

**Universität Ulm**  
**Fakultät für Mathematik und Naturwissenschaften**



ulm university universität  
**uulm**

**Numerische Optimierung  
von Schiffen mit VSP:  
Parametrische Modellierung  
und Simulation**

Dissertation zur Erlangung des Doktorgrades Dr. rer. nat.  
der Fakultät für Mathematik und Wirtschaftswissenschaften  
der Universität Ulm

vorgelegt von  
Juan Carlos Matutat  
aus Göppingen

Ulm 2012

**Tag der Promotion**  
21. Februar 2013

**Erstgutachter**  
Prof. Dr. Karsten Urban

**Zweitgutachter**  
Prof. Dr. Stefan Funken

**amtierender Dekan**  
Prof. Dr. Paul Wentges



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>4</b>
1.1	Das Projekt SimuVSP . . . . .	5
1.2	Der Voith-Schneider-Propeller . . . . .	6
1.3	Kapitelübersicht . . . . .	8
<b>2</b>	<b>Geometriemodellierung</b>	<b>9</b>
2.1	Mathematische Modelle . . . . .	9
2.2	B-Spline Kurven und NURBS Kurven . . . . .	12
2.3	Effiziente Schnittpunktberechnung von NURBS Kurven . . . . .	16
<b>3</b>	<b>Vernetzung und vmesh</b>	<b>19</b>
3.1	Das Konzept des Vernetzers . . . . .	19
3.2	Ablauf des Vernetzungsprozesses . . . . .	21
3.3	Die Besonderheiten des Vernetzungsverfahrens . . . . .	23
3.4	Cubit . . . . .	25
<b>4</b>	<b>Strömungssimulation</b>	<b>26</b>
4.1	Herleitung der Erhaltungsprinzipien . . . . .	26
4.2	Finite-Volumen-Methode . . . . .	28
4.2.1	Approximation der Oberflächenintegralen . . . . .	29
4.2.2	Approximation der Volumenintegrale . . . . .	29
4.2.3	Interpolations- und Differentiationsmethoden . . . . .	30
4.3	Implementierung der Randwerte . . . . .	31
4.4	Das algebraische Gleichungssystem . . . . .	31
4.5	Strömungslöser Comet . . . . .	31
<b>5</b>	<b>Parametrischer Schiffskörpermodellierer</b>	<b>33</b>
5.1	Was ist PaShiMo? . . . . .	33
5.2	Voraussetzungen an den Modellierer . . . . .	35
5.3	Grundlegendes Mathematisches Modell und das Konzept von PaShiMo	36
5.4	Parametrische Modellierung einer Spantkurve . . . . .	39
5.4.1	Parametrisierung der Teilkurven . . . . .	41
5.5	Designkurven . . . . .	42
5.6	Globale Parameter . . . . .	44

5.7	Parametrisierung der gesamten Schiffsgeometrie . . . . .	44
5.8	Berechnung der Teilkurven . . . . .	45
5.9	Umsetzung der Parametrisierung in PaShiMo . . . . .	48
5.10	Aufbereitung der Geometrie für die Weiterverarbeitung (stl Generierung)	50
5.11	Modellierte Schiffsgeometrien . . . . .	56
<b>6</b>	<b>Simulation und Optimierung</b>	<b>59</b>
6.1	Simulation . . . . .	59
6.2	Optimierung . . . . .	63
6.3	Direkte Suchverfahren . . . . .	65
6.4	Gradientenbasierte Optimierungsverfahren . . . . .	67
6.5	Optimierung auf Netzbasis . . . . .	67
<b>7</b>	<b>Ergebnisse</b>	<b>69</b>
7.1	Beispiele von umgesetzten Schiffskörpermodellen in PaShiMo . . . . .	69
7.1.1	Wigley Schiff . . . . .	69
7.1.2	Einfache Schlepper Geometrie . . . . .	71
7.1.3	Schlepper Geometrie mit Finne . . . . .	72
7.1.4	Schlepper Geometrie mit komplexen Anbauteilen . . . . .	74
7.1.5	STL Triangulierung von Schiffen . . . . .	75
7.2	Strömungssimulation . . . . .	78
7.2.1	Simulation . . . . .	78
7.3	Geometrieoptimierung . . . . .	82
7.3.1	Studie von Designvariationen . . . . .	82
7.3.2	Geometrieoptimierung . . . . .	87
7.4	Netzbasierte Formoptimierung . . . . .	90
<b>8</b>	<b>Ausblick</b>	<b>93</b>

# Abbildungsverzeichnis

1.1	VSP Funktionsweise . . . . .	7
1.2	VSP am Schiff . . . . .	7
2.1	Kubische Splinekurve auf einem Gitter . . . . .	13
2.2	Kubische B-Spline Basisfunktionen und kubische B-Spline Kurve . . . . .	13
3.1	Strömungsquader mit Schiff, innerer und äußerer Schicht . . . . .	20
3.2	Strömungsquader mit komplexer Geometrie, innerer und äußerer Schicht . . . . .	21
3.3	Vernetzungsvorgang . . . . .	22
3.4	Berechnung der Pseudonormalen . . . . .	22
3.5	Blockweiser Aufbau des Gitters . . . . .	24
5.1	Übersicht über PaShiMo . . . . .	34
5.2	Spantkurven bestehend aus Teilkurven verschiedener Regularität . . . . .	38
5.3	Orientierung des Schiffes . . . . .	39
5.4	Teilkurve im begrenzenden Quader . . . . .	40
5.5	Spant bestehend aus verschiedenen Teilkurven . . . . .	41
5.6	Parameter für die Teilkurven . . . . .	42
5.7	Designlinie für den Kielverlauf entlang des Schiffes . . . . .	43
5.8	Grundidee der Triangulierung . . . . .	52
5.9	Beschriftung der Punkte für die Triangulierung . . . . .	53
5.10	Erste Erweiterung der Triangulierung . . . . .	53
5.11	Zweite Erweiterung der Triangulierung . . . . .	54
5.12	Auflösung der Übergänge verschiedener Flächen . . . . .	55
5.13	Auflösung der Übergänge verschiedener Flächen . . . . .	55
5.14	Auflösung der Übergänge verschiedener Flächen . . . . .	56
5.15	Spantmodell für den einfachen Wassertrecker . . . . .	57
5.16	Spantmodell für den Wassertrecker mit Finne . . . . .	58
5.17	Spantmodell für das Modell Wigleykind . . . . .	58
6.1	Visuelles Ergebnis einer Simulation mit verwendetem Rechengitter . . . . .	62
6.2	Visuelles Ergebnis der Simulation mit Strömungslinien . . . . .	62
6.3	Designstudie mit instabilen Ergebnissen . . . . .	64
7.1	Vom Wigleyschiff abgeleitete Geometrie . . . . .	69

7.2	Wigleyschiffgeometrie von der Seite . . . . .	70
7.3	Vom Wigleyschiff abgeleitete Geometrie, Sicht von oben . . . . .	70
7.4	Geometrie eines Schleppers, Sicht von unten auf die Bodenplatte . . . .	71
7.5	Geometrie eines Schleppers mit grober Vordiskretisierung, Sicht auf den Bug des Schiffes . . . . .	71
7.6	Geometrie des Schleppers mit Finne, Ansicht von der Seite . . . . .	72
7.7	Geometrie des Schleppers mit Finne, Ansicht von oben auf das Heck . .	73
7.8	Geometrie des Schleppers mit Finne, Ansicht von unten auf das Heck .	73
7.9	Geometrie des Schleppers mit Finne und Bodenplatte, Ansicht von unten	74
7.10	Experimentelle Geometrie für einen Schlepper mit Finne, Ansicht von unten . . . . .	74
7.11	Gleichmäßig triangulierte Geometrie des Schleppers mit Finne . . . . .	75
7.12	Ungleichmäßig triangulierte Geometrie des Schleppers mit Finne . . . .	76
7.13	Nahaufnahme der gleichmäßig triangulierten Geometrie: Beginn der Bodenplatte . . . . .	76
7.14	Nahaufnahme der gleichmäßig triangulierten Geometrie: zusammenfallende Teilflächen . . . . .	77
7.15	Nahaufnahme der gleichmäßig triangulierten Geometrie: Schanzkleid . .	77
7.16	Pavingnetz für eine Schlepper Geometrie . . . . .	78
7.17	Simulationsergebnisse für die Schlepper Geometrie . . . . .	79
7.18	Vergleich der Simulationsergebnisse . . . . .	80
7.19	Vergleich der generierten Gitter . . . . .	81
7.20	Designvariationen von <i>nurbs_ship_v1_107</i> . . . . .	84
7.21	Vergleich der Simulationsergebnisse für die Variationen . . . . .	85
7.22	Designvariationen von <i>nurbs_ship_v1_108</i> . . . . .	86
7.23	Veranschaulichung der Ergebnisse der zweiten Parameterstudie . . . . .	86
7.24	Erstes Optimierungsproblem in einer Variable . . . . .	87
7.25	Zweites Optimierungsproblem in zwei Variablen . . . . .	88
7.26	Optimierungsverlauf bei Kopplung zweier Parameter . . . . .	89
7.27	Ergebnisse bei Kopplung zweier Parameter . . . . .	90
7.28	Oberflächennetz einer Geometrie bei großer Deformation . . . . .	90
7.29	Verlauf der Formoptimierung . . . . .	91
7.30	Ergebnis der netzbasierten Optimierung: Verformung der Zellen . . . . .	92
7.31	Ergebnis der netzbasierten Optimierung: Vergleich der beiden Oberflächen	92

# Zusammenfassung

Diese Dissertation entstand im Rahmen des Projektes SimuVSP in enger Zusammenarbeit der Universität Ulm und der Firma Voith Turbo Schneider Propulsion in Heidenheim. Das interdisziplinäre Projekt wurde vom Bundesministerium für Wirtschaft und Technologie (BMWi) gefördert. Im Fokus des Projektes stand die numerische Simulation und Optimierung von Schiffskörpergeometrien mit Voith-Schneider-Propeller (VSP). Das Gesamtprojekt bestand aus vier Teilprojekten: Die parametrische Modellierung von Schiffskörpergeometrien, die automatische Generierung von dreidimensionalen Netzen für die Simulation, die Strömungsberechnung und die Optimierung auf Basis der Ergebnisse der Strömungssimulation. Diese Dissertation befasst sich mit der parametrischen Modellierung der Geometrie und deren Simulation und Optimierung.

Es erfolgt zunächst die Beschreibung der mathematischen Grundlagen und der bereits vorhandenen Ansätzen aus der Literatur. Die Entwicklung einer solchen Modellierung erfolgt mit Hinblick auf Schiffsformen von Schiffen mit VSP.

Für die Simulation der Strömung um solche Geometrien wurde die Finite-Volumen-Methode eingesetzt. Diese benötigt eine Zerlegung des Strömungsraumes in kleine Volumina. Es werden die hierzu benötigten Grundlagen und eingesetzten Methoden aus dem Bereich der Gittergenerierung und der Strömungssimulation beschrieben.

Es wird die Entwicklung des parametrischen Modellierungswerkzeuges und die Aufbereitung der generierten Geometrie für die Verwendung in einer Simulationskette beschrieben. Das Modellierungswerkzeug ermöglicht eine modulare Entwicklung der Modellierung von einfachen bis hin zu komplexen Geometrien. Die Modellierung der Geometrie erfolgt über einen Ansatz, welcher die Geometrie in Teilflächen zerlegt und diese einzeln modelliert. Die Steuerung der Erzeugung erfolgt hierbei mit Hilfe von Parametern, welche die Form beschreiben. Die Aufbereitung der Geometrie erfolgt über die Erzeugung eines Oberflächennetzes, welches die Teilflächen und die Knickkanten auflöst. Ansätze für die Modellierung und Oberflächennetzgenerierung solcher Geometrien sind aus der Literatur nicht bekannt.

Die Simulation erfolgt auf Grundlage eines automatisch generierten Volumennetzes um eine solche Geometrie. Das auf Basis des Modellierers, des Vernetzers und der Strömungssimulation erarbeitete Verfahren der Simulation und Optimierung einer Geometrie wird beschrieben. Abschließend erfolgt die Vorstellung der erzielten Ergebnisse aus Modellierung, Simulation und Optimierung.

# Abstract

This dissertation was the result of the project SimuVSP. This project is a cooperation of Ulm University and Voith Turbo Schneider Propulsion in Heidenheim, Germany. It was founded by the Federal Ministry of Economics and Technology and is an interdisciplinary project. Focus of the project is the numerical simulation and optimisation of ship hull geometries with Voith-Schneider-Propeller (VSP). The whole project is divided in four partial projects. Namely the parametric modelling of ship hull geometries, the automated mesh generation, the simulation and the optimisation based on the results of the simulation. This dissertation is covering the parametric modelling and the simulation and optimisation.

In the beginning the fundamentals of mathematical modelling of geometries are summarized. The development of the geometry model is done with respect to ship hull geometries with VSP.

For the simulation of the fluid the finite volume method is used. This method requires the decomposition of the fluid volume in small volumes via a mesh. The methods used for the grid generation and the simulation are described.

The geometric model of the resulting tool and the preparation of the data for the simulation is explained. The tool implements a modular approach for modeling ship hull geometries for even complex geometries. The method divides the whole geometry into several surfaces. The control of the model is based on parameters which describe the form of the geometry. The geometry is passed to the meshing tool as surface grid. This grid is capable of resolving the edges on the ship surface. Approaches for modeling and surface grid generation of such kind of geometries are not known in the literature.

The simulation is done on an automatic generated volume grid based on the surface grid of the geometry. The process of simulating and optimising the ship geometry is described. In the end the results of the modelling, the simulation and the optimisation are shown.

# Kapitel 1

## Einleitung

In der heutigen Zeit nimmt die Verwendung von Rechnern in der Produktentwicklung stetig zu. Die rasend schnelle Entwicklung von Rechnerleistungen sowie die sinkenden Beschaffungskosten für Rechner spielen hierfür eine große Rolle. Aber auch die zunehmende Entwicklung von Softwareprodukten, mit deren Hilfe komplexe Fragestellungen beantwortet werden können, ist ein entscheidender Punkt für diese Entwicklung, da erst durch die Verfügbarkeit derartiger Softwareprodukte die vorhandenen Rechnerkapazitäten genutzt werden können.

Speziell die Entwicklung von Softwareprodukten, welche die parallele Architektur moderner Computerprozessoren ausnutzt, ist Schlüssel für den Erfolg des High Performance Computing (HPC). Unter diesem Begriff wird das Sachgebiet des Rechnens auf Computern mit großer Rechenleistung zusammengefasst.

Leisteten sich noch vor einigen Jahren nur Institute von Universitäten und große Forschungseinrichtungen Rechenzentren, so finden diese Rechenzentren heute auch Verbreitung in kleinen und mittelständischen Firmen.

Dieses wachsende Interesse an Rechenzentren ist begründet in der zunehmenden Bedeutung der Simulation von physikalischen und chemischen Vorgängen. In der Forschung und Entwicklung in Industrie und Wirtschaft dienen Simulationen zum besseren Verständnis von Prozessen in verschiedensten Bereichen, wie z.B. dem Automobilbau, dem Maschinenbau und der chemischen Industrie. Speziell im Schiffbau steht die Berechnung von Strömungen in Wasser und Luft im Mittelpunkt. Bei dieser Berechnung der Strömung werden verschiedenste physikalische Vorgänge simuliert, welche sich mit Hilfe von partiellen Differentialgleichungen beschreiben lassen. Diese wiederum können numerisch mit Hilfe verschiedener Methoden gelöst werden, u.a. mit Finiten-Elementen-Methoden oder Finiten-Volumen-Methoden. Die grundlegende Idee hinter den Finite-Volumen-Methoden ist die Zerlegung des Strömungsraumes in viele kleine endliche Volumina. Eine solche Zerlegung bezeichnen wir als Gitter oder Rechennetz. Abhängig von der zugrundeliegenden Problemstellung wird hierzu ein angepasstes Rechennetz benötigt.

Zur Erzeugung solcher Gitter wird zunächst ein Volumen (Strömungsgebiet) benötigt, in welchem die Strömung simuliert wird. Dieses Volumen ist durch die Geometrie des umströmten Gegenstandes und die beschränkende Umrandung des Strömungsgebiets

gegeben. Aus dessen geometrischer Beschreibung wird mit Hilfe von mathematischen Methoden ein Rechengitter erzeugt, auf welchem die Gleichungen, welche die Strömung beschreiben, gelöst werden.

Für die Optimierung von umströmten Körpern spielt daher die Erzeugung und die Beschreibung der geometrischen Form eine entscheidende Rolle. Vor allem die automatische Erzeugung dieser und die anschließende effiziente automatische Gittergenerierung auf Basis dieser Geometrie stellt ein wichtiges Kettenglied in der Optimierungskette dar. Die Beschreibung einer Geometrie hat auf das zu erwartende Ergebnis einen großen Einfluss, da diese die Formvariation ungewollt einschränken kann.

Die Beschreibung von geometrischen Formen spielt in der Mathematik der Neuzeit eine wichtige Rolle. Durch die zunehmende Verfügbarkeit und Verbreitung von computergestützten Entwurfssystemen (CAD) und die zunehmende Bedeutung computergestützter Fertigungsanlagen (CAM) nimmt das Bedürfnis nach mächtigen Modellierungswerkzeugen weiter zu. Aber auch die Unterhaltungsindustrie bedarf computerbasierter Modellierungsverfahren zur Erzeugung visueller Spezialeffekte, [8].

Die Entwicklung der Beschreibung von Flächen und Formen in den 1960er Jahren wurde motiviert durch die Einführung der Computergrafik. Auch heute noch ist die Computergrafik eine treibende Kraft in der Entwicklung neuer Algorithmen und Formbeschreibungen. Die Möglichkeit, mit Hilfe von Computern modellierte Formen nun auch automatisiert herstellen zu können, liefert die entscheidende Verbindung von Entwurf bzw. Entwicklung und Konstruktion. Durch diese Verbindung spielt die Modellierung eine wichtige Rolle in der Produktentwicklung. Vor allem wegen der Möglichkeit der Simulation wird der Produktanalyse in der Produktentwicklungskette eine größere Bedeutung zugeschrieben.

Damit die Aussage von Simulationen jedoch den Ansprüchen von Entwicklern gerecht wird und an Realitätstreue gewinnt, wird eine ausreichende Analyse und Vorhersage der Fehlergüte benötigt. Speziell in der numerischen Strömungssimulation spielt daher die Anpassung der Gitterfeinheiten eine große Rolle.

Bisherige Ansätze zur Simulation und Optimierung von Produkten bedurften das Eingreifen des Benutzers.

In dieser Arbeit wird auf diesen Punkt der vollautomatischen Simulation eingegangen.

## 1.1 Das Projekt SimuVSP

Das Projekt SimuVSP ist eine enge Kooperation des Instituts für Numerische Mathematik der Universität Ulm mit Voith Turbo Schneider Propulsion in Heidenheim. Es wurde gefördert durch das Bundesministerium für Wirtschaft und Technologie. Ziel des Vorhabens ist die Entwicklung einer Simulationstechnologie zur Optimierung von Schiffskörpern mit Voith-Schneider-Propeller.

Ein Teilvorhaben in diesem Projekt stellt die Optimierung von VSP-relevanten Schiffskörperformen dar. In diesem Rahmen entstand die vorliegende Dissertation.

Das Ziel des Teilvorhabens war die Entwicklung von Simulations- und Optimierungsverfahren für das Gesamtsystem Schiffskörper mit Voith-Schneider-Propeller (VSP).

Mit Hilfe mathematisch fundierter Optimierungsansätze wurde eine Optimierung der Gesamtkonfiguration, bestehend aus Schiffskörper und VSP unter Vorgabe verschiedener schiffsrelevanter Zielfunktionale durchgeführt. Die Ergebnisse dieses Teilvorhabens fließen direkt in die Simulationsmodelle des Partners Voith Turbo Schneider Propulsion ein, so dass eine in sich geschlossene Simulationstechnologie entsteht. Diese wird sowohl für die Produktentwicklung als auch für die Ermittlung vertragsrelevanter Leistungsgrößen des VSP eingesetzt werden.

Die hydrodynamischen Wechselwirkungen zwischen Schiffskörper und Propulsor, insbesondere die Ausbildung des Nachstromfeldes als hydrodynamisches Arbeitsumfeld des Propulsors führen zur Notwendigkeit einer strömungsdynamischen Optimierung der Schiffsform. Neben der ganzheitlichen hydrodynamischen Simulation des Gesamtsystems Schiff-Propulsor kommt somit der Realisierung optimaler Schiffslinien eine große Bedeutung zu.

Ein einzusetzender Optimierungsalgorithmus für Schiffskörper basiert auf gezielten Geometrievariationen unter Berücksichtigung bestimmter vorgegebener Optimierungsziele. Um diese unter der Randbedingung wirtschaftlicher Rechenzeiten auf Großrechnern automatisiert durchführen zu können, müssen geeignete Parameter spezifiziert werden. Diese Parametrisierung stellt das erste Teilproblem dar. Die Entwicklung eines Vernetzers zur Generierung eines geeigneten Elementnetz für den CFD-Löser stellt das zweite Teilproblem dar. Dieser Vernetzer soll insbesondere die Formspezifika von VSP-getriebenen Schiffen berücksichtigen und weitestgehend automatisiert ablaufen. Das dritte Teilproblem ist die Kopplung dieser beiden Werkzeuge mit dem CFD-Löser. Diese Kopplung stellt eine Notwendigkeit dar, da der Optimierungsprozess automatisiert erfolgen soll. Die Entwicklung des Optimierungsverfahrens ist das letzte Teilproblem.

Diesem Projekt sind bereits eine Reihe von anderen Projekten mit Voith Turbo Schneider Propulsion vorausgegangen. Diese haben jeweils zu signifikanten Verbesserungen des Wirkungsgrades des VSP geführt. Ein Projekt befasste sich mit der Optimierung der Flügelwinkelkurve des VSP, ein weiteres mit der Optimierung der Flügelblätter des VSP, [23].

In der Dissertation von M. Hopfensitz, welche ebenfalls innerhalb dieses Projektes entstand, ist der automatische Vernetzer *vmesh* beschrieben, [19].

Im Rahmen dieses Projektes erfolgten ebenfalls Veröffentlichungen, [20].

## 1.2 Der Voith-Schneider-Propeller

Der Voith-Schneider-Propeller, welcher auch Zykloidalpropeller genannt wird, geht auf die Erfindung von Ernst Schneider von 1925 zurück. Er ist ein Vertikalachsrotor bei dem sich die im Wasser stehenden Flügel in einer Kreisbahn um eine vertikale Drehachse bewegen. Diese Drehachse liegt orthogonal zur Hauptanströmung. Zusätzlich führen die Flügel eine Schwingbewegung aus, welche so gesteuert wird, dass ein gerichteter Schub entsteht (Abb. 1.1). Die Richtung des Schub ist variabel um  $360^\circ$  und ermöglicht enge Schiffsmanöver wie z.B. Drehen auf engstem Raum. Ausserdem kann bei konstanter

Drehzahl des Rotors der Betrag und die Richtung des Schubs stufenlos variiert werden. Somit entfällt ein separates Steuerorgan und der VSP ist sowohl Antrieb als auch Steuerungsorgan zugleich, [22].

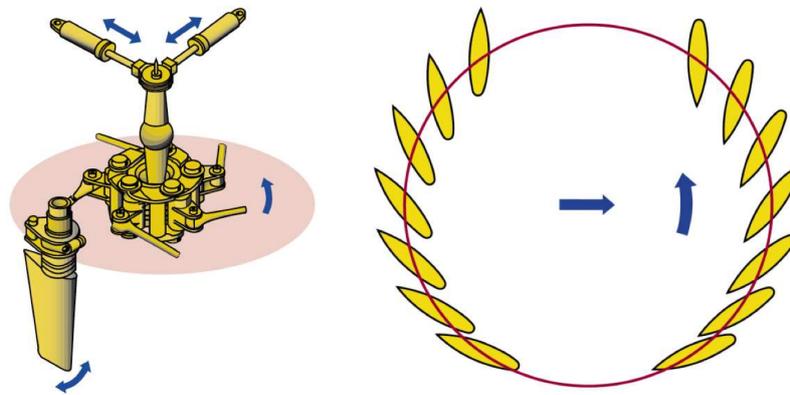


Abbildung 1.1: Rotation eines Flügels (links), Flügelstellung bei der Schuberzeugung (rechts), [2]

Seit über 75 Jahren werden VSP für Spezialschiffe verwendet, welche eine hohe Anforderung an Sicherheit und Manövrierfähigkeit stellen. VSP werden heute u.a. bei Schleppfahrzeugen, Doppelendfähren, Fahrgastschiffen und Schwimmkränen eingesetzt. Diese werden bei Schleppschiffen in der Regel in Bugnähe unterhalb des Schiffeskörpers montiert (Abb. 1.2).



Abbildung 1.2: VSP (links), VSP am Schiffsrumpf (rechts), [45]

Die Funktionsweise des VSP kann interaktiv auf der Homepage von Voith getestet werden, [45].

### 1.3 Kapitelübersicht

In Kapitel 2 wird eine Einführung in die Modellierung von Geometrien gegeben. Es werden Ansätze zur Schiffskörpermodellierung aus der Literatur zusammengefasst. Außerdem wird ein effizienter Algorithmus zur Auswertung von NURBS und B-Spline Kurven beschrieben.

In Kapitel 3 wird das Vernetzungskonzept vorgestellt, auf welchem die automatische Gittergenerierung beruht. Es beinhaltet die Erklärung des Konzepts und des Ablaufs der Vernetzung.

Kapitel 4 beinhaltet die Grundlagen der Strömungsmechanik und stellt die Finite-Volumen-Methode zur Lösung von Strömungsproblemen vor.

In Kapitel 5 wird das Modellierungswerkzeug PaShiMo vorgestellt, welches für die Modellierung von Geometrien umgesetzt wurde. Hierzu wird anhand der Vorstellung des Konzeptes die Funktionsweise der Software dargelegt. Außerdem wird das erstellte Werkzeug zur Oberflächentriangulierung vorgestellt. Es erfolgt ebenfalls eine Vorstellung der umgesetzten Geometrien mit dem Werkzeug PaShiMo.

Kapitel 6 bespricht die entwickelte Simulationmethode und erklärt das hieraus abgeleitete Optimierungsverfahren für die Geometrieoptimierung.

In Kapitel 7 werden die Ergebnisse aus Modellierung, Simulation und Optimierung zusammengefasst.

Abschließend erfolgt ein Ausblick in Kapitel 8.

# Kapitel 2

## Geometriemodellierung

Das rechnerunterstützte Konstruieren ist eine Disziplin, die in vielen Bereichen des Ingenieurwesens nicht mehr wegzudenken ist und zunehmend auch Einfluss auf die Produktion gewinnt. Dieser mathematische Anwendungsbereich wird üblicherweise unter dem Begriff Computer Aided Geometric Design, kurz CAGD, zusammengefasst. Man versteht unter diesem Begriff die Techniken, welche beim Entwurf von Produkten am Rechner, meist aus einem technisch-industriellen Umfeld, Verwendung finden. Produkte, welche auf diese Art und Weise hergestellt werden, basieren auf einer mathematischen Beschreibung der geometrischen Form. Diese erlaubt es, Zeichnungen zu erstellen oder Fertigungsbefehle für computergesteuerte Werkzeugmaschinen zu generieren. Allerdings erfordert der Einsatz des Computers die Entwicklung neuer Methoden zur Beschreibung von Geometrien.

Ziel dieses Kapitels ist die Darlegung einiger grundlegender Methoden, auf welchen sich das in diesem Projekt erarbeitete mathematische Modell stützt. Ausserdem werden die Anforderungen an das Modell dargelegt und eine kurze Übersicht über den aktuellen Stand der Methoden in der Literatur gegeben.

### 2.1 Mathematische Modelle

Ein zentraler Punkt beim rechnergestützten Modellieren ist die Manipulation, die Analyse und die Generierung von geometrischen Objekten. In unserem Anwendungsfall bestehen die Objekte aus Kurven und Flächen. Als problematisch erweist sich die komplexe Topologie, welche berücksichtigt werden muss. In diesem Fall ist notwendig, dass das mathematische Modell eine vollständige Definition des Objektes erlaubt, da dieses die Grundlage für nachfolgende Manipulationen bildet und Verwendung in der Simulation findet.

Es gibt verschiedene Arten, Kurven und Flächen mathematisch darzustellen. Diese lassen sich folgendermaßen klassifizieren:

#### 1. Darstellung über Funktionen

Bei diesem Ansatz werden skalarwertige Funktionen  $f : \Omega \rightarrow \mathbb{R}$  über einem

Parametergebiet  $\Omega \subseteq \mathbb{R}^2$  betrachtet. Mit deren Hilfe können Flächen

$$\{(x, y, f(x, y)) \in \mathbb{R}^3 : (x, y) \in \Omega\} \subseteq \mathbb{R}^3 \quad (2.1.1)$$

beschrieben werden.

Diese Art der Flächendarstellung birgt den Nachteil, per Definition eine starke Einschränkung bei der Modellierung zu besitzen. Es lassen sich beispielsweise geschlossene Flächen, wie Kugeln, im dreidimensionalen Raum nicht in obiger Form angeben.

Dennoch finden diese im Schiffbau Verwendung. Ein verbreitetes Beispiel ist das Wigleyschiff, welches über die folgende Funktion definiert ist, [27, 47],

$$y(x, z) = 2 \cdot B \cdot \frac{x}{L} \left(1 - \frac{x}{L}\right) \cdot \left(1 - \left(\frac{z}{D}\right)^2\right), \quad (2.1.2)$$

wobei  $\frac{x}{L} \in [0, 1]$  und  $\frac{z}{D} \in [-1, 0]$ . Das Schiff liegt in einem kartesischen Koordinatensystem, in welchem die  $x$ - $y$ -Ebene die Konstruktionswasserlinie enthält. Der Parameter  $L$  beschreibt die Länge, der Parameter  $B$  die Breite und  $D$  den Tiefgang des Schiffes. In der Literatur werden die Parameter  $L$ ,  $B$  und  $D$  oft wie folgt gewählt

$$B = 12, \quad D = 7.5, \quad L = 120. \quad (2.1.3)$$

## 2. Parametrische Darstellung

Bei diesem Ansatz werden Kurven oder Flächen mit Hilfe vektorwertiger Abbildungen dargestellt,

$$x : \Omega \rightarrow \mathbb{R}^3, \quad u \mapsto x(u) := (x_1(u), x_2(u), x_3(u))^T, \quad (2.1.4)$$

wobei  $\Omega \subset \mathbb{R}$  oder  $\Omega \subset \mathbb{R}^2$  ein Parametergebiet ist. Typischerweise haben diese Abbildung folgende Gestalt

$$x(u) = \sum_{i=0}^n c_i B_i(u), \quad (2.1.5)$$

mit  $c_i \in \mathbb{R}^3$  für  $i = 0, \dots, n$  und  $B_i : \Omega \rightarrow \mathbb{R}$ ,  $i = 0, \dots, n$ . Ein Beispiel für diese Darstellung sind B-Splines und NURBS (**N**on-**U**niform **R**ational **B**-**S**plines), [36]. Aufgrund der hohen Flexibilität und Freiheit dieser Art der Modellierung hat diese sich in der CAGD weitestgehend durchgesetzt und ist in nahezu jedem Modellierungswerkzeug zu finden. Im Schiffbau wurde dieser Ansatz schon früh in den achtziger und neunziger Jahren aufgegriffen und in kommerziellen Systemen zur Modellierung von Schiffsgeometrien integriert.

## 3. Implizite Darstellung

Bei dieser Art der Darstellung wird eine Kurve oder Fläche implizit über eine Nullstellenmenge einer skalarwertige Funktion  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  dargestellt

$$\{x \in \mathbb{R}^n : f(x) = 0\}. \quad (2.1.6)$$

Für  $n = 3$  ist durch diese Beschreibung eine zweidimensionale Mannigfaltigkeit gegeben. Dank der Darstellung (2.1.6) kann durch eine einfache Punktauswertung festgestellt werden, ob ein Punkt über oder unter der Fläche liegt. Auskunft hierüber gibt das Vorzeichen der Funktion  $f$ . Oftmals wird für die Funktion  $f$  ein Polynom ausgewählt.

Das Generieren und Beschreiben von Schiffslinien, welche die Geometrie beschreiben, war schon immer eine Herausforderung für Ingenieure. Der schwedische Schiffbauer Frederik Henrik Chapman führte die Planung von Schiffen auf Basis von wissenschaftlichen und mathematischen Grundlagen ein, [9]. Seine Entwürfe basierten auf Linienrissen aus drei orthogonalen Ansichten. Die Linienrisse ergeben sich aus dem Schnitt der Geometrie mit drei zueinander orthogonalen Ebenen. Traditionell wurden diese Linien von Hand mit einem Holzspan (wooden spline) angepasst, [25, 26]. Die Konsistenz der drei orthogonalen Ansichten musste dabei per Hand erreicht und angepasst werden. Mit der Einführung des Rechners in den Schiffsentwurf vereinfachte sich dieser Schritt und entfiel später vollständig. Auch wenn das manuelle Ausgleichen der Linienrisse weggefallen ist, so ist die Vorgehensweise mit der Darstellung über Linienrisse bis heute erhalten geblieben, [5, 46].

Die heutigen Techniken im Schiffbau lassen sich wie folgt kategorisieren:

### 1. Klassischer Linienplanentwurf

Bei diesem Ansatz werden die Schiffslinien ohne das Ziel einer mathematischen Oberflächenbeschreibung modelliert. Diese Vorgehensweise orientiert sich stark an dem klassischen Vorgehen aus der Zeit vor der Einführung von CAGD Systemen.

### 2. Freiformoberflächenmodellierung

Parametrische Oberflächenbeschreibungen wie B-Splines, werden benutzt um Geometrien zu erzeugen. Das Modellieren ist hierbei meist ein interaktiver Prozess. Die Linienrisse werden letztlich mit Hilfe von Computerprogrammen aus der Oberflächenbeschreibung generiert.

### 3. Formparametermodellierung

Kurven und Flächen werden hierbei auf Basis von Parametern generiert. Hierbei können diese Parameter physikalische Eigenschaften oder geometrische Eigenschaften der Hülle beschreiben.

Der in dieser Arbeit verwendete Ansatz bildet eine Mischung aus der Freiformoberflächenmodellierung und der Formparametermodellierung. Die ersten Systeme zur Modellierung von Schiffsrümpfen mit Hilfe von Freiformoberflächen geht auf die neunziger und achtziger Jahre zurück. Diese basieren auf B-Splines und NURBS Kurven- und Flächendarstellungen. Zu Systemen die B-Splines verwenden gehören z.B. AutoShip, Fast Ship und Rhino, [1, 13, 38]. Alle diese Systeme haben das Ziel, die Hülle geschlossen als B-Spline oder NURBS darzustellen.

Andere Systeme wie NAPA (Naval Architecture Package) hingegen basieren auf einer Modellierung der Schiffshülle aus einzelnen Patches, [31]. Diese einzelnen Patches werden von einem Netz aus Kurven modelliert und ergeben zusammengesetzt die

vollständige Geometrie. Bestimmte Segmente des Schiffes werden hierbei unabhängig voneinander modelliert. Auf ähnliche Weise versucht das Entwurfssystem 4 (E4), welches auf Arbeiten von Söding und Rabin aufbaut, mit Hilfe von einzelnen Patches die Geometrie zu modellieren, [17].

Der Ansatz, die Schiffshülle über beschreibende Formparameter zu modellieren, geht auf die Siebziger Jahre zurück. Erste Arbeiten zu diesem Thema liegen von Kwik und Buczkowski vor, [6, 26]. Diese Arbeiten benutzten einfache Kurven und versuchen diese mit Hilfe von Formparametern zu beschreiben. Die Arbeit von Reed und Nowacki hingegen erweitert diesen Ansatz, [37]. Zum Einen erhöhen sie die Freiheiten der Formbeschreibung durch Zuhilfenahme von Polynomen höherer Ordnung. Zum anderen führten sie in den Prozess der Liniengenerierung computergestützte Methoden ein. Diese Methoden sahen automatische Approximationen der Kurven vor, um Oszillationen in der Oberflächenmodellierung zu vermeiden, welche durch das Verwenden von Polynomen höherer Ordnung im Modell entstehen. In späteren Arbeiten benutzten Munchmeyer, Schubert und Nowacki B-Splines an Stelle von Polynomen höherer Ordnung um Oszillationen zu vermeiden, [30].

## 2.2 B-Spline Kurven und NURBS Kurven

Wie aus dem vorherigen Abschnitt hervorgeht, spielen B-Splines und NURBS in dem Arbeitsumfeld der Geometriemodellierung eine bedeutende Rolle. In diesem Abschnitt werden einige mathematischen Hintergründe aufgezeigt und wichtige Eigenschaften und Definitionen zusammengefasst. Der Vollständigkeit wegen wird an dieser Stelle auf die Arbeiten von Piegl und deBoor verwiesen, [10, 36].

Eine B-Spline Kurve ist definiert als stetig differenzierbare und stückweise polynomiale Kurve. Diese ergibt sich als Linearkombination von  $n$  Kontrollpunkten

$$\mathbf{P}_0, \dots, \mathbf{P}_{n-1} \in \mathbb{R}^d$$

mit korrespondierenden Funktionen  $N_{i,p}(\cdot)$ , welche B-Splines (Basis Splines) vom Grad  $p$  genannt werden. Die sich ergebende Kurve besitzt folgende Gestalt

$$s(u) = \sum_{i=0}^{n-1} \mathbf{P}_i N_{i,p}(u) \quad (2.2.1)$$

und ist eine parametrische Kurve mit Parameter  $u$ , siehe Abb. 2.1. Es handelt sich um eine eindimensionale Kurve im  $\mathbb{R}^d$ . Die Definition der B-Splines erfolgt über ein Gitter

$$\Delta = \{u_0, \dots, u_m\}$$

mit diskreten Gitterpunkten  $u_0, \dots, u_m \in \mathbb{R}$  aus denen mit Hilfe von Def. 2.2.1 die B-Spline Basisfunktionen rekursiv berechnet werden. Eine B-Spline Basisfunktion ist eine stetig differenzierbare und stückweise polynomiale Funktion, [10].

Ein Grund für die Beliebtheit von B-Spline Basen in der CAGD ist deren Lokalität. Da die B-Spline Funktionen lokal linear unabhängig sind und einen lokalen Träger

$$\text{supp } N_{i,p} = [u_i, u_{i+p+1}) \quad (2.2.2)$$

besitzen, haben die B-Spline Basen einen lokalen Einfluss auf den Kurvenverlauf. Somit ist es möglich, mit den einzelnen Kontrollpunkten den lokalen Verlauf der Kurve zu steuern.

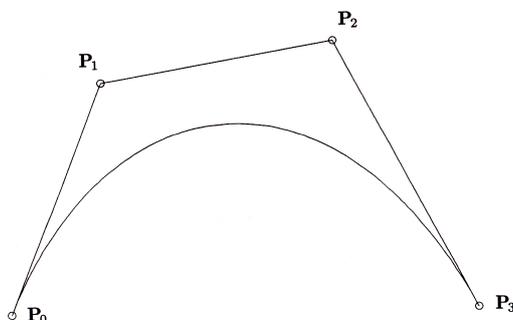


Abbildung 2.1: Kubische Splinekurve auf dem Gitter  $\Delta = \{0, 0, 0, 0, 1, 1, 1, 1\}$  mit den Kontrollpunkten  $P_0, \dots, P_3$ , [36]

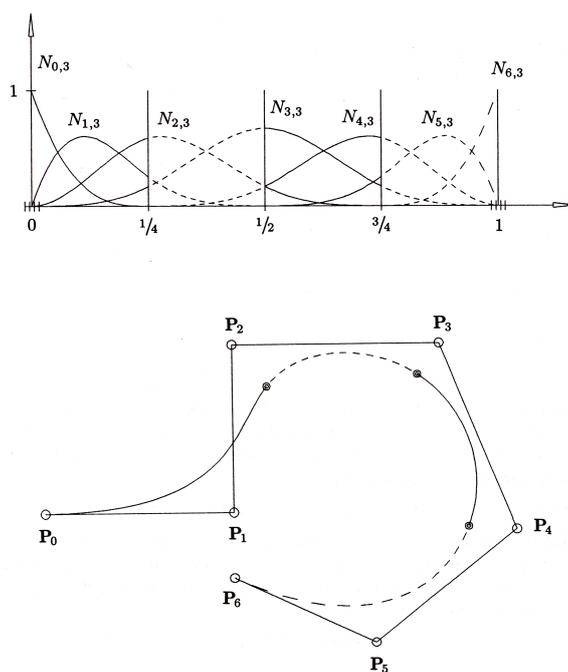


Abbildung 2.2: Kubische B-Spline Basisfunktionen und eine kubische Kurve, welche diese benutzt, [36]

Diese Eigenschaft der Lokalität wird mit Hilfe von Abb. 2.2 näher erläutert. Auf dieser Abbildung sind die kubischen Basisfunktionen auf dem Gitter

$$\Delta = \{0, 0, 0, 0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1, 1, 1, 1\} \tag{2.2.3}$$

zu sehen. Eine Basisfunktion ist eine aus kubischen Polynomen stückweise zusammengesetzte Funktion. Nicht alle Basisfunktionen sind auf dem gesamten Definitionsbereich  $[0, 1]$  der Kurve ungleich Null und haben somit nicht auf den gesamten Kurvenverlauf Einfluss. Durch die abschnittsweise Betrachtung der B-Spline Kurve erkennt man welche Basisfunktionen auf welchen Teil der Kurve Einfluss haben. Dementsprechend können die Kontrollpunkte verwendet werden, um Kurvenabschnitte einzeln zu kontrollieren bzw. modellieren.

Nach dieser einführenden anschaulichen Betrachtung von B-Splines folgt nun die mathematische Definition eines B-Spline, welche in der Arbeit verwendet wird.

**Definition 2.2.1** Sei  $\Delta = \{u_0, \dots, u_m\}$  mit  $m \in \mathbb{N}$  ein Gitter mit den Gitterpunkten  $u_i$  auf dem Intervall  $I = [u_0, u_m] \subset \mathbb{R}$ . Die  $i$ -te B-Spline Basisfunktion ( $i = 0, \dots, m - 1$ ) vom Grad  $p$  wird mit  $N_{i,p}(u)$  bezeichnet und ist definiert als

$$N_{i,0}(u) = \begin{cases} 1, & u_i \leq u < u_{i+1}, \\ 0, & \text{sonst,} \end{cases} \quad (2.2.4)$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u). \quad (2.2.5)$$

Aufgrund obiger Definition gilt  $N_{i,p}(u) = 0$ , falls  $u$  ausserhalb von  $[u_i, u_{i+p+1}]$  liegt. Mit Hilfe von Def. 2.2.1 wird eine B-Spline Kurve wie folgt definiert.

**Definition 2.2.2** Sei  $p \in \mathbb{N}$  und  $\Delta$  ein Gitter auf  $[a, b] \subset \mathbb{R}$ . Eine B-Spline Kurve vom Grad  $p$  ist definiert als

$$s(u) = \sum_{i=0}^n P_i N_{i,p}(u), \quad a \leq u \leq b, \quad (2.2.6)$$

mit den Kontrollpunkten  $P_i \in \mathbb{R}^d$  und den B-Spline Basisfunktionen vom Grad  $p$  über dem Gitter  $\Delta$ .

Es ist bekannt, dass die Basisfunktionen  $N_{i,p}(u)$  nicht negativ sind und dass gilt

$$\sum_{j=i-p}^i N_{j,p}(u) = 1 \quad (2.2.7)$$

für alle  $u \in [u_i, u_{i+1})$ , vgl. [36].

Aus der nicht negativität der Basisfunktionen und (2.2.7) ergibt sich die Eigenschaft, dass eine B-Spline Kurve stets in der konvexen Hülle der Kontrollpunkte  $P_i$  ( $i = 0, \dots, n$ ) verläuft. Diese Eigenschaft besagt, dass jeder Wert der B-Spline Kurve  $s(u)$  im Wertebereich durch eine Konvexkombination der Kontrollpunkte  $P_i$  ( $i = 0, \dots, n$ ) gegeben ist.

In dieser Arbeit werden die Oberflächen der Geometrie mit Hilfe von B-Spline Kurven

auf einem nicht-uniformen Gitter modelliert. Diese werden auch NUBS (**N**on-**U**niform **B**asic **S**pline) genannt. Die Designkurven, welche später eingeführt werden, werden mit Hilfe von NURBS (**N**on-**U**niform **R**ational **B**asic **S**pline) modelliert. NURBS werden ebenfalls mit Hilfe von Def. 2.2.1 definiert.

**Definition 2.2.3** Sei  $p \in \mathbb{N}$  und  $\Delta$  ein Gitter auf  $[a, b] \subset \mathbb{R}$ . Eine NURBS Kurve vom Grad  $p$  ist definiert als

$$n(u) = \frac{\sum_{i=0}^n \omega_i P_i N_{i,p}(u)}{\sum_{i=0}^n \omega_i N_{i,p}(u)}, \quad a \leq u \leq b, \quad (2.2.8)$$

mit den Kontrollpunkten  $P_i \in \mathbb{R}^d$ , den Gewichten  $\omega_i$  ( $i = 0, \dots, n$ ) und den B-Spline Basisfunktionen vom Grad  $p$  über dem Gitter  $\Delta$ .

Im Folgenden wollen wir (2.2.8) umschreiben in die Form

$$n(u) = \sum_{i=0}^n P_i R_{i,p}(u) \quad (2.2.9)$$

mit den rationalen Basisfunktionen

$$R_{i,p}(u) = \frac{\omega_i N_{i,p}(u)}{\sum_{i=0}^n \omega_i N_{i,p}(u)}. \quad (2.2.10)$$

Aus dieser Darstellung ist leicht ersichtlich, dass B-Spline Kurven ein Spezialfall von NURBS Kurven sind ( $\omega_i = 1 \forall i$ ). Zusammenfassend besitzen NURBS und somit auch B-Spline Kurven die folgenden geometrischen Eigenschaften, vgl. [36]:

- Der Anfangs- und Endkontrollpunkt wird von der Kurve interpoliert:

$$P_0 = s(a) = n(a) \quad \text{and} \quad P_n = s(b) = n(b). \quad (2.2.11)$$

- Affine Invarianz: Eine affine Transformation der Kurve erfolgt über die Anwendung der Transformation auf die Kontrollpunkte der Kurve. In der Computergrafik wird sogar die Tatsache ausgenutzt, dass NURBS Kurven invariant gegenüber perspektivischen Projektionen sind.

$$As(u) + v = \sum_{i=0}^m (AP_i + v) N_{i,p}(u) \quad (2.2.12)$$

für eine B-Spline Kurve

$$s(u) : \mathbb{R} \rightarrow \mathbb{R}^d, \quad (2.2.13)$$

mit Kontrollpunkten  $P_i \in \mathbb{R}^d$ , eine lineare Abbildung  $A \in \mathbb{R}^{d \times d}$  und einem Vektor  $v \in \mathbb{R}^d$ . Für eine NURBS Kurve

$$n(u) : \mathbb{R} \rightarrow \mathbb{R}^d \quad (2.2.14)$$

gilt

$$An(u) + v = \frac{\sum_{i=0}^n \omega_i (AP_i + v) N_{i,p}(u)}{\sum_{i=0}^n \omega_i N_{i,p}(u)}. \quad (2.2.15)$$

- Konvexe-Hülle-Eigenschaft: Falls  $u \in [u_i, u_{i+1})$  dann liegt die NURBS Kurve in der konvexen Hülle der Kontrollpunkte  $P_{i-p}, \dots, P_i$ .

$$s(u)|_{[u_i, u_{i+1})} \in \text{conv}\{P_{i-p}, \dots, P_i\} \quad (2.2.16)$$

und

$$n(u)|_{[u_i, u_{i+1})} \in \text{conv}\{P_{i-p}, \dots, P_i\} \quad (2.2.17)$$

mit

$$\text{conv}(X) = \left\{ \sum_{i=0}^n \alpha_i \cdot x_i : x_i \in X, \sum_{i=0}^n \alpha_i = 1, \alpha_i \geq 0, n \in \mathbb{N} \right\}. \quad (2.2.18)$$

- NURBS Kurven sind unendlich oft differenzierbar im Inneren eines Knotenintervalls und  $p - k$  mal differenzierbar am Knoten der Vielfachheit  $k$ .
- Keine Ebene besitzt mehr Schnittpunkte mit der NURBS bzw. B-Spline Kurve als mit dem zugehörigen Kontrollpolygonzug.
- Lokalität: Eine Änderung des Kontrollpunkt  $P_i$  oder des Gewicht  $\omega_i$  beeinflusst die Kurve nur im Intervall  $[u_i, u_{i+p+1})$

$$\text{supp } N_{i,p} = [u_i, u_{i+p+1}). \quad (2.2.19)$$

## 2.3 Effiziente Schnittpunktberechnung von NURBS Kurven

In diesem Abschnitt wird die Herleitung eines effizienten Algorithmus zur Berechnung von Schnittpunkten von NURBS Kurven betrachtet. Die Berechnung von Schnitten spielt eine wichtige Rolle in der Modellierung von Schiffsgeometrien, da diese für die Darstellung der Geometrie als Linienriss benötigt wird.

Im Folgenden besitze die NURBS Kurve folgende Gestalt:

$$n(u) = \begin{pmatrix} n_1(u) \\ n_2(u) \\ n_3(u) \end{pmatrix} \in \mathbb{R}^3, \quad u \in \mathbb{R}, \quad (2.3.1)$$

mit Parameter  $u$  über dem Gitter

$$\Delta = (u_0, u_1, \dots, u_m) \quad , \quad m \in \mathbb{N}, \quad (2.3.2)$$

den Kontrollpunkten

$$\mathbf{P}^{(i)} = \begin{pmatrix} P_1^{(i)} \\ P_2^{(i)} \\ P_3^{(i)} \end{pmatrix} \in \mathbb{R}^3 \quad , \quad i \in \{1, \dots, n\} \quad (2.3.3)$$

und den Gewichten

$$\omega_i, \quad i \in \{1, \dots, n\}. \quad (2.3.4)$$

Es gilt für die Dimension  $n$ , den Grad  $p$  und die Kardinalität des Gitters  $m + 1$

$$n = m + p. \quad (2.3.5)$$

Gesucht wird der Kurvenpunkt, welcher in einer Komponenten  $j \in \{1, 2, 3\}$  des NURBS (2.3.1) einen bestimmten gegebenen Wert  $x_j \in \mathbb{R}$  annimmt. Da dieser Punkt nicht eindeutig definiert ist, kann es mehrere solcher Schnittpunkte geben. Anschaulich handelt es sich hierbei um den Schnitt der Kurve mit einer Hyperebene des  $\mathbb{R}^3$ .

Ziel ist somit das Lösen der Gleichung

$$n_j(u^*) = x_j \quad (2.3.6)$$

für  $u^* \in [u_0, u_m]$  und eine Komponente  $j$  des NURBS.

Eine Komponente des NURBS (2.3.1) besitzt folgende Gestalt

$$n_j(u) = \frac{\sum_{i=1}^n \omega_i P_j^{(i)} N_{i,p}(u)}{\sum_{i=1}^n \omega_i N_{i,p}(u)}, \quad j = 1, 2, 3, \quad (2.3.7)$$

wobei  $N_{i,p}(\cdot)$  die  $i$ -te Basisfunktion vom Grad  $p$  über dem Gitter  $\Delta$  ist.

Nach Einsetzen von (2.3.7) in (2.3.6) und Umstellen der Gleichung ergibt sich

$$\sum_{i=1}^n \omega_i (P_j^{(i)} - x_j) N_{i,p}(u^*) = 0. \quad (2.3.8)$$

Aus dieser Darstellung folgt, dass die Berechnung Schnittpunktes äquivalent ist zum Lösen eines Schnittproblems eines Splines mit den Koeffizienten

$$\tilde{c}_i := \omega_i (P_j^{(i)} - x_j), \quad i = 1, \dots, n. \quad (2.3.9)$$

Zur Lösung der Gleichung (2.3.6) wird daher die Gleichung

$$\sum_{i=1}^n \tilde{c}_i N_{i,p}(u^*) = 0 \quad (2.3.10)$$

und die B-Spline Kurve

$$s(u) = \sum_{i=1}^n \tilde{c}_i N_{i,p}(u) \quad (2.3.11)$$

betrachtet.

Der Kontrollpolygonzug dieser Kurve ist definiert über die Punkte

$$c_i := \begin{pmatrix} u_i^* \\ \tilde{c}_i \end{pmatrix} = \begin{pmatrix} \frac{1}{p} \sum_{k=1}^p u_{i+k} \\ \omega_i (P_j^{(i)} - x_j) \end{pmatrix}. \quad (2.3.12)$$

Nun wird der kleinste Index  $i_0 \in \{1, \dots, n\}$  bestimmt, für welchen gilt

$$\tilde{c}_{i_0} \cdot \tilde{c}_{i_0+1} \leq 0. \quad (2.3.13)$$

Dieser Index existiert, da die NURBS Kurve und somit der Kontrollpolygonzug mindestens einen Schnittpunkt mit der Achse besitzt und dem Mittelwertsatz.

Der Schnittpunkt des Polygonzugs mit der Hyperebene ist

$$u_0^* = -\tilde{c}_{i_0} \left( \frac{u_{i_0+1}^* - u_{i_0}^*}{\tilde{c}_{i_0+1} - \tilde{c}_{i_0}} \right) + u_{i_0}^*. \quad (2.3.14)$$

Mit Hilfe dieser ersten Approximation werden die Newton-Iterationen durchgeführt

$$u_{n+1}^* = u_n^* - \frac{s(u_n^*)}{s'(u_n^*)} = u_n^* - \frac{\sum_{i=1}^n \tilde{c}_i N_{i,p}(u_n^*)}{\sum_{i=1}^n \tilde{c}_i N'_{i,p}(u_n^*)}, \quad (2.3.15)$$

bis die gewünschte Genauigkeit

$$\varepsilon = \|n_j(u^*) - x_j\| \quad (2.3.16)$$

erreicht ist.

Obiges Verfahren stellt ein einfaches und schnelles zweistufiges Verfahren zur Bestimmung von Schnittpunkten einer NURBS Kurve mit einer Hyperebene des  $\mathbb{R}^3$  dar. Dieser Ansatz lässt sich auch zum Lösen von Schnittproblemen für B-Spline Kurven verwenden.

In [36] ist eine allgemeinere Version dieses Problems zu finden. Dieses ist in der Literatur als *point inversion* bekannt. Das Problem besteht darin, den korrespondierenden Parameter zu einem Punkt  $P$  zu finden, von welchem bekannt ist, dass dieser auf einer NURBS Kurve  $n(u)$  liegt.

Dieses Problem kann jedoch weiter verallgemeinert aufgefasst werden als das Auffinden des nächsten Kurvenpunktes zu einem Punkt  $P$ , welcher nicht zwangsweise auf der Kurve liegen muss. Es gilt in diesem Fall das folgende Problem zu lösen

$$\min_u \|n(u) - P\|. \quad (2.3.17)$$

Dieses Problem ist äquivalent mit der Nullstellenbestimmung von

$$f(u) = n'(u) \cdot (n(u) - P) \quad (2.3.18)$$

und kann ebenfalls mit Newton-Iterationen mit geeigneten Abbruchbedingungen schnell und effizient gelöst werden, [36].

# Kapitel 3

## Vernetzung und *vmesh*

In dem Projekt der Simulation von Schiffen mit VSP ist die Strömungssimulation ein Teilproblem. Für die Simulation der Umströmung der Geometrie werden Verfahren eingesetzt welche eine Zerlegung des Strömungsgebietes benötigen. Insbesondere für die numerische Strömungssimulation wird ein Rechengitter des Strömungsraums benötigt. Dieses Rechengitter wird mit Hilfe des Werkzeugs *vmesh* aus der Geometriebeschreibung des Schiffsrumpfes erzeugt. Die Erzeugung erfolgt hierbei automatisiert und kann mit Hilfe von Parametern gesteuert werden. Da mit den Ergebnissen der Simulation Optimierungen durchgeführt werden, müssen die Ergebnisse der Strömungssimulationen die Veränderung der Geometrie abbilden können.

In diesem Kapitel wird die hierbei entwickelte Methode zur automatisierten Generierung eines Rechengitters zusammenfassend vorgestellt. Eine genauere Beschreibung des automatischen Vernetzers *vmesh* ist in der Dissertation von M. Hopfensitz zu finden, [19].

### 3.1 Das Konzept des Vernetzers

Ziel des zweiten Arbeitspaketes im Projekt SimuVSP war die Erstellung eines automatischen Vernetzers, welcher ein Elementenetz für den CFD-Solver generiert und dabei die Formspezifika von Schiffen mit VSP berücksichtigt. Das Strömungsgebiet ist bei diesem Konzept durch einen Quader, in welchem sich die Geometrie des Schiffes befindet, gegeben. Das eigentliche Strömungsgebiet ist das Gebiet zwischen der Quaderberandung und dem Schiffskörper. Es gilt dieses Gebiet mit Kontrollvolumen zu füllen.

Damit dieses Netz auch vernünftige Ergebnisse bei der numerischen Simulation liefert, müssen gewisse Voraussetzungen an das Volumennetz gestellt werden. Zu Beginn des Projektes wurden daher existierende Möglichkeiten der Vernetzung evaluiert und diskutiert. Der Entschluss fiel zugunsten eines hybriden Netzes, welches aus Hexaeder- und Tetraederelemente besteht.

Das hybride Netz bestehend aus verschiedenen Schichten und erfüllt die folgenden Voraussetzungen:

- Die Zellen besitzen nach Möglichkeit die Form von Heaxaedern.

- Die Existenz einer äquidistanten Schicht, welche dem Rumpf direkt anliegt und die Strömung um den Rumpf ausreichend auflöst.
- Die Gitterlinien der rumpfnahen Schicht verlaufen möglichst orthogonal bzw. parallel zum Rumpf.
- Für jeden Knick auf der Schiffsaussenhaut wird es eine Gitterlinie geben, welche entlang dieses Knickes verläuft.
- Die Gitterlinien entlang der Wasseroberfläche verlaufen möglichst horizontal.
- Hängende Knoten sind nicht erlaubt.
- Das Gesamtgitter besitzt Quaderform.

Das resultierende Schichtenmodell besteht aus drei Schichten. Die erste rumpfnahen Schicht wird als innere Schicht bezeichnet. Die daran anschliessende mittlere Schicht dient als Übergang zur äusseren Schicht, welche an die Quaderberandung angrenzt. Jede dieser drei Schichten besitzt eine andere Charakteristika und beinhaltet immer nur Zellen des gleichen Typs.

Die innere Schicht liegt dem Schiffsrumpf direkt an und besteht aus Hexaedern. Die äussere Schicht besteht ebenfalls aus Hexaedern und deckt den größten Bereich des Strömungsquaders ab. Die mittlere Schicht besteht aus Tetraedern und passt sich somit den geometrischen Formen des Schiffes optimal an. Dies gibt dem Vernetzer mehr Freiheiten beim Anpassen komplexer Geometrien. Insgesamt ergibt sich das folgende Bild für das Vernetzungskonzept, siehe Abb. 3.1.

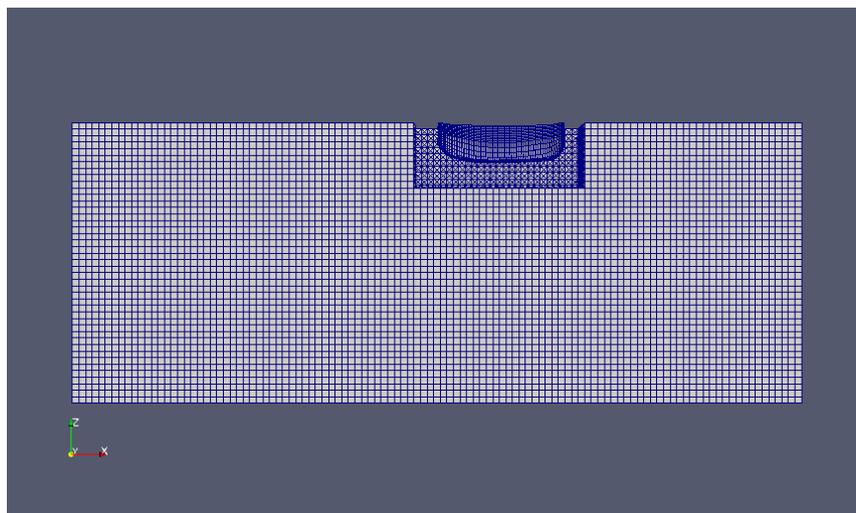


Abbildung 3.1: Strömungsquader mit Schiff, innerer und äußerer Schicht

In Abb. 3.1 ist der Strömungsquader mit den verschiedenen Schichten zu sehen. Die äußere Schicht bestehend aus Hexaedern nimmt das größte Volumen ein. Direkt am

Schiff ist die innere Schicht, bestehend aus Hexaedern, zu finden. Lediglich die mittlere Schicht ist nicht abgebildet. Diese besitzt die Aufgabe, den Übergang zwischen innere und äußere Schicht auszugleichen.

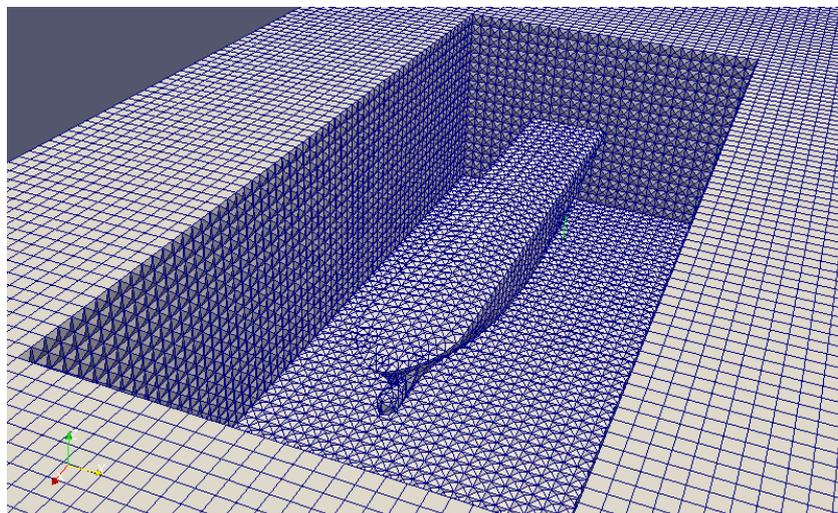


Abbildung 3.2: Strömungsquader mit komplexer Geometrie, innerer und äusserer Schicht

Auf Abb. 3.2 ist das zu vernetzende Volumen für die mittlere Schicht abgebildet. Aufgrund der komplexeren Geometrie des Schiffes ist offensichtlich, wieso eine Vernetzung des mittleren Volumens mit Hexaedern schwer möglich ist und auf Tetraedern zurückgegriffen wird. Es ist auf diesem Bild zu sehen, wie auf den Zellwänden der Hexaedern Pyramiden aufgesetzt werden, um das Entstehen hängender Knoten zu vermeiden.

## 3.2 Ablauf des Vernetzungsprozesses

In diesem Abschnitt wird auf den Prozess des Vernetzens eingegangen, siehe Abb. 3.3. Es können verschiedene Geometrien mit dem Werkzeug *vmesh* vernetzt werden. Die Geometrien müssen in den Formaten STL oder ACIS vorliegen. Geometrien im STL Format liegen als mit Dreiecken triangulierte Geometrien vor und bestehen zum Teil aus mehreren Oberflächen. Diese werden zunächst vernäht, damit eine geschlossene Geometriebeschreibung vorliegt. Anschliessend wird mit Hilfe eines Paving Algorithmus ein Oberflächennetz generiert, [3, 39]. Hierzu wird die Software Cubit eingesetzt. Diese Software berücksichtigt zudem, dass bei zusammengesetzten Oberflächen die Gitterlinien stets entlang der Naht zweier Teilflächen verläuft, [19]. Auf Basis dieses Oberflächennetzes wird mit Hilfe eines Offset-Algorithmus die innere Schicht generiert. Diese besteht ausschließlich aus Hexaedern-Elementen. Der Offset-Algorithmus ist eine eigene Implementierung. Zur Berechnung der Hexaedern wird mit einem Schema für jeden Knoten des Oberflächennetzes eine Pseudonormale berechnet, siehe Abb. 3.4. Diese dient als Richtungsvektor für die Grenzschichtzeugung. Die Erzeugung dieser Grenzschicht wird mit Parametern gesteuert, [19].

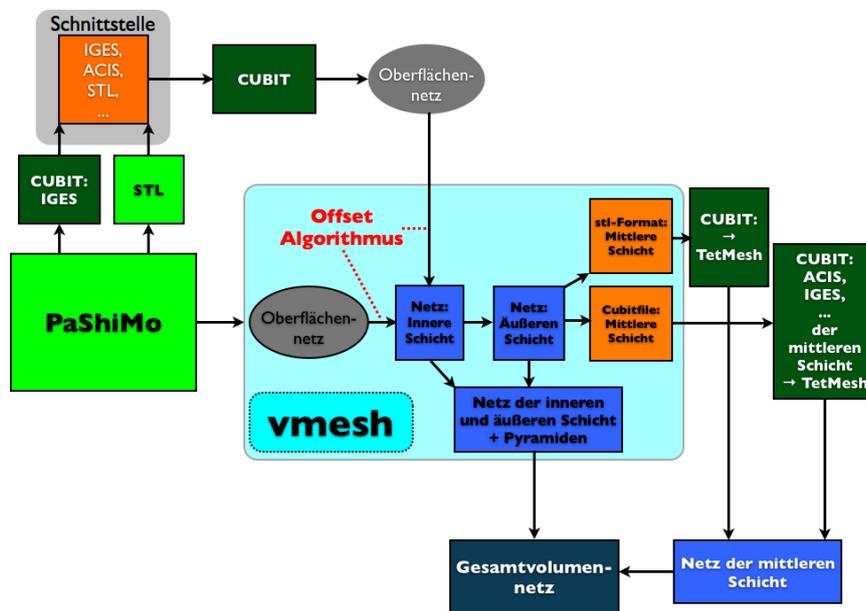


Abbildung 3.3: Vernetzungsvorgang, [19]

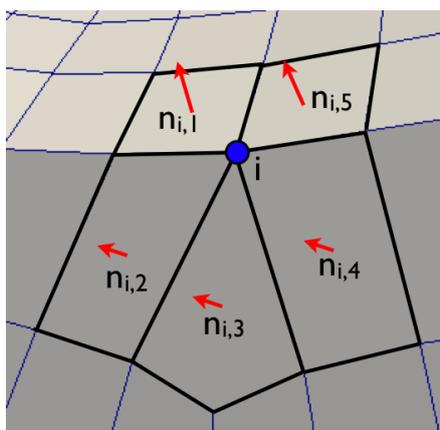


Abbildung 3.4: Berechnung der Pseudonormalen, [19]

Im Anschluss wird die äußere Schicht generiert. Diese besteht ebenfalls nur aus Hexaedern. Auf die innere und äußere Schicht werden nun Pyramiden aufgebracht, welche das Entstehen von hängenden Knoten im Übergang zu der mittleren Schicht vermeidet, siehe Abb. 3.2. Das sich nun ergebende Volumen für die mittlere Schicht wird mit Hilfe von Cubit mit Tetraedern gefüllt.

Diese Tetraeder werden mit dem TetMesher von Cubit generiert, welcher auf dem Delauney-Algorithmus basiert, [42]. Grundlage für diese Vernetzung bildet die Beschreibung der Grenzschicht als STL Geometrie. Hierbei verwendet man die Tatsache, dass die Grenzflächen der mittleren Schicht ausschliesslich aus Dreiecken bestehen. Dies erreicht man durch das Anbringen von Pyramiden auf die Zellwände der Hexader. Unter einer Delauney-Triangulierung versteht man eine Triangulierung, welche die Delauney Eigenschaft besitzt, [4, 44]. In einem letzten Schritt werden die einzelnen Schichten zu einem einzigen Volumennetz zusammengeführt.

An dieser Stelle ist zu bemerken, dass eine Erzeugung des Netzes sowohl als Vollmodell als auch als Halbmodell umgesetzt werden kann. Ein Nachteil bei der Verwendung des Vollmodells ist die Entstehung eines deutlich größere Netz. Aus dieser Tatsache ergeben sich auch die deutlich längeren Rechenzeiten für die Simulation mit CFD Methoden eines solchen Vollmodells. Allerdings besitzt das Vollmodell den Vorteil, die Karmannsche Wirbelstraße hinter dem Schiff aufzulösen, [19].

Die Steuerung des kompletten Vernetzungsvorgangs erfolgt über ein Script welches die einzelnen Programme aufruft. Die Einstellungen für Cubit und für die Vernetzungsparameter erfolgt ebenfalls über dieses Script. Der Offset-Algorithmus und der Algorithmus zur Erzeugung der äusseren Schicht wurden in ein Programm in C++ umgesetzt und werden über interne Parameter gesteuert. Der genaue Aufbau des Scripts für die Steuerung des Vernetzungsprozesses sowie die Funktionsweise der umgesetzten Algorithmen ist in [19] beschrieben.

### 3.3 Die Besonderheiten des Vernetzungsverfahrens

Grundlegend bei dieser Art der Vernetzung ist die Verwendung eines Oberflächen-gitters als Grundlage. Auf Basis dieses Ausgangsnetzes läßt sich ein glattes Oberflächennetz erzeugen, welches für komplexe Geometrien hochwertige Grenzschichten ermöglicht. Die Berechnung der sogenannten Pseudonormalen wird erst durch diesen Schritt ermöglicht. Versuche, mit kommerziellen Vernetzern für gleiche Geometrien automatisiert derartige Grenzschichten zu generieren, scheiterten oder waren nur mit viel Handarbeit möglich, [19]. Dabei ist die Existenz einer solchen hochwertigen Grenzschicht notwendig, um gute Ergebnisse der Strömungssimulation zu bekommen. Ausserdem untersuchte man mit Hilfe von Gitterkriterien die innere Schicht, um die Qualität nachzuweisen. Man betrachtete dabei unter anderem das Diagonalenverhältnis und die skalierte Jacobi-Determinante.

Damit die Strömung um das Schiff herum gut abgebildet wird und das Gitter möglichst automatisiert generiert werden kann, entschied man sich für einen blockweisen Aufbau der äusseren Schicht, siehe Abb. 3.5. Die Verwendung von hexaeder Zellen ergab sich

hierbei aufgrund der horizontalen Anströmrichtung.

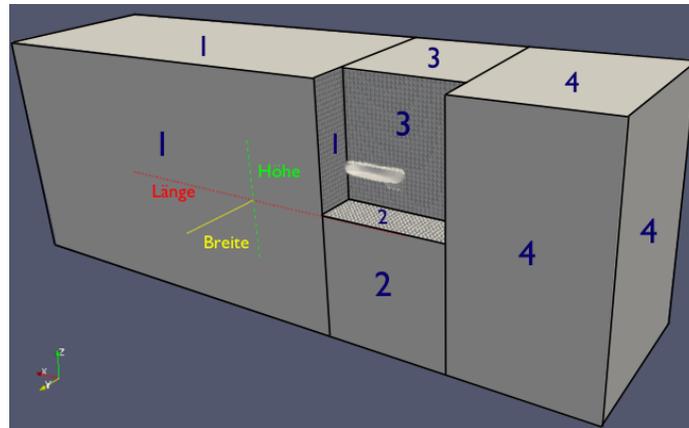


Abbildung 3.5: Blockweiser Aufbau des Gitters, [19]

Da die Vernetzung des Zwischenraums von innerer und äusserer Schicht ebenfalls automatisiert erfolgt, entschied man sich für eine Delauney-Triangulierung. Diese verspricht bei einem automatisierten Vorgehen brauchbare Ergebnisse zu liefern und tolerant gegenüber schwierigen Vernetzungssituationen zu sein, welche sich bei komplexen Geometrien ergeben.

Insgesamt ergab sich somit folgender Ablauf:

1. Einlesen einer triangulierten Geometrie als STL Geometrie.
2. Erzeugung eines geglätteten Oberflächennetzes auf Basis eines Paving-Algorithmus durch Cubit.
3. Berechnung der Pseudonormalen des vorherigen Oberflächennetzes in jedem Knoten und Generierung der Grenzschicht mit *vmesh*.
4. Erstellung der äusseren Schicht auf Basis der Dimension der Geometrie und des Strömungsraumes mit *vmesh*.
5. Generierung von Übergangselementen in Form von Pyramiden auf den vorherigen Schichten.
6. Erzeugung einer Volumenbeschreibung des mittleren Volumens als STL Geometrie.
7. Erzeugung einer Delauney-Triangulierung des mittleren Volumens mit Cubit.
8. Vereinigung aller drei Schichten zu einem Gitternetz.

Durch die Verwendung einer STL Geometrie als Grundlage der Vernetzung wird der Aufwand der Schnittstellenimplementierung zwischen dem geometrischen Modellierer und dem Vernetzer gering gehalten.

## 3.4 Cubit

Zum Vernetzen wurde das kommerzielle Vernetzungstool CUBIT eingesetzt. Diese Software wurde von Sandia National Laboratories entwickelt, [40]. In dieser wurden sowohl zwei- als auch dreidimensionale Vernetzungsalgorithmen umgesetzt. Zur Oberflächenvernetzung stehen Algorithmen für Vernetzung mit Dreiecken und Vierecken zur Verfügung. Außerdem werden verschiedenste Vernetzer für dreidimensionale Volumen angeboten. Diese lassen sich mit Hilfe von Skripten automatisieren.

Für die automatische Vernetzung wurden ein Paving-Algorithmus für die Oberflächenvernetzung und ein Tetraedervernetzer (TetMesher) aus dieser Software verwendet.

# Kapitel 4

## Strömungssimulation

Für die Optimierung von Schiffskörpergeometrien ist es notwendig die Umströmung um die Geometrie zu beschreiben. Die hierfür benötigten Grundlagen und Methoden werden in diesem Kapitel zusammengefasst. Die Grundlage bilden die Arbeiten von Ferziger und Peric, [14].

Zur Berechnung der Strömung mit numerischen Methoden werden partielle Differentialgleichungen und geeignete Anfangs- und Randbedingungen benötigt. Dabei werden die physikalischen Vorgänge in Fluiden mit Hilfe von partiellen Differentialgleichungen beschrieben. Der Beschreibung der Strömung liegen die physikalischen Erhaltungsprinzipien für Impuls und Masse zu Grunde, aus welchen die Massen- und Impulserhaltungsgleichung hergeleitet werden. Diese Gleichungen können abhängig von der betrachteten Strömung durch zusätzliche Erhaltungsgleichungen ergänzt werden. Allerdings liefern diese Grundgleichungen nicht genügend Informationen, um eine eindeutige Lösung zu liefern. Daher werden passende Anfangs- und Randbedingungen benötigt, um die Eindeutigkeit der Lösung zu garantieren.

### 4.1 Herleitung der Erhaltungsprinzipien

Die Erhaltungsprinzipien können über die Betrachtung einer gegebenen Menge Materie und ihrer *extensiven* Eigenschaften hergeleitet werden. Diese betrachtete Menge Materie wird Kontrollmasse (KM) genannt. Unter der extensiven Eigenschaft einer Menge von Materie versteht man Eigenschaften, welche abhängig von der Menge der betrachteten Materie sind, wie die Masse, der Impuls und die Energie. Hingegen verstehen wir unter den *intensiven* Eigenschaften diejenigen Eigenschaften, welche unabhängig von der Menge der betrachteten Materie sind. Als Beispiel sind Dichte und Geschwindigkeit zu nennen.

In der Dynamik ist die Vorgehensweise zur Beschreibung der Bewegung von Festkörpern über die Betrachtung der Kontrollmasse verbreitet. Jedoch ist es bei Fluidströmungen, wie sie hier betrachtet werden, schwierig, ein solche festgelegte Fluidmenge zu verfolgen. Daher wird die Strömung in einem vorgegeben Raumvolumen untersucht. Dieses Volumen wird Kontrollvolumen (KV) genannt. Von dieser Bezeichnung leitet sich der Begriff Kontrollvolumenmethode für die daraus resultierende Untersuchungsmethode

ab.

In dieser Untersuchungsmethode werden hauptsächlich die extensive Eigenschaften Masse und Impuls betrachtet. Hierbei bedienen wir uns der Tatsache, dass das Erhaltungsgesetz einer extensiven Eigenschaft die Änderungsrate der Menge dieser Eigenschaft in der Kontrollmasse ins Verhältnis zu äusseren Einwirkungen setzt. Da in Strömungen Masse weder erzeugt noch vernichtet wird, ergibt sich folgende Erhaltungsgleichung für die Masse:

$$\frac{dm}{dt} = 0, \quad (4.1.1)$$

wobei  $m$  die Masse der betrachteten Kontrollmasse und  $t$  die Zeit bezeichnet. Der Impuls der Kontrollmasse wiederum kann durch die Wirkung von Kräften geändert werden und wird durch das zweite Newtonsche Bewegungsgesetz beschrieben:

$$\frac{d(mv)}{dt} = \mathbf{f} \quad (4.1.2)$$

Hierbei steht  $v$  für die Geschwindigkeit und  $\mathbf{f}$  für die Kräfte welche auf die Kontrollmasse wirken.

Es erfolgt nun die Umformulierung dieser Gesetze in eine Kontrollvolumenform. Hierbei werden als Variablen die instensiven statt die extensiven Eigenschaften verwendet. Es bezeichne  $\phi$  eine beliebige intensive Erhaltungseigenschaft (Massenerhaltung:  $\phi = 1$ , Impulserhaltung:  $\phi = v$ ). Dann lässt sich die entsprechende extensive Eigenschaft  $\Phi$  folgendermaßen beschreiben:

$$\Phi = \int_{V_{KM}} \rho \phi dV, \quad (4.1.3)$$

wobei  $V_{KM}$  das Volumen der Kontrollmasse bezeichnet.

Mit Hilfe dieser Definition erhält man die Kontrollvolumengleichung, welche ebenfalls als Reynolds-Transporttheorem bezeichnet wird, [43]:

$$\frac{d}{dt} \int_{V_{KM}} \rho \phi dV = \frac{d}{dt} \int_{V_{KV}} \rho \phi dV + \int_{S_{KV}} \rho \phi (v - v_b) \cdot n dS, \quad (4.1.4)$$

wobei  $V_{KV}$  das KV-Volumen,  $S_{KV}$  die Oberfläche des Kontrollvolumens und  $n$  der nach außen gerichtete Einheitsvektor senkrecht zu  $S_{KV}$  ist. Des weiteren bezeichnet  $v$  die Fluidgeschwindigkeit und  $v_b$  die Geschwindigkeit mit der sich die Oberfläche des KV bewegt. Diese Gleichung beschreibt die Änderungsrate der Menge der extensiven Eigenschaft in der Kontrollmasse als die Summe aus der Änderungsrate der Menge dieser Eigenschaft im KV und dem Nettozufluss durch die KV-Oberfläche. Dieser Nettozufluss wird hervorgerufen durch die Fluidbewegung relativ zur KV-Oberfläche. Dieser Term, welcher diesen Vorgang beschreibt, wird als konvektiver Fluss bezeichnet, [35].

Durch Setzung von  $\phi = 1$  in (4.1.4) erhält man die Integralform der Erhaltungsgleichung der Masse, welche auch als Kontinuitätsgleichung bezeichnet wird:

$$\frac{\partial}{\partial t} \int_V \rho dV + \int_S \rho v \cdot n dS = 0, \quad (4.1.5)$$

wobei der Einfachheit das Kontrollvolumen als fest angenommen wird und  $V$  das Volumen des KV und  $S$  dessen Oberfläche darstellt.

Mit  $\phi = v$  in (4.1.4) und (4.1.2) lässt sich die Integralform der Impulserhaltungsgleichung herleiten, [35],

$$\frac{\partial}{\partial t} \int_V \rho \phi dV + \int_S \rho v v \cdot ndS = \int_S T \cdot ndS + \int_V \rho b dV, \quad (4.1.6)$$

wobei  $T$  den Spannungstensor des newtonschen Fluid bezeichnet und  $b$  die Körperkräfte pro Masseneinheit erfasst. Hierzu analog erhält man die Erhaltungsgleichung für eine skalare Variable  $\phi$ :

$$\frac{\partial}{\partial t} \int_V \rho \phi dV + \int_S \rho \phi v \cdot ndS = f_\phi, \quad (4.1.7)$$

wobei  $f_\phi$  den Transport von  $\phi$  durch alle Mechanismen außer Konvektion sowie jeglicher Quellen oder Senken des Skalars bezeichnet. In ruhenden sowie stehenden Fluiden ist diffuser Transport immer vorhanden und wird oft mit Gradienten-Approximation beschrieben. Zu diesen Gradienten-Approximationen zählt z.B. das Fick-Gesetz zur Massendiffusion

$$f_\phi^d = \int_S \Gamma \nabla \phi \cdot ndS, \quad (4.1.8)$$

wobei  $\Gamma$  den Diffusionskoeffizienten für  $\phi$  bezeichnet. Mit (4.1.8) und (4.1.7) folgt die Integralform der generischen Erhaltungsgleichung

$$\frac{\partial}{\partial t} \int_V \rho \phi dV + \int_S \rho \phi v \cdot ndS = \int_S \Gamma \nabla \phi \cdot ndS + \int_V q_\phi dV, \quad (4.1.9)$$

wobei  $q_\phi$  die Quellen und Senken von  $\phi$  bezeichnet.

In der numerischen Strömungsmechanik werden die Kontinuitätsgleichung und die Impulserhaltungsgleichung als Navier-Stokes-Gleichungen bezeichnet.

## 4.2 Finite-Volumen-Methode

Die Grundlage der Finite-Volumen-Methode bilden die Erhaltungsgleichungen in Integralform aus dem vorherigen Abschnitt. Zur Vereinfachung werden zeitunabhängige Probleme betrachtet, so dass der instationäre Teil vernachlässigt wird. Somit besitzt die generische Erhaltungsgleichung für eine Größe  $\phi$  die Form

$$\int_S \rho \phi v \cdot ndS = \int_S \Gamma \nabla \phi \cdot ndS + \int_V q_\phi dV. \quad (4.2.1)$$

In einem ersten Schritt erfolgt die Zerlegung des Strömungsraumes, welcher auch als Lösungsgebiet bezeichnet wird, in endlich viele Kontrollvolumen. Die Zerlegung erfolgt mit Hilfe eines Gitters. Die Rechenknoten bilden die Schwerpunkte der einzelnen KVs. Die integrale Erhaltungsgleichung (4.2.1) gilt sowohl für jedes KV, als auch für das gesamte Lösungsgebiet. Durch Aufsummieren der Gleichungen aller KVs erhält man die

Erhaltungsgleichung für das gesamte Lösungsgebiet. Durch diese Tatsache ist die physikalisch geforderte globale Konservativität garantiert und fest in der Methode eingebaut. Es ergeben sich somit Bilanzgleichungen für jedes Kontrollvolumen, welche durch numerische Approximation der Volumen- und Oberflächenintegrale der Erhaltungsgleichung (4.2.1) algebraische Gleichungen liefern. Die Auswertung der Variablenwerte auf der KV-Oberfläche erfolgt im Schwerpunkt der jeweiligen KV-Oberfläche durch Interpolation. Die Berechnung der Oberflächen- und Volumenintegrale erfolgt mit Hilfe geeigneter Quadraturformeln, [35].

### 4.2.1 Approximation der Oberflächenintegralen

Der Nettofluss durch den KV-Rand ist gleich der Summe der Integrale über die Seiten  $S_k$  des KV-Volumens. Dadurch ergibt sich für diesen:

$$\int_S f dS = \sum_k \int_{S_k} f dS, \quad (4.2.2)$$

wobei  $f$  die Komponente des konvektiven ( $\rho\phi v \cdot n$ ) oder des diffusen ( $\Gamma\nabla\phi \cdot n$ ) Flussvektor in Richtung der Normalen zur KV-Seite bezeichnet. Das Geschwindigkeitsfeld und die Fluideigenschaften können als bekannt angenommen werden. Somit ist einzig  $\phi$  unbekannt.

Zur Berechnung des Flächenintegrals in (4.2.4) ist es notwendig den Integranden  $f$  auf der Oberfläche  $S_k$  zu kennen. Dies ist jedoch nicht der Fall, da nur die Knotenwerte von  $\phi$  im KV-Zentrum berechnet werden. Daher ist eine Approximation notwendig. Die einfachste Approximation stellt die Mittelpunkregel dar, welche das Integral als Produkt aus dem Integranden im Schwerpunkt der KV-Seite  $f_k$  und der Fläche der KV-Seite  $A_{S_k}$  annähert:

$$F_k = \int_{S_k} f dS \approx f_k A_{S_k} \quad (4.2.3)$$

Der Wert von  $f$  im Integrationspunkt muss hierbei durch Interpolation gewonnen werden.

### 4.2.2 Approximation der Volumenintegrale

Die Approximation der Volumenintegrale kann ebenfalls über die Mittelpunkregel erfolgen. Diese stellt die einfachste Approximation dar und approximiert das Volumenintegral durch das Produkt aus dem Wert des Integranden im KV-Zentrum und dem KV-Volumen  $\Delta V$ :

$$Q_P = \int_V q dV = \bar{q} \Delta V \approx q_P \Delta V, \quad (4.2.4)$$

wobei  $q_P$  den Wert von  $q$  im KV-Zentrum und  $\bar{q}$  den Mittelwert von  $q$  im KV bezeichnet.

### 4.2.3 Interpolations- und Differentiationsmethoden

Da für die Approximation der Integrale Variablenwerte in anderen Punkten als den Rechenpunkten (KV-Zentren) benötigt werden, müssen diese mit Hilfe von geeigneten Approximationen gewonnen werden.

Zur Berechnung der konvektiven und diffusiven Flüsse wird der Wert von  $\phi$  und die Ableitung von  $\phi$  in Richtung senkrecht zur KV-Seite auf der KV-Oberfläche benötigt. Diese müssen durch die Knotenwerte im KV-Zentrum mit Hilfe von Interpolation ausgedrückt werden. Es gibt zahlreiche Methoden, dies zu erreichen. An dieser Stelle werden daher nur einige weit verbreitete Möglichkeiten vorgestellt.

#### Aufwind-Interpolation

Hierbei erfolgt die Approximation von  $\phi_k$  durch den Wert von  $\phi$  im KV-Zentrum stromauf des betrachteten KV. Diese Approximation wird als *Auffwinddifferenz* bezeichnet, obwohl es sich nicht wirklich um eine Differenz handelt. Es ist eine Interpolation der Form:

$$\phi_k = \begin{cases} \phi_P & \text{falls } (v \cdot n)_k > 0 \\ \phi_K & \text{falls } (v \cdot n)_k < 0 \end{cases} \quad (4.2.5)$$

Dies ist die einzige Approximation, welche die Beschränktheitsannahme erfüllt und somit nie oszillierende Lösungen liefert. Jedoch ist diese *numerisch diffusiv*, d.h. sie bringt erhebliche künstliche Diffusion in die Lösung ein, [35].

#### Lineare Interpolation

Dies ist eine weitere einfache Approximation für den Wert im Mittelpunkt der KV-Seite. Es handelt sich um die lineare Interpolation zwischen zwei benachbarten Rechenknoten (KV-Zentren). Es gilt:

$$\phi_k = \phi_K \lambda_k + \phi_P (1 - \lambda_k) \quad (4.2.6)$$

mit einem Interpolationsfaktor

$$\lambda_k = \frac{x_k - x_P}{x_K - x_P} \quad (4.2.7)$$

Diese Interpolation hat eine Genauigkeit 2.Ordnung, welche mit Hilfe der Taylor-Reihen-Entwicklung gezeigt werden kann, [35]. Die Lösungen, welche mit dieser Approximation gewonnen werden, zeigen im Vergleich zur Auffwinddifferenz Oszillationen. Die Approximation des Gradienten unter Annahme eines linearen Profils erfolgt über

$$\left( \frac{\partial \phi}{\partial x} \right)_k \approx \frac{\phi_k - \phi_P}{x_K - x_P} \quad (4.2.8)$$

Auch diese Approximation hat eine Genauigkeit 2.Ordnung, [35].

### Gemischte Verfahren

Ein besonderes Verfahren, welches die Vorteile der Beschränktheit der Aufwindifferenzen mit der Genauigkeit höherer Ordnung der linearen Interpolation verbindet, ist das Blended-Diskretisierungsschema, [21]. Dabei handelt es sich um eine Linearkombination beider obiger Verfahren:

$$\phi_\lambda = (1 - \lambda)\phi_{UD} + \lambda\phi_{CD}, \quad (4.2.9)$$

wobei  $\lambda$ ,  $0 \leq \lambda \leq 1$  als Blendingfaktor bezeichnet wird,  $\phi_{UD}$  der Approximierte Wert mit den Aufwindifferenzen und  $\phi_{CD}$  den mit linearer Interpolation berechneten Wert bezeichnet.

## 4.3 Implementierung der Randwerte

Die Flüsse durch die KV-Seiten, welche am Rande des Lösungsgebiets liegen, bedürfen einer besonderen Betrachtung. Entweder müssen diese Randwerte bekannt sein oder als Kombination der unbekannt inneren Werte und den bekannten Randdaten ausgedrückt werden. Da es keine Rechenpunkte außerhalb des Randes gibt, beruhen diese Approximationen auf einseitigen Differenzen oder Extrapolation.

Daher werden normalerweise konvektive Flüsse an den Einstromrändern vorgegeben. Sonderfälle stellen undurchlässige Wände oder Symmetrieebenen dar. An diesen sind die konvektiven Flüsse gleich null.

An Ausstromrändern werden meist Aufwind-Approximationen verwendet, da normalerweise die Ableitungen aller Variablen in Strömungsrichtung als gleich null angenommen werden.

In unserer Anwendung wird von diffusen Flüssen an Wänden abgesehen.

## 4.4 Das algebraische Gleichungssystem

Die algebraische Gleichung für ein KV ergibt sich durch Aufsummierung aller Flusapproximationen und Quellterme. Diese Gleichungen setzen hierbei die Variablenwerte im KV-Zentrum in Relation zu den Werten in benachbarten KVs. Die Anzahl der Gleichungen entspricht der Anzahl der Unbekannten und ist gleich der Anzahl der KVs. Aus dieser Tatsache ergibt sich die gute Kondition des Systems.

Das Gleichungssystem für das gesamte Lösungsgebiet besitzt Matrixform. Die Matrixstruktur hängt von der Gitterart ab und ist am einfachsten für strukturierte Gitter mit vierseitigen oder sechsseitigen KVs. Die Lösung eines solchen algebraischen Gleichungssystems erfolgt mit iterativen Methoden.

## 4.5 Strömungslöser Comet

In diesem Projekt wurde der Strömungslöser Comet (Comet Mechanics Engineering Tool) eingesetzt. Dieses Programm führt CFD Berechnung auf Basis der Finiten-

Volumen-Methode durch. Es wurde in der Version 2.360.Z in diesem Projekt eingesetzt. Die Durchführung einer Berechnung erfolgt in zwei Schritten. In einem ersten Schritt wird über den Preprozessor das Gitter eingelesen und die Simulation konfiguriert. Hierbei werden Randbedingungen gesetzt und Diskretisierungsschemata und Lösungsmethoden für die Gleichungssysteme ausgewählt. Es gibt Parameter für

- das Gitter,
- die Fluideigenschaft,
- die Gleichungslöser,
- die Diskretisierungsschemata,
- die Randbedingungen
- und die Parallelisierung.

Abschließend erstellt dieser Preprozessor die Konfiguration.

Im Anschluss ermöglicht das Usercoding die Ergänzung des Programms um eigene Routinen, so dass zusätzliche Daten während der Berechnung generiert und ausgegeben werden können.

Die Berechnung des Berechnungsfalls übernimmt das Berechnungsprogramm *Comet* nach manuellem Start. Die Berechnung kann entweder automatisch und manuell abgebrochen werden.

Der Postprozessor ermöglicht nach Abschluss der Berechnung eine Visualisierung der Ergebnisse und einen Export der Daten in andere Formate für weitere Analysen.

Die Eingabe der Daten für die Konfiguration über den Preprozessor und die Analyse mit Hilfe des Postprozessors erfolgt über eine Shell oder über ein Skript. Die Benutzung von *Comet* ist in [21] beschrieben.

# Kapitel 5

## Entwicklung einer geeigneten Schiffkörperparametrisierung (PaShiMo)

In diesem Kapitel wird das Vorgehen bei der Schiffkörperparametrisierung vorgestellt. Das Ergebnis ist ein computergestütztes Werkzeug mit dem Namen PaShiMo (**P**arametric **S**hip **M**odeler), welches im Rahmen des Forschungsprojekts SimuVSP entstand. Ziel dieses Kapitels ist die Vorstellung der theoretischen Hintergründe der mathematischen Modellierung der Schiffkörpergeometrie sowie die Beschreibung der Funktionsweise des entwickelten Werkzeugs PaShiMo.

### 5.1 Was ist PaShiMo?

Im Rahmen des Projektes SimuVSP sollen Schiffkörper mit VSP hinsichtlich ihrer Strömungsverluste analysiert und optimiert werden. Dies geschieht in enger Kooperation mit Voith Turbo Schneider Propulsion. Dieses Gesamtproblem der Simulation und Optimierung teilt sich wie bereits beschrieben in vier Teilprobleme auf: die parametrische Modellierung der Schiffkörpergeometrie, die automatische Netzgenerierung für die erzeugten Geometrien, die Strömungsberechnung und die Optimierung.

Hierbei ist die Aufgabe von PaShiMo die Erzeugung von zulässigen Schiffkörpergeometrien auf Basis von Parametern. Ziel war die Umsetzung eines parametrischen Modellierers, welcher automatisiert eine Schiffkörpergeometrie auf Basis gegebener Parameterwerte generiert. Diese Geometrie wird von dem automatischen Vernetzer weiterverarbeitet. Dieser Vernetzer vernetzt anschließend automatisch das Gebiet zwischen der Geometrie und einer umgebenden Box, welche um das Schiff gelegt wird. Für diesen Schritt wird eine lückenlose Geometriebeschreibung des Schiffes benötigt.

Mit dem Wissen der Mitarbeiter von Voith Turbo Schneider Propulsion, vor allem mit Dipl.-Ing. Michael Palm, wurden Kriterien erarbeitet, welche der Modellierer erfüllen muss, damit eine anschließende Vernetzung garantiert werden kann.

Um diese Kriterien zu erfüllen, wurde ein mathematisches Konzept für den Modellierer

entwickelt, welches im Abschnitt 5.3 genauer erläutert wird. Damit diese Geometrie weiterverarbeitet werden kann, wurden Methoden zur Triangulierung von Oberflächen entwickelt und umgesetzt. Diese Methoden werden in Abschnitt 5.10 vorgestellt. Zu Testzwecken und zur Veranschaulichung der Modellierung mit Hilfe des Geometriemodellierers PaShiMo wurden verschiedene Modelle von Schiffen umgesetzt. Diese werden in Abschnitt 5.11 vorgestellt.

Insgesamt entstand ein Modellierungswerkzeug, welches es ermöglicht, komplexe Geometrien auf Basis von Parametern zu erzeugen. Bei der Optimierung von Schiffen mit VSP ist dieses Werkzeug in die Optimierungskette eingebaut.

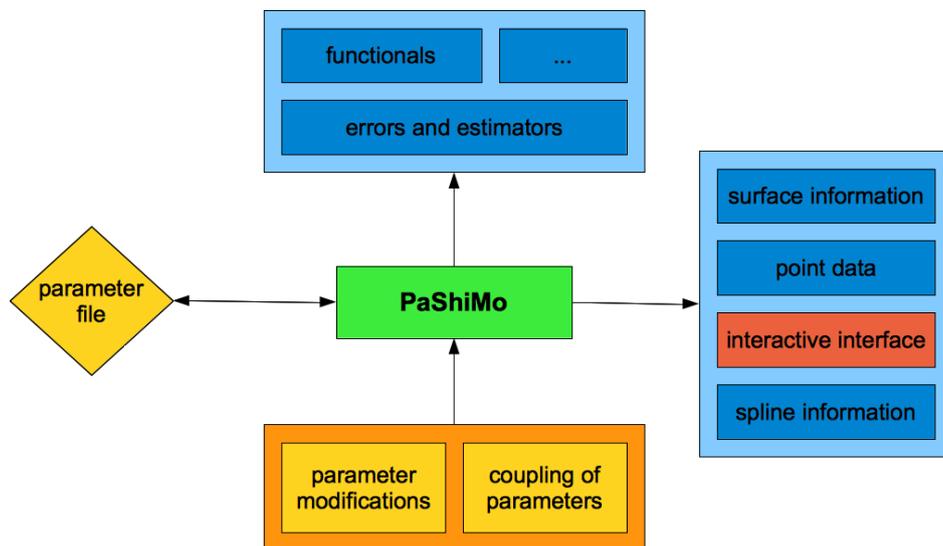


Abbildung 5.1: Übersicht über PaShiMo

In Abb. 5.1 ist die Funktionsweise von PaShiMo grafisch dargestellt. PaShiMo ist eine C++ Bibliothek, welche verschiedene Möglichkeiten zur geometrischen Modellierung in Form von Funktionen und Methoden implementiert und über eine API (**A**pplication **P**rogramming **I**nterface) dem Benutzer zur Verfügung stellt. Das Modellierungswerkzeug PaShiMo ist hierbei in der Mitte in grün abgebildet. Das Werkzeug bietet verschiedene Möglichkeiten zur Erzeugung und Manipulation von Geometrien. Grundsätzlich arbeitet PaShiMo auf der Basis von Parametern, welche verschiedene Eigenschaften der Schiffskörpergeometrie beschreiben. Ein vollständiger Parametersatz kann in Form einer Datei gegeben sein, welche von PaShiMo eingelesen wird. Alternativ können die Parameter über eine Programmierschnittstelle an PaShiMo übergeben werden. Die in PaShiMo berechnete Geometrie kann auf verschiedene Arten ausgelesen werden. Es besteht die Möglichkeit, Informationen über bestimmte Punkte auszulesen und eine vollständige Beschreibung der Geometrie in Form einer Datei in einem gängigen CAD Format zu erhalten. Die interaktive Kommunikation mit dem Werkzeug spielt für die Optimierung eine entscheidende Rolle, da der Optimierer eine Auswertung sowie eine Modifikation der Geometrie automatisiert durchführen muss.

Ein wichtiges Werkzeug in dem Optimierungskontext ist die Kopplung verschiedener Parameter, da die Modifikation nur eines Parameters sowie die unabhängige Änderung von mehreren Parametern in der Regel nicht ausreicht, um eine echte und sinnvolle Verbesserung der Geometrie zu erzielen.

## 5.2 Voraussetzungen an den Modellierer

Zu Beginn des Projekts wurden bisher existierende Möglichkeiten zur Beschreibung von Geometrien herausgearbeitet (siehe Kap. 2). Es wurde untersucht, ob diese für dieses Projekt in Frage kommen.

Um vollen Zugriff auf den Modellierungsvorgang zu haben, wurde entschieden, dieses Modellierungswerkzeug eigenständig umzusetzen, um größtmögliche Kontrolle über die Parameter und deren Einfluss auf die Geometriegenerierung zu besitzen. Unter diesen Voraussetzungen muss der Modellierer in der Optimierung nicht als Blackbox Programm angesehen werden.

### Allgemeine Voraussetzungen an das Geometriewerkzeug:

- Die erzeugte Geometrie darf keine Lücken in der Geometriebeschreibung besitzen.
- Die Geometrie soll möglichst straken, d.h. einen möglichst glatten Verlauf besitzen.
- Die Geometrieerzeugung soll möglichst stabil auf Parameteränderungen reagieren.
- Knicke auf dem Schiffsrumpf sollen modelliert werden können.
- Hydrostatische Eigenschaften der Geometrie sollen durch Parameter beschrieben werden können.
- Geometrische Eigenarten von Schiffen mit Voith-Schneider-Propeller sollen abgebildet werden können.
- Anbauteile sollen modellierbar sein.
- Die Geometrie soll Auswertungen von Punkten auf der Oberfläche der Geometrie erlauben.
- Die Geometrie sollte in einem Format ausgegeben werden können, welches eine Weiterverarbeitung erlaubt.
- Die Steuerparameter sollen für Schiffsbauingenieure anschaulich sein.
- Das Werkzeug soll flexibel und modular erweiterbar sein.

Auf Basis dieser Kriterien wurde das Konzept des Modellierers herausgearbeitet. Aufgrund des Wunsches nach einer flexiblen Erweiterbarkeit des Werkzeugs entschied man sich für eine Umsetzung als Programmbibliothek, welche die umgesetzten Methoden dem Nutzer in Form einer API zur Verfügung stellt. Diese Umsetzung erfolgte vollständig in C++ mit Standardbibliotheken, so dass eine Portierung auf beliebige Plattformen möglich ist.

Der Benutzer besitzt über die Verwendung der implementierten Methoden hinaus die Möglichkeit, diese Bibliothek um neue Methoden zu erweitern oder bereits existierende Methoden nach seinen Bedürfnissen anzupassen. Die Grundlage der Modellierung bilden NURBS und B-Splines. Bei der Beschreibung der Spanten werden B-Splines eingesetzt und bei der Beschreibung der Designkurven NURBS. NURBS bieten den Vorteil mehr Freiheiten bei der Modellierung von Kurven zu besitzen als B-Splines.

Die automatisierte Erzeugung der Geometrie wird mit Hilfe von Parametern gesteuert. Ziel war es die Anzahl der benötigten Parameter gering zu halten, da die Anzahl dieser Parameter die Anzahl der Freiheitsgrade in der Optimierung bestimmt.

### 5.3 Grundlegendes Mathematisches Modell und das Konzept von PaShiMo

Das Ziel dieses Abschnittes ist die kurze Erklärung des mathematischen Modells zur automatisierten Erzeugung einer parametrisierten Oberfläche. Hierzu wurde ein Ansatz gewählt, welcher sich stark am klassischen Schiffsbau orientiert. Die gesamte Beschreibung basiert auf der spantenweise Betrachtung der Geometrie.

Das Ziel ist die Erzeugung einer isogeometrischen Beschreibung der Geometrie. Die Erzeugung dieser Geometrie wird durch Parameter gesteuert. Das Ergebnis der Erzeugung ist eine endliche Menge von parametrischen Flächen. Jede dieser Flächen beschreibt einen Teil der vollständigen Geometrie. Dieser flexible und modulare Aufbau der Geometriebeschreibung erlaubt es, die Geometrie um zusätzliche Geometrieteile zu ergänzen. Aufgrund dieses Vorgehens lassen sich geometrische Abhängigkeiten der Teilflächen modellieren. Aber auch notwendige geforderte Konsistenzen der erzeugten Geometrie können gewährleistet werden.

Im klassischen Schiffsbau werden Schiffe auf Basis von Spanten beschrieben, [12]. Die Spanten beschreiben die Form von Rumpfqerschnitten an bestimmten Positionen des Schiffes. Aufgrund der Symmetrie des Schiffes ist es ausreichend, eine Hälfte des Schiffes mit Hilfe von Spanten zu beschreiben. Da das Schiff einen möglichst glatten Geometrieverlauf besitzen soll, wird die Geometrie des Querschnitts zwischen den Spanten interpoliert mit Rücksicht auf möglichst glatte Übergänge. Dieser glatte Übergang wurde früher mit Hilfe der Straklatten gewährleistet. In Zeiten der rechnergestützten Konstruktion wird dies mit Hilfe von Splines erreicht. Es ist bei der Modellierung der Spanten zu berücksichtigen, dass einige Schiffsgeometrien fertigungsbedingt Knicke im Geometrieverlauf besitzen. Dies ist insbesondere bei Schiffen der Fall, welche aus Stahl

gefertigt werden, da die einzelnen Stahlbleche, aus welchen sich die Geometrie zusammensetzt, nicht beliebig gebogen werden können. Die in diesem Projekt untersuchten Geometrien besitzen Knickverläufe in der Geometrie.

Die Spanten von Schiffen mit VSP werden als Kurven aufgefasst und in dieser Arbeit als Spantkurven bezeichnet. Eine solche Spantkurve ist für eine feste Spantposition eine zwei-dimensionale Kurve. Sie ist stetig und setzt sich zusammen aus Kurven unterschiedlicher Regularität, siehe Abb. 5.2. Bei der Erarbeitung des Konzepts von PaShiMo muss diese Eigenart einer Spantkurve berücksichtigt werden. Die sich auf der Spantkurve ergebenden Ecken bilden die Kanten auf der Schiffskörpergeometrie. Die parametrische Beschreibung muss diesen Verlauf von Kanten entlang des Schiffes abbilden können.

Für das restliche Kapitel gilt folgende Konvention für die Orientierung der Geometrie (siehe Abb. 5.3). Das Schiff liegt in einem kartesischen Koordinatensystem mit den drei Achsen  $x$ ,  $y$  und  $z$ . Aufgrund der Symmetrieeigenschaft betrachten wir nur den halben Schiffskörper. Die Symmetrieebene bildet die  $x$ - $z$ -Ebene. Die Punkte auf der Geometrieoberfläche besitzen stets positive  $x$ - und  $y$ -Komponenten. Die Spanten des Schiffes liegen auf zur Symmetrieebene orthogonalen Ebenen, d.h. auf zur  $y$ - $z$ -Ebene parallelen Ebenen. Die  $z$ -Komponente der Oberflächenpunkte darf auch negative Werte annehmen. Der Bug des Schiffes liegt auf der  $y$ - $z$ -Ebene. Das Schiff wird in positiver  $x$ -Achsenrichtung aufgebaut. Es gilt die folgende Notation für Punkte  $P$  in diesem Koordinatensystem

$$P = (P^1, P^2, P^3) = (P^x, P^y, P^z) \in \mathbb{R}^3. \quad (5.3.1)$$

Es wird zunächst die mathematische Definition einer Spantkurve eingeführt, welche später die Erklärung der Parametrisierung und der Funktionsweise des implementierten Computerwerkzeuges erleichtert. Die Länge, Breite und Tiefe des Schiffes ist zunächst auf das Einheitsintervall normiert, d.h. die maximale Länge, Breite und Tiefe ist 1.

**Definition 5.3.1 (Spantkurve)** *Sei die Länge des Schiffes 1, so ist  $I = [0, 1] \subset \mathbb{R}$  das zulässige Intervall für Spantpositionen und die Spantkurve an der Position  $x_0 \in I$  ist eine zweidimensionale Kurve*

$$s_{x_0} : \mathbb{R} \rightarrow [0, 1]^2, \quad v \mapsto s_{x_0}(v), \quad (5.3.2)$$

für einen Laufparameter  $v \in \mathbb{R}$ .

Zunächst wird ein Modell für die zusammengesetzte Spantkurve hergeleitet, welche obiger Definition genügt. Diese Spantkurve wird mit Hilfe von Parametern  $p \in \mathbb{N}$  beschrieben. Diese werden in dem Vektor  $\omega \in \Omega \subset \mathbb{R}^p$  zusammengefasst, wobei  $\Omega$  als der Spantparameterraum bezeichnet wird.

Der Hintergrund ist die Erarbeitung einer automatisierten Methode, welche es erlaubt für einen Parametersatz  $\omega$  die Spantkurve automatisiert zu generieren. Die Erzeugung

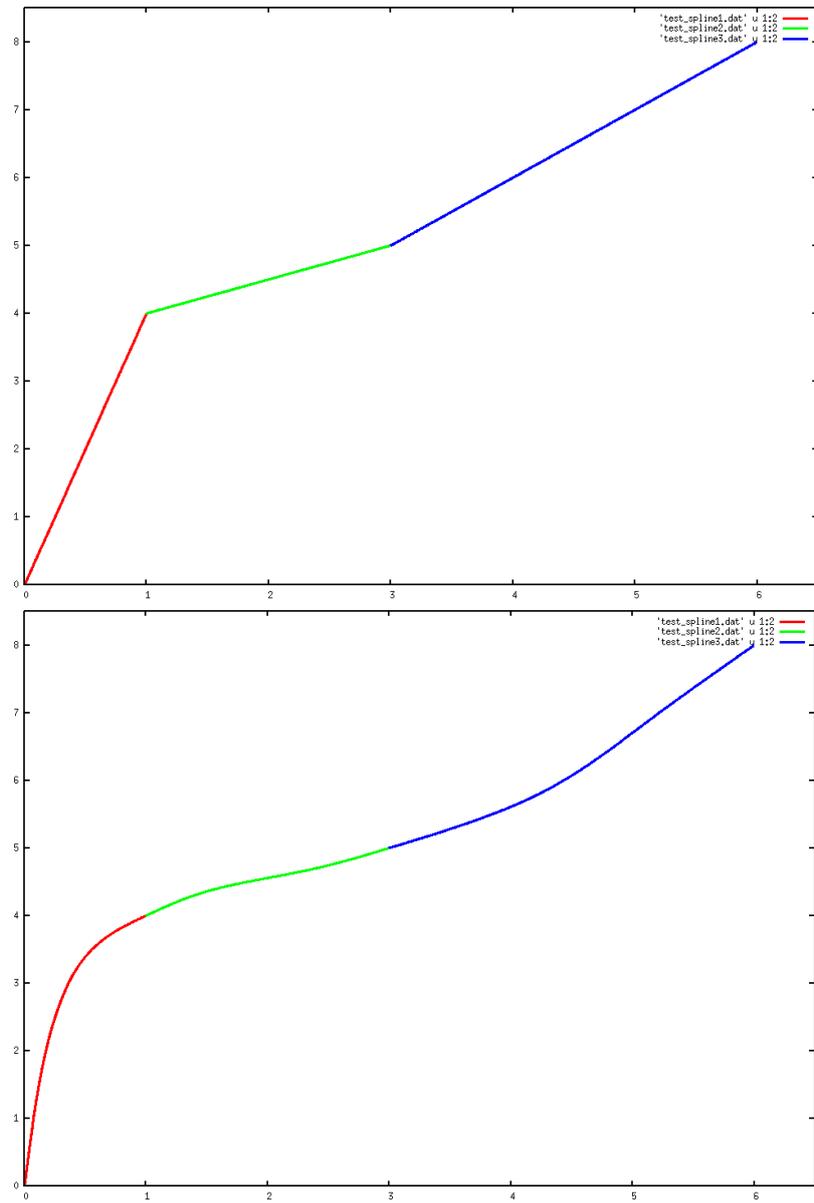


Abbildung 5.2: Spantkurven bestehend aus Teilkurven veschiedener Regularität

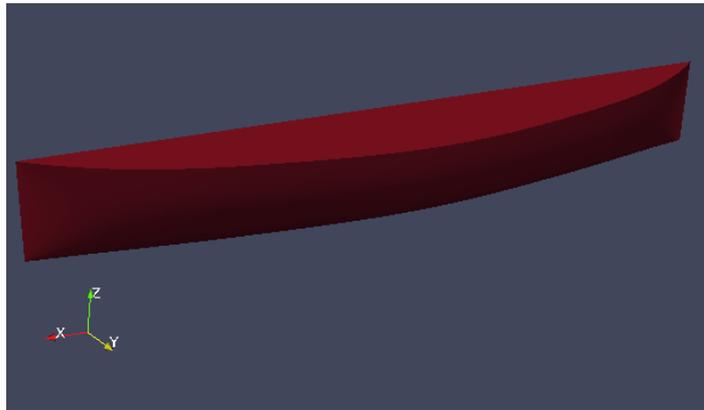


Abbildung 5.3: Orientierung des Schiffes

der vollständigen Geometriebeschreibung basiert auf dieser Methodik.

## 5.4 Parametrische Modellierung einer Spantkurve

Die Spantkurve setzt sich zusammen aus  $n \in \mathbb{N}$  Teilkurven unterschiedlicher Regularität.

**Definition 5.4.1** Sei die Spantkurve aus  $n \in \mathbb{N}$  Teilkurven zusammengesetzt und sei  $\omega \in \Omega \subset \mathbb{R}^p$  der Vektor, in welchem die  $p \in \mathbb{N}$  Parameter zur Beschreibung der Spantkurve zusammengefasst werden. Die  $i$ -te Teilkurve für  $i = 1, \dots, n$  ist gegeben als

$$s_i : [0, 1] \times \Omega \rightarrow \mathbb{R}^2, \quad (v, \omega) \mapsto s_i(v, \omega), \quad (5.4.1)$$

für einen Parameter  $v \in [0, 1] \subset \mathbb{R}$ .

In obiger Definition hängt die Kurve nicht mehr von der Spantposition ab, sondern nur von dem Parametervektor  $\omega$ .

Es ist zu beachten, dass die Kurve  $s_i$  ein Tupel mit zwei Komponenten ist

$$s_i(v, \omega) = (s_i^1(v, \omega), s_i^2(v, \omega)) = (s_i^y(v, \omega), s_i^z(v, \omega)) \in \mathbb{R}^2. \quad (5.4.2)$$

Jede Teilkurve  $s_i$  wird in einem lokalen Koordinatensystem modelliert. Eine Modellannahme ist, dass die Kurve  $s_i$  vollständig im Inneren des begrenzenden Quader verläuft, welcher durch den Anfangspunkt  $P_i$  und den Endpunkt  $P_{i+1}$  eindeutig bestimmt ist, siehe Abb. 5.4.

Durch diese Modellannahme beschränkt sich die Konstruktion der Teilkurven auf das zweidimensionale Einheitsintervall  $[0, 1] \times [0, 1] \subset \mathbb{R}^2$ . Die Umrechnung der Teilkurven in das Koordinatensystem der Spantkurve erfolgt mit Hilfe folgender Transformation.

**Definition 5.4.2 (Teilkurven Transformation)** Seien die Anfangs- und Endpunkte der  $i$ -ten Teilkurve für  $i = 0, \dots, n - 1$  durch die Punkte  $P_i$  und  $P_{i+1}$  aus dem  $\mathbb{R}^2$

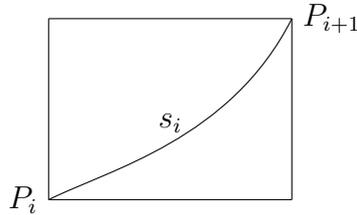


Abbildung 5.4: Teilkurve im begrenzenden Quader

gegeben und sei  $s_i$  die im Inneren von  $[0, 1] \times [0, 1] \subset \mathbb{R}^2$  verlaufende  $i$ -te Teilkurve der Spantkurve. So ist diese Teilkurve im Koordinatensystem des Schiffes gegeben als

$$s_i(v) = \begin{pmatrix} s_i^1(v) \\ s_i^2(v) \end{pmatrix}^\top = \begin{pmatrix} P_i^1 + s_i^1(v)(P_{i+1}^1 - P_i^1) \\ P_i^2 + s_i^2(v)(P_{i+1}^2 - P_i^2) \end{pmatrix}^\top. \quad (5.4.3)$$

Mit Hilfe dieses Konzepts wird die Parametrisierung der gesamten Spantkurve über die Parametrisierung einer Teilkurve aufgebaut. Ausgehend von der Position der Anfangs- und Endpunkte einer jeden Teilkurve werden einzelne Blöcke angelegt, in welchen je eine Kurve generiert wird. Da die in diesem Block generierte Kurve die Anfangs- und Endpunkte interpoliert, ist die Stetigkeit der Spantkurve garantiert. Die Beschreibung des Blocks jeder Teilkurve erfolgt durch die Beschreibung der Anfangs- und Endpunkte der Teilkurvenstücke.

Die Koordinaten dieser Punkte können auf zwei verschiedene Arten beschrieben werden. Zum Einen können diese absolut angegeben werden und zum anderen mit Hilfe der Polarkoordinatendarstellung. Bei jeder dieser beiden Beschreibungen werden zwei Parameter benötigt. Eine effizientere Variante zur Beschreibung der Eckpunkte der Blöcke ergibt sich durch eine Mischung beider Verfahren. Die absolute Angabe der Koordinaten birgt für den Benutzer den Vorteil, sich keine Gedanken über die Umrechnung der Koordinaten machen zu müssen. Dies bringt allerdings ein Problem beim Konstruieren von konsistenten Geometrien mit sich. Die Abhängigkeiten von Teilflächen zueinander können nicht modelliert werden. Dies führt dazu, dass erzeugte Geometrien schnell unzulässig werden, da die Teilkurven zwangsweise nicht mehr stetig verbunden sind.

Daher existiert in PaShiMo die Möglichkeit, die Anfangs- und Endpunkte der Teilkurven relativ zueinander zu konstruieren. Der beschreibende Koordinatenvektor  $\vec{p} \in \mathbb{R}^2$  eines Punktes beschreibt bei dieser Methode den Verschiebungsvektor zwischen dem neuen konstruierten Punkt und dem vorher modellierten Koordinatenpunkt. Diese Konstruktion ermöglicht einen konsistenten Aufbau der Geometrie. Aus diesen Überlegungen heraus ergibt sich der als relationäre Blocktopologie bezeichnete Aufbau der Spantkurventopologie.

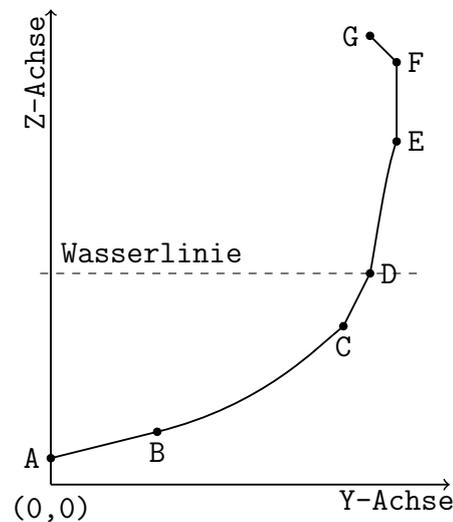


Abbildung 5.5: Spant bestehend aus verschiedenen Teilkurven

**Definition 5.4.3 (Relationäre Blocktopologie)** Seien  $n \in \mathbb{N}$  die Anzahl der Teilkurven und  $\vec{p}_i \in \mathbb{R}^2$  für  $i = 0, \dots, n-1$  die Differenzvektoren zwischen den Anfangs- und Endpunkten der Teilkurven. Die durch diese sich ergebende Folge von Koordinatenpunkte wird relationäre Blocktopologie genannt. Die Koordinaten  $P_i$  für  $i = 0, \dots, n-1$  der Eckpunkte der Blöcke lassen sich wie folgt berechnen.

$$P_i = \begin{pmatrix} P_i^1 \\ P_i^2 \end{pmatrix}^\top = \sum_{k=0}^i \vec{p}_k. \quad (5.4.4)$$

Die Differenzvektoren aus der obigen Definition werden entweder absolut oder in Form von Polarkoordinaten gegeben

$$\vec{p}_k = (r_k \cdot \cos(\alpha_k), r_k \cdot \sin(\alpha_k)), \quad (5.4.5)$$

für Radien  $r_k \geq 0$  und Winkel  $\alpha_k \in [0, 2\pi)$ . Unabhängig von der Beschreibungsmethode werden pro Differenzvektor  $\vec{p}_k$  zwei Informationen benötigt, welche in dem Parametervektor  $\omega$  für einen Spant abgelegt werden.

### 5.4.1 Parametrisierung der Teilkurven

Die Kurvenstücke  $s_i$  der Spantkurve (siehe Abb. 5.6) werden nun mit Hilfe von Parametern beschrieben. Hierbei ist zu beachten, dass für die Teilkurven je nach gewünschter Komplexität eine verschiedene Anzahl von Parameter benötigt wird.

Die Teilkurven werden durch den Anfangs- und Endwinkel  $T_1$  bzw.  $T_2$  beschrieben. Hierbei ergibt sich modellabhängig eine weitere Parameterreduktion. Falls zwei Teilkurven stetig differenzierbar verbunden sind, so wird nur eine Winkelinformation benötigt.

Da die Kurvenstücke möglichst glatt sein soll, wird die Krümmung am Anfang und am Ende der Teilkurven auf 0 gesetzt.

Zudem besitzt man im Modellierungswerkzeug die Möglichkeit, den Flächeninhalt  $A$  der eingeschlossenen Fläche mit den Achsen und der Wasserlinie zu kontrollieren. Die auf diese Art und Weise eingeführten Parameter werden in dem Parametervektor  $\omega$  gespeichert. Hierbei ist zu beachten, dass  $\omega$  die Parameter aller Teilkurven enthält.

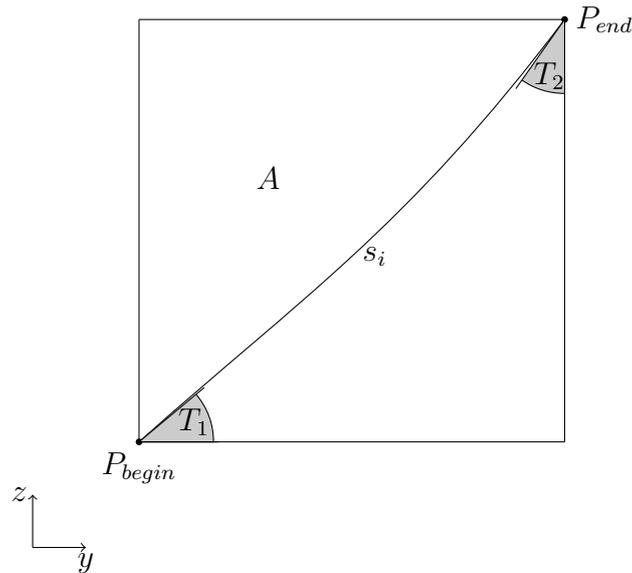


Abbildung 5.6: Parameter für die Teilkurven

## 5.5 Designkurven

Durch die obige Methode wird die Spantkurve an einer festen  $x$ -Position mit Hilfe eines Parametersatzes  $\omega \in \mathbb{R}^p$  zu beschreiben.

Der naive Ansatz, Parametersätze für eine hinreichend große Anzahl von Spanten zu speichern, führt bei der Beschreibung einer Spantengeometrie zu einer sehr großen Anzahl von Parametern. Daher wird ein Parametersatz  $\omega$  als Auswertung einer  $p$ -dimensionalen Funktion  $Q$  aufgefasst

$$Q : [0, 1] \rightarrow \Omega, \quad u \mapsto Q(u), \quad (5.5.1)$$

wobei  $u \in [0, 1] \subset \mathbb{R}$  die jeweilige Spantposition und  $\Omega \subset \mathbb{R}^p$  den Parameterraum beschreibt. Die Parameter für die mathematische Beschreibung der Geometrie ergeben sich nun durch die Parametrisierung von  $Q$ .

$Q$  wird als NURBS Kurve auf einem festen Gitter auf dem Intervall  $[0, 1]$  modelliert. Die Kontrollpunkte und die Gewichte dieses NURBS sind die Parameter der Parametrisierung.

Die einzelnen Komponenten von  $Q$  beschreiben Eigenschaften der einzelnen Teilkurven der Spantkurve entlang des Schiffes. Diese Tatsache ist die Motivation für den Namen Designkurve. In Abb. 5.7 ist dies am Beispiel der Kiellinie veranschaulicht. Diese Designlinie modelliert die Position des Anfangspunktes der Spantkurven.

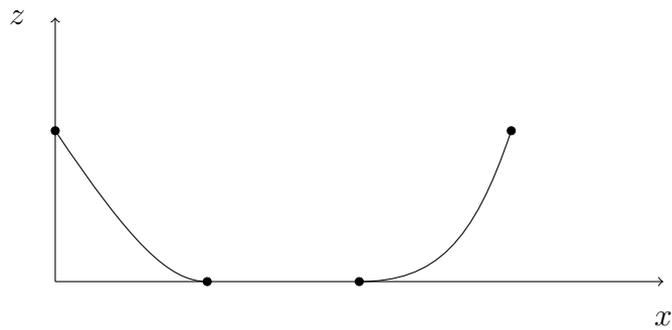


Abbildung 5.7: Designlinie für den Kielverlauf entlang des Schiffes

## 5.6 Globale Parameter

Es werden globale Parameter welche die Länge, die Breite und den Tiefgang des Schiffes beschreiben, in das Modell eingeführt. Diese Parameter fließen in das Modell über eine Skalierungsfunktionen ein. Wie zuvor in diesem Kapitel beschrieben, erfolgt die Konstruktion jeder Teilkurve im zweidimensionalen Einheitsintervall  $[0, 1] \times [0, 1] \subset \mathbb{R}^2$ . Die Berechnung der tatsächlichen Koordinaten wird an dieser Stelle noch einmal verallgemeinert.

**Definition 5.6.1 (Totale Skalierung)** Sei  $L \in \mathbb{R}$  die Länge des Schiffes,  $B \in \mathbb{R}$  die Breite des Schiffes und  $T \in \mathbb{R}$  der Tiefgang des Schiffes. Die Skalierung der Punkte  $P \in \mathbb{R}^3$  auf der Spantkurve in das endgültige Koordinatensystem erfolgt über

$$Scal_{total} : \mathbb{R}^3 \rightarrow \mathbb{R}^3, \quad P = (P^1, P^2, P^3) \mapsto (L \cdot P^1, B \cdot P^2, T \cdot P^3). \quad (5.6.1)$$

Mit Hilfe dieser Skalierung wurden Parameter eingeführt, welche zwar keinen Einfluss auf den Geometrieverlauf, jedoch aber auf die Skalierung der Geometrie besitzen. Mit Hilfe dieser können Einflüsse auf die Dimensionen des Schiffes vorgenommen werden, ohne die Geometriebeschreibung zu ändern.

## 5.7 Parametrisierung der gesamten Schiffsgeometrie

Mit Hilfe der Konstruktionen aus den vorherigen Abschnitten kann nun die vollständige Geometrie des Schiffes beschrieben werden.

Die sich ergebenden Einzelflächen  $S_i$  lassen sich beschreiben über

$$S_i : [0, 1] \times [0, 1] \rightarrow \mathbb{R}^3, \quad (u, v) \mapsto Scal_{total} \left( u, s_i(v, Q(u)) \right). \quad (5.7.1)$$

Hierbei ist  $i$  der Index der  $i$ -ten Teilfläche, welche die  $i$ -te Teilkurve des Spant enthält. Der Parameter  $u$  beschreibt die  $x$ -Position des Spantes und der Parameter  $v$  ist der Parameter der  $i$ -ten Teilkurve.

Aus dieser Konstruktion ergibt sich eine Blocktopologie für die Schiffkörpergeometrie. Die Geometrie besteht aus einzelnen isogeometrischen Flächen  $S_i$  welche nach Konstruktion stetig verbunden und jeweils auf dem mehrdimensionalen Intervall

$$D = [0, 1] \times [0, 1] \subset \mathbb{R}^2$$

definiert sind. Die Erzeugung der einzelnen Flächen wird gesteuert über die Kontrollpunkte der NURBS Kurve  $Q$ . Aufgrund der Variabilität der Anzahl von Teilkurven eines Spant und der Einführung der NURBS Kurve  $Q$  kann das Modell vereinfacht oder erweitert werden.

## 5.8 Berechnung der Teilkurven

In diesem Abschnitt wird die Berechnung der Teilkurven aus den Parametern dargestellt. Für eine lineare Teilkurve ist die Teilkurve  $s$  in dem lokalen Koordinatensystem gegeben durch

$$s(v) = \begin{pmatrix} v \\ v \end{pmatrix} \quad (5.8.1)$$

mit Parameter  $v \in [0, 1]$ .

Die allgemeine Definition einer Spantkurve erfolgt mit Hilfe einer NUBS-Funktion. Es wurde eine NUBS-Funktion verwendet, da hierdurch die Modellierung zusätzlich durch die Wahl des Gitters beeinflusst werden kann. Bei einigen Modellen, bei denen ein stärkerer Einfluss der Kontrollpunkte im Randbereich der Teilkurven benötigt wird, wird ein *nicht uniformes* Gitter verwendet.

**Definition 5.8.1** Sei  $k$  eine NUBS-Funktion der Dimension  $b$  und dem Grad  $p$  auf einem Gitter  $\Delta$  auf  $[0, 1] \subset \mathbb{R}$  mit den Koeffizienten  $\mathbf{c}_i$

$$k(t) = \sum_{i=1}^b \mathbf{c}_i N_{i,p,\Delta}(t). \quad (5.8.2)$$

Zusätzlich erfüllt  $k$  die Bedingungen

$$k(0) = k(1) = 0, \quad (5.8.3)$$

$$k''(0) = k''(1) = 0. \quad (5.8.4)$$

Dann ist die Teilkurve  $s$  gegeben als

$$s(t) = \begin{pmatrix} t + k(t) \\ t - k(t) \end{pmatrix} \quad (5.8.5)$$

mit dem Parameter  $t \in [0, 1]$ .

Diese Teilkurve  $s$  interpoliert den Anfangs- und Endpunkt. Die Randbedingungen an die erste Ableitung, sowie die Flächeninhaltsbedingung an die Teilkurve  $s$  werden nun über  $k$  eingearbeitet.

Für ein festes Gitter  $\Delta$  von  $k$  ergibt sich über die Bedingungen

$$\begin{aligned} k(0) &= p_1 \\ k(1) &= p_2 \\ k'(0) &= p_3 \\ k'(1) &= p_4 \\ k''(0) &= p_5 \\ k''(1) &= p_6 \end{aligned}$$

ein lineares Gleichungssystem für die Koeffizienten  $\mathbf{c}$  von  $k$

$$\mathbf{A}_k \mathbf{c} = \bar{\mathbf{p}}. \quad (5.8.6)$$

Die rechte Seite  $\bar{\mathbf{p}}$  ist über die Spantkurvenparameter  $\omega$  gegeben. Die Matrix  $\mathbf{A}_k$  hängt von dem Gitter  $\Delta$  und dem Grad  $p$  ab. Die Einträge  $\mathbf{a}_{m,n}$  der Matrix  $\mathbf{A}_k$  bestehen aus Auswertungen der B-Spline Basisfunktionen von  $k$ .

$$k(0) = \sum_{i=1}^b \mathbf{c}_i N_{i,p,\Delta}(0) = \mathbf{c}_1 \underbrace{N_{1,p,\Delta}(0)}_{\mathbf{a}_{1,1}} + \dots + \mathbf{c}_b \underbrace{N_{b,p,\Delta}(0)}_{\mathbf{a}_{1,b}} \quad (5.8.7)$$

$$k(1) = \sum_{i=1}^b \mathbf{c}_i N_{i,p,\Delta}(1) = \mathbf{c}_1 \underbrace{N_{1,p,\Delta}(1)}_{\mathbf{a}_{2,1}} + \dots + \mathbf{c}_b \underbrace{N_{b,p,\Delta}(1)}_{\mathbf{a}_{2,b}} \quad (5.8.8)$$

$$k'(0) = \sum_{i=1}^b \mathbf{c}_i N'_{i,p,\Delta}(0) = \mathbf{c}_1 \underbrace{N'_{1,p,\Delta}(0)}_{\mathbf{a}_{3,1}} + \dots + \mathbf{c}_b \underbrace{N'_{b,p,\Delta}(0)}_{\mathbf{a}_{3,b}} \quad (5.8.9)$$

...

Diese Matrix muss für jede Teilkurve nur einmal aufgestellt werden.

Es gilt für die Anzahl der Gitterkontrollpunkte  $\ell + 1$  des Gitters  $\Delta$ , dem Grad  $p$  und die Anzahl der Gleichungen  $N$

$$p + 1 + \ell = N. \quad (5.8.10)$$

Wegen der Konstruktion von  $s$  mit Hilfe eines NUBS  $k$  lässt sich das Modell um weitere Bedingungen erweitern. Um die Flächeninhaltbedingung an die Teilkurve einzubringen, wird das Gleichungssystem um die Bedingung

$$\int_0^1 k(t) dt = p_7 \quad (5.8.11)$$

ergänzt.

Nun wird der Zusammenhang zwischen der Funktion  $s$  und den Koeffizienten  $\mathbf{c}$  des NUBS  $k$  genauer betrachtet. Ziel ist es eine Darstellung von  $s$  in Abhängigkeit der Koeffizienten von  $k$  zu erhalten.

Die lineare Abbildung  $s$  lässt sich als Hintereinanderschaltung folgender linearer Transformationen schreiben:

### 1. Spiegelung an der $y$ -Achse

Diese lineare Abbildung lässt sich durch folgende Matrix beschreiben:

$$A_1 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad (5.8.12)$$

### 2. Rotation um $\phi = \frac{\pi}{2}$

Diese lineare Abbildung lässt sich durch folgende Matrix beschreiben:

$$A_2 = \begin{pmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{pmatrix} = \begin{pmatrix} \frac{1}{2}\sqrt{2} & -\frac{1}{2}\sqrt{2} \\ \frac{1}{2}\sqrt{2} & \frac{1}{2}\sqrt{2} \end{pmatrix} \quad (5.8.13)$$

### 3. Streckung mit Faktor $\sqrt{2}$

Diese lineare Abbildung lässt sich durch folgende Matrix beschreiben:

$$A_3 = \begin{pmatrix} \sqrt{2} & 0 \\ 0 & \sqrt{2} \end{pmatrix} \quad (5.8.14)$$

Der Skalierungsfaktor von  $\sqrt{2}$  ergibt sich aus der Länge der Diagonalen.

Die Hintereinanderausführung dieser linearen Abbildungen ergibt:

$$T = A_3 A_2 A_1 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \quad (5.8.15)$$

Nun wird die Definition des Kontrollpolygonzug eines NUBS betrachtet, [11].

**Definition 5.8.2** Sei  $k$  eine NUBS Funktion vom Grad  $p$ , der Dimension  $b$ , mit den Gitterknotenpunkten  $\{\tau_1, \dots, \tau_{b+p+1}\}$  und den Koeffizienten  $\{c_0, \dots, c_b\}$ . Dann ist der Kontrollpolygonzug von  $k$  gegeben durch die Punkte

$$P_i = (\tau_i^*, c_i) \text{ für } i = 0, \dots, b \quad (5.8.16)$$

mit

$$\tau_i^* = \frac{1}{p} \sum_{j=1}^p \tau_{i+j}. \quad (5.8.17)$$

Es gilt folgende Identität

$$\begin{pmatrix} t \\ k(t) \end{pmatrix} = \sum_{i=1}^b P_i N_{i,p,\Delta}(t) = \sum_{i=1}^b \begin{pmatrix} \tau_i^* \\ c_i \end{pmatrix} N_{i,p,\Delta}(t). \quad (5.8.18)$$

Mit (5.8.15) ergibt sich für  $s$

$$s(t) = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \sum_{i=1}^b \begin{pmatrix} \tau_i^* \\ c_i \end{pmatrix} N_{i,p,\Delta}(t) = \sum_{i=1}^b \underbrace{\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} \tau_i^* \\ c_i \end{pmatrix}}_{M_i} N_{i,p,\Delta}(t). \quad (5.8.19)$$

Aus diesem Ergebnis lässt sich direkt aus dem NUBS  $k$  eine Darstellung von  $s$  gewinnen. Ausgehend von einem gegebenen Parametersatz  $\omega$  ergibt sich folgender Ablauf für die die Berechnung der Parameterkurvendarstellung der Teilkurven:

1. Assemblierung der Matrix  $\mathbf{A}_k$  des linearen Gleichungssystems (5.8.6) für jede Teilkurve.
2. Bestimmung einer L-R-Zerlegung für die Matrix  $A_k$  von (5.8.6).
3. Auswertung und Umrechnung der Parameter  $\omega$  in das lokale Koordinatensystem der Teilkurven. Diese bilden die rechten Seiten  $\bar{\mathbf{p}}$ .
4. Berechnung der Koeffizienten des NUBS  $k$  mit Hilfe der L-R-Zerlegung und  $\bar{\mathbf{p}}$ .
5. Berechnung des Kontrollpolygonzugs von  $k$  mit Hilfe von (5.8.17).
6. Berechnung der NUBS Kontrollpunkte  $M_i$  in (5.8.19).

Für die NUBS Kontrollpunkte  $M_i^*$  des globalen Koordinatensystems gilt

$$M_i^* = P_{begin} + \begin{pmatrix} (P_{end}^1 - P_{begin}^1) & 0 \\ 0 & (P_{end}^2 - P_{begin}^2) \end{pmatrix} M_i. \quad (5.8.20)$$

Dieser Ablauf zur Bestimmung der Spantkurve aus einem Parametersatz  $\omega$  wurde in dem Computerprogramm PaShiMo umgesetzt.

## 5.9 Umsetzung der Parametrisierung in PaShiMo

Das Programm PaShiMo ist eine C++ Bibliothek. Das Programm wurde mit Hilfe von Klassen strukturiert.

Die Schnittstelle von PaShiMo für den Benutzer ist in der Klasse *Ship* umgesetzt. Diese Klasse enthält die Methoden zum Einlesen und Auslesen der NURBS Informationen für die Designlinien  $Q$ . Auch die Auswahl des Schiffsmodells erfolgt über Methoden dieser Klasse. Ausserdem können Informationen über die Geometrie und Punktauswertungen über die Geometrie über Methoden in dieser Klasse abgefragt werden. Im Folgenden werden einige ausgewählte Methoden dieser Klasse vorgestellt.

- *open\_NURBS*  
Diese Methode liest die NURBS Informationen für die Designkurve aus einer Datei ein.
- *save\_NURBS*  
Diese Methode schreibt die aktuellen NURBS Informationen der Designkurve in eine Datei.
- *set\_type*  
Mit Hilfe dieser Methode kann das jeweilige Schiffsmodell ausgewählt werden. Erst durch Aufruf dieser Methode wird festgelegt, wie die NURBS Informationen interpretiert und weiterverarbeitet werden.

- *get\_type*  
Diese Methode liefert Informationen über das aktuelle verwendete Schiffsmodell.
- *eval*  
Diese Methode ermöglicht die Auswertung von einzelnen Punkten der Geometrie.
- *plot\_3Dpoints*  
Diese Methode schreibt eine Punktwolke von Punkten auf der Oberfläche in eine Datei.
- *getParam\_NURBS*  
Diese Methode liefert den Wert eines Parameters, d.h. den Koordinatenwert eines Kontrollpunktes.
- *modParam\_NURBS*  
Diese Methode modifiziert den Wert eines Parameters, d.h. den Koordinatenwert eines Kontrollpunktes.
- *getWeights\_NURBS*  
Diese Methode liefert den Wert eines Gewichtes des NURBS für die Designkurve.
- *modWeights\_NURBS*  
Diese Methode modifiziert den Wert eines Gewichtes des NURBS für die Designkurve.
- *set\_glob*  
Diese Methode setzt den Wert eines globalen Parameters.
- *get\_glob*  
Diese Methode liefert den Wert eines globalen Parameters.

Die NURBS Informationen sind in einer ASCII Datei abgelegt. Diese besitzen die Endung *.ship.dat*. In der ersten Zeile dieser Datei sind die Werte für die globalen Parameter abgelegt. In der darauf folgenden Zeile findet sich die Anzahl der NURBS Kurven, welche eingelesen werden sollen. Anschließend folgen die NURBS Kurven nacheinander. Jeder Datenblock für eine Kurve enthält in der ersten Zeile die Information über die Anzahl der Gitterknotenpunkte. Im Anschluss folgt die Aufzählung dieser. In der nächsten Zeile stehen die Informationen über die Ordnung und die Koordinatendimension  $N \in \mathbb{N}$  des NURBS. In den den nächsten  $N$  Zeilen folgen die Koordinaten der Kontrollpunkte. Die letzte Zeile jedes Blocks enthält die Werte der Gewichte der NURBS Kurve. Es ist anzumerken, dass ein solcher Datensatz nur mit einer entsprechenden Modellierung Sinn macht.

## 5.10 Aufbereitung der Geometrie für die Weiterverarbeitung (stl Generierung)

Die Aufbereitung der Geometrie sieht die Erzeugung einer triangulierten Oberfläche aus dem analytischen Modell der Beschreibung vor. Die Triangulierung wird im stl Format in eine Datei ausgegeben. Das stl Format speichert das Dreiecksnetz mit Hilfe der Eckpunkte und der Normalen ab. In Listing 5.1 ist ein Beispiel für eine solche stl Datei zu sehen. Eine zusammenhängende Geometrie wird zwischen *solid* und *endsolid* beschrieben, welche in diesem Fall den Namen *shape* besitzt. Die Dreiecke werden jeweils zwischen *facet* und *endfacet* durch die Normale und die Eckpunkte beschrieben, wobei die Koordinaten der Normale in der Zeile mit dem Begriff *normal* und die Koordinaten der Eckpunkte in der Zeile mit dem Begriff *vertex* abgelegt werden.

Listing 5.1: stl Beispiel

```

solid shape
...
  facet normal      n1  n2  n3
  outer loop
    vertex  p1  p2  p3
    vertex  q1  q2  q3
    vertex  r1  r2  r3
  endloop
  endfacet
...
endsolid shape

```

Die auf diese Weise beschriebene Geometrie kann von vielen CAD Programmen eingelesen und weiterverarbeitet werden.

Im Internet sind kostenlose und quelloffene Programme zur Visualisierung solcher triangulierten Daten zu finden. Ein solches Programm ist Paraview von Sandia National Laboratories. Dieses wurde in dem Projekt SimuVSP als Betrachter für die erzeugte Geometrie eingesetzt. Hierdurch entfällt die Umsetzung eines eigenen Betrachters für die Geometrie.

Die Gründe für die Umsetzung eines Oberflächentriangulierers für die Geometrien aus PaShiMo sind:

- Visualisierung der Geometrie durch einen bereits existierenden Betrachter.
- Weiterverwertung der Geometrie durch andere CAD Werkzeuge.
- Weiterverwertung der Geometrie durch den automatischen Vernetzer *vmesh*.

Bei der Erzeugung der Triangulierung müssen hängende Knoten vermieden werden. Zur Charakterisierung der Teilflächen wird folgende Definition eingeführt.

**Definition 5.10.1** Für die Oberfläche  $S_i$  in (5.7.1) wird

$$m_i(u) = \int_0^1 \|s'_i(v, Q(u))\| dv \quad (5.10.1)$$

als die Bogenlänge der Teilkurve in  $u \in [0, 1]$  bezeichnet.

Um eine Methode zur Triangulierung dieser Flächen herzuleiten, werden die folgenden Referenzflächen eingeführt. Hierzu seien  $a, b \in [0, 1]$  mit  $a < b$ .

1. **Referenzfläche vom Typ 1:**  
 $m_i(a) = 0, m_i(b) > 0$  und  $m'_i(u) > 0 \quad \forall u \in (a, b)$ .
2. **Referenzfläche vom Typ 2:**  
 $m_i(a) > 0, m_i(b) = 0$  und  $m'_i(u) < 0 \quad \forall u \in (a, b)$ .
3. **Referenzfläche vom Typ 3:**  
 $m_i(a) > m_i(b) > 0, m'_i(u) < 0 \quad \forall u \in (a, b)$ .
4. **Referenzfläche vom Typ 4:**  
 $m_i(b) > m_i(a) > 0, m'_i(u) > 0 \quad \forall u \in (a, b)$ .
5. **Referenzfläche vom Typ 5:**  
 $m_i(b) = m_i(a), m'_i(u) = 0 \quad \forall u \in (a, b)$ .

Es gilt folgende Ergebnis:

**Ergebnis 5.10.2** Jede durch PaShiMo erzeugte Teilfläche lässt sich in Flächen zerlegen, welche einer der obigen Referenzflächen entspricht.

Eine Triangulierungsmethode für die Teilflächen wird mit Hilfe dieses Ergebnisses hergeleitet. Da jede Teilfläche in kleinere Flächen vom Typ einer Referenzfläche zerlegt werden kann, kann jede dieser kleinen Flächen einzeln betrachtet werden. Daher ist es ausreichend, eine Strategie zur Vernetzung der fünf Referenzflächen zu erarbeiten.

Es wird die folgende geometrische Beobachtung verwendet: Es werden zwei parallel gegenüberliegende Seiten eines Parallelogramms betrachtet. Für jede dieser beiden Seiten sei je ein Gitter gegeben. Die Anzahl der verwendeten Gitterpunkte beider Gitter unterscheidet sich maximal um eins. Ausserdem befindet sich das Gitter mit der um eins größeren Anzahl von Gitterpunkten auf der längeren der beiden Seiten. Dieser Sachverhalt ist in Abb. 5.8 dargestellt. Nun verbindet man die Gitterknoten wie auf dieser Abbildung zu sehen. Diese Idee bildet die Grundlage des Algorithmus. Die Bezeichnung der Punkte wird wie in Abb. 5.9 gewählt. Die Menge der Knoten bezeichnen wir mit  $V$  und die Menge der Dreieckszellen mit  $T$ . Mit dieser Grundüberlegung erhält man den Grundalgorithmus (siehe Algorithmus 1). Mit diesem Grundalgorithmus können Referenzflächen vom Typ 5 trianguliert werden.

Um die übrigen Referenzflächen zu vernetzen, wendet man folgende Strategie an: Es wird von Spantposition zu Spantposition die Anzahl der Gitterpunkte um eins erhöht.

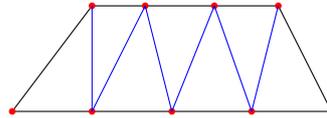


Abbildung 5.8: Grundidee der Triangulierung

Hierbei ergibt sich ein zusätzliches Dreieck  $\Delta_1$ . Dies ist in Abb. 5.10 veranschaulicht. Mit dieser Strategie vernetzt man Referenzflächen vom Typ 1 und 4.

Für die Triangulierung der Referenzflächen vom Typ 2 und 3 wendet man die Anordnung von Dreiecken, wie in Abb. 5.11 gezeigt, an. Hierbei wird von Spantposition zu Spantposition die Gitterauflösung um eins erniedrigt.

In PaShiMo wurden diese Triangulierungsmethoden als Funktionen in der Datei *jship-tools.cpp* implementiert. Diese fünf Grundverfahren werden gesteuert über folgende Parameter:

- Zwei Parameter für die Anfangsspantposition und Endspantposition des Spant.
- Zwei Parameter für die jeweilige Auflösung an der Anfangsspant- und Endspantposition.
- Einem Parameter zur Festlegung der Teilfläche auf welche die Triangulierung angewandt werden soll.
- Einem Parameter für die Auflösung in Längsrichtung des Schiffes, welche für Triangulierungen von Referenzflächen vom Typ 5 benötigt werden.

Zur Triangulierung der Geometrie werden diese fünf verschiedenen Triangulierungsverfahren eingesetzt. Um hängende Knoten beim Übergang von einer Teilfläche zur nächsten Teilfläche zu vermeiden wird mit denjenigen Teilflächen begonnen, welche sich in Flächen zerlegen lassen vom Referenztyp 1 oder 2. Diese in Frage kommende Teilflächen werden beginnend mit dem kleinsten Spantkurvenindex  $i$  mit der passenden Methode trianguliert. Die sich hierbei ergebende Verteilung von Gitterknoten in Längsrichtung des Schiffes hängt von der gewählten Auflösung auf der Teilkurve  $s_i$  ab. Diese Gitterknotenverteilung in Längsrichtung des Schiffes wird auf die verbleibenden Teilflächen angewandt. Diese lassen sich in Flächen des Referenztyps 5 zerlegen. Für Flächen diesen Typ kann jede beliebige Gitterknotenverteilung angewandt werden, da deren Längsverteilung nicht von der Auflösung auf dem Spant abhängt.

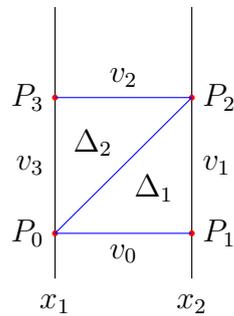


Abbildung 5.9: Beschriftung der Punkte für die Triangulierung

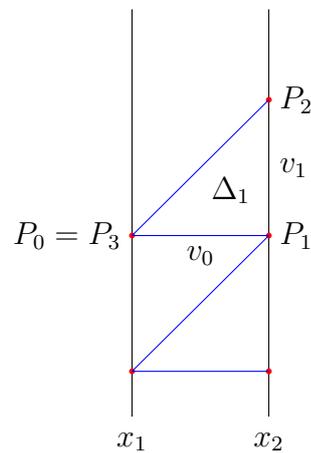


Abbildung 5.10: Erste Erweiterung der Triangulierung

---

**Algorithm 1** Grundalgorithmus
 

---

```

for all  $i \in V$  do
  compute nodes  $P_i$ 
  compute vectors  $v_i$ 
end for
for all  $j \in T$  do
  compute normals  $n_j$  for  $\Delta_j$ 
end for
for all  $j \in T$  do
  write nodes  $P_i$  for  $\Delta_j$ 
  write normals  $n_j$  for  $\Delta_j$ 
end for

```

---

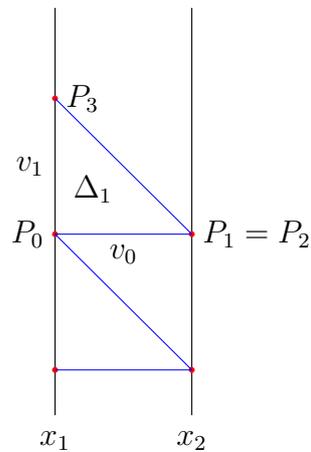


Abbildung 5.11: Zweite Erweiterung der Triangulierung

Auf diese Weise wird aus der analytischen Beschreibung der Geometrie eine triangulierte Oberfläche erzeugt, welche beliebig fein aufgelöst werden kann. Diese Vordiskretisierung erleichtert es dem Vernetzer, die Geometrie zu rekonstruieren. Da diese Aufgabe von PaShiMo durchgeführt wird, ist außerdem gesichert, dass die Geometrie richtig aufgebaut wird. Die Ausgabe der diskretisierten Geometrie erfolgt in Form der oben beschriebenen stl Datei.

In Abb. 5.12, 5.13 und 5.14 sind die Ergebnisse der aufbereiteten Geometrie abgebildet. Der große Vorteil dieses Vorgehens ist die Wiederverwendbarkeit einer einmal parametrisierten Triangulierung. Für eine automatische Geometrieerzeugung heißt dies, dass die Erzeugung der Vordiskretisierung automatisiert erfolgt, falls ein Parameter der Schiffkörpermodellierung seinen Wert ändert.

Außerdem erleichtert diese Vordiskretisierung den Austausch der erzeugten Geometrien mit CAD-Programmen. Somit entfällt die Implementierung eines Betrachtungsprogramms für die Geometrie.

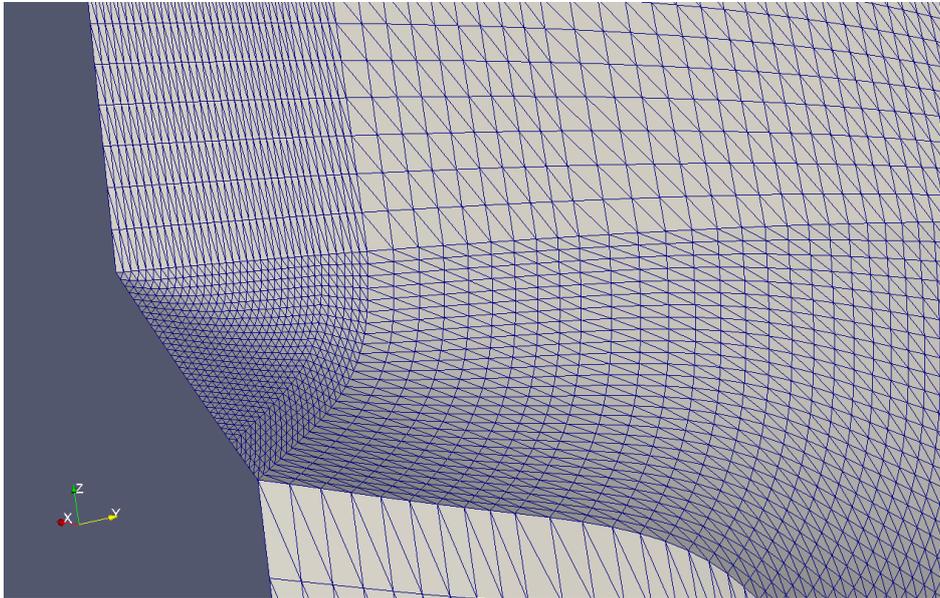


Abbildung 5.12: Auflösung der Übergänge verschiedener Flächen

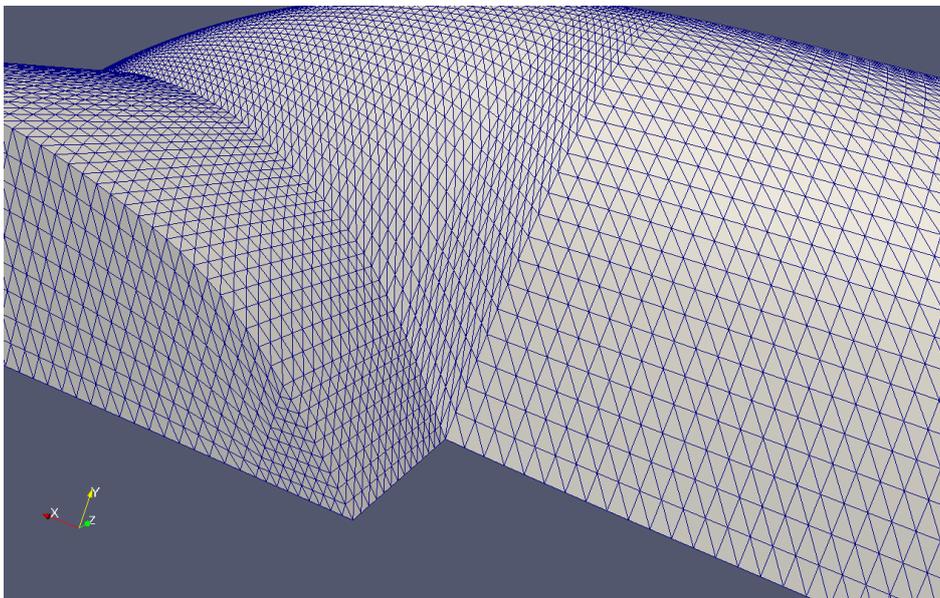


Abbildung 5.13: Auflösung der Übergänge verschiedener Flächen

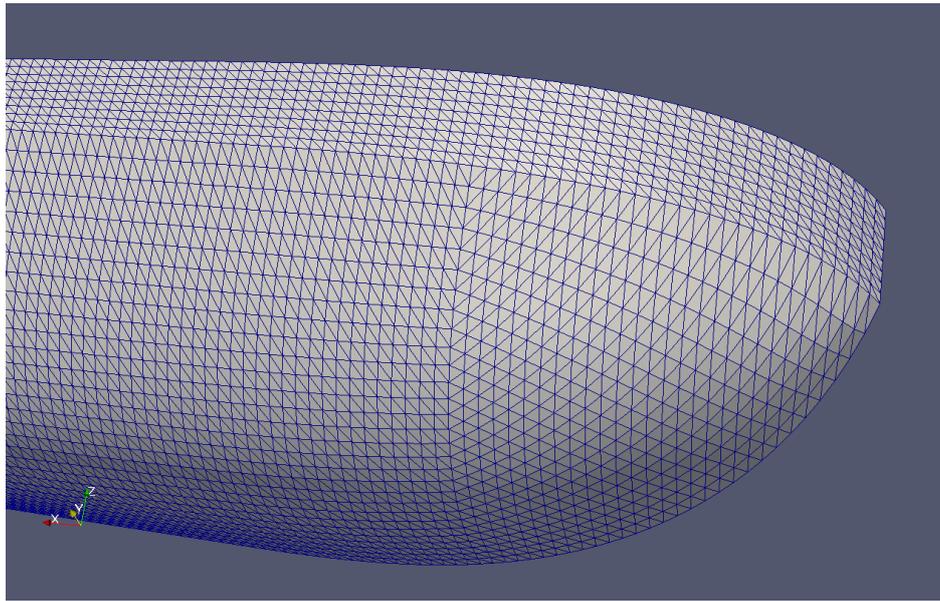


Abbildung 5.14: Auflösung der Übergänge verschiedener Flächen

## 5.11 Modellierte Schiffsgeometrien

In diesem Abschnitt werden die mit PaShiMo modellierten Schiffsmodelle vorgestellt. Ein wichtiges Ziel im Vorhaben SimuVSP war die Modellierung von Schiffen mit VSP. Eine Besonderheit von Schiffen mit VSP ist die ebene Bodenplatte, an welcher der VSP integriert wird.

Es wurden die folgenden Modelle mit Hilfe von PaShiMo modelliert:

- einfacher Wassertrecker
- einfacher Wassertrecker mit Volumenparameter
- Wassertrecker mit Finne
- Wigleyschiff

Das erste Modell, welches in PaShiMo umgesetzt wurde, war das einfache Modell des Wassertreckers. Als einfach wird in diesem Fall der Wassertrecker ohne Anbauteile verstanden. Anbauteile können beispielsweise eine Finne oder eine Schutzpatte sein. In Abb. 5.15 ist der Grundspant eines solchen Wassertrecker abgebildet. Der Spant dieses Modells besteht aus fünf Teilflächen. Die fünfte Teilfläche ergibt sich durch den Abschluss der Geometrie, welche durch die gestrichelte Abschlusslinie angedeutet ist. Es wurden zwei Variationen dieser Geometrie umgesetzt. Da der Punkt  $D$  für den Wassertrecker oberhalb der Wasseroberfläche liegt und der Widerstand der Luft gegenüber dem Widerstand im Wasser vernachlässigt werden kann, wurde ein Modell dieses Wassertreckers umgesetzt, welches bereits mit Punkt  $D$  abschliesst. Aus dieser Unterscheidung entstanden beide Variationen. Das Modell dieses Wassertreckers

besitzt eine Bodenplatte, welche mit der Kurve zwischen den Punkten  $A$  und  $B$  modelliert wird. Danach folgt ein gebogenes Stück, welches sowohl über Wasser als auch unter Wasser verläuft. Zwischen den Punkten  $C$  und  $D$  verläuft ebenfalls ein lineare Teilkurve. Das Schanzkleid wird modelliert durch die lineare Teilkurve zwischen den Punkten  $D$  und  $E$ . In diesem einfachen Modell des Wassertrecker wird die Teilkurve zwischen den Punkten  $B$  und  $C$  durch die Anfangs- und Endpunkte sowie dem Tangentialwinkel am Anfang und am Ende der Teilkurve modelliert.

Eine erste Erweiterung dieses Modells erfolgte durch das Hinzufügen eines Parameters, welcher den Flächeninhalt der Fläche zwischen der  $z$ -Achse der Wasserlinie und der Spantkurve beschreibt. Dieser Flächeninhalt, integriert über die Länge des Schiffes, beschreibt die Verdrängung des Schiffes. Dies ist ein wichtiger Parameter für das Schiff, da dieser die maximale Traglast des Schiffes definiert. Für eine spätere Simulation spielt dieser Parameter eine Rolle, da eine Optimierung ohne Festhalten der Verdrängung zu einem Schiff mit möglichst geringer Verdrängung führt. Das resultierende Schiff wird ohne eine solche Restriktion auf der Symmetrieebene verschwinden.

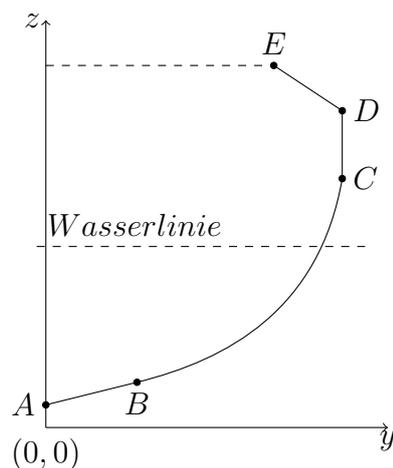


Abbildung 5.15: Spantmodell für den einfachen Wassertrecker

Das nächste umgesetzte Modell war der Wassertrecker mit Anbauteil. Das Anbauteil war in diesem Fall die Finne. In Abb. 5.16 ist ein Spantmodell dieses Modells dargestellt. Die Position des Punktes  $B_1$  wird durch drei Designlinien kontrolliert. Mit Hilfe dieser ist ebenfalls der Punkt  $A_1$  bestimmt. Analog zum einfachen Wassertrecker werden  $A$  und  $B$  durch Parameter beschrieben. Die Position des Punktes  $C$  wird mit Hilfe der Projektion von  $B_1$  auf die Teilkurve zwischen den  $B$  und  $D$  berechnet. Die Teilkurve zwischen  $B$  und  $D$  wird hierbei analog zum einfachen Modell zusätzlich durch den Anfangs- und Endwinkel gesteuert. Die Position des Punktes  $D$  und  $E$  wird durch Parameter beschrieben. Dieses Modell kann maximal aus sechs Teilflächen bestehen.

Ein weiteres umgesetztes Modell ist dem Wigleyschiff ähnlich. Es besitzt eine einfache Geometrie. Bei diesem Modell werden die Punkte  $A$  und  $B$  durch Parameter

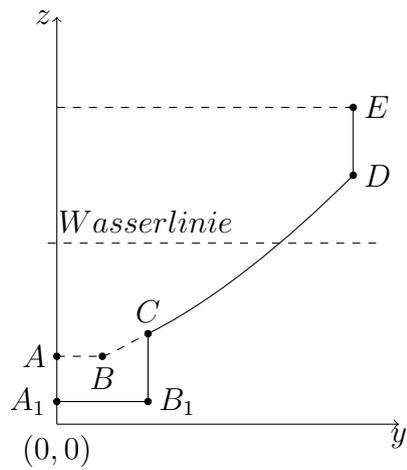


Abbildung 5.16: Spantmodell für den Wassertrecker mit Finne

beschrieben. Ausserdem wird der Anfangs- und Endwinkel des Kurvenstücks zwischen den Punkten  $A$  und  $B$  durch Parameter kontrolliert. Dieses Modell wurde um einen

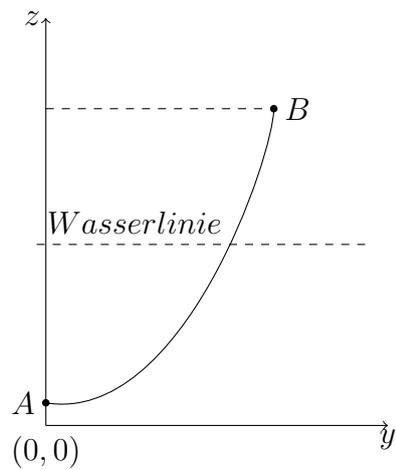


Abbildung 5.17: Spantmodell für das Modell Wigleykind

Parameter zur Steuerung des Flächeninhalts erweitert.

# Kapitel 6

## Simulation und Optimierung

In diesem Kapitel wird auf die Simulation und Optimierung von Schiffskörpergeometrien in dem Vorhaben SimuVSP eingegangen. Für eine erfolgreiche Optimierung wird eine robuste Simulationsmethode benötigt, welche Informationen zur Bewertung der Geometrie liefert. Auf Basis dieser Bewertung erfolgt die Optimierung. Ziel der Optimierung ist die Verbesserung der Geometrie. Damit jedoch eine Optimierung automatisiert erfolgen kann, muss ein Simulationsablauf gefunden werden, welcher robust automatisiert werden kann. Unter robust wird in diesem Fall verstanden, dass der Durchlauf einer solchen Simulation ohne manuellen Eingriff erfolgt. Die Teilschritte, welche zu einer solchen erfolgreichen automatisierten Optimierung führen, werden in den nächsten Abschnitten dieses Kapitel besprochen.

### 6.1 Simulation

Unter der Simulation wird ein klar definierter Prozess verstanden, welcher sich in verschiedene Schritte unterteilt. Ausgang für eine Berechnung stellt immer eine parametrische Geometriemodellierung mit dem Parametersatz einer Realisierung dar. Unter einer Realisierung ist eine Geometrie zu verstehen, welche mit einem Modell umgesetzt wurde und von einem Parametersatz beschrieben wird. Ausgehend von einem solchen Datensatz einer Realisierung wird mit Hilfe von PaShiMo die Geometrie berechnet. Diese generierte Geometrie wird anschliessend dem Vernetzer übergeben, welcher auf Basis dieser Geometrie ein dreidimensionales Rechennetz des Lösungsraums erstellt. Dieses Rechennetz wird im Anschluss dem Strömungslöser übergeben. Dieser liest das Gitter ein und erstellt das Strömungsproblem für gegebene Einstellungen für den Löser. Nach erfolgreicher Erstellung des Strömungsproblem wird die Berechnung mit Comet gestartet. Da bei der Lösung des Strömungsproblems iterative Methoden benutzt werden, erfolgt nach einer vorgegebenen Anzahl von Iterationen der Abbruch der Berechnung. Nach Abbruch der Berechnung erfolgt die Berechnung der relevanten Größe für die Bewertung der Geometrie. Die für unsere Analyse wichtige Widerstandskraft wird über die Drücke auf der Schiffsaussenhaut in Bewegungsrichtung des Schiffes, d.h. in  $x$ -Richtung, berechnet. Diese Berechnung erfolgt mit Hilfe des Usercodings von Comet (siehe Kap. 4). In einem letzten Schritt erfolgt die Ausgabe dieser berechneten Größe,

welche die Bewertung der Geometrie darstellt.

Da der beschriebene Ablauf der Simulation in eine Optimierungskette eingebaut werden soll, muss dieser nun automatisiert werden. Hierzu wurden verschiedene Möglichkeiten betrachtet und experimentell untersucht. Das größte Problem stellte hierbei die Übergabe der Geometrieinformationen des Schiffmodells an den Vernetzer dar. Die Übergabe der Geometrie des Schiffskörpers an den Vernetzer erfolgte mit Hilfe eines Oberflächennetzes. Es handelt sich hierbei um eine Beschreibung der Geometrieoberfläche durch Dreiecke in Form einer STL Geometrie (siehe Kap. 5). Diese STL Geometrie kann ohne weitere Bearbeitungsschritte von dem Vernetzungswerkzeug *vmesh* eingelesen werden. Es ergibt sich somit folgende Zusammenfassung der einzelnen Schritte der Simulation:

1. Geometrieerstellung und Ausgabe als Oberflächennetz
2. Vernetzung des Strömungsraumes auf Grundlage des Oberflächennetzes
3. Konfiguration des Strömungsproblems über den Preprozessor
4. Lösung des Strömungsproblems
5. Abbrechen der Berechnung und Auswertung des Zielfunktional auf Basis der Lösungsdaten (erfolgt nach jeder Iteration mit Hilfe einer Erweiterung des Löser)

Die Steuerung dieser Simulation erfolgt über ein bash shell Skript, welches die einzelnen Programme aufruft. Dies ist mit Hilfe eines Scripts möglich, da sowohl Geometriemodellierer und Vernetzer als auch Strömungslöser über Kommandozeilenbefehle gesteuert werden können, siehe Listing 6.1. Die vollständige Simulation erfolgt mit Hilfe des Skripts automatisiert und bedarf keinen manuellen Eingriff von außen.

Listing 6.1: Perl Skript Simulationskette

```
#!/usr/bin/perl
# SETTING THE PATHS
use Cwd;
my $abs_path = getcwd();
$ENV{tmp_work_dir} = $abs_path;
my $comet_calculation = "$abs_path/comet_calculation";

# GENERATE STL SURFACE WITH pashimo
chdir("./pashimo_tool") or die "ERROR: _path_setting";
system("make");
system("./pashimo_stl");

# RESET DIRECTORY
chdir($abs_path);

# GENERATE MESH WITH vmesh
chdir("./vmesh") or die "ERROR: _path_setting";
system("make");
```

```

system(" ./vmesh_automation_auto" );
system(" cubit -nogui -nographics -input .../ file.jou" );
system(" ./vmesh_v29_2_par" );
system(" cubit -nogui -nographics -input .../ middle_layer" );
system(" ./vmesh_complete" );

# RESET DIRECTORY
chdir($abs_path);

# COPY DATA FOR COMET
system(" cp .../ vsp.echo .../ actual.echo" );
system(" cp .../ vmesh.cel .../ vmesh.cel" );
system(" cp .../ vmesh.vrt .../ vmesh.vrt" );

# CONFIGURE COMET BY PREPROCESSOR
chdir(" comet_calculation") or die "ERROR: _path_setting";
system(" Cometpp -nogui << actual.echo" );
system(" lcomet << lcomet_commands.txt" );

# START COMET
system(" ./Comet << casefile" );

# RESET DIRECTORY
chdir($abs_path);

# STOP COMET BY CRITERIA AND GET VALUE
chdir(" comet_start_stop") or die "ERROR: _path_setting";
system(" make" );
system(" ./opti_chain" );

# RESET DIRECTORY
chdir($abs_path);

```

In Abb. 6.1 sind die Ergebnisse einer solchen Simulation mit Rechengitter zu sehen, welche mit Hilfe der hier beschriebenen Methode gewonnen wurden. Auf dem Bild ist ebenfalls ein Schnitt durch das Gitter auf der Symmetrieebene zu sehen. Um die Umströmung des Rumpfes mit Wasser sichtbar zu machen, wurden die Stömungslinien in Rumpfnähe visualisiert, siehe Abb. 6.2.

Beide Bilder wurden mit dem Programm Paraview erstellt. Dieses Programm entstand aus einem Projekt von Kitware Inc. und Los Alamos National Laboratory in den USA, [41]. Es dient zur Visualisierung großer Datenmengen aus Simulationen und ist für viele verschiedene Plattformen erhältlich. Zur Generierung der Bilder mit Paraview mussten in einem separaten Schritt die Daten aus dem Strömungslöser in ein anderes Datenformat (EnSight) umgewandelt werden. Es ist anzumerken, dass Paraview ein open-source Projekt ist und somit frei zugänglich ist.

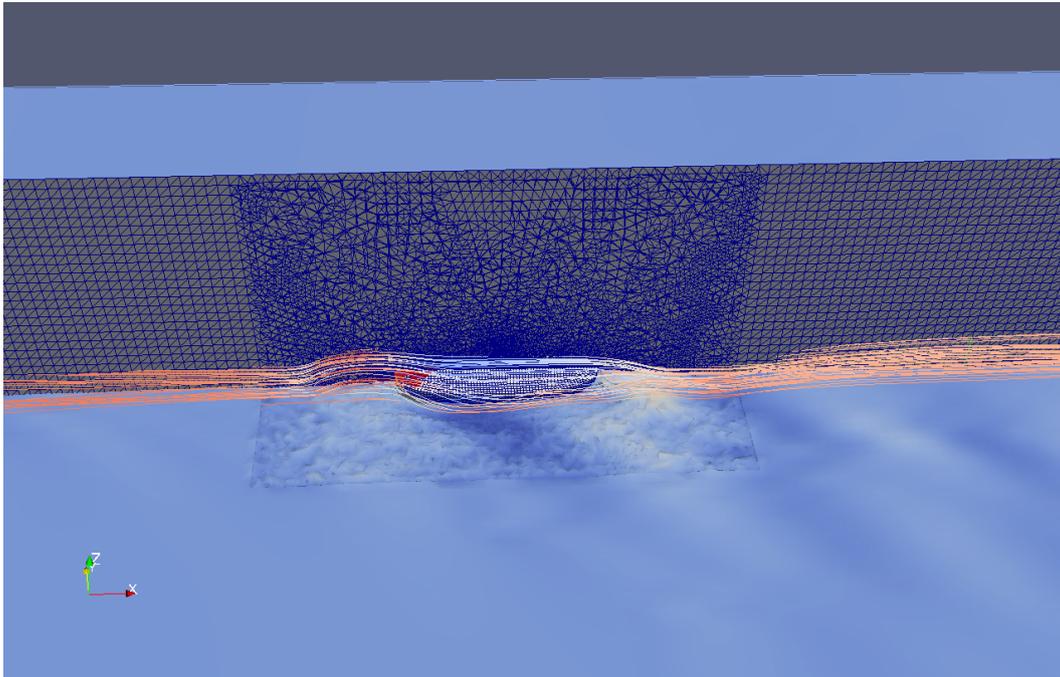


Abbildung 6.1: Visuelles Ergebnis einer Simulation mit verwendetem Rechengitter auf der Symmetrieebene und den visualisierten Strömungslinien um die Geometrie

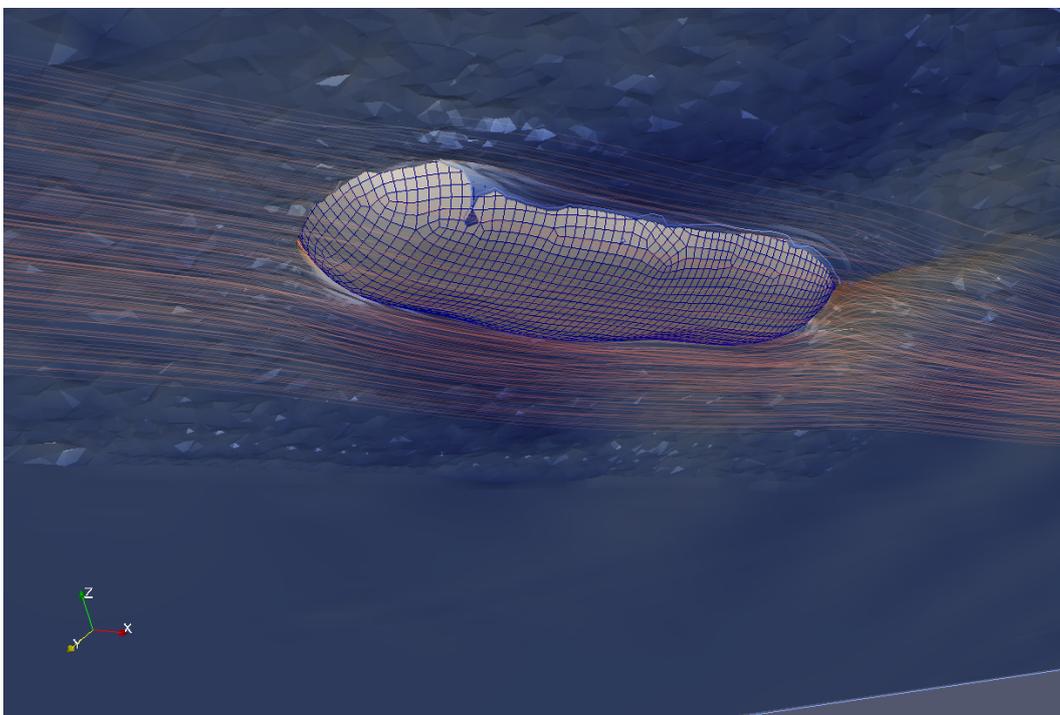


Abbildung 6.2: Visuelles Ergebnis der Simulation mit Strömungslinien

## 6.2 Optimierung

Die Optimierung erfolgt auf Grundlage der Simulationskette die in dem vorherigen Abschnitt vorgestellt wurde. Generell stellt sich das Problem der Optimierung als das Auffinden eines Minimums oder Maximums von einem Zielfunktion  $F$  in einem Suchraum  $\Omega \subset \mathbb{R}^n$  dar

$$\min_{x \in \Omega} F(x) \quad \text{oder} \quad \max_{x \in \Omega} F(x) \quad (6.2.1)$$

In diesem Fall wird nach einer optimalen Geometrie gesucht. Da jedoch eine Geometrie allgemein schwer zu fassen ist, wurde in diesem Projekt eine parametrische Beschreibung der Geometrie erarbeitet. Diese Beschreibung einer Geometrie mit Hilfe von Parametern  $\omega \in \mathbb{R}^n$  ermöglicht es, diese Geometrie für eine mathematische Optimierung zu benutzen. Das Zielfunktional ergibt sich in diesem Fall über die Verarbeitung der Ergebnisse einer Strömungssimulation. Verschiedenste über die Lösungsdaten der Simulation konstruierten Funktionale können eingesetzt werden. Abhängig von der Problemstellung ergibt sich eine Minimum- oder Maximumsuche.

In dieser Arbeit fiel die Entscheidung auf die Minimierung der Widerstandskraft einer solchen Geometrie, d.h.

$$\min_{x \in \Omega} F(x), \quad (6.2.2)$$

wobei in diesem Fall  $F$  die Kraft in Fahrtrichtung des Schiffes ( $x$ -Richtung) ist.

Die Berechnung der Widerstandskraft des Schiffes bzw. der Geometrie erfolgt mit Hilfe der obigen Simulation. Diese liefert für eine korrekt konfigurierte Simulation für genügend große Geometrieänderungen Ergebnisse mit registrierbaren Abweichungen. Unter einer korrekt konfigurierten Simulation wird ein Simulationsproblem verstanden, welches mit Hilfe der im vorherigen Abschnitt beschriebenen Methode simuliert wird. Ausgehend von dieser Tatsache können sich dennoch zwei Szenarien ereignen:

- Die Geometrieänderung ist ausreichend groß, jedoch sind die Auswertungen der Zielfunktionale fast identisch.
- Die Geometrieänderung ist klein, jedoch liegen die Auswertungen der Zielfunktionale weit auseinander.

Beide Phänomene wurden in diesem Projekt beobachtet und stellen ein großes Problem dar, da diese nicht vorhersehbar sind und eine erfolgreichen Optimierung erschweren. In Abb. 6.3 ist dieser Sachverhalt dargestellt. Diese beide Beobachtungen lassen sich an Hand von zwei Beispielen erklären. Eine Veränderung der Geometrie kann eine neue Geometrie entstehen lassen, welche einen Wirbel erzeugt, der den Widerstand an einer Stelle der Geometrie minimiert, jedoch an einer anderen Stelle der Geometrie stromabwärts einen Wirbel verursacht, welcher die Verbesserung kompensiert. Ebenso kann durch eine minimale Veränderung der Geometrie ein Strömungsabriss verursacht werden, welcher einen Sprung in der Lösung generiert.

Beide Beispiele sind der Tatsache geschuldet, dass bis heute keine Aussagen über die globale Existenz und Eindeutigkeit der dreidimensionalen Navier-Stokes-Gleichungen

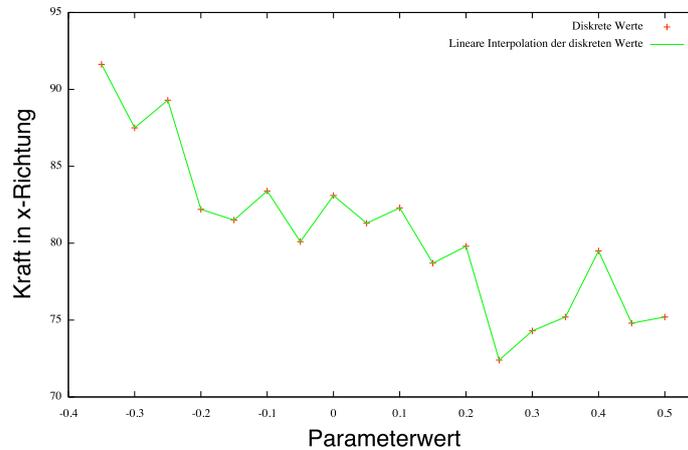


Abbildung 6.3: Designstudie mit instabilen Ergebnissen

existieren. Dieses Problem gehört zu einem der sieben festgesetzten Millennium Probleme vom Clay Mathematics Institut in Cambridge (Massachusetts), [7]. Insbesondere liegt in diesem Fall ein Stabilitätsproblem vor, welches sich aus der verwendeten Lösungsmethode für die Navier-Stokes-Gleichungen ergibt.

Die Widerstandskraft in  $x$ -Richtung des Schiffes erhält man über Summation der Kraftwerte in  $x$ -Richtung einer Lösung eines Zeitschritts, [19]. Da es sich hier um die Simulation eines instationären Problems handelt, wird der für die Optimierung relevante Wert über eine Mittelwertbildung der letzten  $n \in \mathbb{N}$  Zeitschritte gewonnen. Hierbei wird die Tatsache verwendet, dass nach einer genügend großen Anzahl  $N \in \mathbb{N}$  ( $N > n$ ) von Zeitschritten die Lösung als auskonvergiert gilt

$$\tilde{F} = \frac{1}{n} \sum_{i=0}^{n-1} F_{N-i}, \tag{6.2.3}$$

wobei  $\tilde{F}$  die gemittelte Widerstandskraft und  $F_i$  die Widerstandskraft im  $i$ -ten Zeitschritt bezeichnet. Der auf diese Weise berechnete Wert  $\tilde{F}$  der Widerstandskraft stellt das zu minimierende Zielfunktional dar. Der Suchraum ist gegeben durch die Parameter der Modellierung. Da in der Regel nur jeweils eine Teilmenge der Parameter gleichzeitig verändert werden und somit frei sind in der Optimierung ergibt sich das folgende Optimierungsproblem.

**Definition 6.2.1 (Optimierungsproblem)** Sei  $\Omega \subset \mathbb{R}^P$  der gesamte Parameterraum,  $\omega \in \Omega$  eine Realisierung der Geometriemodellierung und  $I = \{i_1, \dots, i_k\}$  ( $|I| = k < P$ ) die Indexmenge der zu betrachteten Optimierungsparameter, dann lautet das Optimierungsproblem

$$\min_{x \in \tilde{\Omega}} \tilde{F}(x), \tag{6.2.4}$$

wobei  $x = (\omega_{i_1}, \dots, \omega_{i_k})$  aus dem verkleinerten Parameterraum  $\tilde{\Omega} \subset \mathbb{R}^k$  und  $\tilde{F}$  die gemittelte Widerstandskraft über die letzten Zeitschritte ist.

Aufgrund der Mittelwertbildung und der nicht voraussetzbaren Glattheit der Lösung des Strömungsproblems kann  $\tilde{F}$  ebenfalls nicht zwingend als glatt angenommen werden. Dies muss bei der Wahl des Optimierungsverfahrens berücksichtigt werden.

## 6.3 Direkte Suchverfahren

Bei dieser Klasse von Optimierungsverfahren wird die Glattheit des Zielfunktionals nicht benötigt. Daher stellen Verfahren aus dieser Klasse gute Kandidaten für das in dieser Arbeit betrachtete Optimierungsproblem dar. Im Folgenden werden einige Methoden aus dieser Klasse beschrieben.

### Verfahren von Hooke und Jeeves

Das direkte Suchverfahren von Hooke and Jeeves beruht auf sequentiellen Auswertungen des Zielfunktionals, [18].

Ausgehend von einer Startlösung  $x_{(0)}$  aus dem Suchraum  $\mathbb{R}^n$ , einer Schrittweite  $\delta > 0$  und den Suchrichtungen  $e_i \in \mathbb{R}^n$ , welche die  $i$ -ten Einheitsvektoren darstellen, wird in der Nachbarschaft nach einem fest vorgegeben Schema, dem sogenannten Pattern, nach einer besseren Lösung gesucht. Bei diesem Vorgehen werden sukzessiv neue Lösungen  $x_{(i)} \in \mathbb{R}^n$  erzeugt und es ergibt sich bei der Minimasuche eine Folge von Lösungen

$$\tilde{F}(x_{(0)}) > \tilde{F}(x_{(1)}) > \dots > \tilde{F}(x_{(i)}) > \tilde{F}(x_{(i+1)}) > \dots \quad (6.3.1)$$

Das Verfahren bricht ab, falls keine neue Verbesserung gefunden werden, oder eine vorgegebene Genauigkeit für ein  $\varepsilon > 0$  erreicht ist

$$|\tilde{F}(x_{(i)}) - \tilde{F}(x_{(i+1)})| < \varepsilon. \quad (6.3.2)$$

Das Auffinden einer verbesserten Lösung ist in zwei Schritte unterteilt. In dem Erkundungsschritt werden ausgehend von einer Lösung  $x_{(i)}$  nacheinander die Nachbarpunkte

$$x_{(i)} + e_i \cdot \delta_1 \text{ und } x_{(i)} - e_i \cdot \delta_1 \text{ für } i = 1, \dots, n \quad (6.3.3)$$

mit der Schrittweite  $\delta_1 = \delta$  ausgewertet und nach Verbesserungen untersucht. Sobald in diesem Schritt eine Verbesserung gefunden wird, setzen wir diese Lösung als neue temporäre Lösung  $x'_{(i)}$  mit

$$\tilde{F}(x_{(i)}) > \tilde{F}(x'_{(i)}). \quad (6.3.4)$$

Kann keine verbesserte Lösung für alle  $i$  gefunden werden, so wird dieser Erkundungsschritt mit den Schrittweiten  $\delta_k = \frac{\delta}{2^{(k-1)}}$  mit  $k = 2, \dots$  wiederholt bis eine Verbesserung  $x'_{(i)}$  gefunden wird oder das Verfahren für  $\delta_k < \varepsilon$  abbricht, da die Genauigkeit  $\varepsilon$  erreicht ist. Die letzte temporäre Lösung in einem solchen Erkundungsschritt ergibt die neue Lösung  $x_{(i+1)}$ . Nach dem Auffinden einer verbesserten Lösung  $x_{(i+1)}$  wird die erfolgreiche Suchrichtung  $e^*$  erneut getestet bevor mit der neuen Lösung  $x_{(i+1)}$  ein Erkundungsschritt durchgeführt wird.

Der Schritt in welchem entweder die Verkleinerung der Schrittweite oder das erneute Ausführen der erfolgreichen Richtung durchgeführt wird, heißt Fortschrittsschritt.

### Verfahren von Nelder und Mead

Bei dem Verfahren von Nelder-Mead erfolgt die Optimierung mit Hilfe eines Simplexes. Ausgehend von endlich vielen Punkten

$$x_{(1)}, \dots, x_{(n+1)} \in \mathbb{R}^n, \quad (6.3.5)$$

welche geometrisch anschaulich einen Simplex aufspannen des  $\mathbb{R}^n$ , werden diese Punkte nach bestimmten Regel verschoben. Dies verursacht eine Verschiebung, Deformation oder Skalierung des Ausgangssimplexes bis sich ein hinreichend kleiner Simplex ergibt, [32]. Über die Ecken dieses Simplex erhält man die Lösung und den optimalen Wert des Zielfunktional

$$\tilde{F}^* = \frac{1}{n} \sum_{i=1}^n \tilde{F}(x_{(i)}) \quad (6.3.6)$$

Dieser Algorithmus arbeitet sehr effizient und kommt mit wenigen Funktionsauswertungen aus. Allerdings kann es während des Optimierungsvorgangs zu einer Deformation des Simplex kommen, so dass dieses entartet und keine Konvergenz eintritt.

### Genetische Algorithmen

In dieser Klasse der genetischen Algorithmen werden viele verschiedene Verfahren zusammengefasst. Allgemein versteht man unter einem genetischen Algorithmus ein heuristisches Suchverfahren, welches sich an Vorgängen der natürlichen Evolution orientieren. Diese Verfahren finden häufig Anwendung bei komplexen Fragestellungen, bei welche keine Stetigkeit oder Differenzierbarkeit des Zielfunktional vorausgesetzt werden kann. Genetische Algorithmen gehören zu der größeren Klasse evolutionärer Algorithmen. Diese emittieren die Vererbung, Mutation, Selektion und Kreuzung in der biologischen Evolution mit Hilfe von künstlichen Populationen, [29, 33].

### Simulated Annealing

Das Simulated Annealing ist eine stochastische Metaheuristik zur Bestimmung globaler Optimas. Es ist ein Verfahren welches mit Hilfe stochastischer Permutationen versucht Optimas zu bestimmen. Dabei toleriert das Verfahren ebenfalls vorübergehende Verschlechterungen um lokale Optimas zu verlassen und bessere Lösungen zu finden. Das Verfahren simuliert hierbei den Härtungsprozess in Stahl oder Glas. Daher leitet sich auch der Name dieses Verfahrens ab. Die Toleranz zur Akzeptanz von schlechteren Werten wird in diesem Verfahren über einen Abkühlungsparameter geregelt. Je weiter der Optimierungsvorgang voranschreitet, desto eher werden Verschlechterungen in der Lösung nicht toleriert, [24].

Auch das Simulated Annealing setzt keine Differenzierbarkeit der Zielfunktion voraus.

## 6.4 Gradientenbasierte Optimierungsverfahren

Diese Klasse von Optimierungsverfahren garantieren eine schnellere und effizientere Bestimmung von Optimas, setzen aber  $\tilde{F} \in \mathcal{C}^1(\mathbb{R}^n)$  voraus. Dies ist in diesem Fall nicht gegeben.

Ein Ausweg bietet der Ansatz über das Automatische Differenzieren. Dieses Verfahren kommt aus der Informatik und angewandten Mathematik und sieht die Berechnung der Ableitung über eine Ergänzung des Computerprogramms vor. Hierbei werden Ableitungen mit Hilfe der Kettenregel aus dem Programmcode generiert. Die Grundlage bildet hierbei die Tatsache, dass eine Funktion in einem Programm als eine Abfolge von Zwischenschritten beschrieben werden kann, welche für sich auf differenzierbare Elementarfunktionen wie  $\cos$ ,  $\sin$ ,  $\log$  beruhen. Die Berechnung der Ableitung erfolgt über die Verkettung der einzelnen Ableitungen der Elementarfunktionen, [16].

Mit Hilfe einer auf diese Weise berechneten Funktion lassen sich Suchverfahren wie das Newton Verfahren und das Verfahren von Fletcher und Reeve anwenden, [15, 34].

Zu der Anwendung von Automatischer Differentiation in der Strömungssimulation liegen bereits erste Ergebnisse vor, [28].

## 6.5 Optimierung auf Netzbasis

Außer dem parametergetriebenen Ansatz der Geometrieoptimierung wurde auch die Optimierung auf Netzbasis betrachtet. Diese Variante der Optimierung benutzt als Grundlage der Optimierung nicht die Geometriebeschreibung auf Basis der Parameter sondern ein beliebiges Rechengitter. Die Generierung einer neuen Geometrie erfolgt hierbei durch Verschiebung der Gitterknoten. Allerdings kennt diese Art der Geometriemodifikation ihre Grenzen, da allzu große Verzerrungen in den Zellen ein Gitter ergeben, welches nicht mehr berechnet werden kann. Für kleine Veränderungen hingegen liefert diese Methode einen guten Ansatz für eine Optimierung. Ein großer Vorteil ist hierbei, dass die Netztopologie unverändert bleibt.

Ein Netz ist hierbei gegeben über die Knoten  $R_i$  mit ihren Koordinaten  $\vec{r}_i \in \mathbb{R}^3$  ( $i = 1, \dots, N$ ) und wird in dem Knotenvektor

$$R = \left( \vec{r}_1, \dots, \vec{r}_N \right) \quad (6.5.1)$$

zusammengefasst. Durch Anwenden einer Transformation

$$T = \left( \vec{t}_1, \dots, \vec{t}_N \right) \quad (6.5.2)$$

mit Verschiebungsvektoren  $t_i \in \mathbb{R}^3$  für jeden Knoten  $R$  erhält man das neue Rechengitter

$$R' = \left( \vec{r}_1 + \omega_1 \vec{t}_1, \dots, \vec{r}_N + \omega_N \vec{t}_N \right) \quad (6.5.3)$$

mit geeigneten Faktoren  $\omega_i \in \mathbb{R}$  ( $i = 1, \dots, N$ ). Diese Faktoren bilden in diesem Ansatz die Optimierungsvariablen.

Eine geeignete Transformation ist gegeben mit

$$\vec{t}_i = \left\langle \vec{d}_k, \frac{\vec{n}_i}{\|\vec{n}_i\|} \right\rangle (1 - e^{-\frac{\lambda}{\|\vec{n}_i\|}}) \vec{n}_i \quad (6.5.4)$$

für eine normierte Deformationrichtung  $\vec{d}_k$  in einem festen Knoten  $R_k$  und Differenzvektor  $\vec{n}_i = (\vec{r}_i - \vec{r}_k)$ . Angewandt wird diese auf Knoten  $R_i$  welche sich innerhalb einer Kugel mit einem zu wählendem Radius  $\delta > 0$  um den Knoten  $R_k$  befinden. Mit der Variable  $\lambda$  wird der Einfluss des Abstandes eines Knoten  $R_i$  vom Deformationsknoten  $R_k$  gesteuert. Als Wert dieses Parameters  $\lambda$  wurde 1 gewählt.

Bei dieser Wahl der Deformation kommen nur die Faktoren  $\omega_i$  als Optimierungsvariablen in Frage für welche eine Deformation  $t_i$  definiert ist. Nach Auswahl von  $n$  Optimierungsvariablen ergibt sich auch in diesem Fall eine Problem der Gestalt

$$\min_{\omega \in \mathbb{R}^n} \tilde{F}(\omega). \quad (6.5.5)$$

# Kapitel 7

## Ergebnisse

In diesem Kapitel werden die Ergebnisse der Arbeit, welche in dem Projekt SimuVSP entstanden, zusammengestellt.

### 7.1 Beispiele von umgesetzten Schiffskörpermodellen in PaShiMo

Mit Hilfe von PaShiMo wurden verschiedene Schiffsmodelle umgesetzt. Zum Einen wurde die Geometrie eines einfachen klassischen Schleppers umgesetzt. Dieses wurde um ein komplexes Anbauteil, eine Finne, erweitert. Ausserdem wurde eine dem Wigley Schiff ähnliche Geometrie umgesetzt, welche zu Vergleichszwecken herangezogen wurde.

#### 7.1.1 Wigley Schiff

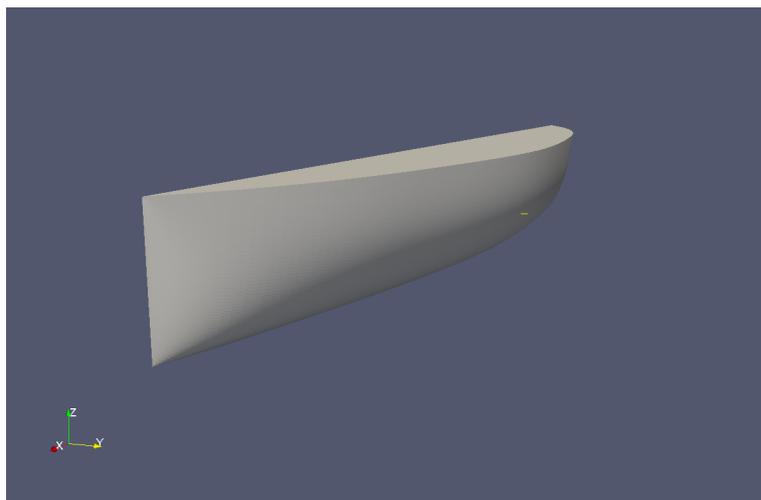


Abbildung 7.1: Vom Wigleyschiff abgeleitete Geometrie

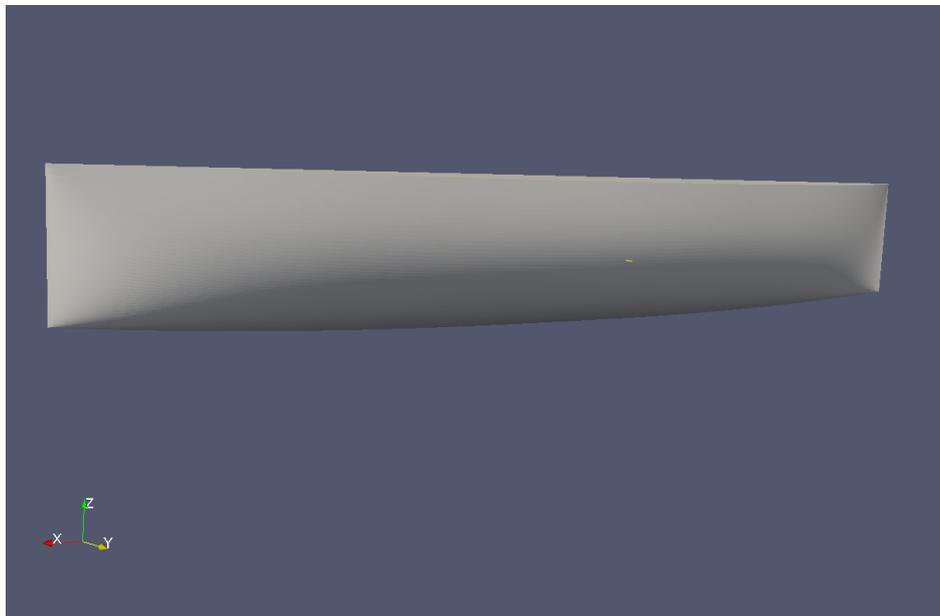


Abbildung 7.2: Wigleyschiffgeometrie von der Seite

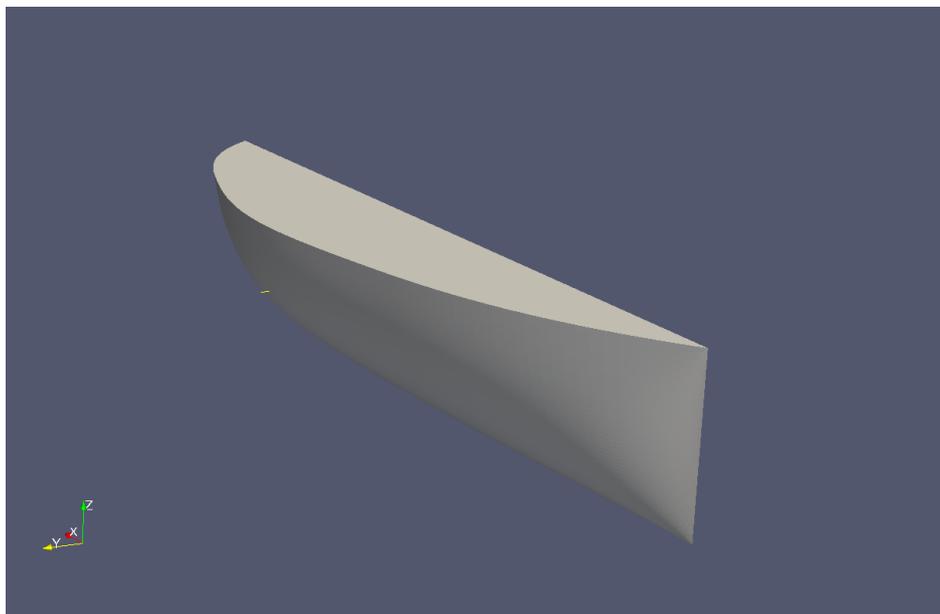


Abbildung 7.3: Vom Wigleyschiff abgeleitete Geometrie, Sicht von oben

### 7.1.2 Einfache Schlepper Geometrie

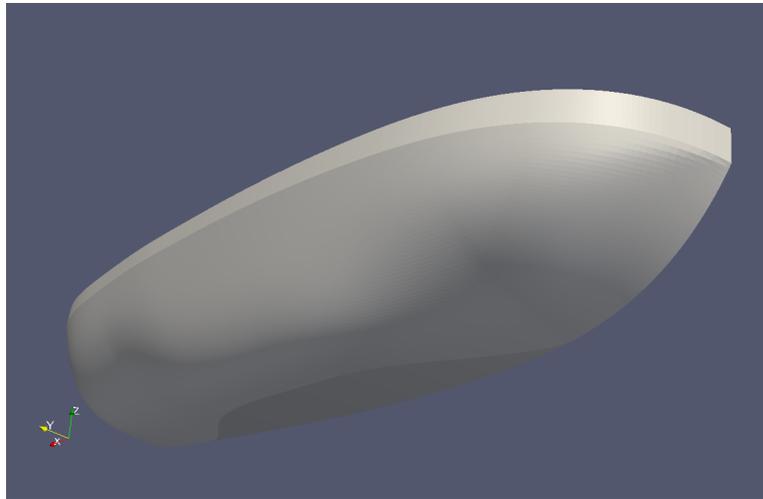


Abbildung 7.4: Geometrie eines Schleppers, Sicht von unten auf die Bodenplatte

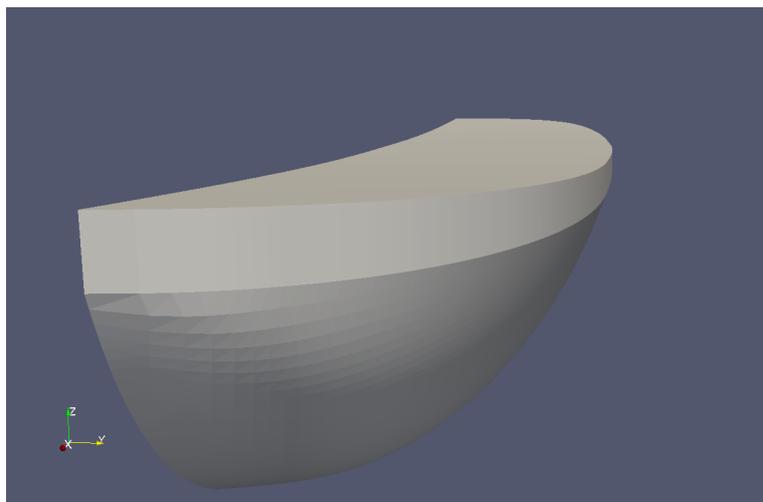


Abbildung 7.5: Geometrie eines Schleppers mit grober Vordiskretisierung, Sicht auf den Bug des Schiffes

### 7.1.3 Schlepper Geometrie mit Finne

Der in diesem Abschnitt gezeigte Schlepper basiert auf dem Modell des einfachen Schleppers. Allerdings wurde dieses Modell zusätzlich um eine Finne erweitert. Eine Finne ist für gewöhnlich an jedem Schlepper zu finden und ist daher ein Anbauteil, welches in jedem Fall mit dem Geometriemodellierung dargestellt werden muss. Dieses Modell ist von besonderer Bedeutung, da es zeigt wie flexibel der Modellierer erweitert werden kann.

In Abb. 7.6, 7.7 und 7.8 ist die Komplexität dieser Geometrie zu sehen. Insgesamt setzt sich die Geometrie dieses Modells aus sechs Teilflächen zusammen, welche zusammenhängend modelliert wurden.

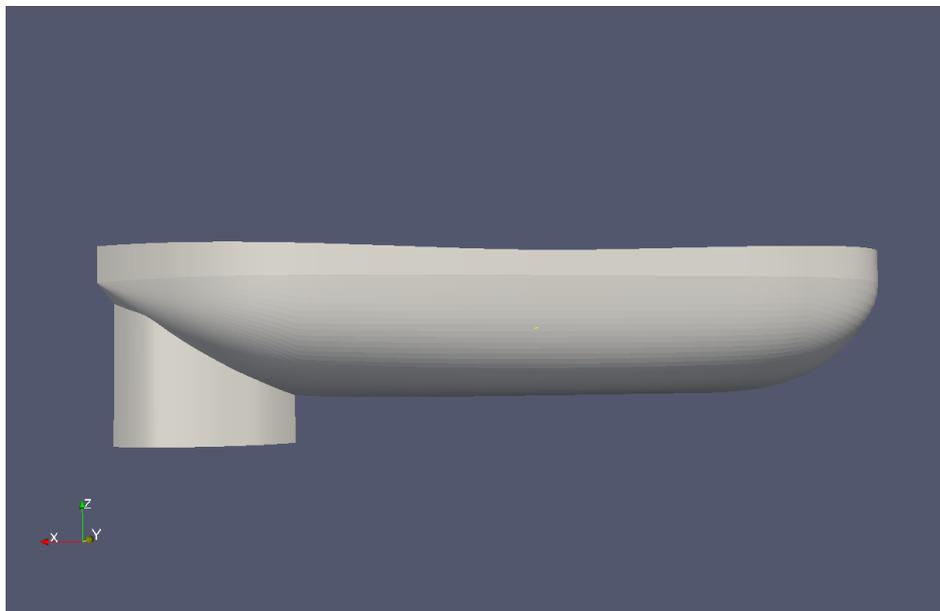


Abbildung 7.6: Geometrie des Schleppers mit Finne, Ansicht von der Seite



Abbildung 7.7: Geometrie des Schleppers mit Finne, Ansicht von oben auf das Heck

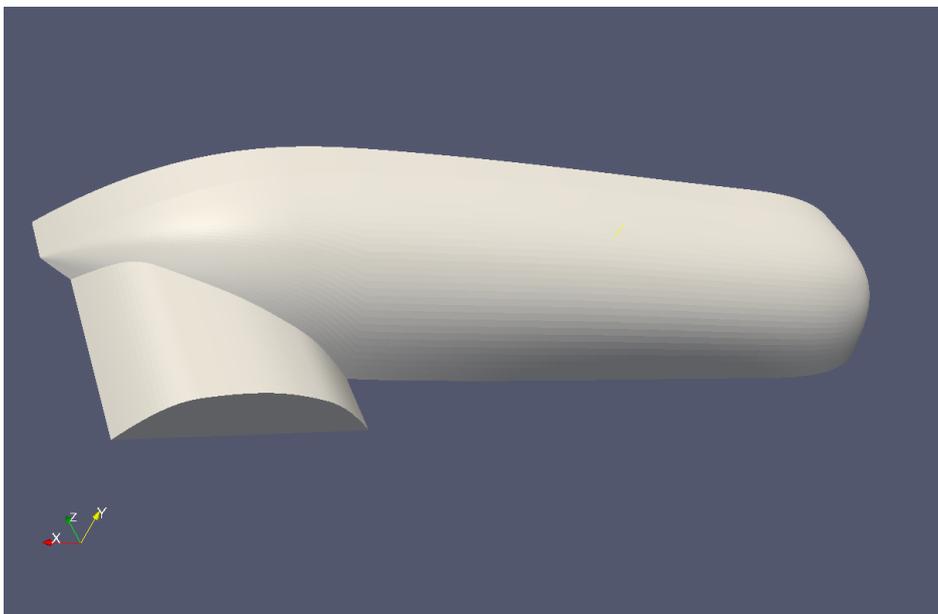


Abbildung 7.8: Geometrie des Schleppers mit Finne, Ansicht von unten auf das Heck

### 7.1.4 Schlepper Geometrie mit komplexen Anbauteilen

Das in diesem Abschnitt vorgestellte Modell ist eine Erweiterung des Schleppers mit Finne. Es soll demonstrieren, wie Anbauteile modelliert werden können, welche per Definition aus zwei Teilen bestehen, welche nicht verbunden sind. Die hierbei stark vereinfachte Bodenplatte dient zum Schutz der Voith Schneider Propeller. Durch dieses Modell wurde gezeigt, wie solche Geometrieerweiterungen mit dem Modellierungswerkzeug gehandhabt werden können. Mitunter wurde eine experimentelle Geometrie mit Knick in der Finne umgesetzt, siehe Abb. 7.10.

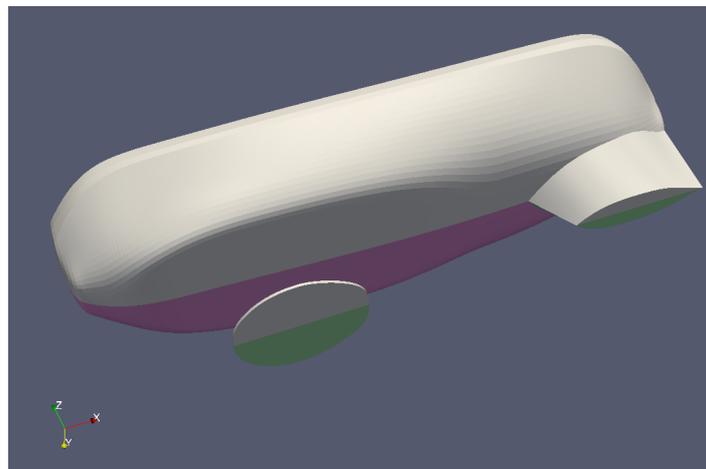


Abbildung 7.9: Geometrie des Schleppers mit Finne und Bodenplatte, Ansicht von unten

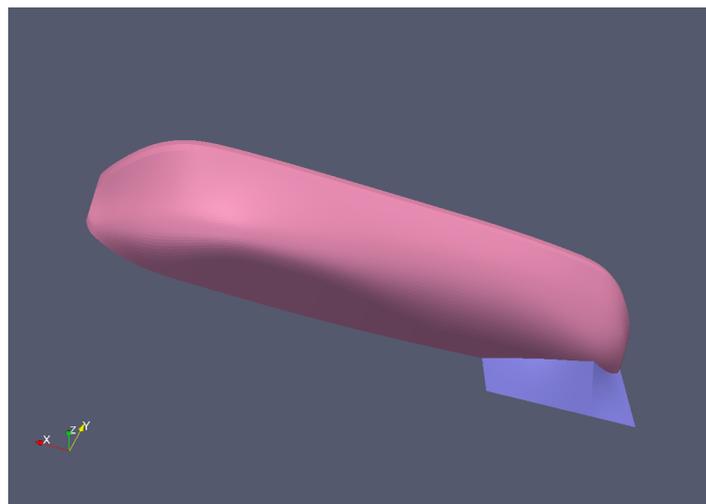


Abbildung 7.10: Experimentelle Geometrie für einen Schlepper mit Finne, Ansicht von unten

### 7.1.5 STL Triangulierung von Schiffen

In diesem Abschnitt werden die triangulierten Geometrien vorgestellt. In Abb. 7.11 ist eine gleichmäßige Triangulierung der Oberfläche zu sehen. Für die Simulationskette wird die triangulierte Geometrie benötigt, da diese eine geschlossene Geometrie garantiert und von dem Vernetzungswerkzeug eingelesen werden kann. In Gegensatz hierzu ist in Abb. 7.12 eine ungleichmäßige Triangulierung zu sehen. Diese kann nicht von dem Vernetzer eingelesen werden, da die Teilflächen nicht miteinander verbunden sind. Um eine derartige Geometrie für eine Simulation mit der in dieser Arbeit vorgestellten Simulationskette zu verwenden, muss diese zunächst vernäht werden.

Um die Güte des Oberflächennetzes zu visualisieren, sind in Abb. 7.13, 7.14 und 7.15 Nahaufnahmen des Netzes zu sehen. Alle Übergänge werden ohne hängende Knoten aufgelöst.

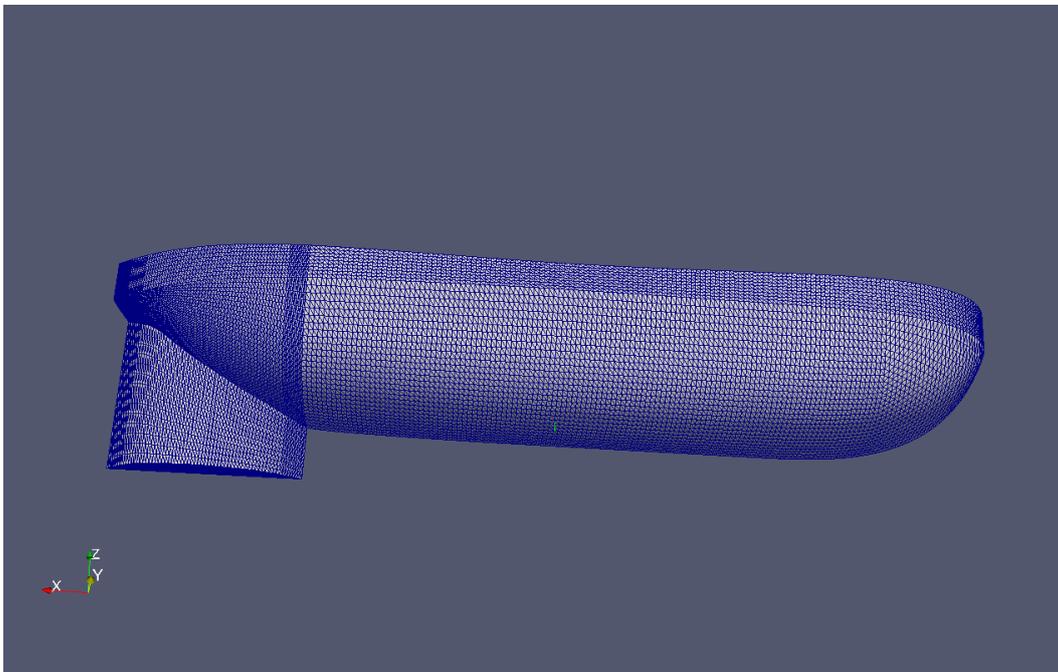


Abbildung 7.11: Gleichmäßig triangulierte Geometrie des Schleppers mit Finne

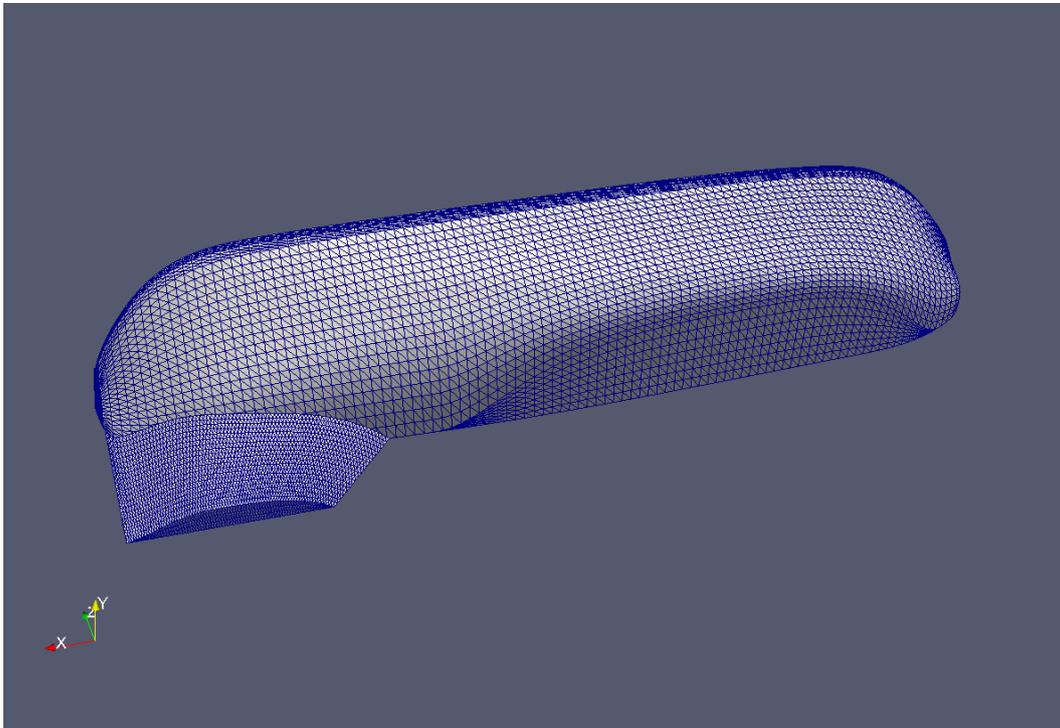


Abbildung 7.12: Ungleichmäßig triangulierte Geometrie des Schleppers mit Finne

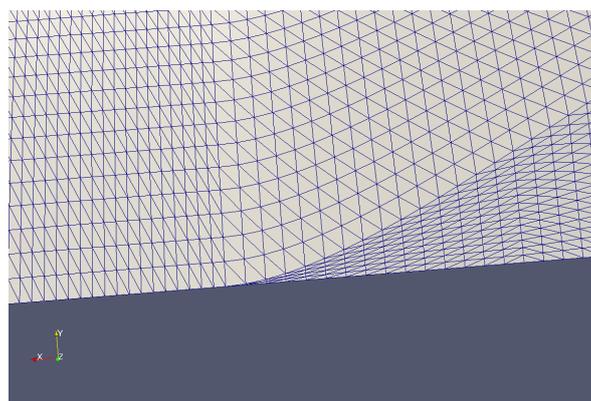


Abbildung 7.13: Nahaufnahme der gleichmäßig triangulierten Geometrie: Beginn der Bodenplatte

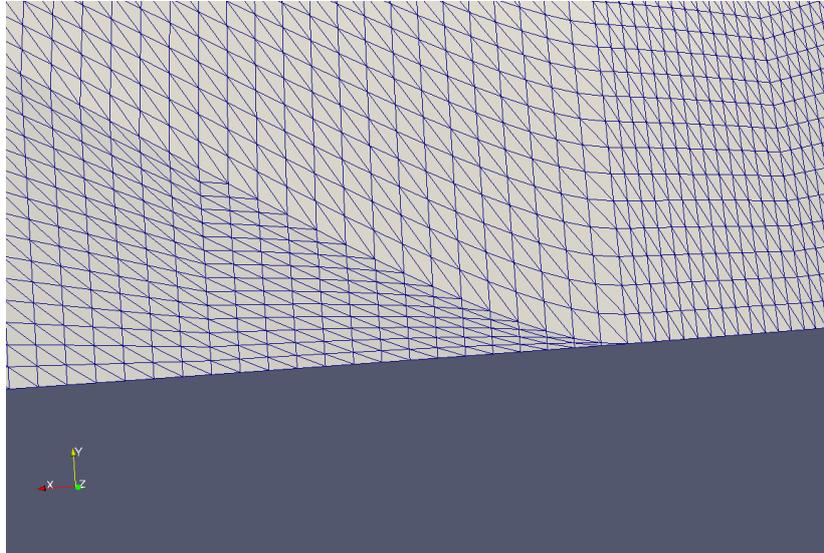


Abbildung 7.14: Nahaufnahme der gleichmäßig triangulierten Geometrie: zusammenfallende Teilflächen

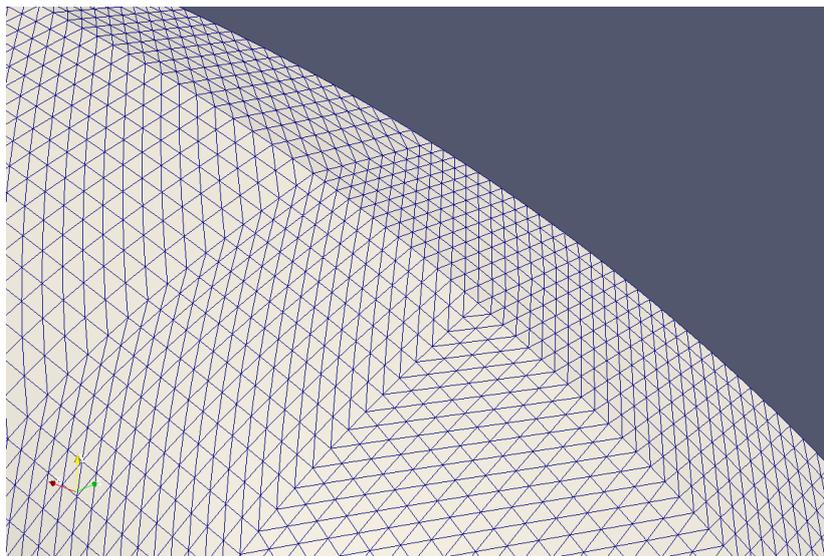


Abbildung 7.15: Nahaufnahme der gleichmäßig triangulierten Geometrie: Schanzkleid

## 7.2 Strömungssimulation

### 7.2.1 Simulation

An dieser Stelle werden die Ergebnisse der Simulation vorgestellt. In Abb. 7.16 ist das Oberflächennetz zu sehen, welches mit Hilfe des Paving Algorithmus auf Basis der triangulierten Geometrie erstellt wurde. Dieses Oberflächennetz dient als Grundlage für die Erstellung der inneren Schicht und basiert auf der gleichmäßigen Triangulierung der Schiffskörpergeometrie, welche in dem Abschnitt zuvor gezeigt wurden.

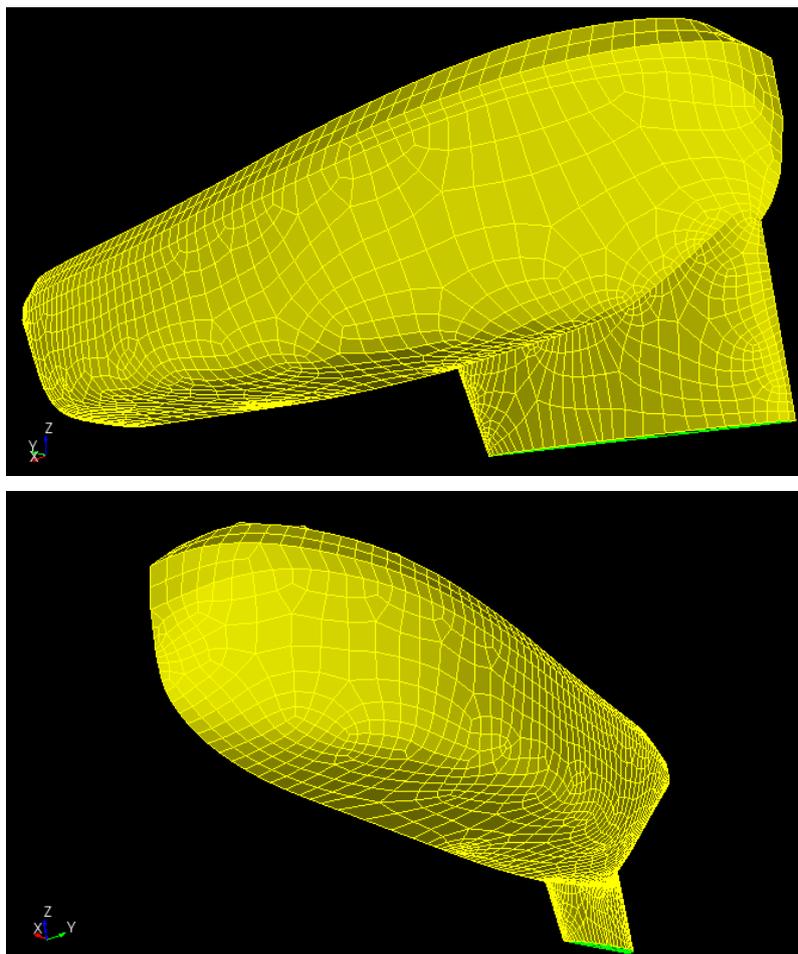
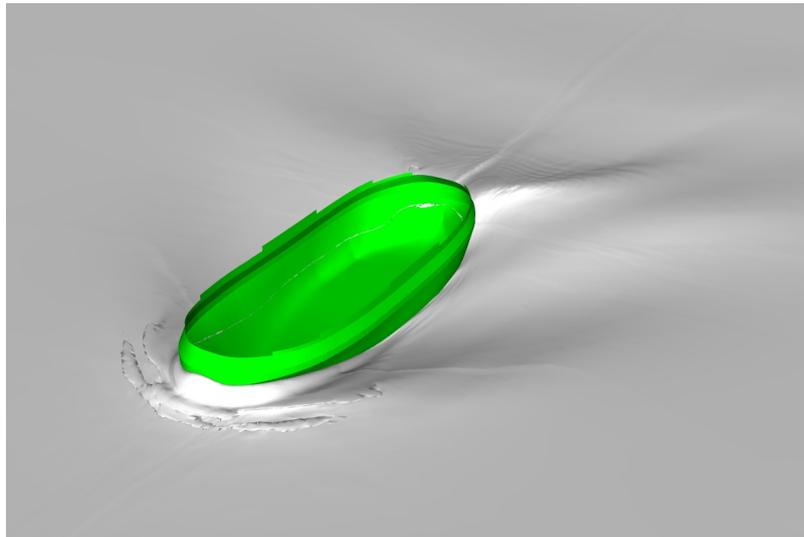


Abbildung 7.16: Pavingnetz für eine Schlepper Geometrie

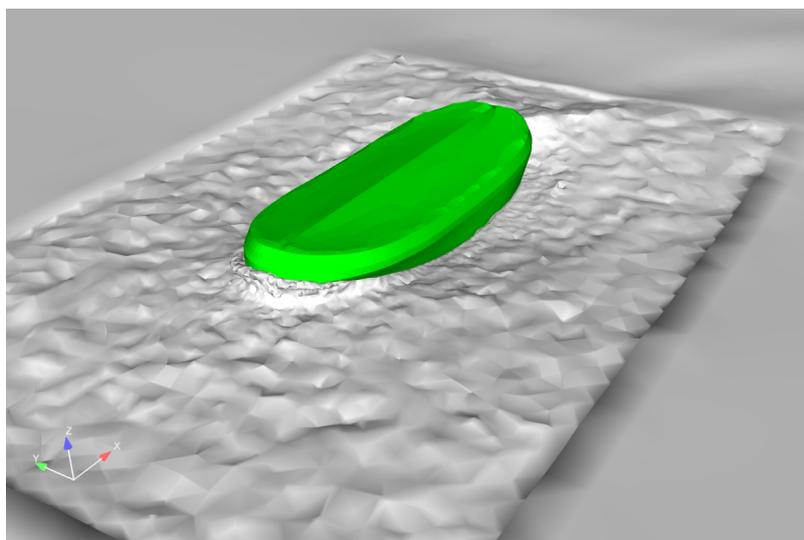
Um die Qualität der Simulation, welche eine Abhängigkeit von dem Netz besitzt, zu beurteilen, wurden Vergleichsrechnungen mit Simulation von Voith durchgeführt. Es wurden die folgenden Szenarien verglichen:

- Die Simulation mit der in dieser Arbeit vorgestellten Simulationskette.
- Die Simulation mit einem manuellen Netz, welches von Voith zur Verfügung gestellt wurde mit identischen Lösereinstellungen.

In beiden Szenarien wird die identische Geometrie eines Schleppers mit Finne verwendet. Die Geometrie des Schleppers wurde auf Modellgröße skaliert. Im Fall der manuellen Vernetzung wurden zwei Netze mit unterschiedlicher Anzahl von Zellen (3.6 Mio. Zellen und 1.7 Mio. Zellen) generiert. Mit Hilfe des automatischen Vernetzers wurde ein Netz mit 1 316 000 Zellen generiert. In allen Fällen handelt es sich um ein Vollmodell. In Abb. 7.17 sind die Ergebnisse dieser Strömungssimulationen zu sehen. Auf dem oberen Bild ist die Lösung für das feine Netz zu sehen, wohingegen auf dem unteren Bild die Lösung mit automatischen Netz abgebildet ist. Der Unterschied der Auflösung ist deutlich erkennbar.



(a) Simulation mit einem feinen Netz von Voith



(b) Simulation mit einem Netz des automatischen Gittergenerierers

Abbildung 7.17: Simulationsergebnisse für die Schlepper Geometrie

In Abb. 7.18 sind die Ergebnisse für die Widerstandskraft der beiden Simulationen

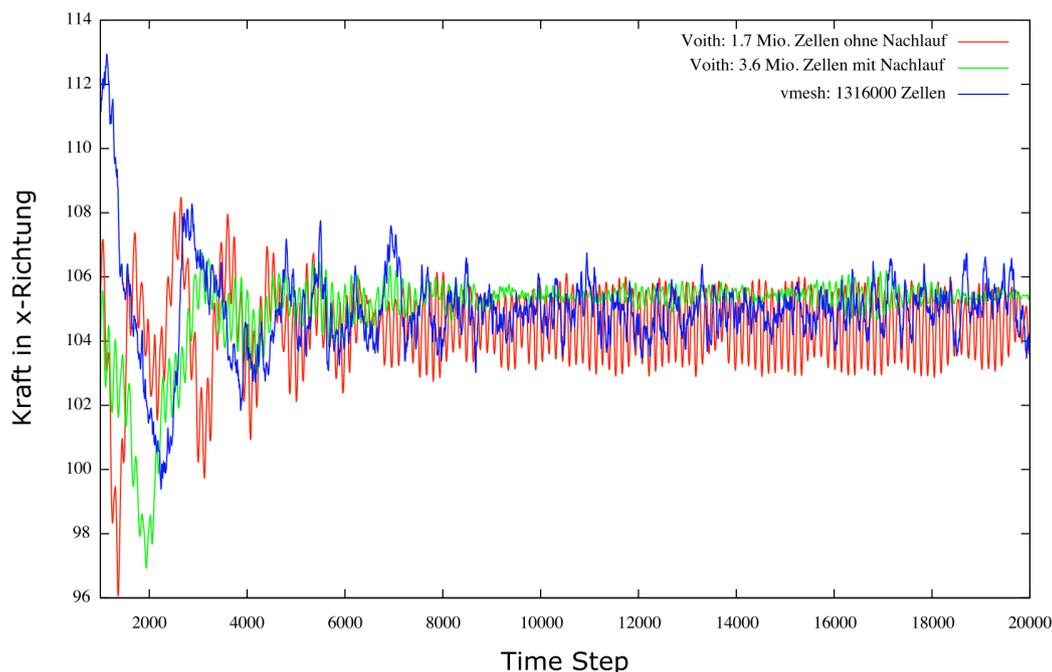


Abbildung 7.18: Vergleich der Simulationsergebnisse

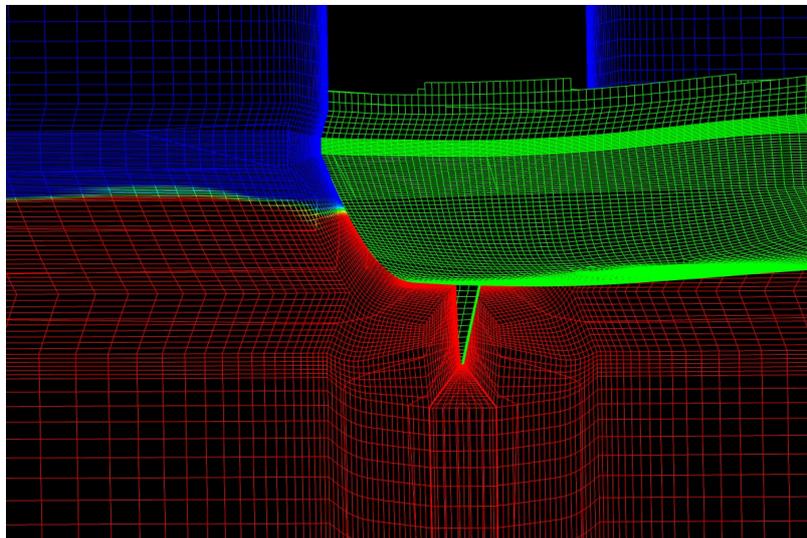
zu sehen. Auf der  $x$ -Achse sind die Zeitschritte aufgetragen. Die Widerstandskräfte sind in  $y$ -Richtung aufgetragen, welche aus dem jeweiligen Zeitschritt aus der Lösung berechnet werden. Ein Zeitschritt wurde dabei als 0.001 Sekunden gewählt, d.h. die gesamte Berechnung erfolgte über 20 Sekunden.

Die Kurve in grün zeigt die Widerstandskraft für das Modell mit 3.6 Mio. Zellen. Die Widerstandskraft für das Modell mit 1.7 Mio Zellen wird durch die rote Kurve beschrieben. Die blaue Kurve ist die Widerstandskraft aus der automatischen Simulation mit einem Modell mit ungefähr 1 316 000 Zellen. Es ist anzumerken, dass bei der manuellen Erstellung der Netze ein unterschiedlicher Abstand zum Auslauf gewählt wurde. In dem Modell mit 3.6 Mio. Zellen wurde ein größerer Abstand zwischen Schiff und Auslauf des Strömungsquaders gegenüber dem Modell mit 1.7 Mio. Zellen gewählt.

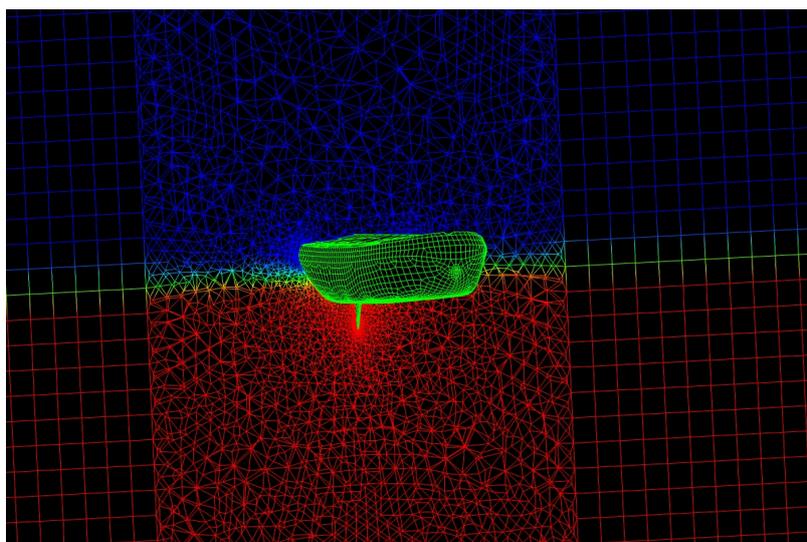
Bei der Betrachtung dieser Ergebnisse ist zu sehen, wie bereits mit einer geringen Auflösung gute Ergebnisse erzielt werden. Als Referenz für gute Ergebnisse wird das vorliegende manuell erstellte Netz mit 3.6 Mio. betrachtet. Da die Ergebnisse dieser Simulation für die spätere Optimierung verwendet werden, ist die Analyse dieser Ergebnisse wichtig.

Es wurde untersucht, inwieweit eine weitere Verfeinerung des Netzes eine Verbesserung der Ergebnisse liefert. Hierzu wurde an den Parametern für die automatische Vernetzung gedreht und beobachtet. Durch Modifikation dieser Parameter kann die Anzahl der Zellen beeinflusst werden. Es wurde untersucht ob durch eine Erhöhung der Zellen weitere Verbesserungen erzielt werden können. Es zeigte sich, dass eine Erhöhung

der Anzahl der Zellen keine weiteren Verbesserungen mit sich bringt, jedoch eine Reduktion der Zellanzahl schlechtere Ergebnisse liefert, [19]. Bei Betrachtung von Abb.



(a) Manuell erstelltes Rechengitter



(b) Automatisch generierte Rechengitter

Abbildung 7.19: Vergleich der generierten Gitter

7.17 fällt die unterschiedliche Güte bei der Auflösung des sich ergebenden Wellenbildes auf. Das manuelle Netz kann das Wellenbild deutlich feiner und glatter auflösen. Dies liegt daran, dass bei der manuellen Vernetzung speziell darauf geachtet wurde, dass die Grenzschicht beim Phasenübergang zwischen Wasser und Luft gut aufgelöst wird. Dies wird bei dem automatischen Verfahren in der Simulationsskette nicht berücksichtigt. In Abb. 7.19 ist dieser Sachverhalt zu sehen.

## 7.3 Geometrieoptimierung

### 7.3.1 Studie von Designvariationen

Auf Basis der Ergebnisse aus dem vorherigen Abschnitt wurde die Geometriemodellierung umgesetzt. Hierzu wurde zunächst ein einfaches Modell herangezogen und eine Optimierung auf Basis von Designvariationen durchgeführt. Es wurden die folgenden Einstellungen für die Netzparameter in der Simulation gewählt:

Parametername	Wert
Zellgröße: zg	170
Deckel: oben	3378
Boden: unten	-6622
Seite: seite	7000
Vordere Fläche: vorne	20500
Hintere Fläche: hinten	-7000
pu	-1500
ps	1500
pv	4000
ph	-1500
Genauigkeit vmerge: eps	0.5
Skalierung: scaling	1
Pyramidenrichtung: change	-1
Dicke einer Zelle: thickness	1
Anzahl der Zellschichten: num of lay	16
Streckung: dilation	1.09
Offset-Algorithmus: p	1
Paving	1.7 (auto)
Glättung des Oberflächennetzes	Laplace

Als Ausgangsgeometrie wurde das Modell *nurbs\_ship.v1\_107* gewählt. Bei diesem Schiff handelt es sich um einen einfachen Schlepper ohne Anbauteile. Intern hat dieser Schiffstyp die Identifikationsnummer 8 und entspricht dem Typ des einfachen Wassertrecker aus Kap. 2. Das Einlesen der Geometrie und die Erzeugung der Oerflächentriangulierung wird über das in Listing 7.1 gezeigte Programm realisiert. Hierbei wurde die gleichmäßige Triangulierung der Oberfläche (typ 9) benutzt.

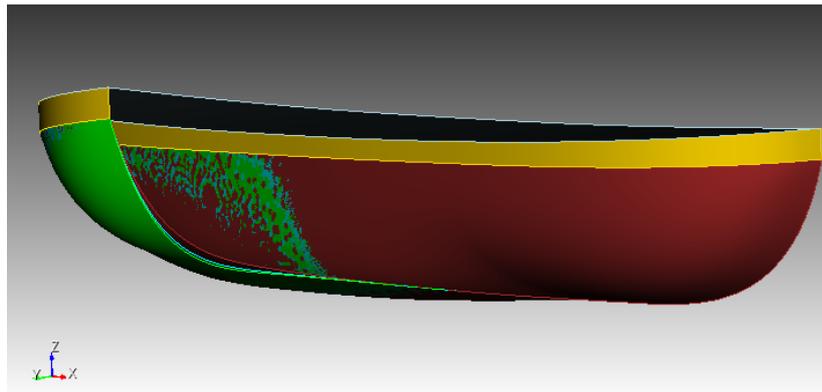
Listing 7.1: Programmcode Einlesen und generieren der STL Triangulierung

```
...  
#include <shipclasses/shipclasses.h>  
  
using namespace std;  
using namespace jshiptools;  
  
int main()  
{  
    string pashimo = getenv("SHIP_HOME");  
    pashimo = pashimo + "_savedDL/simu_run.ship_dat";  
    int ship_type = 8;  
  
    Ship ship;  
    ship.open_NURBS(pashimo);  
    ship.set_type(ship_type);  
  
    jshiptools::stl_net_file_9(ship, ". /stl_complete", 1);  
  
    return 0;  
}
```

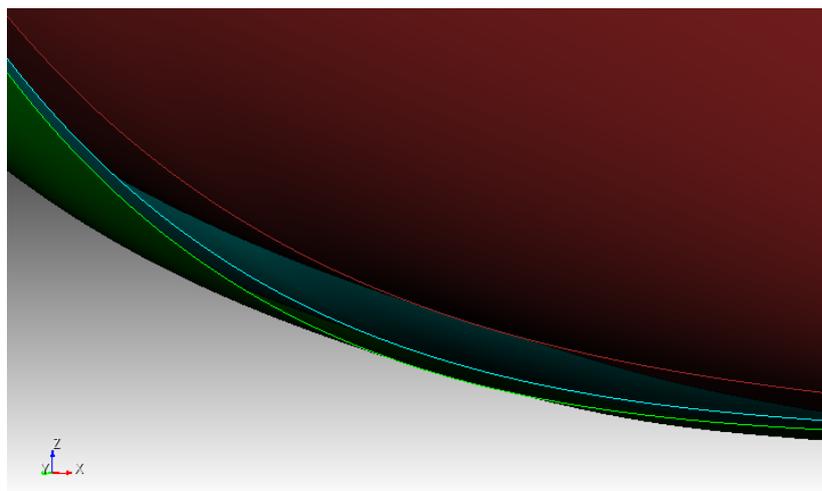
Es wurden zwei Variationen der Geometrie durch die Modifikation eines Parameters erzeugt, welcher den Verlauf der Kiellinie des Schiffes beeinflusst. Ziel war es, eine geeignete Einstellung für die Simulation zu finden, so dass die Modifikation der Geometrie in den Ergebnissen der Strömungssimulation abgebildet wird. Die Geometrievariationen sind in Abb. 7.20 abgebildet. Es wurden bewusst Veränderungen verschiedener Größe gewählt, um an diesen die Sensitivität der Lösung zu beobachten. Die Ausgangsgeometrie ist in grün dargestellt. Die Modifikationen sind rot und blau eingefärbt. Beide Modifikationen sind Verschlinkungen der Ursprungsgeometrie und verschieden stark ausgeprägt.

Bei Betrachtung der Ergebnisse der Simulation für diese Designvariationen werden die Abweichungen der Widerstandskräfte jeder Variation sichtbar, siehe Abb. 7.21. Die Kurven beschreiben hierbei den Verlauf der Widerstandskraft über die einzelnen Zeitschritte. Die rote Kurve gehört zu der Ausgangsgeometrie, wohingegen die grüne und die blaue Kurve zu den Modifikationen gehören. Es ist klar zu erkennen, dass die Geometriemodifikationen in den Kurven abgebildet werden. Auch die gemittelten Werte für den Widerstand können die Modifikationen abbilden. Die Mittelung erfolgt hierbei über die letzten 5000 Zeitschritte.

Modifikation	Gemittelter Widerstandswert
Ausgangsgeometrie Zellgröße:	88,6
kleine Veränderung	86,8
große Veränderung	80,4



(a) Ansicht auf das halbe Schiff



(b) Vergrößerung der wichtigen Stelle an der Geometrie

Abbildung 7.20: Designvariationen von *nurbs\_ship\_v1\_107*

Die berechneten Widerstandskräfte über die Simulation sind ebenfalls plausibel bezüglich der Modellierung des Problems. Es ist eine Verringerung des Widerstandes für die modifizierten Geometrien zu erwarten, welcher durch die Ergebnisse abgebildet wird. Außerdem wird die Größe der Veränderung der Geometrie durch die Ergebnisse abgebildet. Diese Resultate bestätigen somit die Tauglichkeit der getroffenen Einstellungen in der Simulationskette für eine Optimierung.

Des weiteren wurde eine Designstudie für das Modell *nurbs\_ship\_v1\_108* betrachtet. Hierbei handelt es sich um einen Schlepper mit Bodenplatte (typ 10). In diesem Fall

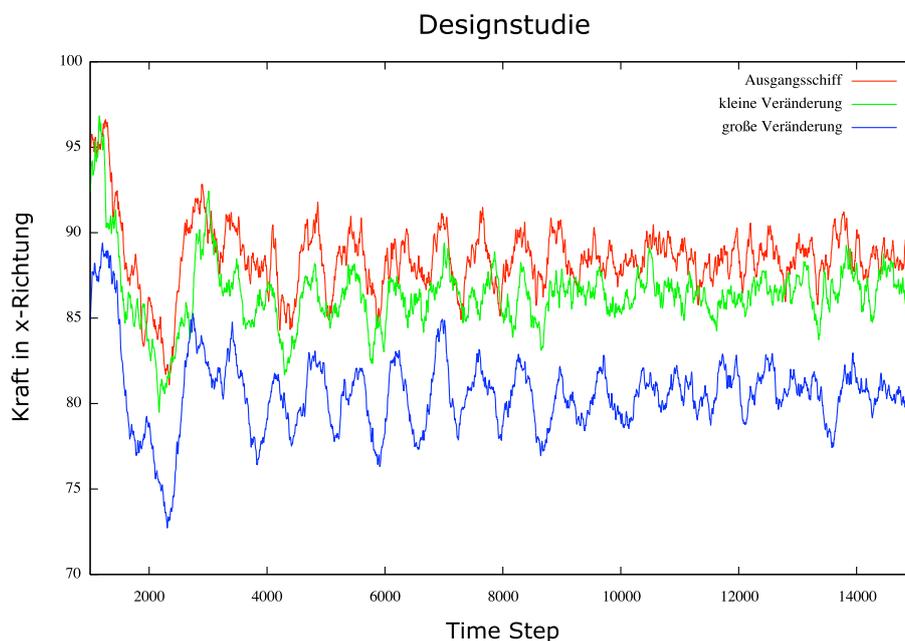
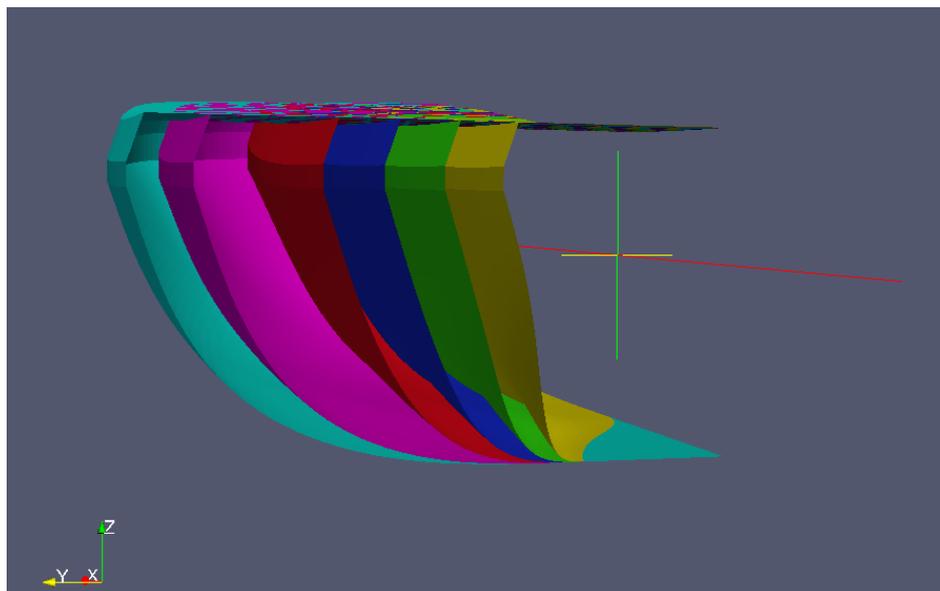


Abbildung 7.21: Vergleich der Simulationsergebnisse für die Variationen

wurde untersucht, inwiefern die Einstellungen von obigen Studien auf andere Modelle und Modifikationen übertragen werden können. Es wurde der Parameter, welcher den Verlauf der Seitenwand des Schiffes beeinflusst, ausgewählt. Hierzu wurden fünf Variationen der Ausgangsgeometrie erzeugt, siehe Abb. 7.22. Damit die Abweichung in der Geometrie besser sichtbar werden, handelt es sich bei dieser Abbildung um einen Schnitt durch übereinandergelegten Geometrievariationen in der Mitte des Schiffes. Die blau eingefärbte Geometrie ist die Ausgangsgeometrie. Bei dem Vergleich der Simulationsergebnisse werden nur die gemittelten Widerstandskräfte betrachtet, da ein Vergleich der Verlaufskurven sehr unübersichtlich ist.

Ausgehend von dem Parameterwert 0,90996 wurde dieser mit Schrittweite 0,15 modifiziert. Es ergeben sich bei der Mittelung die folgenden Werte für die Widerstandskraft in Abhängigkeit des Parameterwertes, welcher modifiziert wurde:

Parameterwert	Gemittelter Widerstandswert
0,60996	96,2071
0,75996	119,141
0,90996 (Ausgangswert)	152,226
1,05996	171,052
1,20996	187,687
1,35996	233,107
1,50996	248,354

Abbildung 7.22: Designvariationen von *nurbs\_ship\_v1\_108*

In Abb. 7.23 ist der Sachverhalt grafisch veranschaulicht. Hierzu wurde in einem Graphen die gemittelten Widerstandskraft gegen den Parameterwert aufgetragen und anschließend noch linear interpoliert. In dieser Grafik ist gut zu erkennen, wie die Simulation klar unterscheidbare Werte für die Widerstandskraft generiert und somit die Tauglichkeit der Einstellungen für eine Optimierung bestätigt.

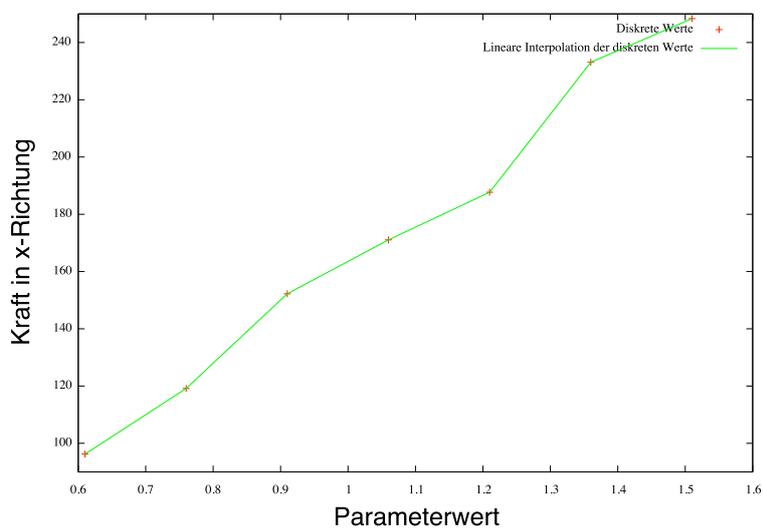


Abbildung 7.23: Veranschaulichung der Ergebnisse der zweiten Parameterstudie

### 7.3.2 Geometrieoptimierung

Auf Grundlage der Erkenntnisse der Designstudien wurden erste Optimierungen durchgeführt. Zunächst wurde mit einem Parameter der Modellierung die Wasserlinie des Schiffes hinsichtlich der Widerstandskraft optimiert. In diesem Fall lautet das Optimierungsproblem

$$\min_{x \in \mathbb{R}} \tilde{F}(x), \quad (7.3.1)$$

wobei  $\tilde{F}$  die gemittelte Widerstandskraft über den letzten 5000 Zweitschritten ist und  $x$  der Parameter der Geometrie. Wir schränken den Suchraum ein auf ein Intervall

$$I = [d - \lambda d, d + \lambda d] \subset \mathbb{R}, \quad (7.3.2)$$

wobei  $d$  den Ausgangswert des Parameters aus einem Geometriemodell bezeichnet und  $\lambda$  ein Streckungsparameter für den Suchraum beschreibt. Es wurde  $\lambda = 0.3$  gewählt. Die Einstellungen für die Simulation entsprechen den aus dem Abschnitt zuvor für die Designstudie. Als Optimierungsverfahren wurde das Hooke und Jeeves Verfahren verwendet mit einer Startschrittweite  $\delta = 0.15$ . Als Ausgangsgeometrie wählte man *nurbs\_ship\_v1\_107*. Es wurde ein Parameter ausgewählt welcher den Verlauf der Kiellinie des Schiffes beschreibt (Curve 0, Dim 2, Pos 5). Es gilt für den Ausgangswert

$$d = 0. \quad (7.3.3)$$

Daher wählen wir in diesem Fall das Suchintervall

$$I = [-0.6, 0.6] \subset \mathbb{R}. \quad (7.3.4)$$

Das sich ergebende Problem ist visuell in Abb. 7.24 dargestellt. Die Ausgangsgeometrie ist rot eingefärbt. Die blaue Geometrie ist die Geometrie zum Parameterwert 0.6. Die gelbe Geometrie ist die Geometrie zum Parameterwert -0.6.

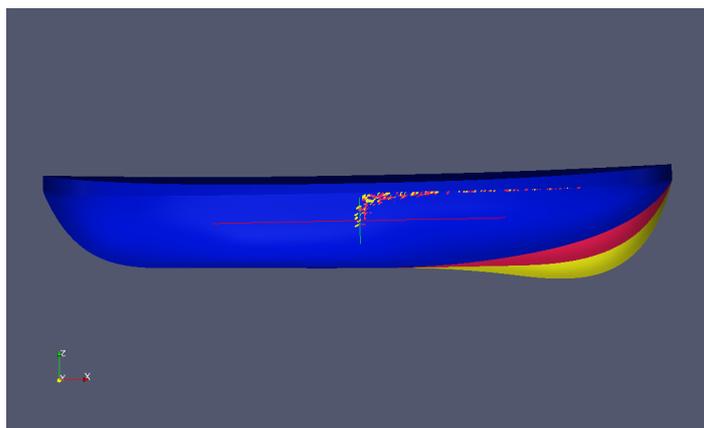


Abbildung 7.24: Erstes Optimierungsproblem in einer Variable

Die Optimierung liefert den Optimalwert 0.6 mit einer Widerstandskraft von 92.2. Die

optimale Geometrie entspricht der blauen Geometrie in Abb. 7.24. Zum Auffinden der Lösung wurden vier Auswertungen benötigt. Die Durchführung der Optimierung zeigt, dass eine Optimierung erfolgreich durchgeführt werden kann.

Es wurde ein komplizierterer Fall betrachtet. In diesem erfolgt die Optimierung über zwei Parameter und das Problem lautet

$$\min_{x \in \mathbb{R}^2} \tilde{F}(x), \quad (7.3.5)$$

wobei  $\tilde{F}$  erneut die gemittelte Widerstandskraft ist und  $x$  der Parametervektor mit den Geometrieparametern.

Es werden in diesem Fall zwei Parameter ausgewählt, welche die Kiellinie beschreiben. Zusätzlich zu dem Parameter aus dem einfachen Optimierungsfall wird der Nachbar-kontrollpunkt für die Kiellinie ausgewählt (Curve 0, Dim 2, Pos 4). Dieser Parameter besitzt den Startwert 0. Dementsprechend wählt man das Suchintervall

$$I = [-0.6, 0.6] \times [-0.6, 0.6] \subset \mathbb{R}^2. \quad (7.3.6)$$

Das sich ergebende Problem ist visuell auf Abb. 7.25 dargestellt. Die blaue Geometrie gehört zum Parametertupel  $(0.6, 0.6)$ , die rote Geometrie ist die Ausgangsgeometrie und die gelbe Geometrie die Realisierung der Parameter  $(-0.6, -0.6)$ .

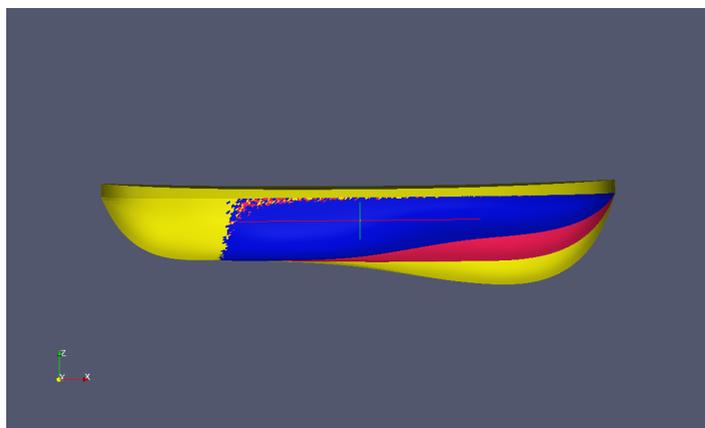


Abbildung 7.25: Zweites Optimierungsproblem in zwei Variablen

In diesem Fall wird das Verfahren von Hooke und Jeeves verwendet, welches neun Auswertungen benötigte. Dieses liefert als optimale Lösung  $(0.6, 0.6)$  mit einem Widerstandswert von 83.7.

Anschließend wurde eine Abwandlung des zweidimensionalen Optimierungsfalles durchgeführt.

Es wurde die Kopplung zweier Parameter betrachtet. In diesem Fall erfolgt die Optimierung mit einer Nebenbedingung für die Parameter. Diese lautet

$$g(x) = g^*, \quad (7.3.7)$$

wobei  $g$  eine Funktional auf der Geometrie darstellt und  $g^*$  ein vorgegebener Wert ist. In diesem Fall wurde  $g$  als die Funktion, welche uns das Volumen der Geometrie berechnet, gewählt. Für  $g^*$  wählen wir das Volumen der Ausgangsgeometrie. Diese Funktion  $g$  ist in dem Modellierungswerkzeug implementiert und wird an dieser Stelle verwendet. Anschaulich erfolgt eine Optimierung der Geometrie unter der Bedingung, dass sich das Volumen dieser nicht ändert. Die Kopplung der Parameter ist wie folgt realisiert. Für einen fest gegebenen Wert  $x_1^*$  der ersten Variable wird  $x_2^*$  so gewählt, dass gilt

$$g(x_1^*, x_2^*) = g^*. \quad (7.3.8)$$

Die Lösung dieser Bedingung erfolgt in einem eigenen Schritt unabhängig von der Simulation. Es ergibt sich ein eindimensionales Problem, wobei die Variable  $x_1$  als Optimierungsparameter gewählt wird. Das Suchintervall für  $x_1$  wird gewählt als

$$I = [-0.6, 0.6] \subset \mathbb{R}. \quad (7.3.9)$$

Als Optimierungsverfahren wurde das Verfahren von Hooke und Jeeves mit einer Anfangsschrittweite  $\delta = 0.1$  eingesetzt. Der Verlauf der Optimierung ist in Abb. 7.26 zu sehen. In diesem Bild ist die Widerstandskraft in  $x$ -Richtung abgetragen gegen die beste Lösung jedes Iterationsschritts. Die Erkundungsschritte sind hierbei nicht abgebildet.

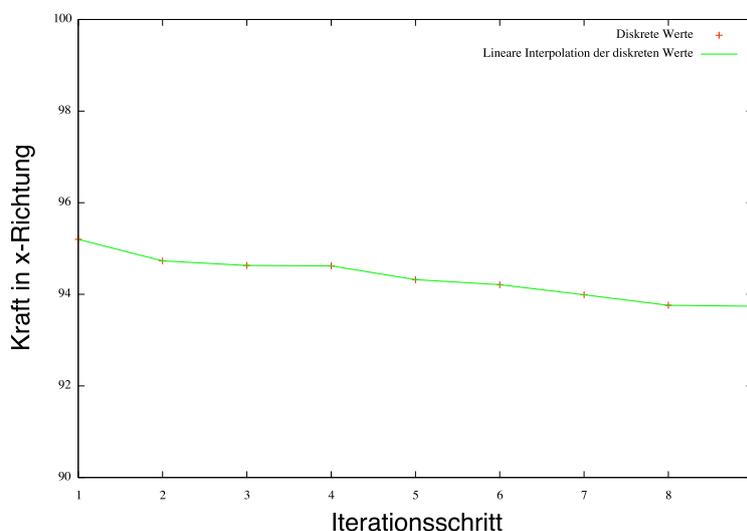


Abbildung 7.26: Optimierungsverlauf bei Kopplung zweier Parameter

Das Optimum ist  $(0.135, -0.075)$  mit einer Widerstandskraft von 93.74. Die sich ergebende Geometrie ist in Abb. 7.27 gezeigt. Die blaue Geometrie ist hierbei das optimale Ergebnis der Optimierung und die rote Geometrie die Ausgangsgeometrie. Es ist zu sehen, wie sich die Geometrien schneiden.

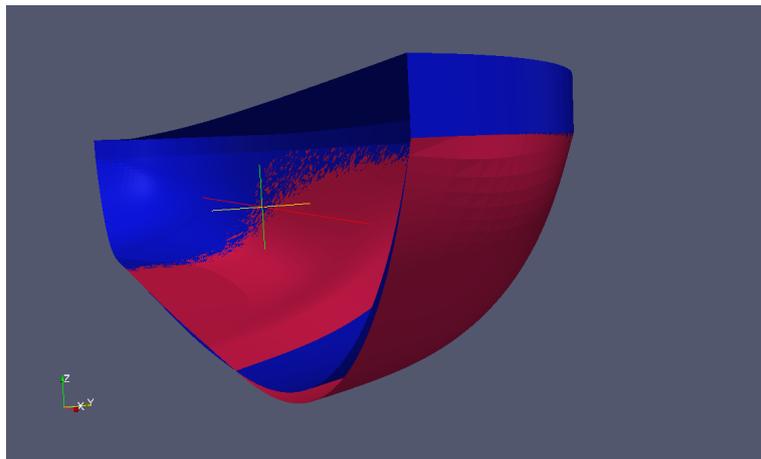


Abbildung 7.27: Ergebnisse bei Kopplung zweier Parameter

## 7.4 Netzbasierte Formoptimierung

Bei der Umsetzung der netzbasierten Formoptimierung wurde ein vom Vernetzungswerkzeug generiertes Netz mit 1 316 000 Zellen verwendet. Auswahlkriterium für dieses Netz war die Existenz von guten Ergebnissen auf diesem Netz. Dieses Kriterium wurde gewählt, da bei diesem Optimierungsverfahren das Netz deformiert wird und daher die Qualität des Netzes sich zunehmend verschlechtert.

Zunächst wurden Analysen für die Parameter  $\omega$  der Deformation durchgeführt. Anschließend wurde eine Deformationsrichtung  $\vec{d}_k$  und ein Deformationszentrum  $R_k$  für die Deformation ausgewählt, [19].

Ziel der Analyse war es festzustellen wie das Intervall der Parameter  $\omega$  zu wählen ist, damit keine Überlappung von Zellen in dem Netz entstehen. In Abb. 7.28 ist die Anwendung einer Deformation auf das Oberflächennetz zu sehen.

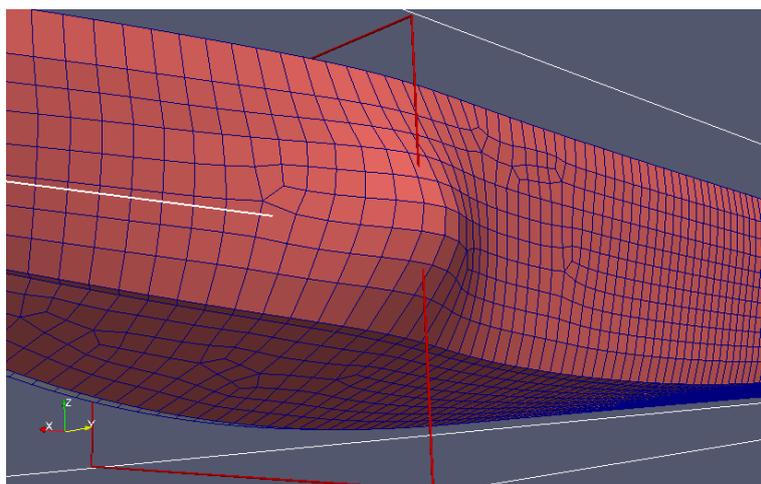


Abbildung 7.28: Oberflächennetz einer Geometrie bei großer Deformation, [19]

Es wurde zum Optimieren das Verfahren von Hooke und Jeeves verwendet, [19]. In Abb. 7.29 ist der Verlauf der Optimierung dargestellt. In dem Graphen ist die Widerstandskraft in  $x$ -Richtung in jedem Iterationsschritt der Optimierung zu sehen. Die Ergebnisse der netzbasierten Optimierung sind auf Abb. 7.30 und 7.31 zu sehen.

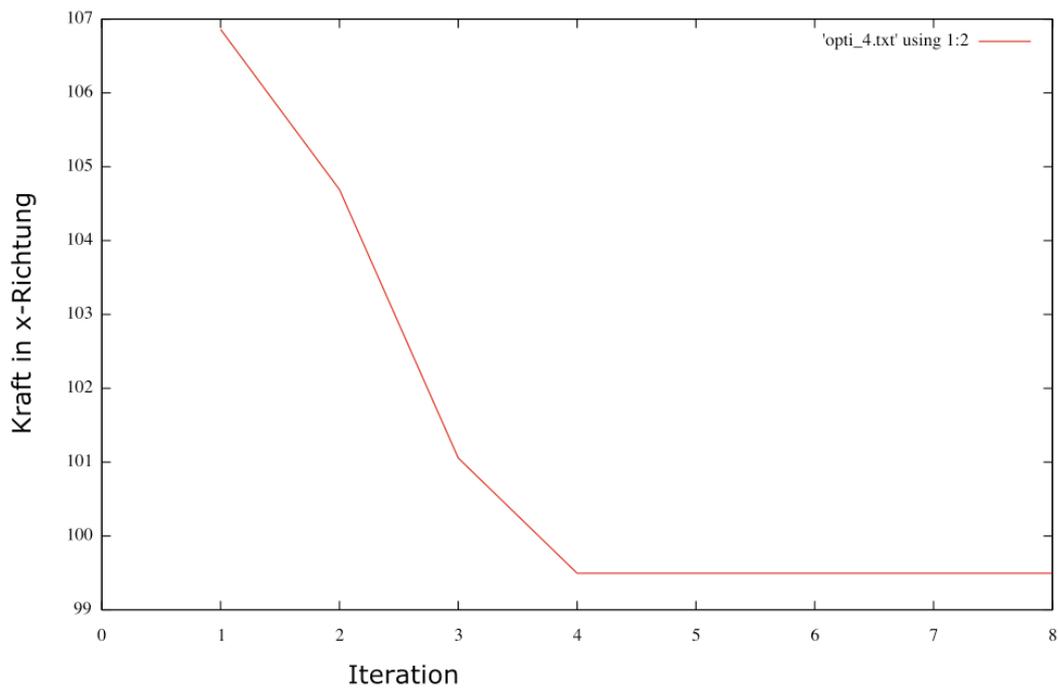


Abbildung 7.29: Verlauf der Formoptimierung, [19]

Hierbei ist das gelbe Netz das Ausgangsnetz der Optimierung und das blaue Netz das optimierte Oberflächennetz der Geometrie. Auf Abb. 7.30 sind die Verformungen der Zellen auf der Oberfläche gut zu erkennen.

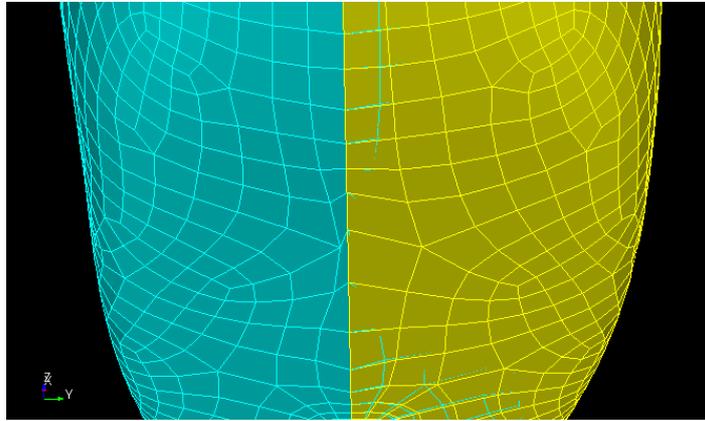


Abbildung 7.30: Ergebnis der netzbasierten Optimierung: Verformung der Zellen auf der Oberfläche, [19]

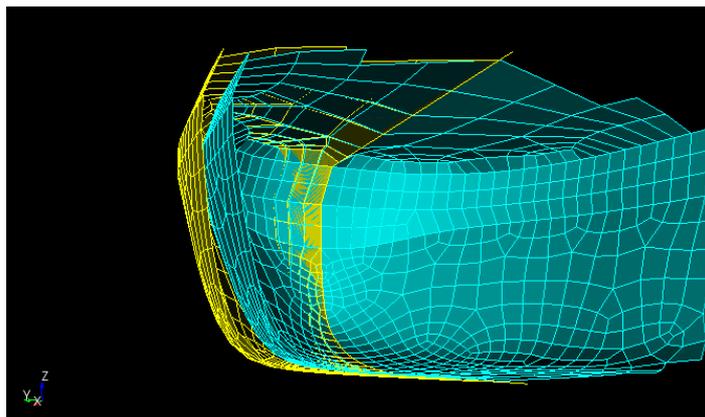


Abbildung 7.31: Ergebnis der netzbasierten Optimierung: Vergleich der beiden Oberflächen

# Kapitel 8

## Ausblick

Bei der Erarbeitung der Methoden zur Modellierung, Simulation und Optimierung von Schiffsgeometrien ergaben sich neben den erzielten Ergebnissen neue Fragestellungen für weitere Arbeiten.

Das Modellierungswerkzeug PaShiMo kann um zusätzliche Modelle von Geometrien für Schiffe erweitert werden. Hierbei ist sogar die Modellierung von komplexen Anbauteilen denkbar.

Interessant dürfte auch die Verbesserung der Oberflächentriangulierung sein. Diese muss zur Zeit für jeden Modelltyp manuell eingestellt werden. Hier ist eine automatische Steuerung über Parameter denkbar, so dass der Bediener sich auf die reine Modellierung beschränken kann.

Bei der Vernetzung kann eine Verbesserung des Auflösungsvermögens des Wellenbildes erzielt werden durch Einführen von hexaeder Zellen in Grenzschichtnähe.

Bei der Simulation kann durch geeignete Wahl der Parameter für die Vernetzung und die Strömungssimulation eine weitere Verbesserung der Ergebnisse erzielt werden. Insbesondere die Rechenzeit einer Simulation könnte durch eine gute Wahl der Einstellungen reduziert werden.

Durch Einbringen von Methoden des Automatischen Differenzierens ist ebenfalls der Einsatz von gradientenbasierten Verfahren in der Optimierung denkbar.

Auch die Einführung von 'billigen' Suchschritten auf groben Gittern bei der Optimierung ist denkbar. Dies würde eine deutliche Verbesserung der Laufzeit der Optimierung bringen.

# Danksagung

An dieser Stelle möchte ich mich bei allen bedanken, welche sowohl durch ihre fachliche als auch persönliche Unterstützung zum Gelingen dieser Dissertation beigetragen haben.

Vor allem danken möchte ich meinem Doktorvater Prof. Karsten Urban für seine sehr gute Betreuung und seine hilfreichen Anregungen.

Ebenfalls Bedanken möchte ich mich bei den Mitarbeitern der Firma Voith Turbo Schneider Propulsion für die hilfreiche Unterstützung bei kniffligen Fragen. Im Besonderen gilt hier der Dank Dirk Jürgens, Michael Palm, Sebastian Singer und David Bendl.

Mein Dank gilt auch den Kolleginnen und Kollegen des Institut für numerische Mathematik die mich bei der täglichen Arbeit unterstützten und begleiteten. Ein besonderer Dank geht an Michael Hopfensitz für seine Hilfsbereitschaft und Kollegialität.

Meinen Eltern, Kurt und Dina, sowie meiner Schwester Natalie möchte ich herzlich danken für ihre Unterstützung.

Meiner Freundin Michaela Mezler-Andelberg gilt mein ganz besonderer Dank für ihren Rückhalt und ihr Verständnis während dieser Zeit.

# Index

- Affine Invarianz, 15
- API, 36
- Aufwinddifferenz, 30
- Automatischer Vernetzer, 19
  
- B-Spline Kurve, 12
- Bilanzgleichungen, 29
- Blended-Diskretisierungsschema, 31
- Blendingfaktor, 31
  
- C++, 36
- CAGD, 9
- Comet, 31
- Computer Aided Geometric Design, 9
- CUBIT, 25
  
- Designkurve, 43
- diffuser Transport, 28
- Diffusionskoeffizient, 28
  
- extensive Eigenschaft, 26
  
- Fick-Gesetz, 28
- Finite-Volumen-Methode, 28
  
- generischen Erhaltungsgleichung, 28
- globale Parameter, 44
- Gradienten-Approximation, 28
  
- Halbmodell, 23
  
- Impulserhaltung, 28
- intensive Eigenschaft, 26
- Interpolationsmethoden, 30
- Isogeometrische Beschreibung, 36
  
- Karmannsche Wirbelstraße, 23
- Konservativität, 29
- Kontinuitätsgleichung, 27
- Kontrollmasse, 26
  
- Kontrollvolumen, 26
- Kontrollvolumenmethode, 26
- Konvektion, 28
- Konvexe-Hülle-Eigenschaft, 15
  
- Lösungsgebiet, 28
- Lokalität, 16
  
- Mittelpunktregel, 29
  
- Navier-Stokes-Gleichungen, 28
- Newtonsche Fluid, 28
- Numerische Diffusion, 30
- NURBS Kurve, 15
  
- Oberflächenintegral, 29
- Offset-Algorithmus, 23
- Oszillation, 30
  
- Paraview, 61
- PaShiMo, 33
  
- Randwerte, 31
- Rechenknoten, 28
- relationäre Blocktopologie, 40
- Reynolds-Transporttheorem, 27
  
- Schichtenmodell, 20
- Schnittpunkte von NURBS Kurven, 16
- SimuVSP Projekt, 5
- Spanten, 36
- Spantkurve, 37
- Spantparameterraum, 37
- STL Triangulierung, 50
- Straklatten, 36
  
- Teilkurve, 39, 40
  
- vmesh, 6, 19
- Voith-Schneider-Propeller, 6

Vollmodell, 23

Volumenintegral, 29

Vordiskretisierung, 50

VSP, 6

Wigleyschiff, 10

# Literaturverzeichnis

- [1] Autoship. <http://www.autoship.com>, 23. November 2012.
- [2] J.-E. Bartels and D. Jürgens. The Voith Schneider Propeller: Current applications and new developments. Technical report, Voith Turbo Marine GmbH, 2006.
- [3] T. D. Blacker and M. B. Stephenson. Paving: A new approach to automated quadrilateral mesh generation. *International Journal for Numerical Methods in Engineering*, 32(4):811–847, 1991.
- [4] H. Borouchaki, F. Hecht, and E. S. P. George. Reasonably efficient Delaunay based mesh generator in 3 dimensions. In *In the 4th International Meshing Roundtable*, pages 3–14, 1995.
- [5] A. Brix. *Praktischer Schiffbau*. Salzwasser-Verlag, 2010.
- [6] L. Buczkowski. Mathematical construction, approximation and design of ship body form. *Journal of Ship Research*, 13(3):185–206, 1969.
- [7] J. Carloson, A. Jaffe, and A. Wiles. *The Millennium Prize Problems*. Clay Mathematics Institute, American Mathematical Society, 2006.
- [8] E. E. Catmull. Introduction of compute technology in the film industry. In *COMP-CON'82, Digest of Papers*, pages 34–, 1982.
- [9] F. H. Chapman. *Architectura Navalis Mercatoria*. Dover Pubn Inc, 2006.
- [10] C. DeBoor. *A Practical Guide to Splines*. Springer Verlag, 2001.
- [11] P. Dierckx. *Curve and Surface Fitting with Splines*. Oxford Science Publications, 1995.
- [12] A. Dudzus and E. Danckwardt. *Schifftechnik, Einführung und Grundbegriffe*. VEB Verlag Technik Berlin, 1982.
- [13] Fastship. <http://www.proteusengineering.com/fastship.htm>, 23. November 2012.
- [14] J. Ferziger and M. Peric. *Numerische Strömungsmechanik*. Springer Verlag, 2008.
- [15] R. Fletcher and C. M. Reeves. Function minimization by conjugate gradients. *The Computer Journal*, 7(2):149–154, 1964.

- [16] A. Griewank and A. Walther. *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*. SIAM, 2008.
- [17] S. Harries. *Parametric Design and Hydrodynamic Optimization of Ship Hull Forms*. PhD thesis, Technische Universität Berlin, 1998.
- [18] R. Hooke and T. A. Jeeves. Direct search solution of numerical and statistical problems. *Journal of the ACM*, 8(2):212–229, 1961.
- [19] M. Hopfensitz. *Numerische Optimierung von Schiffen mit VSP: Automatische Gittergenerierung*. PhD thesis, Universität Ulm, 2011.
- [20] M. Hopfensitz, J. C. Matutut, and K. Urban. Numerical modelling and simulation of ship hull geometries. In *Progress in Industrial Mathematics at ECMI 2010*, pages 465–471. Springer Berlin Heidelberg, 2012.
- [21] Institute of Computational Continuum Mechanics GmbH. *COMET Version 2.00 - User Manual*, 2001.
- [22] D. Jürgens. *Theoretische und experimentelle Untersuchungen instationärer Tragflügelumströmungen und Entwicklung eines Berechnungsverfahrens für Vertikalachsmotoren*. PhD thesis, Rostock, 1994.
- [23] D. Jürgens, M. Palm, S. Singer, and K. Urban. Numerical optimization of the Voith-Schneider Propeller. *ZAMM - Journal of Applied Mathematics and Mechanics*, 87(10):698–710, 2007.
- [24] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.
- [25] C. Kuo and K. MacCallum. Computer aided applications in ship technology. *Computers in Industry*, 5(3):211–219, 1984.
- [26] K. H. Kwik. *Zur Mathematischen Darstellung von Schiffslinien*, volume 245 of *Schriftenreihe Schiffbau*. Institut für Schiffbau, Universität Hamburg, 1969.
- [27] L. Landweber. Wigley parabolic hull group discussion. *Proceedings of the Workshop on Ship Wave-Resistance Computations*, 1:51–66, 1979.
- [28] R. Leidenberger. Automatic differentiation in flow simulation. Master’s thesis, Universität Ulm, (Diplomarbeit), 2007.
- [29] M. Mitchell. An introduction to genetic algorithms. *Cambridge, MA: MIT Press*, 1996.
- [30] F. C. Munchmeyer, C. Schubert, and H. Nowacki. Interactive design of fair hull surfaces using b-splines. *Computers in Industry*, 1(2):77–86, 1979.
- [31] NAPA. <http://www.napa.fi/products-services/ship-design>, 23. November 2012.

- [32] J. A. Nelder and R. Mead. A simplex method for function minimization. *The Computer Journal*, 7(4):308–313, 1965.
- [33] V. Nissen. *Einführung in evolutionäre Algorithmen*. Vieweg, 1997.
- [34] J. Nocedal and S. Wright. *Numerical Optimization*. Springer, 2006.
- [35] M. Peric. *A Finite Volume Methode for the prediction of three-dimesional Fluid Flow in complex ducts*. PhD thesis, Imperial College, University of London, 1985.
- [36] L. Piegel and W. Tiller. *The NURBS book*. Springer, 1997.
- [37] A. M. Reed and H. Nowacki. Interactive creation of fair ship lines. *Journal of Ship Research*, 18(2):96–112, 1974.
- [38] Rhino. <http://www.rhino3d.com/marine.htm>, 23. November 2012.
- [39] D. R. White and P. Kinney. Redesign of the paving algorithm: Robustness enhancements through element by element meshing. *6th International Meshing Roundtable*, pages 323–335, October 1997.
- [40] Sandia National Laboratories. <http://cubit.sandia.gov/>, 23. November 2012.
- [41] Sandia National Laboratories. <http://www.paraview.org/>, 23. November 2012.
- [42] Sandia National Laboratories. *CUBIT 11.1 User Documentation*, 2008.
- [43] K. Stolle. Berechnung der Umströmung von Schiffen mit dem CFD-Code Open-FOAM. Master’s thesis, Universität Ulm, (Diplomarbeit), 2010.
- [44] J. F. Thompson, B. K. Soni, and N. P. Weatherill. *Handbook of Grid Generation*. CRC Press, 1999.
- [45] Voith. <http://www.voith.com>, 23. November 2012.
- [46] W. H. White. *Handbuch für den Schiffbau*. Salzwasser-Verlag, 2010.
- [47] W. Wigley. A comparison of experiment and calculated wave profiles and wave resistance for a form having parabolic waterlines. *Proceeding of Royal Society London A*, 144(851):144–159, 1934.

## **Erklärung**

Ich versichere hiermit, daß ich die Arbeit selbständig angefertigt habe und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie die wörtlich oder inhaltlich übernommenen Stellen als solche kenntlich gemacht habe.

Ulm, den

.....

Vorname Nachname / Unterschrift