)}^{\min(p,k)} \frac{\binom{p}{j} \binom{q}{k-j}}{\binom{p+q}{k}} \mathbf{c}_j \cdot \mathbf{d}_{k-j} \\ &= \sum_{j=\max(0,k-q)}^{\min(p,k)} \frac{\binom{p}{j} \binom{q}{k-j}}{\binom{p+q}{k}} (c_j^x d_{k-j}^x + c_j^y d_{k-j}^y + c_j^z d_{k-j}^z). \end{aligned}$$

Cross Product: The cross product of two vectors, also referred to as outer product,

$$\mathbf{v}_1 \times \mathbf{v}_2 = \begin{pmatrix} v_1^y v_2^z - v_1^z v_2^y \\ -v_1^x v_2^z + v_1^z v_2^x \\ v_1^x v_2^y - v_1^y v_2^x \end{pmatrix},$$

applied to two Bézier curves yields the following expression:

$$\begin{aligned} \mathbf{z}(u) &= \mathbf{c}(u) \times \mathbf{d}(u) = \left(\sum_{i=0}^p N_i^p(u) \mathbf{c}_i \right) \times \left(\sum_{i=0}^q N_i^q(u) \mathbf{d}_i \right) = \\ &= \sum_{k=0}^{p+q} \binom{p+q}{k} (1-u)^{p+q-k} u^k \mathbf{z}_k \end{aligned}$$

with the coefficients

$$\begin{aligned} \mathbf{z}_k &= \sum_{j=\max(0,k-q)}^{\min(p,k)} \frac{\binom{p}{j} \binom{q}{k-j}}{\binom{p+q}{k}} \mathbf{c}_j \times \mathbf{d}_{k-j} \\ &= \sum_{j=\max(0,k-q)}^{\min(p,k)} \frac{\binom{p}{j} \binom{q}{k-j}}{\binom{p+q}{k}} \begin{pmatrix} c_j^y d_{k-j}^z - c_j^z d_{k-j}^y \\ -c_j^x d_{k-j}^z + c_j^z d_{k-j}^x \\ c_j^x d_{k-j}^y - c_j^y d_{k-j}^x \end{pmatrix}. \end{aligned}$$

Construction of Orthogonal Vector Fields: We consider a vector field $\mathbf{t}(u)$ in Bézier representation of degree p . To construct another field $\mathbf{f}(u)$ in Bézier representation of degree q which is orthogonal to $\mathbf{t}(u)$, we employ Equation (A.2) and demand all of the resulting coefficients z_k to be zero:

$$\begin{aligned} \forall k \in (1, \dots, p+q) : \\ 0 &= \binom{p+q}{k} z_k = \sum_{j=\max(0,k-q)}^{\min(p,k)} \binom{p}{j} \binom{q}{k-j} (t_j^x f_{k-j}^x + t_j^y f_{k-j}^y + t_j^z f_{k-j}^z). \end{aligned} \tag{A.3}$$

This results in a linear system of $p+q+1$ homogeneous equations for $q+1$ unknown control points \mathbf{f}_i , each of which possesses 3 coordinates. The corresponding matrix of dimensions $(p+q+1) \times 3(q+1)$ has band structure of maximum width $3(\min(p,q)+1)$ in horizontal direction and, thus, maximum

width $(\max(p, q) + 1)$ in vertical direction :

$$\begin{pmatrix} 0 \\ \dots \\ 0 \end{pmatrix} = \begin{pmatrix} \mathbf{t}_0 \cdot \\ \binom{p}{1} \binom{q}{0} \mathbf{t}_1 \cdot & \binom{p}{0} \binom{q}{1} \mathbf{t}_0 \cdot & & \\ \binom{p}{2} \binom{q}{0} \mathbf{t}_2 \cdot & \binom{p}{1} \binom{q}{1} \mathbf{t}_1 \cdot & \binom{p}{0} \binom{q}{2} \mathbf{t}_0 \cdot & \\ & \dots & & \\ & \binom{p}{p} \binom{q}{q-2} \mathbf{t}_p \cdot & \binom{p}{p-1} \binom{q}{q-1} \mathbf{t}_{p-1} \cdot & \binom{p}{p-2} \binom{q}{q} \mathbf{t}_{p-2} \cdot \\ & & \binom{p}{p} \binom{q}{q-1} \mathbf{t}_p \cdot & \binom{p}{p-1} \binom{q}{q} \mathbf{t}_{p-1} \cdot \\ & & & \mathbf{t}_p \cdot \end{pmatrix} \begin{pmatrix} \mathbf{f}_0 \\ \dots \\ \mathbf{f}_q \end{pmatrix}. \quad (\text{A.4})$$

Solvability for this homogeneous system is always provide if $q > p/2 - 1$.

Example A.2.1. For the case of $p = 3$ and $q = 2$, the equation system stemming from Eq. (A.3) is the following:

$$\begin{aligned} k=0: \quad 0 &= \binom{3}{0} \binom{2}{0} \mathbf{t}_0 \cdot \mathbf{f}_0 \\ k=1: \quad 0 &= \binom{3}{1} \binom{2}{0} \mathbf{t}_1 \cdot \mathbf{f}_0 + \binom{3}{0} \binom{2}{1} \mathbf{t}_0 \cdot \mathbf{f}_1 \\ k=2: \quad 0 &= \binom{3}{2} \binom{2}{0} \mathbf{t}_2 \cdot \mathbf{f}_0 + \binom{3}{1} \binom{2}{1} \mathbf{t}_1 \cdot \mathbf{f}_1 + \binom{3}{0} \binom{2}{2} \mathbf{t}_0 \cdot \mathbf{f}_2 \\ k=3: \quad 0 &= \binom{3}{3} \binom{2}{0} \mathbf{t}_3 \cdot \mathbf{f}_0 + \binom{3}{2} \binom{2}{1} \mathbf{t}_2 \cdot \mathbf{f}_1 + \binom{3}{1} \binom{2}{2} \mathbf{t}_1 \cdot \mathbf{f}_2 \\ k=4: \quad 0 &= \binom{3}{3} \binom{2}{1} \mathbf{t}_3 \cdot \mathbf{f}_1 + \binom{3}{2} \binom{2}{2} \mathbf{t}_2 \cdot \mathbf{f}_2 \\ k=5: \quad 0 &= \binom{3}{3} \binom{2}{2} \mathbf{t}_3 \cdot \mathbf{f}_2 \end{aligned}$$

or equivalently:

$$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} \binom{3}{0} \binom{2}{0} \mathbf{t}_0 \cdot & & & & & \\ \binom{3}{1} \binom{2}{0} \mathbf{t}_1 \cdot & \binom{3}{0} \binom{2}{1} \mathbf{t}_0 \cdot & & & & \\ \binom{3}{2} \binom{2}{0} \mathbf{t}_2 \cdot & \binom{3}{1} \binom{2}{1} \mathbf{t}_1 \cdot & \binom{3}{0} \binom{2}{2} \mathbf{t}_0 \cdot & & & \\ \binom{3}{3} \binom{2}{0} \mathbf{t}_3 \cdot & \binom{3}{2} \binom{2}{1} \mathbf{t}_2 \cdot & \binom{3}{1} \binom{2}{2} \mathbf{t}_1 \cdot & & & \\ & \binom{3}{3} \binom{2}{1} \mathbf{t}_3 \cdot & \binom{3}{2} \binom{2}{2} \mathbf{t}_2 \cdot & & & \\ & & \binom{3}{3} \binom{2}{2} \mathbf{t}_3 \cdot & & & \end{pmatrix} \begin{pmatrix} \mathbf{f}_0 \\ \mathbf{f}_1 \\ \mathbf{f}_2 \end{pmatrix}.$$

Note that we have $3 * 3 = 9$ unknown control point coordinates for six linear

List of Figures

I.1	Differential Operator	1
I.2	A Model Problem	2
I.3	Optimization Loop	12
I.4	Parametric Model: Rectangle	13
I.5	Parametric Model: Partitioned Rectangle	14
I.6	The Eleonora Maersk Container Ship	15
I.7	A Bulbous Bow	16
I.8	Container Ship Characteristics	17
II.1	A sketch of the control curve network, the wireframe.	20
II.2	Frames and Ribbons	21
II.3	Two G^1 Continuous Patches	22
II.4	Global Parameters	23
II.5	MidShip Parameters	25
II.6	Bow Parameters	26
II.7	Stern Parameters	28
II.8	Catia-Model: CCN with Ribbons	31
II.9	Catia-Model: Refined CCN with Ribbons	32
III.1	Hermite Interpolation	38
III.2	Hermite Interpolation, Scaling Comparison	39
III.3	Hermite Interpolation, Scaling Comparison, Degree 2	40
III.4	Quadratic Hermite Interpolation, Tangent Length.	40
III.5	Hermite Interpolation, Scaling Comparison, Degree 3	42
III.6	Cubic Hermite Interpolation, Tangent Lengths.	43
III.7	Interpolation, G^1 Joint	44
III.8	Knots Distribution	47
III.9	Knots Distribution, Example 2	47
III.10	Interpolation Illustration	49
III.11	Interpolation: Knots	50
III.12	Interpolation Parameters: Parametrization	51
III.13	Interpolation: Approximation Weight	52
III.14	Interpolation: DOF	53
III.15	Tangent Plane	55
III.16	Framing Example with Compatibility Conditions	57
III.17	The RMF and the Frenet Frame	63
III.18	RMF: Double Reflection Method	64
III.19	RMF with Boundary Conditions	64
III.20	Framing Example, Bézier	66
III.21	Framing Example, B-Spline	67
III.22	Framing Example with Compatibility Conditions	68
III.23	Minimal Twist Vector	76

LIST OF FIGURES

III.24	RMF: Double Reflection Method	77
III.25	Ribboning Example, Bézier	79
III.26	Ribboning Example, B-Spline	79
III.27	Bézier Curves Loop with Frames	84
III.28	Bilinearly Blended Coons Patch from Bézier Curves Loop	84
III.29	Bézier Curve Loop with Ribbons	84
III.30	Bicubically Blended Coons Patch from Bézier Curves Loop	85
III.31	B-Spline Curve Loop with Frames	86
III.32	B-Spline Curve Loop with Ribbons	87
III.33	Bicubically Blended Coons Patch from B-Spline Curve Loop	87
IV.1	Diagram of the Stages of the Wireframe	90
IV.2	MidShip Curve Network	94
IV.3	Bow Profile Curve Data	95
IV.4	Bulb Root Determination	96
IV.5	The Stages of the Wireframe, Bow	97
IV.6	The Stages of the Wireframe, Stern	100
IV.7	Stern Back Section Construction	103
IV.8	Shaft Curve Construction	107
IV.9	Patch Network, MidShip	112
IV.10	Patch Network, MidShip and Bow Joint	115
IV.11	Bulb Bottom Patch Parametrisation Comparison	117
IV.12	Patch Network, Bow	118
IV.13	Patch Network, MidShip and Stern Joint	120
IV.14	Patch Network, Stern	123
V.1	Initial Parameter Set, Global Views	126
V.2	Initial Parameter Set, Bow Views	127
V.3	Initial Parameter Set, Stern Views	127
V.4	Parameter Set “Long”	128
V.5	Parameter Set “Short”	129
V.6	Parameter Set “Wide”	130
V.7	Parameter Set “Narrow”	130
V.8	Parameter Set “High”	131
V.9	Parameter Set “Low”	132
V.10	Parameter Set “Long Parallel Midship”	132
V.11	Parameter Set “Short Parallel Midship”	133
V.12	Parameter Set “Long Bulb”	134
V.13	Parameter Set “Short Bulb”	135
V.14	Parameter Set “Broad Bulb”	135
V.15	Parameter Set “High Bulb”	136
V.16	Parameter Set “Low Bulb”	137
V.17	Parameter Set “Long Transom”	137
V.18	Parameter Set “Long Shaft”	139
V.19	Parameter Set “High Shaft”	139
V.20	Parameter Set “Low Shaft”	140
VI.1	Topological arrangement in SmoothPM	145
VII.1	Zero Sectional-Curvature Curves.	151

References

- [1] C. Abt, S.D. Bade, L. Birk, S. Harries. Parametric Hull Form Design - A Step Towards One Week Ship Design. *8th International Symposium on Practical Design of Ships and Other Floating Structures (PRADS 2001)*, Shanghai, China, September 2001.
- [2] C. Abt, L. Birk, S. Harries. Parametric Hull Design: The FRIENDSHIP-Modeler. *International Conference on Ship and Shipping Research, NAV 2003, Palermo*, 2003.
- [3] C. Abt, S. Harries. FRIENDSHIP-Framework - Integrating Ship-Design Modelling, Simulation, and Optimisation. *The Naval Architect, RINA*, January 2007.
- [4] M. Aigner, L. Gonzalez-Vega, B. Jüttler, M.L. Sampoli. Computing isophotes on free-form surfaces based on support function approximation. *The Mathematics of Surfaces (MoS XIII 2009)*, LNCS 5654, Springer, 2009, 1 - 18, 2009.
- [5] J.J.M. Baar, W.G. Price. Developments in the Calculation of the Wave Making Resistance of Ships. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, Vol. 416, No. 1850, pp. 115-147, 8 Mar 1988.
- [6] A. Baer, C. Eastman, M. Henrion. Geometric modelling: a survey. *Computer-Aided Design*, Volume 11, Issue 5, 1979.
- [7] R.E. Barnhill, A.J. Worsey. Smooth interpolation over hypercubes. *Computer-Aided Geometric Design* 1, 101-113, 1984.
- [8] R. Brard. The Representation of a Given Ship Form by Singularity Distributions when the Boundary Condition on the Free Surface is Linearized. *J. Ship Res.*, 16, 79-82, 1972.
- [9] K.A. Belibassakis, Th.P. Gerostathis, C.G. Politis, P.D. Kaklis, A.-A.I. Ginnis, D.N. Mourkogiannis. A novel BEM-ISOGOMETRIC method with application to the wavemaking resistance problem of bodies at constant speed. *13th Congress of Intl. Maritime Assoc. of Mediterranean, IMAM 2009, Istanbul, Turkey*, 12-15 Oct. 2009
- [10] K.A. Belibassakis, Th.P. Gerostathis, K.V. Kostas, C.G. Politis, P.D. Kaklis, A.-A.I. Ginnis, C. Feurer. A BEM-ISOGOMETRIC method with application to the wavemaking resistance problem of ships at constant speed.

REFERENCES

- 30th International Conference on Offshore Mechanics and Arctic Engineering, OMAE2011*, Rotterdam, 2011.
- [11] K.A. Belibassakis, Th.P. Gerostathis, K.V. Kostas, C.G. Politis, P.D. Kaklis, A.-A.I. Ginnis, C. Feurer. A BEM-isogeometric method for the ship wave-resistance problem. *Ocean Engineering*, 2013.
- [12] V. Bertram. Practical Ship Hydrodynamics. *Butterworth-Heinemann, Oxford*, 2000.
- [13] D.P. Bertsekas. Nonlinear Programming: 2nd Edition, 2nd Printing. *Athena Scientific*, 2003.
- [14] R.L. Bishop. There is More than One Way to Frame a Curve. *The American Mathematical Monthly*, Vol. 82, No. 3, pp. 246-251, March 1975.
- [15] W. Böhm, G. Farin. A survey of curve and surface methods in CAGD. *Computer-Aided Geometric Design*, 1984, 1-60, 1984.
- [16] W. Böhm. Visual Continuity. *Computer-Aided Geometric Design*, 1988, volume 20, number 6, 1988.
- [17] D.-Y. Cho, K.-Y. Lee, T.-W. Kim. Interpolating G1 Bezier surfaces over irregular curve networks for ship hull design. *Computer Aided Design*, 2006.
- [18] B.K. Choi, H. Shin, W.S. Yoo. Visually smooth composite surfaces for an unevenly spaced 3D data array. *Computer Aided Geometric Design*, 1993.
- [19] CFD Workshop 2005 (CFDWS2005). <http://www.nmri.go.jp/cfd/cfdws05/>. *National Maritime Research Institute, Mitaka, Tokyo, Japan*, March 9 - 11, 2005.
- [20] L. Collatz, W. Wetterling. Optimization Problems. *Springer-Verlag New York Inc.*, 1975.
- [21] S.A. Coons. Surfaces for Computer-Aided Design of Space Forms. *Design Division, Mechanical Engineering Dept. Massachusetts Institute of Technology, Cambridge, Massachusetts*, 1967.
- [22] J.A. Cottrell, T.J.R. Hughes, A. Reali. Studies of refinement and continuity in isogeometric structural analysis. *Computer Methods in Applied Mechanics and Engineering*, 196, 41-44, 4160-4183, 2007.
- [23] J.A. Cottrell, T.J.R. Hughes, Y. Bazilevs. Isogeometric Analysis: Toward Integration of CAD and FEA. *Wiley*, 2009.
- [24] J. Dennis, V. Torczon. Direct search methods on parallel machines. *SIAM Journal of Optimization* 1, 4, 448-474, 1991.
- [25] W.H. Du, F.J.M. Schmitt. On the G^1 continuity of piecewise Bézier surfaces: a review with new results. *SIAM Journal of Optimization* 1, 4, 448-474, 1991.
- [26] H. Engl. Integralgleichungen. *Springer Verlag GmbH*, 1997.

-
- [27] L.C. Evans. Partial Differential Equations. *Graduate Studies in Mathematics, American Mathematical Society*, 1998.
- [28] D.J. Eyres. Ship Construction, Fifth Edition. *Butterworth-Heinemann*, 2001.
- [29] Exciting - Exact Geometry Simulation for the Optimized Design of Vehicles and Vessels. <http://exciting-project.eu/>. *An EU project in the FP7-Framework. FP7- SST- 2007- RTD-1 - 218536*, October 2008 - March 2012.
- [30] G. Farin. A Construction for Visual C^1 Continuity of Polynomial Surface Patches. *Computer Graphics and Image Processing*, 1982.
- [31] G. Farin. NURBS Curves and Surfaces. *AK Peters*, Boston, 2nd Edition, 1995.
- [32] G. Farin. Curves and Surfaces for CAGD. *Morgan Kaufmann Publishers*, San Francisco, 5th Edition, 2002.
- [33] R.T. Farouki, V.T. Rajan. Algorithms for Polynomials in Bernstein Form. *Computer Aided Geometric Design*, 1988. 5, I 26., 1988.
- [34] R.T. Farouki, T. Sakkalis. A complete classification of quintic space curves with rational rotation-minimizing frames. *Journal of Symbolic Computation Volume 47, Issue 2, Pages 214-226*, February 2012.
- [35] Friendship Framework. <http://www.friendship-systems.com/>. *FRIENDSHIP SYSTEMS GmbH, Benzstrasse 2, D-14482 Potsdam, Germany*.
- [36] A.-A.I. Ginnis, K.V. Kostas, C. Feurer, K.A. Belibassakis, Th.P. Gerostathis, C.G. Politis, P.D. Kaklis. A Catia Ship-Parametric Model for Isogeometric Hull Optimization with Respect to Wave Resistance. *Pro-ceedings of the ICCAS 2011 International Conference on Computer Applications in Shipbuilding*, Trieste, Italy, 20-22 September 2011.
- [37] W. Gordon. Free-form surface interpolation through curve networks, *GMR-921, General Motors Research Laboratories*, 1969.
- [38] J.A. Gregory, V. K.H. Lau, J. Zhou. Smooth Parametric Surfaces and n-Sided Patches. *GMR-921, General Motors Research Laboratories*, 1969.
- [39] S. Hahmann. Visualization techniques for surface analysis. *Advanced Visualization Techniques, Wiley*, 49-74, (1999).
- [40] S. Hahmann, A. Belyaev, L. Buse, G. Elber, B. Mourrain, C. Rössl. Shape Interrogation. *Shape Analysis and Structuring, Springer*, 1-52, (2007).
- [41] P.J. Hartley, C. Judd. Parametrization and Shape of B-spline Curves for CAD *Computer-Aided Design, Volume 21, Issue 5*, 1980
- [42] N. Hansen, S. Muller, P. Koumoutsakos. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es). *Evolutionary Computation 11, 1, 1-18*, 2003.

REFERENCES

- [43] S. Harries. Parametric Design and Hydrodynamic Optimization of Ship Hull Forms. Ph.D. Thesis. *Technische Universität Berlin*, 1998.
- [44] S. Harries, H. Nowacki. Form parameter approach to the design of fair hull shapes. *10th International Conference on Computer Applications in Shipbuilding (ICCAS '99)*, Massachusetts Institute of Technology, Cambridge, MA, USA, 1999.
- [45] S. Harries. Geometric modeling and optimization. *OPTIMISTIC – Optimization in MarineDesign*, Mensch & Buch Verlag, Berlin, pp.115-138, 2003.
- [46] T.H. Havelock. The Theory of Wave Resistance. *Proc. R. Soc. Lond. A* 1932 138, 1932.
- [47] C.M. Hoffmann, K.-J. Kim. Towards valid parametric CAD models. *Computer Aided Design*, 2000.
- [48] M. Hosaka, F. Kimura. Non-four-sided patch expression with control points. *Computer Aided Geometric Design*, 1984.
- [49] G.C. Hsiao, W.L. Wendland. Boundary Integral Equations. *Springer-Verlag Berlin Heidelberg*, 2008.
- [50] T.J.R. Hughes, J.A. Cottrell and Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement, *Computer Methods in Applied Mechanics and Engineering*, 194, 4135-4195, 2005.
- [51] T.J.R. Hughes. Isogeometric analysis: Progress and Challenges. *International Conference on Mathematical Methods for Curves & Surfaces (MMCS 08)*, Tromso, 2008.
- [52] H.C. Kim. Parametric Design of Ship Hull Forms with a Complex Multiple Domain Surface Topology. Ph.D. Thesis. *Technische Universität Berlin*, 2004.
- [53] R. Kress. Linear Integral Equations. *Springer Verlag Berlin*, 1989.
- [54] H. Lackenby. On the Systematic Geometric Variation of Ship Forms. *RINA Transactions*, Vol. 92, 1950.
- [55] J. Letcher. The Geometry of Ships, The Principles of Naval Architecture Series. *The Society of Naval Architects and marine Engineers, New Jersey, J.R. Paulling (ed.)*, 2009.
- [56] E.T.Y. Lee. Choosing nodes in parametric curve interpolation. *Computer-Aided Design*, Volume 21, Issue 6, July–August 1989.
- [57] K. Lee. Principles of CAD/CAM/CAE Systems. *Addison-Wesley Longman Publishing Co., Inc. Boston, MA*, 1999.
- [58] C.T. Loop, T.D. DeRose. A multisided generalization of Bézier Surfaces. *ACM Transactions on Graphics*, 1989.
- [59] Z. Michalewics. Genetic algorithms + data structures = evolutionary programs. *AI series. Springer-Verlag, New York*, 1992.

-
- [60] L.M. Milne-Thomson. Theoretical Hydrodynamics, 5th Edition. *McMillan*, 1968.
- [61] J.N. Newman. Evaluation of the Wave-Resistance Green Function: Part 1 - The Double Integral, *J. Ship Res.*, 31, 2, 79-90, 1987.
- [62] J.N. Newman. Evaluation of the Wave-Resistance Green Function: Part 2 - The Single Integral on the Centerplane, *J. Ship Res.*, 31, 3, 79-90, 1987.
- [63] H. Nowacki, G. Creutz. Ship Lines Creation by Computer - Objectives, Method and Results. *Symposium on Computer Aided Hull Surface Definition, Annapolis, MD, USA*, 1977.
- [64] H. Nowacki, M.I.G. Bloor, B. Oleksiewicz. Computational Geometry for Ships. *World Scientific Publishing, Singapore*, 2005.
- [65] H. Nowacki. Five decades of Computer Aided Ship, to appear in CAD. *Computer-Aided Design, Volume 42, Issue 11*, 2010
- [66] J. Owen, M.S. Bloor. Neutral formats for product data exchange: The current situation. *Computer-Aided Design, Volume 19, Issue 8, October 1987, Pages 436-443*, 1987./
- [67] F. París, J. Cañas. Boundary Element Methods - Fundamentals and Applications. *Oxford University Press*, 1997.
- [68] F.L. Pérez-Arribas, J.A. Suárez-Suárez, L. Fernández-Jambrina. Automatic Surface Modelling of a ship hull. *Computer-Aided Design*, 2006.
- [69] F. Pérez, J.A. Clemente. Constrained design of simple ship hulls with B-Spline surfaces. *Computer-Aided Design*, 2011.
- [70] L.A. Piegl, W. Tiller. The NURBS Book. *Springer-Verlag, London, 2nd Edition*, 1997.
- [71] L.A. Piegl, W. Tiller. Symbolic operators for nurbs. *Computer Aided Design, vol. 29, no. 5, pp. 361-368*, 1997.
- [72] O. Pironneau. Finite Element Methods for Fluids. *John Wiley & Sons*, 1989.
- [73] N. Pla-Garcia, M. Vigo-Anglada, J. Cotrina-Navau. N-sided patches with B-Spline Boundaries. *Computers & Graphics*, 2006.
- [74] T. Poeschl. Detecting Surface Irregularities Using Isophotes. *Computer Aided Geometric Design, Volume 1, Issue 2, November 1984, Pages 163-168*. 1984.
- [75] C.G. Politis, A.-A.I. Ginnis, P.D. Kaklis, K.A. Belibassakis, C. Feurer. An isogeometric BEM for exterior potential-flow problems in the plane. *SIAM/ACM Joint Conference on Geometric & Physical Modeling, San Francisco*, 2009.
- [76] A.M. Reed, H. Nowacki. Interactiv Creation of Fair Ship Lines. *Journal of Computer Aided Ship Design*, 1974.

REFERENCES

- [77] A. Rodríguez, L. Fernández-Jambrina. Programmed Design of Ship Forms. *Computer-Aided Design*, 2012.
- [78] D.F. Rogers, S.G. Satterfield. B-Spline Surfaces for Ship Hull Design. *Computer Graphics*, 1977.
- [79] M.A. Sabin. Non-rectangular surface patches suitable for inclusion in B-spline surface. *Eurographics* 1983.
- [80] G. Skeie. Ship types and design parameters used in WP2. *Technical Report, BRINO917GES081114-1, Det Norske Veritas (DNV)*, 11 pages., 2008.
- [81] T. Takacs, B. Jüttler. Existence of Stiffness Matrix Integrals for Singularly Parameterized Domains in *Computer Methods in Applied Mechanics and Engineering*, Volume 200, Issues 49–52, December 2011.
- [82] T. Takacs, B. Jüttler. H^2 regularity properties of singular parameterizations in isogeometric analysis. *Graphical Models*, Volume 74, Issue 6, November 2012.
- [83] D.W. Taylor. On Ships' Forms Derived by Formulae. *SNAME Transactions*, pp. 243-270, 1905.
- [84] E.C. Tupper. Introduction to Naval Architecture. *Elsevier Science Ltd*, 1996.
- [85] W. Wang, B. Jüttler, D. Zheng, Y. Liu. Computation of the Rotation Minimizing Frame. *ACM Transactions on Graphics*, Volume 27 Issue 1, 2008.
- [86] D.G.M. Watson. Practical Ship Design. *ELSEVIER OCEAN ENGINEERING BOOK SERIES*, Volume 1, 1998.
- [87] J.J. Van Wijk. Bicubic patches for approximating non-rectangular control-point meshes. *Computer Aided Geometric Design*, 1986.
- [88] S. Zhang, A.L. Thomas. Free Form Surface Construction Using Gregory Patches. *The Third International Conference in Central Europe on Computer Graphics and Visualization*, 1995.
- [89] C. Zhou, D. Liu. The Use of Bézier Surfaces in the Design of a Ship Hull. *Computer Applications in the Automation of Shipyard Operation and Ship Design*, 1985.

Index

A

approximation, 2, 3, 5, 10, 11, 20, 36, 48, 50–54, 65–67, 77, 78, 80, 86, 90, 91, 146.

B

B-spline, 4, 6, 33, 61, 62, 80, 83, 86, 149, 153.
curve, 54, 56, 59, 60, 62, 96, 98, 153.
derivative, 154.
rational, 2, 33, 101, 154.
refinement, 2, 11.
surface, 4.
Bézier, 33, 149.
curve, 54, 153.
product, 149, 155.
boundary element method, 3, 10, 19.
boundary integral equation, 2, 10.
boundary ribbon, *see* ribbon.
bow, 12, 15, 16, 89, 93, 98, 101, 104, 106, 113, 116.
breadth, 15, 17, 24, 25.
bulb, 12, 16, 25, 27, 94, 96, 102, 125, 133, 134, 136.

C

Catia™, 3, 12, 19, 20, 29.
compatibility, 4, 61, 67, 70, 72–76, 78, 80, 83, 86, 91, 93, 98, 99, 101, 102, 104–107, 109, 146, 150.
computer-aided design, 1, 10, 142.
container ship, 3, 5, 6, 12, 14, 16, 17, 141, 150.
Coons patch, 4, 6, 20, 21, 29, 33, 70, 73, 81–83, 86, 98, 108–111, 150.
cross product, 55, 156.
cross tangent ribbon, *see* ribbon.
curve loop, 4, 5, 13, 20, 70, 81, 83, 88, 89, 92, 106, 108, 109, 112,

116, 117, 122, 143, 145, 150.

D

degree elevation, 11, 46, 47, 54, 69, 80, 143, 154.
depth, 15, 92, 129, 131.
dimensionless parameter, 13, 24, 26, 27.
discretisation, 2, 3.
disturbance potential, 7–10.
draft, 15, 17, 25, 28.

E

Exciting, 23.

F

finite element method, 1, 3, 10, 11.
flat-of-bottom, 16, 24, 91, 92, 94.
flat-of-side, 16, 91–93, 98.
forward overhang, 16, 25, 93.
frame, 5, 21, 33, 35, 36, 54, 55, 58, 63, 69, 70, 73, 75, 76, 80–82, 89, 91–93, 96, 98, 99, 101, 102, 104–108, 111, 116, 117, 143, 144, 146, 147, 149.
Frenet frame, 55.

G

G^1 continuity, *see* geometric continuity.
geometric continuity, 4, 5, 20, 29, 33, 35, 37, 42, 54, 83, 98, 105, 106, 109, 125.
GoTools™, 141, 142, 144–147.
Green's function, 3, 9.
Green's identity, 2.

H

Hermite
curve, 89, 91, 96, 99.
function, 37.

- interpolation, *see* interpolation, Hermite.
- I**
interpolation, 5, 14, 33–36, 44–46, 48, 54, 90, 95, 98, 99, 104, 106, 141, 144, 146, 147, 149.
Hermite, 37–39, 41, 42, 90, 92, 93, 99, 102, 104, 107.
isogeometric analysis, 2, 5–7, 11, 19, 110, 152.
isophote curve, 22, 125.
- K**
knot insertion, 11, 47, 59, 143, 154.
KRISO container ship, 17.
- L**
length between perpendiculars, 16, 17, 23, 24, 26, 28, 95, 125, 126, 128.
LU decomposition, 143, 146, 147.
- M**
midship, 15, 16, 89, 91, 109, 110.
parallel, 16, 24, 91, 92, 110, 131, 133.
MS Exciting container ship, 17.
- N**
NURBS, *see* B-spline, rational.
- O**
optimisation, 3, 4, 6, 27, 42, 49, 63, 65, 70, 76, 77, 90, 149.
optimisation loop, 3, 6, 12, 19, 29, 149, 152.
- P**
parametric model, 3–6, 12–14, 17, 33, 86, 125, 141, 149.
- Q**
quadratic programming, 48, 65, 75, 77, 147.
- R**
ribbon, 6, 21, 36, 61, 70, 73, 76, 80, 108, 143, 146, 147.
rotation minimizing frame, 62, 65, 76.
rotation minimizing ribbon, 76, 80.
- S**
scalar product, 36, 55, 56, 58, 61, 70, 71, 155.
Series 60 hull, 3.
shaft, 16, 26, 28, 98, 103, 136, 138, 140.
singular geometry, 6, 109–111, 116, 122.
SmoothPM, 6, 143, 147, 149.
stern, 15, 16, 89, 96, 102, 106, 107, 116, 125.
surface model, 4, 19, 35.
- T**
tensor product surface, 4, 5, 10, 33, 109, 155.
transom, 17, 26–28, 98, 99, 104, 106, 121, 136, 138.
twist vector, 4, 62, 72, 74, 76, 78, 80, 82, 83, 86, 150.
- V**
velocity potential, 7.
- W**
waterline, 8, 10, 15, 19, 21, 24, 27, 95, 96, 102, 104, 112.
wave making resistance, 3, 6, 7, 9, 11, 16, 19, 29.
Wigley hull, 3.
- Z**
zero sectional-curvature curve, 150.

